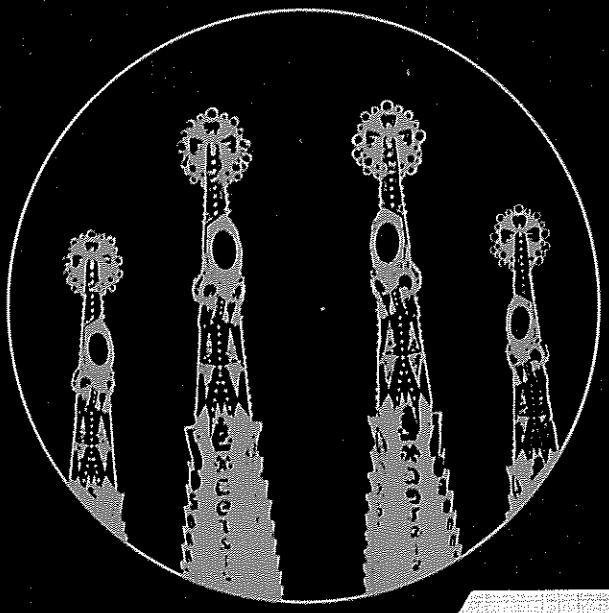
Volume III

SIGNAL PROCESSING V

THEORIES AND APPLICATIONS



L. Torres
E. Masgrau
M.A. Lagunas
editors

Elsevier

Torres, L.;
Masgrau, E.; Lagunas, M.:
5th European Signal Proceeing Conference, Sept. 18-21, 1990, Barcelona. vol.3.

Amsterdam: Elsevier Science Publishers, 1990.

SIGNAL PROCESSING V
THEORIES AND APPLICATIONS

SIGNAL PROCESSING V
THEORIES AND APPLICATIONS

Proceedings of EUSIPCO-90
Fifth European Signal Processing Conference
Barcelona, Spain, September 18–21, 1990

Edited by

Luis TORRES
Enrique MASGRAU
Miguel A. LAGUNAS

Department of Signal Theory and Communications
ETSIT-UPC
Barcelona, Spain

VOLUME III

1990

ELSEVIER
AMSTERDAM • NEW YORK • OXFORD • TOKYO

ELSEVIER SCIENCE PUBLISHERS B.V.
Sara Burgerhartstraat 25
P.O. Box 211, 1000 AE Amsterdam, The Netherlands

Distributors for the United States and Canada:

ELSEVIER SCIENCE PUBLISHING COMPANY INC.
655 Avenue of the Americas
New York, N.Y. 10010, U.S.A.

ISBN: 0 444 88636 2

This book is printed on acid-free paper.

Printed in The Netherlands

FOREWORD

EUSIPCO-90, the European Signal Processing Conference, is the fifth of the International Conferences promoted and organized by EURASIP, the European Association for Signal Processing. This book (in three volumes) presents the Proceedings of the Conference. The conference was held September 18-21, 1990 in Barcelona, Spain.

EUSIPCO-90 consisted of 52 sessions organized in 6 parallel programs. The Scientific Committee reviewed over 710 submitted abstracts to select the 515 that were accepted for presentation at the conference. Each abstract was reviewed by at least 2 independent reviewers. In addition, 11 tutorials were given by well known experts in the areas.

The Technical Sessions were organized in 7 broad topics according to the following distribution:

A. THEORY OF SIGNALS AND SYSTEMS
1. Detection. 2.Estimation. 3.Filtering. 4.Spectral estimation. 5. Adaptive systems.
6. Modelling. 7.Prediction

B. IMAGE PROCESSING
1. Coding. 2.Enhancement. 3.Restoration and reconstruction. 4.Biomedical processing

C. SPEECH PROCESSING
1.Coding. 2.Synthesis. 3.Recognition and understanding. 4.Enhancement.
5. Aids for the handicapped

D. MULTIDIMENSIONAL SIGNAL PROCESSING
1. Array processing. 2.Digital transforms. 3.Digital filtering. 4.Geophysical and seismic processing

E. IMPLEMENTATIONS
1. Hardware. 2.Software. 3.VLSI. 4.Novel architectures

F. KNOWLEDGE ENGINEERING AND SIGNAL PROCESSING
1.Expert systems. 2.Pattern recognition. 3.Signal interpretation. 4.Neural networks

G. APPLICATIONS
1. Radar. 2.Sonar. 3.Communications. 4.Digital audio. 5.Sensing. 6. Robotics

The volumes contents are as follow:
Volume 1: Theory of Signals and Systems. Multidimensional Signal Processing
Volume 2: Image Processing. Speech Processing
Volume 3: Implementations. Knowledge Engineering and Signal Processing. Applications

We would like to thank all the participants of EUSIPCO-90, as well as the sponsor institutions and exhibitors. Without any of them, EUSIPCO-90 would not have had place. Elsevier Science Publishers B.V. (North-Holland) provided all the help and care we needed. Angela Noguera and Silvia Soriano from the Department of Signal Theory and Communications, Barcelona, did an outstanding job in clerical work. Our gratitude to them.

Although many times we asked ourselves "why did we accept to organize EUSIPCO-90?", we have been very pleased to accept this commitment and hope these Proceedings will be helpful to the Signal Processing Community.

Barcelona, September 1990

Luis Torres
Enrique Masgrau
Miguel A. Lagunas

CONFERENCE COMMITTEE

M. A. Lagunas
Chairman

J. Fernández
A. Figueiras
A. Gasull
J.B. Mariño
E. Masgrau
A. Moreno
C. Nadeu
L. Torres
F. Vallverdú
G. Vázquez

SCIENTIFIC COMMITTEE

ACHA, J.
ALMEIDA, L.B.
AMENGUAL, M.
BAJON, J.
BELLANGER, M.
BERTRAN, M.
BIEMOND, J.
BOHME, J.
BOITE, R.
BRACCINI, C.
CARAYANNIS, G.
CASACUBERTA, F.
CASALS, A.
CASAR, J.R.
CASTANIE, F.
CISNEROS, G.
CLARKE, R.J.
COLLA, A.M.
CONSTANTINIDES, A.G.
CORTELAZZO, G.
COWAN, C.F.N.
DOCAMPO, D.
DUBUISSON, B.
ELIAS, A.
FARINA, A.
FARRIER, D.
FETTWEISS, A.
FIGUEIRAS, A.R.
FLANDRIN, P.
GALAND, C.
GARCIA, R.
GARCIA, N.
GAUDENZI, R.
GILGE, M.
GLANGEAUD, F.
GRANLUND, G.H.
GRIFFITHS, J.W.R.
HEES, W.
HEUTE, U.
HOFFMANN, J.C.
JUTTEN, J.C.
KUNT, M.
LABARTA, J.
LACAZE, B.
LACOUME, J.L.
LLABERIA, J.M.
LUCAS, R.
MACCHI, O.
MARCOS, S.
MARTHON, M.
MARTIN, N.
MARTINEZ, J.A.
MECKLENBRAUKER, W.
MUÑOZ, C.
NEUVO, Y.
NIEMANN, H.
NOLL, P.
ORTIGUEIRA, M.
PARDO, J.M.
PICINBONO, B.
RENDAS, M.Y.
RIBEIRO, I.
ROCCA, F.
ROUX, C.
RUBIO, A.J.
SALLENT, S.
SAN FELIU, A.
SANTOS, J.
SANZ, A.
SCAGLIOLA, C.
SCHUSSLER, H.W.
SECILLA, J.P.
SICURANZA, G.L.
SMITH, S.G.
THEODORIDIS, S.
TORRAS, C.
TRANCOSO, I.
VARY, P.
VENTURA, J.
VICENZI, C.
VIDAL, E.
VIOLA, R.
WOLF, D.

ORGANIZATION

Organizer:

European Association for Signal Processing
EURASIP

Sponsors

Dirección General de Investigación Científica y Técnica
Ministerio de Educación

CIDEM Centre d'Informació i Desenvolupament Empresarial
Generalitat de Catalunya

UPC. Universitat Politècnica de Catalunya

CIRIT
Generalitat de Catalunya

IBM

CESELSA

Grupo AMPER

TELETTRA España, S.A.

PAGE IBERICA, S.A.

COMELTA, S.A.

DIGITAL Equipment Corporation. España

TID. Telefónica Investigación y Desarrollo

ELSEVIER- Science Publishers B.V.

HEWLETT PACKARD

EXHIBITORS

BOIXAREU Editores, S.A.
COMELTA, S.A.
CONTROL SYS
DIGIMETRIE
ELSEVIER- Science Publishers B.V.
FUJITSU
HORIZON TECHNOLOGIES
LOUGHBOROUGH SOUND IMAGES Ltd.
NEC Electronics
OROS
SGS - Thomson
SIGNAL TECHNOLOGY Inc.
TEXAS INSTRUMENTS
VTE Digitalvideo GmbH

TECHNICAL PROGRAM SCHEDULE

Tuesday, September 18

Time	Room 7	3	5a	5b	6	8	4
8h 40 9h 10 9h 15	Opening Ceremony						
10h 35 11h 00	L1 Speech Enhancement	L2 Time Delay Estimation	L3 Signal Enhance. and noise Reduction	L4 Geophys. and Seismic Process.	L5 Novel Architec.	L6 Sensing/ Robotics	P1 Image Sequence Coding
13h 00							
14h 40 15h 10		Tutorial 1			Tutorial 2		
15h 15 16h 55 17h 15 18h 15		L7 Neural Nets I	L8 Speech Synthes.	L9 Time-Frequen. Signal Analysis	L10 Radar Signal Process. I	L11 Image Enhance. and Restorat.	P2 Communications I

Wednesday, September 19

Time	Room 3	5a	5b	6	8	4
8h 40 9h 10	Tutorial 3			Tutorial 4		
9h 15 10h 35 11h 00	L12 Adaptive Filtering I	L13 Image Coding	L14 Speech Coding	L15 Underwat. Acoustics	L16 VLSI Implementation	P3 Communications II P4 Pattern Recognit./ Signal Interpr.
13h 00						
14h 40 15h 10	Tutorial 5			Tutorial 6		
15h 15 16h 55 17h 15 18h 15	L17 Neural Nets II	L18 Speech Recognit. in Noise	L19 Echo Cancelling and Deconvol.	L20 Estimation and Identification	L21 Transform Image Coding	P5 Hardware Implem./ Software Tools for VLSI

Thursday, September 20

Time	Room 3	5a	5b	6	8	4
8h 40 9h 10	Tutorial 7			Tutorial 8		
9h 15 10h 35 11h 00	L22 Speech Coding I	L23 Adaptive Filtering II	L24 Array Process.: Spatial-Spectrum Estimat.	L25 Biomedical Image Processing	L26 Dynamic Scene Analysis	P6 Speech Recognition I P7 Hardware and Software for DSP
13h 00						
14h 40 15h 10	Tutorial 9			Tutorial 10		
15h 15 16h 55 17h 15 18h 15	L27 Array Process.	L28 Knowled. Engineer.	L29 Acoustic Echo Control	L30 Speech Processing	L31 Detection	P8 Image Process. and Machine Vision

Friday, September 21

Time	Room 3	5a	5b	6	8	4
8h 40 9h 10	Tutorial 11					
9h 15 10h 35 11h 00	L32 Line Detection	L33 Speech Coding II	L34 Modelling	L35 VLSI for Multidimensional Signal Processing	L36 Speech Recognition II	P9 Multidimens. Filtering P10 Adaptive Filtering III
13h 00						
14h 40 15h 10						
15h 15 16h 55 17h 15 18h 15	L37 Array Process: Adaptive Beamform.	L38 Spectral Estimat.	L39 Radar Signal Process. II	L40 Vector Quantizat. Image Coding	L41 Modelling/ Signal Theory	P11 Filtering

L = Lecture P = Poster

VOLUME I

OPENING SESSION

THEORY OF SIGNALS AND SYSTEMS

TIME - DELAY ESTIMATION

SIGNAL ENHANCEMENT AND NOISE REDUCTION

TIME - FREQUENCY SIGNAL ANALYSIS

ADAPTIVE FILTERING I

ESTIMATION AND IDENTIFICATION

ADAPTIVE FILTERING II

DETECTION

LINE DETECTION

MODELLING

ADAPTIVE FILTERING III

MODELLING / SIGNAL THEORY

FILTERING

MULTIDIMENSIONAL SIGNAL PROCESSING

GEOPHYSICAL AND SEISMIC PROCESSING

ARRAY PROCESSING: SPATIAL-SPECTRUM ESTIMATION

MULTIDIMENSIONAL FILTERING

ARRAY PROCESSING: ADAPTIVE BEAMFORMING

VOLUME II

IMAGE PROCESSING

IMAGE SEQUENCE CODING

IMAGE ENHANCEMENT AND RESTORATION

IMAGE CODING

TRANSFORM IMAGE CODING

BIOMEDICAL IMAGE PROCESSING

DYNAMIC SCENE ANALYSIS

IMAGE PROCESSING AND MACHINE VISION

VECTOR QUANTIZATION IMAGE CODING

SPEECH PROCESSING

SPEECH ENHANCEMENT

SPEECH SYNTHESIS

SPEECH ANALYSIS

SPEECH RECOGNITION IN NOISE

SPEECH CODING I

SPEECH RECOGNITION I

SPEECH PROCESSING

SPEECH CODING II

SPEECH RECOGNITION II

VOLUME III

IMPLEMENTATIONS

NOVEL ARCHITECTURES

VLSI IMPLEMENTATION

HARDWARE IMPLEMENTATION / SOFTWARE TOOLS FOR VLSI

HARDWARE AND SOFTWARE FOR DSP

VLSI FOR MULTIDIMENSIONAL SIGNAL PROCESSING

KNOWLEDGE ENGINEERING AND SIGNAL PROCESSING

NEURAL NETS I

PATTERN RECOGNITION AND SIGNAL INTERPRETATION

NEURAL NETS II

KNOWLEDGE ENGINEERING

APPLICATIONS

SENSING/ROBOTICS

RADAR SIGNAL PROCESSING I

COMMUNICATIONS I

UNDERWATER ACOUSTICS

COMMUNICATIONS II

ECHO CANCELLING AND DECONVOLUTION

ACOUSTIC ECHO CONTROL

RADAR SIGNAL PROCESSING II

AUTHOR INDEX

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Optimizing the CORDIC Algorithm for Processors with Pipeline Architecture

D. König and J.F. Böhme

Department of Electrical Engineering, Ruhr University Bochum, 4630 Bochum, West Germany

The CORDIC algorithm is known being able to calculate a variety of functions including trigonometric and hyperbolic functions as well as multiplication and division. In this paper we investigate a CORDIC processor with pipeline architecture. We present an algorithm that optimizes the parameters of this pipeline structure in order to achieve minimal hardware amount and latency time. Some results of this optimization procedure are given. We also discuss some numerical properties of the CORDIC algorithm.

1. Introduction

The CORDIC algorithm has been developed by Volder [1] in 1959 to calculate iteratively magnitude and phase of a two component vector in Cartesian coordinates or to rotate such a vector by a given angle. Because the iterations require only shift-add operations, they can be carried out easily. Walther [2] introduced generalized coordinates enabling the CORDIC method to calculate a larger set of elementary functions which include trigonometric and hyperbolic functions as well as arithmetic operations like multiplications and divisions.

This variety of operations recommends the use of CORDIC processors in computer graphics, digital signal processing and linear algebra. In particular, millions of plane rotations, often combined with multiplications and additions, have to be executed in real time signal processing. The demand for high computational speed can be satisfied by a CORDIC processor with pipeline architecture. The iterative structure of the CORDIC algorithm permits a pipeline implementation leading to a very high data throughput [3].

2. The CORDIC Algorithm

The CORDIC algorithm is defined by iterative shift-add operations on a three-component vector,

$$\begin{bmatrix} x_{i+1} \\ y_{i+1} \end{bmatrix} = \begin{bmatrix} 1 & -m\sigma_i 2^{-S(i)} \\ \sigma_i 2^{-S(i)} & 1 \end{bmatrix} \cdot \begin{bmatrix} x_i \\ y_i \end{bmatrix}, \tag{1}$$

$$\begin{aligned} z_{i+1} &= z_i - \sigma_i \varepsilon m^{-1/2} \arctan(m^{1/2}\, 2^{-S(i)}) \\ &= z_i - \sigma_i \varepsilon \alpha_{m,i} \qquad (i = 0,1,\ldots,N-1), \end{aligned} \tag{2}$$

where the shift parameters $\{S(i)\}$ are integers, $m \in \{-1,0,1\}$ is the coordinate system parameter, $\sigma_i \in \{-1,1\}$ is the direction of the rotation, and $\varepsilon \in \{-1,1\}$ is the sign of the CORDIC operation. Equation (1) is called microrotation and can be interpreted as a pseudo-rotation of $(x_i, y_i)'$ by the angle $\alpha_{m,i}$ and an additional scaling by

$$k_{m,i} = (1+m2^{-2S(i)})^{1/2}. \tag{3}$$

Executing N iterations leads to

$$\begin{bmatrix} x_N \\ y_N \end{bmatrix} = K_m \begin{bmatrix} \cos(m^{1/2}\alpha_m) & -m^{1/2}\sin(m^{1/2}\alpha_m) \\ m^{-1/2}\sin(m^{1/2}\alpha_m) & \cos(m^{1/2}\alpha_m) \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \end{bmatrix},$$

$$z_N = z_0 - \varepsilon\alpha_m, \tag{4}$$

with scaling factor K_m and macrorotation angle α_m,

$$K_m = \prod_{i=0}^{N-1} k_{m,i}, \tag{5}$$

$$\alpha_m = \sum_{i=0}^{N-1} \sigma_i \alpha_{m,i}. \tag{6}$$

If the angles $\alpha_{m,i}$ satisfy the convergence condition,

$$\alpha_{m,i} - \sum_{j=i+1}^{N-1} \alpha_{m,j} \le \Delta\alpha \qquad (i=0, 1, \ldots, N-2), \tag{7}$$

where $\Delta\alpha$ is a given angle resolution, the choice of $\sigma_i = -\text{sign}\,(x_i y_i)$ or $\sigma_i = \text{sign}\,(\varepsilon z_i)$ forces $y_N \to 0$ or $z_N \to 0$, respectively, for all input data of the region of convergence,

$$C_m = \sum_{i=0}^{N-1} \alpha_{m,i} \ge \begin{cases} |m^{-1/2} \arctan(m^{1/2} y_0/x_0)| & \text{for } y_N \to 0 \\ |z_0| & \text{for } z_N \to 0 \end{cases}. \tag{8}$$

3. Compensation of the Scaling Factor

A disadvantage of the CORDIC algorithm is the undesirable scaling of the output data. Using multiplications by K_m^{-1} would reduce the effeciency of the CORDIC

method. Therefore, different approaches for compensating the scaling factor with shift-add operations have been proposed.

Some authors investigated a repetition of several microrotations. In this way, the scaling factor can be approximated by a power of two. Thus, the compensation of the scaling factor can be carried out by simply shifting the output data. To reduce the number of supplementary microrotations Deprettere [3] suggested double-shift microrotations which offer additional degrees of freedom for optimizing {S(i)}. This concept requires, however, single-shift cells as well as double-shift cells, thus leading to an irregular chip architecture. Therefore, we disregard this concept in the sequel.

Following Haviland and Tuszynski [4], we propose the use of scaling iterations, described by

$$x'_N = G_m^{-1}\, x_N\,, \tag{9}$$

$$y'_N = G_m^{-1}\, y_N \tag{10}$$

with

$$G_m^{-1} = 2^{-T(m,0)} \cdot \prod_{j=1}^{NS} \left[1+\eta(m,j)\cdot 2^{-T(j)} \right] \cong K_m^{-1}. \tag{11}$$

This scaling operation can be carried out by simple shift-add operations like the microrotations. Obviously, the imperfect compensation of the scaling factor produces a residual error, that is characterized by the following quantity representing the erroneous bit,

$$\Delta K_m = \operatorname{sign}\left[\frac{K_m}{G_m} - 1 \right] \cdot \log_2 \left| \frac{K_m}{G_m} - 1 \right| . \tag{12}$$

4. Pipeline Architecture of the CORDIC Processor

The CORDIC algorithm combined with the scaling operation described before is well suited for a pipeline implementation (see Fig. 1) that contains two main components.

- The macrorotation is performed by a chain of N microrotation cells. Because S(i) does not depend on m, simple hardwired shifts are required.
- The scaling operation is executed by a short chain of cells having complexity and computation time similar to those of the microrotation cells. This is a great simplification when comparing our concept with the double-shift one.

In addition we need two supplementary components.

- The initial stage converts the external input data to the internal data format. For numerical reasons, this internal data format differs from the external one by overflowbits and guardbits. Additionally, the initial stage performs an initial rotation by $\pi/2$ for $m = 1$. This rotation can be achieved by a small amount of hardware and extends the convergence region significantly.
- The conversion of the output data into the external data format is executed in the final stage of the CORDIC pipeline.

5. The Optimization Problem

The CORDIC algorithm described above is determined by a set of parameters,

- the shifts of the microrotations S(i), $i \in \{0,1,...,N-1\}$,
- the scaling shift T(m,0), $m \in \{-1,0,1\}$,
- the signs of the scaling iterations $\eta(m,j)$, $m \in \{-1,0,1\}$, $j \in \{1,...,NS\}$ and
- the scaling shifts T(j), $j \in \{1,...,NS\}$.

In the sequel, we call a realization of this set of parameters a CORDIC sequence.

We now describe a new algorithm for searching optimum CORDIC sequences for some data formats of interest. The optimization procedure is based on minimizing

$$NG = N + NS\,, \tag{13}$$

over the set of parameters {S(i), T(m,0), η(m,j), T(j)}.

The reason to use this optimization criterion is that NG is a direct measure for both, the hardware amount and the latency time of the pipeline processor. Therefore, the problem is to find CORDIC sequences for a given data format with minimum NG.

The dimension of the parameter space is extremely large for most data formats. Searching for optimum CORDIC sequences then is an enormous combinatorial problem. Fortunately, the parameter space is restricted by the following requirements.

- The angles of the microrotations as a function of the shifts S(i) have to satisfy the convergence condition given by (7).
- Without loosing optimum CORDIC sequences, we demand

  $S(i) \le S(j)$, $T(i) \le T(j)$ for $i \le j$, $(i,j = 0,...,N)$.

- To guarantee the required angle resolution, the shift of the last microrotation should satisfy

  $S(N-1) \ge NBit$,

  where NBit is the mantissa length of the external data format.

- In trigonometric applications ($m = 1$), the region of convergence often has to cover $[-\pi, \pi]$. Therefore, considering an initial rotation by $\pi/2$ performed in the initial stage, we demand

  $C_1 \ge \pi/2$.

- The influence of the scaling error on the output data in the x-path and the y-path of the CORDIC processor has to be kept within $\pm 1/2$ LSB. This condition is met by the choice of

  $|\Delta K_m| \ge NBit + 1$.

Our optimization algorithm considers all these requirements. The strongest restriction of the parameter space results from the convergence condition. Therefore, we begin with the shift of the last microrotation S(N–1). While constructing all possible sets of shifts in the order S(N–2), ..., S(0), we test the convergence condition. Thus, we only get those angle sequences that satisfy the convergence condition.

For every set of shifts constructed in this way, the resulting scaling factors are computed. Considering (11), we try to compensate these scaling factors with a minimum number of scaling iterations.

6. Results of Optimizations

The algorithm described before has been implemented on a VAX Computer. The program generates CORDIC sequences for all data formats of interest. In this section, we discuss two examples dedicated to 24 mantissa bits. Using the first sequence given in Table 1, all CORDIC functions ($m \in \{-1,0,1\}$) can be calculated. This CORDIC sequence requires the minimum number of NG = 37 microrotation and scaling cells.

$\{S(i)\}$ = {1,2,2,2,2,3,4,5,5,6,6,7,8,9,10,11,12
13,13,14,15,...,24}

$\{T(m,0)\}$ = {0,0,0}, $m \in \{-1,0,1\}$

$\{\eta(-1,j)\}$ = { 1, 1, 0, 1,-1, 0,-1,-1}
$\{\eta(\ 0,j)\}$ = { 0, 0, 0, 0, 0, 0, 0, 0}
$\{\eta(\ 1,j)\}$ = {-1, 1,-1, 1, 0, 1,-1, 0}

$\{T(j)\}$ = { 2, 4, 5, 6, 6,17,20,21}

Table 1: CORDIC sequence for 24 mantissa bits, NG = 37, ΔK_{-1}=–25.36 bit, ΔK_1=–28.06 bit

Another CORDIC sequence is shown in Table 2. In contrast to the first one, it is dedicated to calculate linear and trigonometric CORDIC functions ($m \in \{0,1\}$). This CORDIC sequence requires only NG = 31 cells.

$\{S(i)\}$ = {1,1,2,3,3,4,5,5,6,6,7,8,9,10,...24

$\{T(m,0)\}$ = {0,0}, $m \in \{0,1\}$

$\{\eta(\ 0,j)\}$ = { 0, 0, 0}
$\{\eta(\ 1,j)\}$ = {-1, 1, 1}

$\{T(j)\}$ = { 2, 6,17}

Table 2: CORDIC sequence for 24 mantissa bits, NG = 31, ΔK_1=–29.16 bit

Additionally, we present two results for a processor with 16 mantissa bits.

$\{S(i)\}$ = {1,2,2,2,2,3,4,5,5,6,7,8,9,10,11,12
13,13,14,15,16}

$\{T(m,0)\}$ = {0,0,0}, $m \in \{-1,0,1\}$

$\{\eta(-1,j)\}$ = { 1, 1, 0, 0, 1}
$\{\eta(\ 0,j)\}$ = { 0, 0, 0, 0, 0}
$\{\eta(\ 1,j)\}$ = {-1, 1,-1,-1, 1}

$\{T(j)\}$ = { 2, 4, 6,10,13}

Table 3: CORDIC sequence for 16 mantissa bits, NG = 26, ΔK_{-1}= 19.66 bit, ΔK_1= 20.97 bit

$\{S(i)\}$ = {0,1,2,3,4,5,6,7,8,9,10,...,24}

$\{T(m,0)\}$ = {0,1}, $m \in \{0,1\}$

$\{\eta(\ 0,j)\}$ = { 0, 0, 0, 0, 0}
$\{\eta(\ 1,j)\}$ = { 1,-1, 1, 1, 1}

$\{T(j)\}$ = { 2, 5, 9,10,16}

Table 4: CORDIC sequence for 16 mantissa bits, NG = 22, ΔK_1= 23.05 bit

7. Numerical Properties of the CORDIC algorithm

Obviously, the numerical behaviour of a CORDIC realization is important. We simulated the processor architecture described before. In this section, we present some results.

7.1. Arithmetic Concepts

The use of floating-point arithmetic instead of fixed-point arithmetic improves the accuracy of the output data considerably. Implementing floating-point arithmetic in a pipeline architecture requires postnormalization of the data after each add/sub-operation and therefore leads to a significantly increased hardware amount. For this reason, Yang [5] has investigated the implementation of a modified floating-point arithmetic. Using this concept, the input data and the output data of the CORDIC processor are floating point numbers. After an exponent alignment carried out in the initial stage, the microrotations and the scaling iterations are calculated using fixed-point arithmetic. Compared with pure fixed-point arithmetic, the modified floating-point arithmetic has several advantages, for example the region of convergence is no longer limited for m = 0. Our simulations are based on modified floating-point arithmetic.

7.2. Number of Guardbits and Rounding Procedures

For internal operations, the external data format has to be extended by some guardbits to guarantee the accuracy of the output data. This is a well-known approach [2]. We use five guardbits that are sufficient for all tested data formats up to 24 mantissa bits.

Additionally, a proper rounding procedure is important when converting the internal data format to the external one. Fig. 2 shows that rounding to nearest provides a symmetric error distribution without bias and of small variance. Therefore, this rounding procedure should be implemented in the final stage of the CORDIC processor.

7.3. Scaling Accuracy

We have mentioned the residual error when compensating the scaling factor. Fig. 3 illustrates the influence of this scaling error on the accuracy of the output data. Obviously, the scaling error produces a bias of the error distribution leading to bad results when repeating several CORDIC operations (see Fig. 4). In order to achieve an approximately bias free error distribution, it is necessary to compensate the scaling factor with high accuracy. Therefore we recommend CORDIC sequences with

$$|\Delta K_m| - \mathrm{NBit} \geq 5.$$

For certain data formats, this condition can be met without additional hardware amount. In other cases, one or two supplementary iteration cells are required.

8. Conclusions

The CORDIC method well suits to a pipeline implementation. In this paper, we have investigated a pipeline processor architecture. In contrast to former publications, we avoid double-shift microrotations. An algorithm that

optimizes the CORDIC parameters and results of optimizations have been presented. The modular design with simple cells simplifies the implementation on a chip without leading to an increased latency time of the pipeline. We have also discussed the numerical properties of the CORDIC method. We propose to implement rounding to nearest in the final stage of a CORDIC processor and underline the importance of small scaling errors.

References

[1] Volder, J.E.: The CORDIC Trigonometric Computing Technique, IRE Trans. (1959), Vol. EC–8, No. 3, pp. 330–334

[2] Walther, J.S.: A Unified Algorithm for Elementary Functions, Proc. SJCC (1971), pp. 379–385

[3] Deprettere, E.F. et al.: Pipelined CORDIC Architectures for Fast VLSI Filtering and Array Processing, Proc. ICASSP (1984), pp. 41A.6.1–41A.6.4

[4] Haviland, G.L. and Tuszynski, A.A.: A CORDIC Arithmetic Processor Chip, IEEE Trans. on Computers 29(2), pp. 68–79, 1980

[5] Yang, B., Timmermann, D. et al.: Special Computers: Graphics, Robotics, Proc. CompEuro (1987), pp. 727–730

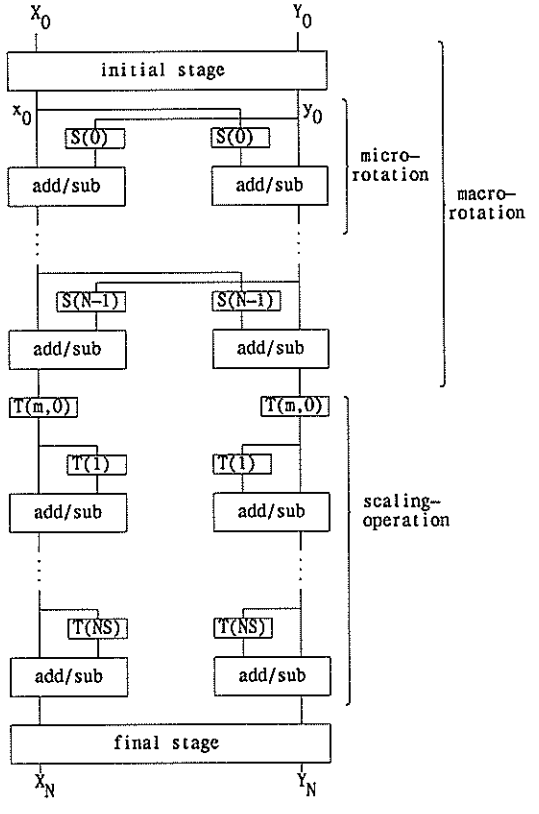


Fig. 1: Pipeline architecture of a CORDIC processor, z–path is not shown

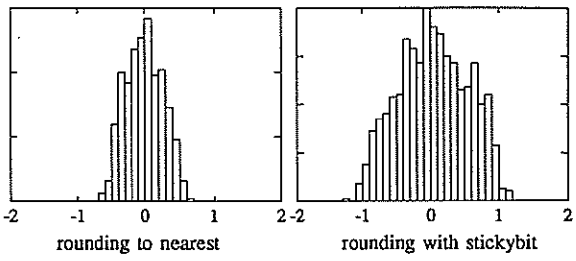


Fig. 2: Error distribution of $\sqrt{X_N^2 + Y_N^2}$ for different rounding procedures ($m = 1$, $z \to 0$)

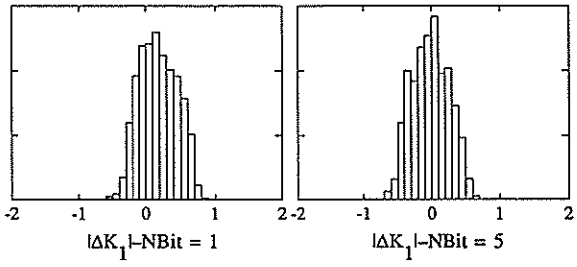


Fig. 3: Error distribution of $\sqrt{X_N^2 + Y_N^2}$ for different scaling errors ($m = 1$, $z \to 0$)

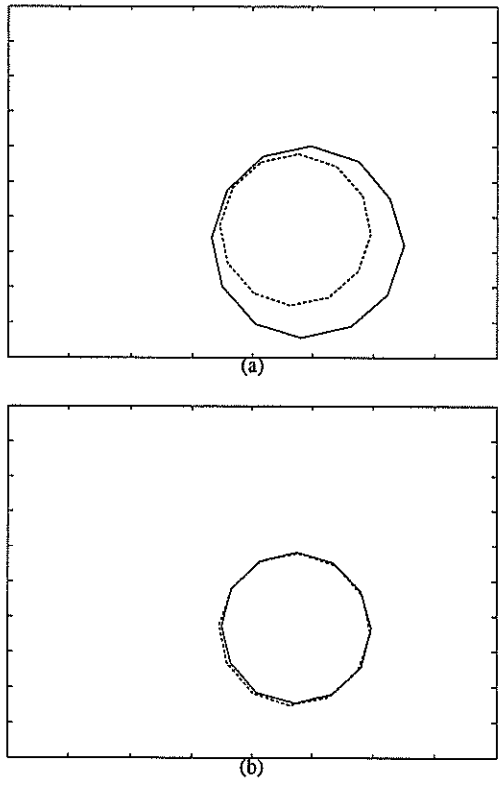


Fig. 4: Influence of scaling error after 2000 CORDIC rotations a) $|\Delta K_1|$–NBit=1, b) $|\Delta K_1|$–NBit=5

– – – ideal result

—— CORDIC result using 12 mantissa bits

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

MULTICOMPUTER FOR PARALLEL PROGRAMMING OF DIGITAL SIGNAL PROCESSING ALGORITHMS

José Parera, Rafael Sarmiento, Juan Santos, Marcelino Veiga
E.T.S.I.Telecomunicación. Polytechnic University of Madrid, Spain
Department of Signals, Systems and Radiocommunications

This paper describes a multicomputer system aimed to general purpose Digital Signal Processing applications. The hardware architecture is based on a binary n-cube topology with specialized processing nodes and a high speed interconnection network. We focus our attention on the interconnection network since this kind of architecture belongs to the message-passing concurrent computer family. Routing schemes for both point to point and broadcast messages as well as data flow control are explained, emphasizing hardware solutions.

1. INTRODUCTION

The purpose of this contribution is to present the development of a multicomputer system aimed to general purpose digital signal processing. This system attempts to make easier the implementation of complex real-time applications; this complexity arising from either the amount of data to be processed or the operations involved in the algorithm.

The design of the system is based on the following main ideas:

* Interconection network with high speed communication channels.

* Specialization of the processing nodes.

* Flexibility to tune the system to application requirements.

* Easy development of applications.

This system is not conceived only as a machine capable of executing applications, but as a true development environment; therefore we specially emphasize programming and debugging aspects as well the monitorization and evaluation of the applications generated. The objective is to get a system that allow the user to develop complex applications in a rapid and easy way.

2. OVERVIEW

The hardware architecture is based on a binary n-cube topology with self-controlled parallel links that perform data routing without intervention of node processors.

Each node has at least two processors: a network controller processor (network manager) and a DSP processor. The network manager is in charge of message sending and receiving, host communication and event recording. In addition, it can collaborate with the other processor in the execution of the DSP algorithm.

Freeing the DSP processor from the tasks related with the network, allows us to concentrate its computational power in the DSP algorithm.

Both processors are part of two independent subsystems (with their own buses, memories and control) and they communicate each other through a standard bus. Other subsystems can be attached to this bus in order to cope with the requirements of each node for the execution of the algorithm. These subsystems could be: shared memory, A/D and/or D/A converters, mass storage devices interfaces, graphics controllers, etc. This way, the system can be tuned to the application, but keeping the nucleus, which consists of the network managers and interconection network, with an homogeneus structure.

The software architecture is split in two big blocks: basic software and CAD/CAE tools.

The basic software is distributed among the host and the multicomputer, while the CAD/CAE tools reside completely in the host.

The main components of the basic software are: real-time operating systems in the node processors, inter and intranode communication package, host communication package, distributed symbolic debugger, event recorder, test package and function library.

While an application is running on the multicomputer, the host has two main tasks assigned to it: the multicomputer control and an operating system service. By means of this operating system service, nodes can use host resources such as: terminal, file system, external network access, etc and/or execute some process. This service allows the host to collect information about the multicomputer with which, after an off-line elaboration, the development of the algorithm can be reproduced. By studying the evolution of the algorithm one can observe the adequation of the tasks to nodes assignment, hot spots in CPUs utilization or in message traffic through the network, etc.

This feedback makes easier the redistribution of tasks in order to obtain a good computational load balancing in the nodes.

Finally, we want to point out that among the CAD/CAE tools a block diagram compiler stands

out. This compiler generates applications with the graphic language usually used by DSP engineers.

A system with the above-mentioned features is referred as a *message-passing concurrent computer*. This naming emphasizes a fundamental aspect of the architecture: the ability to send and receive messages through the interconection network.

This architectural element is specially important if we consider the timing restrictions imposed by real-time DSP algorithms. Though, our objective is to get an interconection network whose performance allow us to reduce to a minimum the time spent to communicate messages.

3. MESSAGES

The described architecture presents a loosely coupling between nodes, though messages are, from the application point of view, the information exchange units between tasks executing on different nodes, as well as the basic mechanism for synchronizing them.

Messages can be classified on two basic types: point to point or broadcast. Messages point to point are generated in one node and have a destination in a different node; those of broadcast type are generated in one node and must reach all the remaining nodes in the network.

Both types of messages are handled automatically by a subsystem named channel controller so that the node processors are involved only when the message has reached his destination. The network is smart enough to decide the message type, to establish the routing path and to notify the network managers.

Messages have variable length, limited only by the amount of memory available at the nodes, so a mechanism must be established so that it allows the channel controller detect the end of a message.

As we will see, a message generated by an application must be delimited by two hardware control structures (header and tail). This control structures are named flits (flow control units).

Any node can generate messages to any other node or to the rest of nodes. Since the network resources are shared and limited, a dead-lock free routing algorithm has to be used. The algorithm chosen for the point to point messages is called *e-cube*.

This algorithm establishes a path between the source and destination nodes such as the message flows through channels that change the most signifyint bit of the difference from the actual address and the destination address. So, a message generated in node 1111 destined to node 0010 will follow the path:

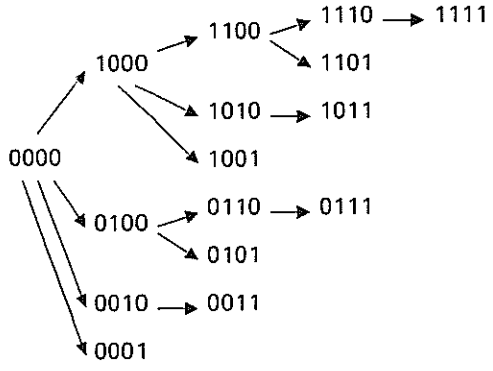
$$1111 \longrightarrow 0111 \longrightarrow 0011 \longrightarrow 0010$$

This routing algorithm is efectively dead-lock free if two conditions are met: a) a message reaching his destination must be consumed, that is, it must free all the network resources assigned to it; and b) once a channel has accepted the header flit of a message, the channel remain assigned to this message until the tail flit arrives.

The *e-cube* algorithm presents some aditional advantages from the point of view of the hardware implementation: it is deterministic and memory-free, in the sense that the path depends only of the destination address.

On the other hand, broadcast messages coexist in the network with point to point messages. The broadcast routing algorithm must not conflict with the *e-cube* algorithm in order to prevent a dead-lock. So, the path followed by broadcast messages has to be compatible with the path followed by point to point messages. The simplest way to acomplish this requirement is to copy the broadcast message in so many messages as the dimension of the net and to route each of these messages through channels that expand complete subcubes, as the message traverses nodes.

This way, a broadcast message generated in node 0000 in a four-dimensional network will be distributed as follows:



It can be seen that all the messages produced by the splitting of the original message follows the same path that a point to point message would follow.

In order to reduce the time spent in sending a broadcast message the channel controller accepts the message from the network manager simultaneously in all channels, so the copy above-mentioned is only conceptual.

There are three main differences in the way the network handles broadcast messages:

a) The routing scheme has memory, in the sense that the channel controller of a node must know what subcubes are to be expanded from the node. This information must appear in the header flit and dinamically be modified as the message traverses the nodes.

b) Broadcast type messages must be consider both destination and transit messages (except at terminal nodes where they are destination only)

and must be simultaneously routed to the network manager and to the selected output channels.

c) Two or more input channels of a node can handle broadcast messages and then request two or more output channels (perhaps the same channels). In this last case a new dead-lock condition can arise. The simplest solution is to implement a prioritized mechanism for assign output channels.

Although net homogeneity seems to be lost due to this priority assignment, it must be considered that the priority of a path between two nodes is modified in the output channel of each node. Priority matches two conditions: a) never decreases; and b) increases as distance to the destination decreases.

4. CHANNELS

A channel consist of a transmitter, a receiver and a physical link. The network manager sees transmitters and receivers as peripherals shared with the network.

Receiver access must be performed only when the channel controller notify to the network manager a message arrival. In order to access a transmitter, a request must be sent. This request will be inmediately granted if it was idle or as soon as the transmitter leave the busy state provided that there is not another higher priority request.

The channel controller is in charge of arbitrating the network resources including blocking, buffering and, of course, forwarding of messages.

The method of message forwarding used, named *wormhole*, resembles circuit switching. So that, once a message path has been established it remains assigned to this message until the tail flit liberates it. Although, the way in which a cuircuit is established and liberated is different from circuit switching.

With *wormhole* method, the header flit opens the path as it can be forwarded, that is to say, if the network resources needed by the message are free. When a resource has been accesed the message begin to flow, becoming blocked in case of the header flit cannot continue. Blocking is only performed in the net without freeing any resource assigned to the message at that moment. This way, no buffering is needed on intermediate nodes; nonetheless, by means of a blocking memory inserted on every channel, the tail flit can advance thus liberating resources more quickly.

We can see that the path is opened and closed dinamically controlled by the messages flits.

From a physical point of view, a message is composed of some number of bytes. So, the link and temporary buffering are 8 bit wide. Aside to the data path exist four control lines that allow channel controllers knowing the state of the link at every moment in order to correctly regulate information flow. By means of these control lines, each node is capable of synchronizing with neighbour nodes since every node in the system works asynchronously.

Communication speed is limited by several factors including: device technology, propagation delays and, mainly, protocols. The protocol that we are using allows sending data without need of acknowledge.

Let us suppose an opened message path with m nodes involved. Each datum is propagated directly through the path from the origin to the destination nodes in only one clock cycle. When node m will not be able to accept more data, it will signal the preceding node which will begin buffering. This process continues until node m can accept again more data or buffer memory is exhausted. In this last case, node m-1 signals node m-2 which begins buffering, and so on.

This mechanism permits a continous flow of data from the origin to the last unblocked node, since block signaling is performed on a neighbour node basis.

No error control is provided at the hardware level; so, we must assure a reliable communication link in an electromagnetic noisy environment. In order to cope with this requirement, the physical link is composed of balanced driver and receiver pairs joined together with a terminated twisted pair cable.

5. CONCLUSIONS

The communication paradigm previously described requires a complex hardware implementation. Simple solutions have been adopted compared with *adaptive routing*, *virtual cut-through*, or other alternatives; nonetheless, these simple solutions are hardwired, obtaining this way higher communication speed than we could get with the other schemes that require software support.

Actually, hardware design is being simulated with a peak performance of 80 Mbits/s which we expect increase above 100 Mbits/s. Including software overhead, we expect a task to task communication speed of at least half the speed in the network. Those numbers are not too far from the performance achieved in bus oriented systems where the data path is usually 32 bits wide. It is important to realize that this performance can be obtained for any message path, no matter the number of nodes involved.

Hardware complexity demands a custom or PGA integration for both transmitter and receiver subsystems. This way we could increase speed easily by means of a smaller clock cycle and a wider data path.

6. REFERENCES

[1] *Interconnection Networks for large scale parallel processing.*
H.J. Siegel. Lexington Books, 1985.

[2] *An Interactive System for Analysis of Hypercube Message Passing Performance.*
Alva L. Couch et al. Report of the Department of Computer Science (Tufts University) 1987.

[3] *Multi-Microprocessors Systems for Real-Time Applications*
G. Conte y D. del Corso Editors. Reidel Publishing Company. Dordrecht, 1985.

[4] *Solving Problems on Concurrent Processors*
G. Fox et al. Prentice-Hall International, Inc. Englewood Cliffs, New Jersey,1988

[5] *Multicomputers: Message Passing Concurrent Computers*
W. Athas y C. Seitz, IEEE Computer, Vol. 21, nº8, August 1988, pp. 9-24

[6] *Deadlock-free Message Routing in Multiprocessor Interconnection Networks*
W. Dally y C. Seitz, IEEE Transactions on Computers, Vol. 36, nº5, May 1987, pp. 547-553

[7] *Graphical Representations of Program Performance on Hypercube Message Passing Multiprocessors*
A. Couch, Ph.D. Dissertation, Department of Computer Science, Tufts University (Boston)

[8] *Data Communications on Hypercubes*
Y. Saad y M. Schultz, Journal of Parallel and Distributed Computing, Vol 6, nº1, February 1989

[9] *Evaluating the Performance of Multicomputer Configurations*
D. Agrawal y V. Janakiram, IEEE Computer, May 1986

[10] *The Third Conference on Hypercube Concurrent Computers and Applications*
ACM Press, 1988

[11] *VLSI Mesh Routing Systems*
C. Flaig, Report 5241:TR:87. Department of Computer Science. California Institute of Technology.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A FLEXIBLE LOW-POWER DIGITAL SIGNAL PROCESSOR BASED ON A CONTENT-ADDRESSABLE MEMORY

M. Ansorge, U. Sjöström, I. Defilippis, P. Balsiger, F. Pellandini

Institut de Microtechnique , Université de Neuchâtel, Rue A.-L. Breguet 2
CH-2000 NEUCHATEL, Switzerland

Abstract : This paper addresses the problem of implementing Linear Time Invariant DSP algorithms on serial-parallel processors. An improved processor architecture is proposed which is programmable and efficient with respect to the power consumption and the computation throughput.

1. Introduction

A new serial-parallel processor architecture for the implementation of Linear Time Invariant Digital Signal Processing (LTI DSP) algorithms is presented in this paper. The processor is especially suitable for low-power applications requiring programmable filters with adjustable filter order and coefficient sets.

The overall performance is improved by a specific data compression technique. Redundant computations are avoided using a data encoder embedded in a bit-serial Content-Addressable Memory (CAM) [1, 2]. The reduction of the computations is obtained by inspection of the data values, where only significant information (i.e. non-zero data bits) generate effective operations. The cyclic and deterministic scheduling in LTI DSP results in a considerable simplification of the CAM unit. As a consequence, the proposed architecture is well adapted to VLSI ASIC implementations due to its simplicity and structural regularity.

2. Basic Inner-Product Partitioning

Inner-product computation is the fundamental operation needed for LTI DSP processing. The N-th order inner-product defined in (1) is expressed in two's complement fixed point number representation. W_c and W_d are respectively defined as the coefficient and signal-data wordlength. By interchanging the summation order, expression (2) is derived, leading to a decomposition very similar to Distributed Arithmetic [3].

$$y = \sum_{i=0}^{N-1} a_i x_i = \sum_{i=0}^{N-1} \Big(\sum_{j=1}^{W_c-1} a_{ij} 2^{-j} - a_{i0} \Big) \cdot x_i \qquad (1)$$

$$y = \sum_{j=1}^{W_c-1} \Big(\sum_{i=0}^{N-1} a_{ij} x_i \Big) \cdot 2^{-j} - \sum_{i=0}^{N-1} a_{i0} x_i \qquad (2)$$

3. Reference Architectures

For comparison purposes, two reference architectures will be considered. The first one (Fig. 1) corresponds to a conventional serial-parallel implementation of expression (2). The inner-product processing is performed by a simple Shift-Accumulator (SA) consisting of a single adder/subtractor and a single step shifter. Obviously, the coefficient set a_i can easily be programmed and is stored in a compact $N \cdot W_c$ bit memory. Despite its simplicity, it should be noticed that this structure provides precisely computed results in comparison to other proposed architectures [4].

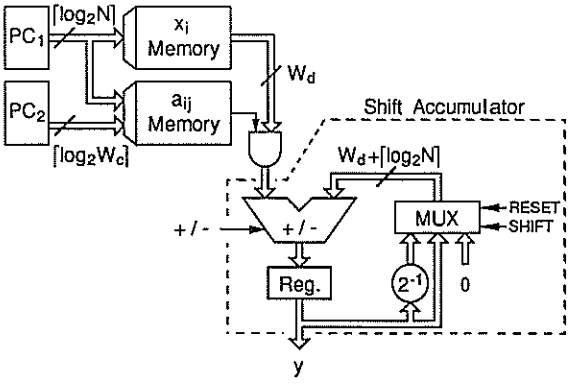


Figure 1 : Conventional Serial-Parallel Architecture

Considering the time-scheduling, the conventional serial-parallel architecture requires $N \cdot W_c$ computation microcycles, since all partial products are systematically computed, even when they are irrelevant (i.e. zero-valued). This results in unnecessary large processing time and power consumption. The number of computation microcycles can be optimized in two successive steps by :

- minimizing the number of relevant partial products; the set of non-zero digits appearing within the coefficients can be minimized using Booth's modified recoding scheme based on the Canonical Signed Digit (CSD) representation [5].
- skipping all unnecessary computations.

This optimization process is usually performed during the system design resulting in so-called *multiplierless* architectures, where a distinction can be made between *hard-wired* [6] and *programmed* multiplierless structures.

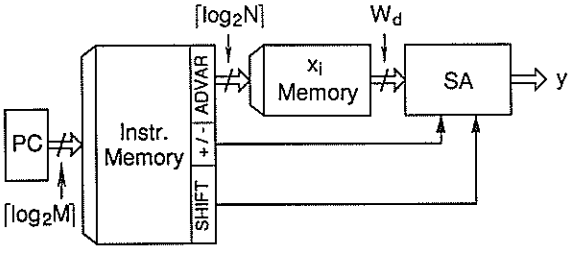


Figure 2 : Programmed Multiplierless Architecture

The second considered reference architecture corresponds to a programmed multiplierless structure (Fig. 2). As expected, only useful operations are performed. The number of processed microcycles is equal to the number of non-zero digits (M) obtained after CSD encoding. It should be noticed that the coefficient set a_i is not any more directly programmable, since it has been converted into pseudo-instructions. The necessary instruction memory space is $M\cdot(2+\lceil \log_2 N \rceil)$ bit large, where $\lceil . \rceil$ denotes the ceiling function.

4. Proposed Architecture

The new processor architecture proposed in this paper (Fig. 3) combines the advantages of both reference structures, by optimizing the number of computed operations in *real-time*. An extended bit-serial Content-Addressable Memory (CAM) is used for that purpose [1, 2].

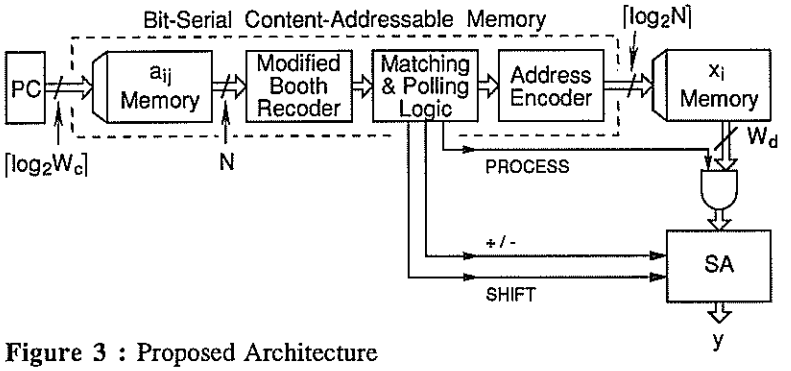


Figure 3 : Proposed Architecture

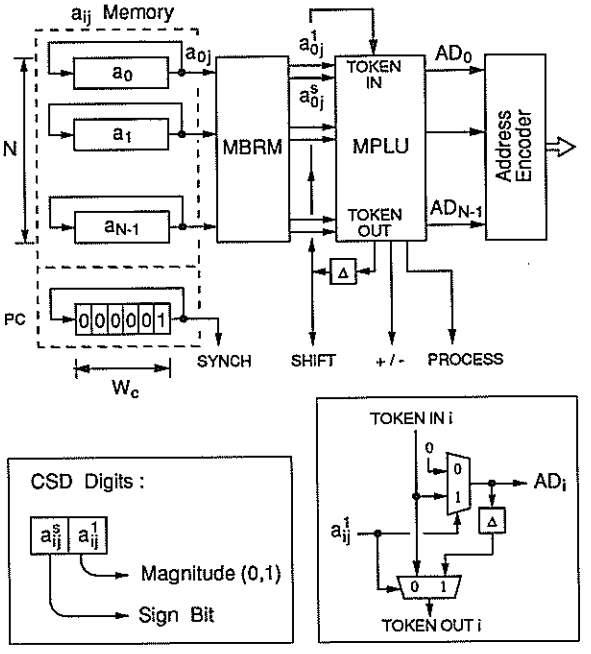


Figure 4 : Detailed CAM Memory Organization

Fig. 4 shows a detailed description of the CAM. The coefficient set is stored in a RAM using a two's complement form for storage compactness and ease of programming reasons. If required, the RAM ($N\cdot W_c$ bit) can be replaced by simple shift registers. All the coefficients are processed concurrently by a bit-serial Modified Booth Recoding Module (MBRM) [5]. They are converted into CSD digits, which in turn are inspected by a Matching and Polling Logic Unit (MPLU) in order to extract the significant digits. The MPLU handles the CSD digits column-wise during stretchable macrocycles. A control *token* is fed into the MPLU at the beginning of a macrocycle. The token skips asynchronously all irrelevant digits and is locked within the MPLU as soon as a non-zero digit has been found. The position of the locked token specifies then the address of the signal-data x_i to be selected for generating a partial product. Once a whole column has been processed, the token comes out at the bottom of the MPLU, and a new macrocycle can be started.

The macrocycle duration is data-dependent and counts as many microcycles as there are significant

digits in a coefficient column. There is one exception : for empty columns, an extra microcycle is necessary, resulting in a small overhead. The whole inner-product computation with no overhead requires M microcycles. Finally, since the proposed architecture has a strong data-driven behavior, it shares various properties with data-flow processors [7].

5. Primary Options

Fig. 3 shows that the coefficient and signal-data memories are basically interchangeable. However some important differences arise depending on whether coefficients or signal-data are stored and processed by the CAM memory. The former structures will be called Coefficient-Oriented Architectures (COA), and the latter Signal-Oriented Architectures (SOA). Both options will be discussed considering an N-th order direct form FIR filter. A single inner-product is then computed per sampling period.

Table 1a	LATENCY TIME T_L	SAMPL. PERIOD T_{Smin}	POWER CONSUMPTION
CONVENTIONAL SERIAL-PARALLEL ARCHITECTURE	$\theta(N \cdot W_c)$	$\Omega(N \cdot W_c)$	$\theta(N \cdot W_c)$
PROGRAMMED MULTIPLIERLESS ARCHITECTURE	$\theta(M)$	$\Omega(M)$	$\theta(M)$
PROPOSED COA ARCHITECTURE	$\theta(M)$	$\Omega(M)$	$\theta(M)$

Table 1b	LATENCY TIME T_L	SAMPL. PERIOD T_{Smin}	POWER CONSUMPTION
CONVENTIONAL SERIAL-PARALLEL ARCHITECTURE	$\theta(N \cdot W_d)$	$\Omega(N \cdot W_d)$	$\theta(N \cdot W_d)$
PROPOSED SOA ARCHITECTURE	$\theta(M)$	$\Omega\left(N \cdot \left\lceil \frac{W_d}{2} \right\rceil\right)$	$\theta\left(\frac{N \cdot W_d}{3}\right)$

Table 1 : Architecture Performance Comparison

The COA processors, which have been discussed so far, are especially interesting for DSP applications. Once a coefficient set is programmed for a specific application, the number of required microcycles per inner-product is fixed, i.e. M is a constant. The latency time T_L, the power consumption, and by extension the minimum achievable sampling period T_{Smin} are then directly given by M (Table 1a). Consequently, COA processors are particularly well-suited for LTI DSP algorithms with a small M factor. This can be obtained by using adequate signal-processing optimization techniques.

The SOA processors are dependent on the signal values. Hence, the number of microcycles M and the latency time T_L are time-varying. The power consumption is now given by the mean value of M, i.e. $\theta(N \cdot W_d/3)$. The power saving can be estimated at 66% (Table 1b). The minimum sampling period T_{Smin} is dependent on the upper bound of M, i.e. $O(N \cdot \lceil W_d/2 \rceil)$. Compared to the conventional serial-parallel architecture, T_{Smin} can only be reduced by a factor of two.

Considering the COA and SOA processors as described in Fig. 3, some supplementary facilities have to be added for the signal-data updating. At this level, SOA structures turn out to be more attractive, since the updating is obtained at no extra cost. Furthermore, due to the signal-orientation, all known filter-structures for Distributed Arithmetic [8] can directly be applied to SOA. This is particularly interesting for DSP algorithms, where symmetries exist in the coefficient set (cf linear phase FIR filters). In addition, SOA are interesting for IIR state-space filters. In that case, multiprocessor implementations can be simplified by sharing a single CAM between all the processors.

6. Secondary Options

Various secondary options are available. At the hardware level, the complexity can be reduced by directly abutting the full CAM to the subsequent memory (Fig. 3). Then, the CAM address encoder and the following address decoder can be suppressed.

The power consumption can be further reduced by adding a simple halt mechanism. With this facility, the COA or SOA processors are automatically set into a stand-by mode as soon as the computations are performed. The processors are then restarted at the beginning of the next sampling period. This halt mechanism is very efficient for multiprocessor implementations.

The processing time can in turn be enhanced by reducing the CAM latency time, especially in the MPLU. Partitioning techniques for recursive algorithms are available for that purpose [9].

Finally, the implemented filter order N is programmable in a limited sense by resetting the unused coefficients to zero.

7. Application Example

A Wave Digital Lattice benchmark Filter (WDLF) has been chosen as an application example. This filter fulfills the requirements of a specific 5th order PCM filter according to CCITT specification G712 PCM. This specification mainly consists of a passband ripple of ±0.125 dB from 0Hz to 3kHz, a stopband attenuation of -14 dB from 4.0kHz, and -32 dB from 4.6kHz.

The sampling frequency is set to 16kHz. The detailed structure of this WDLF is given in [10].

Starting from the basic filter structure, an equivalent state-space form is derived according to the methodology described in [11]. The resulting scaled state-space filter is expressed in a matrix form as a function of the input signal x, the state variables w_k, and the output signal y (3). The necessary coefficient wordlength W_c is 11 bit long. With a global data wordlength W_d of 20 bit, a signal-to-noise ratio of 90dB is achieved.

$$\begin{pmatrix} w_1(n) \\ w_2(n) \\ w_3(n) \\ w_4(n) \\ w_5(n) \\ y(n) \end{pmatrix} = \frac{1}{2^9} \begin{pmatrix} 65 & 592 & 0 & 0 & 0 & 15 \\ -351 & 80 & 0 & 0 & 0 & -81 \\ -348 & 0 & 128 & 0 & 0 & 156 \\ 0 & 0 & 0 & 60 & 336 & 100 \\ 0 & 0 & 0 & -264 & 160 & -440 \\ 58 & 0 & 320 & -176 & 0 & 22 \end{pmatrix} \begin{pmatrix} w_1(n-1) \\ w_2(n-1) \\ w_3(n-1) \\ w_4(n-1) \\ w_5(n-1) \\ x(n) \end{pmatrix} \quad (3)$$

The state-space filter has been simulated for the first and second reference architectures, as well as for COA and SOA processors. For each architecture, a single time-multiplexed processor was considered. The obtained results show that for COA, the overall performance is almost improved by a factor of 5 compared to the first reference architecture (Table 2a). Hence, for a fixed sampling frequency, the power consumption is reduced by 5. Considering multi-channel filtering applications, the number of processed channels can be increased by the same factor, at no extra cost for the power consumption.

Table 2a	SAMPL. PERIOD T_{Smin}	POWER CONSUMPTION
CONVENTIONAL SERIAL-PARALLEL ARCHITECTURE	$\Omega(396)$ → 100%	$\theta(396)$ → 100%
PROGRAMMED MULTIPLIERLESS ARCHITECTURE	21%	21%
PROPOSED COA ARCHITECTURE	21%	21%

Table 2b	SAMPL. PERIOD T_{Smin}	POWER CONSUMPTION
CONVENTIONAL SERIAL-PARALLEL ARCHITECTURE	$\Omega(720)$ → 100%	$\theta(73'440)$ → 100%
PROPOSED SOA ARCHITECTURE	50%	30% to 33%

Table 2 : Performance Comparison for the WDLF

For the SOA processor, according to Table 1b, the minimum achievable sampling period T_{Smin} can be reduced by a factor two (Table 2b). For the estimation of the SOA power consumption, simulations were performed by applying successively 5 different sinusoidal input signals, as well as white and gaussian random noise sources. The simulations show that the power saving is very close to the expected results (Table 2b, Table 1b).

8. Conclusion

In this paper, a new flexible and efficient serial-parallel processor architecture is presented for low-power LTI DSP applications. Considering the available primary and secondary implementation options, the architecture can be adjusted to fit many specific applications, especially for VLSI realizations.

Acknowledgements

This project was supported by the Swiss Foundation for Research in Microtechnology, under Grants FSRM 88/14 and FSRM 87/1. The authors wish also to thank Mr. M. Yazdanpanah for his contribution to the design of the benchmark filter.

References

[1] W. Hilberg, *Digitale Speicher 1 : Elektronische Speicher*, Oldenbourg Verlag, München, FRG, 1987.

[2] L. Chisvin and R. J. Duckworth, "Content-Addressable and Associative Memory : Alternatives to the Ubiquitous RAM", *IEEE Computer*, July 1989, pp. 51-64.

[3] A. Peled and B. Liu, "A New Hardware Realization of Digital Filters", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-22, No. 6, Dec. 1974, pp. 456-461.

[4] A. Theys, E. Dijkstra, and C. Piguet, "Discrete Transform on a Chip : Design and VLSI Implementation", *Proc. Int'l Symp. on Applied Control, Filtering, and Signal Processing '87*, IASTED '87, Geneva, CH, June 1987.

[5] K. Hwang, *Computer Arithmetic : Principles, Architecture, and Design*, McGraw-Hill, New York, USA, 1979.

[6] P. Denyer, and D. Renshaw, *VLSI Signal Processing : A Bit-Serial Approach*, Addison-Wesley Publ., Wokingham, UK, 1985.

[7] J. B. Dennis, "Data Flow Supercomputers", *IEEE Computer*, Vol. 13, No. 11, Nov. 1980, pp. 48 - 56.

[8] S. A. White, "Applications of Distributed Arithmetic to Digital Signal Processing : A Tutorial Review", *IEEE ASSP Mag.*, Vol. 6, No. 3, July 1989, pp. 4-19.

[9] D. J. Kuck, "A Survey of Parallel Machine Organization and Programming", *Computing Surveys*, Vol. 9, No. 1, March 1977, pp. 29-59.

[10] L. Claesen & al., "Automatic Synthesis of Signal Processing Benchmark using the CATHEDRAL Silicon Compilers", *Proc. IEEE Custom Integrated Circuit Conf.*, 1988.

[11] U. Sjöström, I. Defilippis, M. Ansorge, F. Pellandini : "A Methodology for ASIC Implementation of Digital Filters", *Proc. GRETSI-89*, Juan-Les-Pins, France, June 1989, pp. 797-800.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A PARALLEL DSP ARCHITECTURE FOR IMAGE PROCESSING

Fernando A. BELTRAN BLAZQUEZ and Jesús NAVARRO ARTIGAS

Grupo de Electrónica.
Dpto. Ingeniería Eléctrica e Informática.
Centro Politécnico Superior (ETSII / ETSIT).
María de Luna 3, ACTUR, 50015 Zaragoza, Spain.

This paper presents a digital signal processor (DSP) based parallel architecture intended for industrial vision systems.

The authors' proposal is based on the very nature of the low-level image processing (basically series of discrete convolution products) and the arithmetic high-efficency of DSPs for these taks.

The architecture is modularly implemented. This modularity allows it a great flexibility, reconfigurabitily and adaptability to the current tasks.

The used DSPs are the current industrial standard ones: Texas Instruments' TMS 32010. It also allows a low-cost for this industrial vision system.

The first prototype of our system is a 16 DSPs version in a 4x4 configuration: 4 image processing modules each of them comprising 4 DSPs.

Our system is capable of real time processing of 512 x 512 8-bit pixels images and its work is controlled by a host computer via PC bus or VME bus (the user can choose between them without restrictions).

1 Introduction

A relevant digital signal processing field is the image processing one.
The more interesting image processing industrial applications need treatment systems with real-time responses. In the last years, the effort made in developing real-time image processing systems has been more oriented towards software than towards hardware.
The increasing use of integrated digital signal processors (DSPs) has led to the proposal of several DSP parallel architectures for image processing. In these architectures several processors work at a time each one processing a portion of the whole image (subimage). Our work adds up to this hardware research interest and comes to give a step more in it.
The aim of our architecture is to profit the intrinsic capabilities of DSPs in image processing, obtaining a system structurally and functionally configured so that the flexibility and the functional adequacy of our system to the industrial tasks intended for it (robotics, automation, etc.) are guarantied.
The system here presented is so a DSP based parallel architecture which employs a current industrial standard DSP: the Texas Instruments' TMS 32010, whose use gives the system an additional interest: its low-cost.
Along this paper, we'll describe the most relevant technical characteristics of our system and several improvements and modifications with regard to the published ones in our previous papers [1], [2].

2 Architecture description

The system is host computer governed via PC bus or VME bus and has an internal bus (system bus) which includes the lines of both normalized buses and some other control lines.
The different processing modules which configure the system are interconnected through the system bus between them and with the host computer.
There is also an additional module for image acquisition, and initial or processed images monitorization. This module is also host computer governed (with PC bus or VME bus interconnections) and can transfer images to the host computer memory and store images from it. The module memory is used as refreshing memory for image monitorization.
The working method of this architecture is composed by the following steps:

1. One image is acquired, monitorized (for camera calibration) and stored in the host computer memory.
2. The image is divided in a number of subimages equal to the number of the current architecture processors (in our first version, 16 subimages).
3. Each subimage is loaded in the local memory associated with each DSP.
4. The DSPs perform the image processing and each one of them returns a processed subimage.
5. The global processed image is loaded in the host computer memory and displayed in the system monitor.

3 Image memory and memory expansion

The image to be processed is digitized in a 512 x 512 8-bit pixels format (256 gray level resolution) and so it needs 256 Kbytes of RAM memory to be stored.
For a 4x4 configuration (16 subimages), each subimage needs 16 K of memory to be stored, but the TMS 32010 processor has only 12 addressing lines to manage its external memory implying a maximum of 4K addressable external memory from each DSP in its microprocessor working mode. To solve this problem, a memory expansion scheme is implemented using I-O ports [3],[4], making it possible to address 32K of external RAM memory to store the subimage and the programs corresponding to each DSP.
The 32K-words of static RAM are addressed by the lower bits of a 16-bit counter (4 items of 724193). The

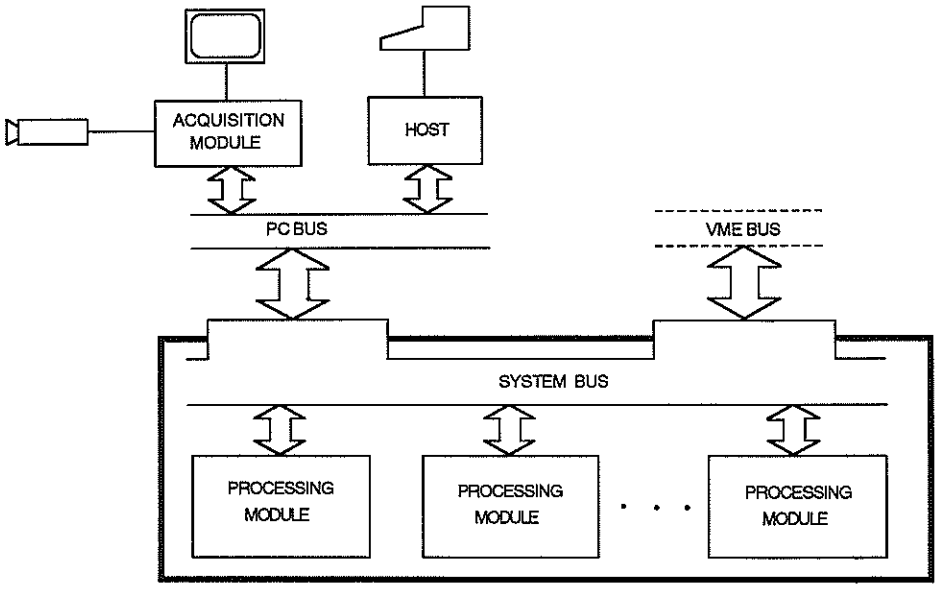


Figure 1. System diagram.

corresponding address is loaded into the counter by an OUT instruction to determinate port (port 0, by example). This action loads the content of the data bus into the counters. The memory can then be sequentially read from or written to another port (port 1, by example) by an IN or OUT instruction. The MSB of the counter determines whether the memory address is incremented or decremented after a reading or writing in the data memory. The sequential access to the memory is guarantied until a new address is loaded into the counter.
The architecture has an overlapped memory organization [1] to do the low-level image processing. Its precedents are the proposals of Naqvi et al. [5],[6].
In our architecture each local memory (addressed by his corresponding DSP) does not only store the subimage to be processed, but this subimage and a fringe around it. The width of this fringe depends on the processing window to be used in the low-level image treatment. When, for example, a 5x5 window is used, the fringe is two pixels wide.
With our overlapping filosophy some image areas are shared by different DSPs local memories but the scarce increment in the needed memory is highly compensated by the no-needed communication between processors in low-level image processing and the subsequent increase of processing speed.
The main difference between our memory organization and another proposed ones lays in its physical partition and flexible access possibilities.
Another authors [5], [6] work mainly upon a single data memory accessed by the different DSPs present in the system, implying a software achieved overlapping. Our system, having as many data memories as subimages to be processed, has a hardware overlapping technique, laying the shared information in the different adjacent local memories.

4 Processing module

This architecture is modularly implemented to obtain a high degree of reconfigurability and system adaptability to the intended tasks.
The system has several processing modules. These modules are identical and its number may be varied depending on the current needs. Its first version has four modules, each of them including (Fig. 2) four processing units in a single card. Each processing unit comprises one DSP, its associated local memory and some digital devices (counters, buffers, etc.) which constitute the expansion memory and the intercommunication circuitries of the DSP with the rest of the processing units and the host computer. The card includes a bus (the card bus) with the same lines as the system bus, being a mere extension of it inside the card. This card bus eases the access from each DSP to its local memory and also the access from the host computer and from the restant DSPs of the system to it (in medium and high level image processing these accesses are needed to the right performance of the system).
Each DSP has a direct access to its local memory. The access of a DSP to the restant local memories of the system (in the same or another processing module) is protocol controlled. When a DSP demands control over the bus, the module control unit checks the free bus condition and gives control to the DSP if this condition is true, interrupting the access of the restant DSPs of the system.
It's also possible the external access to the local memories. In that case the host computer must achieve the control over the system bus too, interrupting the different DPSs whose local memories it is going to access.

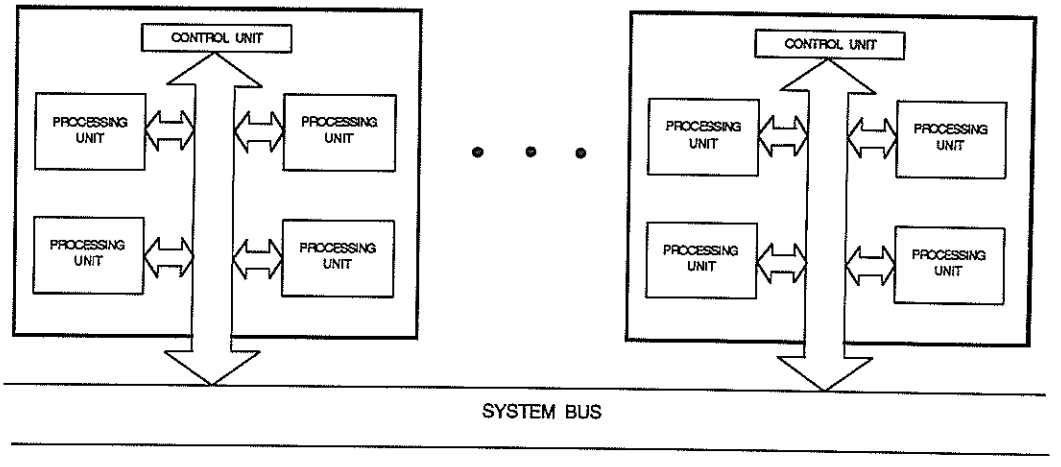


Figure 2. Processing module.

The corresponding protocol governs the process. Once obtained the control by the host, the computer can write in several local memories but only can read from one of them. Finished the external access, the system bus is released and its condition is set free, continuing their work the previously interrupted DSPs. The presence of tri-state buffers in the bus interconnections is the hardware counterpart of the communication protocols which are software implemented over the different control units of the cards and handshake circuitry of the DSPs.

5 Conclusions

This paper presents a low-cost real time DSP based image processing parallel architecture modularly implemented.
Their technical characteristics (as the expandable nature of its modular implementation, and its proposed overlapping memory) give our architecture a great reconfigurability and adaptability making it possible its application to very different industrial tasks.
Our proposal opens a new research way in the software field, easing adapted and optimized programs to be runned over this architecture). We expect to present in a near future new proposals using more advanced DSP generations allowing greater processing speeds and design simplification.
This possible future designs will probably be of greater cost and of more academic or research interest than industrial one, but undoubtably will imply an interesting advance in the possible digital signal processing applications.

Acknowledgements

We should like to thank Miss Isabel Vives and Miss Pilar González without whose unvaluable help would have not been possible the preparation of this paper.

References

[1] F.A. Beltrán, J. Navarro. *A Parallel Architecture for Industrial Vision Systems.* Proceedings International Workshop on Sensorial Integration for Industrial Robots. Zaragoza (Spain), 1989, pp. 251-255.

[2] F.A. Beltrán, J. Navarro. *Arquitectura para Sistemas de Visión Artificial basada en Procesadores Digitales de Señal.* Actas 1er Congreso de la Asociación Española de Robótica. Zaragoza (Spain), 1989, pp. 83-91.

[3] Texas Instruments. *First Generation TMS 320. User's guide.* 1989.

[4] Texas Instruments. *Digital Signal Processing Applications with the TMS 320 Family.* 1988.

[5] A.A. Naqvi, M.B.Sandler. *Image Processing with multiple DSPs.* IEE Colloquium on VLSI for Image Processing, 1987.

[6] A.A. Naqvi, M.B.Sandler. *Performance of the OSMMA Image Processing System.* Proc. Int. Conf. on Parallel Processing for Computer Vision and Display. Leeds, U.K., 1987.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A HIERARCHICAL STRUCTURE FOR REAL TIME PARALLEL PROCESSING

* Guido Castellini, ** Enrico Del Re, ** Ada Fort, * Laura Pierucci

* C.N.R Istituto di Ricerca sulle Onde Elettromagnetiche , via Panciatichi 64 , 50127 - Firenze.
** Dipartimento di Ingegneria Elettronica, Universita' agli Studi di Firenze, via S. Marta 3, 50139 - Firenze, Italy.

Abstract.This paper presents a new digital architecture designed to acquire and process in parallel large bandwidth signals or bidimensional data. Originally the architecture operates as attached processors to an apparatus based on general CPUs and standard buses (VXI) in an experiment of high energy physics but it's interesting also for general applications.

1. INTRODUCTION

In many area of applications computational-intensive speed, from elaboration of specialized algorithms with millions measured data to processing of very complex algorithms in real time is required.
Parallel processing, where many processors take on a problem simultaneously, hold the answer. The performance of such multiprocessor system is mainly determined through performance of used processor elements, performance of network topologies and quality of scheduling/ partitioning algorithms. However the allocation of problems to numerous processors is a difficult work and requires well thought steps, especially designing a mimd-machine based on commercial DSP.

We have studied the problem relative to the implementation of parallel network of DSP (locally interconnected) and designed a specialized apparatus, based on the previous architecture, to acquire and process data coming from a vertex detector in a high energy experiment (ZEUS experiment).

The ZEUS vertex detector (hereafter called VXD) is a multi-wire time expansion chamber designed for the detection of short lived particles that decay in the neighborhood of the HERA beam in Hamburg. The signals from the individual wires must be of relatively short duration, and must be sampled with 4 ns sampling rate.

Therefore in the acquisition structure new commercial devices in ECL technology have been used and the parallel processors are Texas DSPs (TMS320C30).

The readout electronics must be able to store the information corresponding to a time of ≈ 800 ns and to transmit these data to a buffer with a mean rate of 1 ms.

A second requirement of the readout electronics is that it must be able to store up to 15 events and to compress and format the data relative to an event and to transfer them to the main computer.

In future developments the apparatus will execute also pattern recognition tasks and measurement on the acquired data.

In what follows we describe the design of a prototype Multi-Hit Acquisition System (MHAC), which is being tested in our laboratories, and the design of the communication networks within the MHAS and between the MHAS and the control units.

The paper presents also an approach to the evaluation of the proposed apparatus efficiency. We will consider both our specialized application and general application to the digital signal processing.

2. ARCHITECTURE OF THE IMPLEMENTED SYSTEM

A block diagram of the Multi-Hit Acquisition card (MHAC) is shown in fig. 1 and the block diagram of the system, as a whole, is shown in fig. 2.
The system has a tree architecture. The central computer of the experiment and a dedicated station (VAX station from Digital)

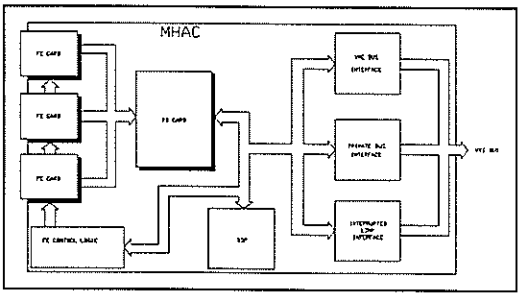


Fig.1 MHAC implemented for Vertex Detector readout electronic

with the man machine interface are at the top level and perform the data collection and the control of the dedicated CPUs (based on Motorola chips). The second level CPUs control

in real time the parallel processing structure at the bottom level and provide the already processed data and command transfer between the top and bottom level.

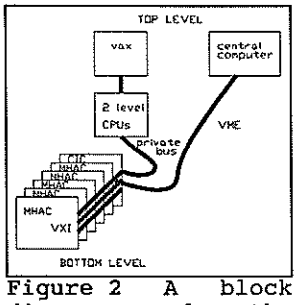


Figure 2 A block diagram of the system as a whole

The implemented apparatus (bottom level device) is composed of three types of 9U cards that are interconnected by a standard VXI bus. The choice of a standard bus is due to the availability of commercial component: however our use of the bus is non standard even if it is compatible to the VXI protocols.

The bus is divided in a standard VME bus, an interrupted link between card and card and a dedicated packet switching bus.

The card types are:

i) Crate Interconnect cards (CIC). These cards are located in slot 0 of each crate, they manage the data transmission,the monitor and control signals in the crate. They are slave for the intercrate communications. Also these cards contain a DSP.

ii) Multi Hit Acquisition Cards (MHAC). The cards receive the parallel signals from the detector. These are constructed in modular sub-units; this structure permits the modification of the system minimizing hardware and software redesign, allowing a more general use of the apparatus.

iii) VME Interconnect Cards. These cards are used for VME data readout.

The modules that constitute the MHAC are:

a) The Mother Board (MB). In the first design version it contains four DSP (fig. 3) to control the various functions of the MHAC (card control, data processing, data output formatting, global trigger service and communication links controls). In the final system we have planned to reduce the number of DSP chips, due to the cost of the apparatus (more than 100 cards will be used).

b) The fast Front End card (FE). It acquires, digitizes and memorizes the analog signals from the detector.The input ECL pipelines clock period is from 3 to 4 ns.

c) The Primary Buffer card (PB). Upon the arrival of a general trigger it receives the data corresponding to one event from three fast front end cards stores it into a buffer(dual port memories). Upon receipt of a global command the two programmable gate arrays(XILINX) on board of this card pre-elaborate and analyze the data, and send it to the DSPs network.

3. MHAC SUB-BLOCK DESCRIPTION

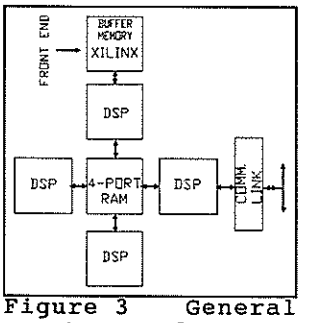


Figure 3 General version of MHAC module with four DSPs

Each Front End board receives: the signals from 24 detector sense wires, two clock signals, analog variable thresholds for the discriminators. The board sends the data to the PB board via a 16 wire internal connection.

The analog signals enter the fast front end card through quad ECL LE CROY discriminators. The thresholds of the discriminators are individually controlled by the DSP on the MB by means of a DAC. The signals are then split into two and are sampled at the frequency of 125 MHz and sent to four 4x1024 bit μPB100474 memories (write cycle 6 ns) which works at half the sampling rate of the system. Memory addressing on the MB is set up so that the memories are filled cyclically and it's possible to preset the cyclically buffer length.

The Primary Buffer card receives 24 data wires from 3 Front End cards and stores them into CY7C132 2Kx8 dual port memories.

As soon as the transfer of the data is completed the data is clocked into a Programmable Gate Arrays XC3042 that implement preprocessing algorithms. This data is then transferred to the DSPs.

The mother board card comprises the four DSPs and multiport memory structure, the control circuits of the acquisition cards and the bus and communication links drivers. It uses the standard VME part of the VXI bus, and subdivides the remaining VXI pins into two parts with separate functions for the DSPs interconnections: a private bus and an interrupted link.

Four DSPs communicate respectively with the PB board, with the output Dual Port Memory that is connected with the VME transfer bus, with the interfaces of the private bus and of the interrupted link.

Three synchronization schemes are present on the card:

a) A data flow structure between the four DSPs. The common memory on the card and the memory on the interrupted bus may be mapped as data or as program so that one DSP may influence the program flow of the other neighbour DSPs.

b) An interrupt driven structure to manage the private bus and VME communication links.

c) A semaphore memory cell to synchronize the slow communication on the VME bus.

The VME interface is formed by a Dual Port Memory (DPM) and a Logic Block (LB) able to process the VME standard protocol signals. From the VME side the DPM is divided in two parts: the first part is read as a one cell address Read Only Memory, the scanning of this memory is implemented by means of a local counter chained to all 30 MHAC cards; the second part of the memory is a standard Read-Write Memory with standard VME interface which permits direct connection to the DSP on the card.

4. COMMUNICATION LINKS IN THE SYSTEM

The parallel processing structure transfers data and exchanges control signals by means of three independent communication links (fig. 4). The first is a standard VME bus which is completely dedicated to fast data transfer to the second level CPUs. The second link is an ECL/TTL bus (private bus) which uses the synchronization lines of the standard VXI. This bus uses a packet switching protocol with variable length packets.

The third interconnection net is based on multiple access memories: the neighbour processors can divide the computed data without the necessity of a data transfer. The partially processed data blocks flow through the computing structure, moved by the tasks developed by each node.

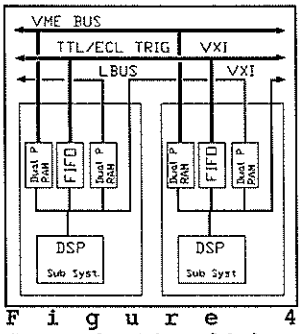


Figure 4
Communication links

The processors work as independent nodes: the described links allow the global data exchange along the network and the distribution of broadcast information.

The private bus uses the standard TTL/ECL trigger lines to transmit data, addresses and control signals.

Same dedicated lines are used for a bi-directional daisy chain which serves for interrupts issued by the MHAC cards.

The interface between the private bus and the DSP is composed of two parts:

i) A Dual Port Memory 2KX16. The memory is divided into four partitions that correspond to the four different operations to be performed. The length of each partition is fixed and all partitions have the same size.

ii) A logic unit (LUN) that interconnects the bus to the DPM and that decodes the bus protocol.

5. TOOLS TO MANAGE THE EQUIPMENT

The DSPs have a variety of functions, these can be divided in four main categories: power up, set up, debugging and run.

To provide tools for the debugging of the system and to implement fault diagnosis and fault tolerant characteristic each card has a stand alone monitor and debugger program. The DSPs on the MHAC can load the programs from all the bus in the system even if the standard way is the VME bus. The auto-diagnostic features of the monitor can individuate the faults on the cards and each card can be excluded from the normal operation of the apparatus.

At power up the DSPs perform a self check, check the communication links between individual DSPs and the main CPU of the system, boot and check the PGAs, check the primary buffer memories and Front End cards.

During set up the DSPs measure the errors in the acquisition sub-system, set up the discriminator thresholds, transmit the set up data to the system controller.

During the run phase the DSPs establish the priority between function and send error messages, receive and process the data from the Programmable Gate Arrays generate the information needed for the data transfer on the VME bus, process and format the data for block transfer mode.

6. PRELIMINARY PERFORMANCE EVALUATION

To evaluate the performance of a multiprocessor system it is necessary to determine the efficiency of the network topologies and quality of the scheduling and partitioning of the used algorithms.

Parallel solutions always involve a trade-off between communication and computation, which are characterized by their granularity. A fine-grain solution creates many small tasks,while a coarse-grain solution creates a few large ones.If the tasks is too small, the processes spend too much time communicating results and getting new tasks. If it's too big then some processes might sit idle waiting for others to finish.

The implemented parallel apparatus is optimized for the specialized tasks in the high energy physics experiments. The pulse signals recovery, energy analysis, data compression and data formatting are parallel operations due to the parallel structure of the vertex detector.

Future applications to pattern recognition analysis will be easily carried out because the object of interest are confined in small areas of the vertex detector.

It is interesting to evaluate the performance of the proposed architecture to more general digital signal processing tasks.

First of all to achieve this goal we have

considered an unlimited extension of the four DSP node and we have used the FFT algorithm as a benchmark to understand and analyze the interactions between partitioning/scheduling of the algorithm, node processor performance and interconnection network.

The smaller sub_tasks of the FFT is the radix-2 butterfly:

$$C=A+B*W^{-i}$$
$$D=A-B*W^{-i}$$

There are three different optimal mapping of N point FFT (through four port memory) to the parallel processing structure, avoiding communication overheads. They depend from numbers P of processors used to map the FFT radix-2 basic elements (butterflies).

The first one implies that each basic element is executed by one processor node using $(N/2)\log_2(N)$ processors, only usable for low point FFT's. The second propose is a mapping of the FFT columns to processor nodes using $P*k<=\log_2(N)$ processors (where k is the number of columns mapped on each processing node), but needs continuously data (pipelining) for effective work.

The third one maps the rows of the FFT algorithm to processor nodes using $P*k<=N/2$ processors (where k is the number of rows mapped on each processor node), allowing non-continuously work and more freedom to choose the number of processors because N/2 is greater then log(N) in most of applications.

With the TMS 320C30 as node element and the proposed network structure a 16 point FFT can be processed in 1 microsecond, allowing sampling rates until 16 MHz.(fig. 5)

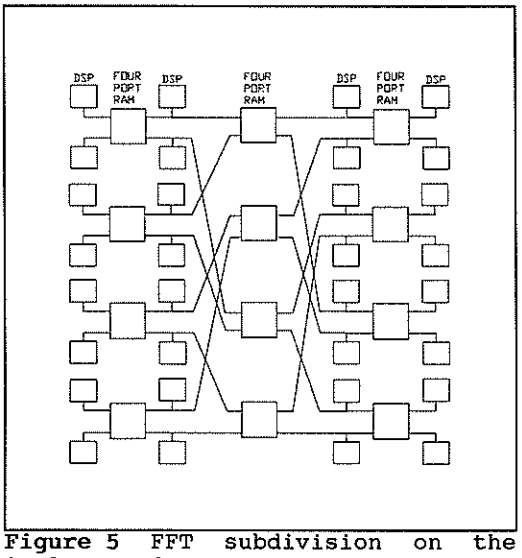


Figure 5 FFT subdivision on the implemented structure

The disadvantage due to a more complicated connection scheme of processor nodes and to the increasing number of necessary links per node if N increases may be avoided by implementing a dynamic connection network.

Researches about the use and the limits of the intercard communication links (interrupted bus, private bus) are under development.

7. CONCLUSION

The implemented system is an innovative solution to the problem of acquiring and processing data in an high energy experiment.

The architecture of the single card may be adapted also to different parallel signal processing use, due to is modularity and programmability.

However mapping digital signal processing algorithms, different from FFT, to our structure will be a future research, because until now there are no efficient tools for partitioning and optimizing the mapping operation by means of tools for rapid prototyping of software for such a structure.

The presented research work has been supported by the italian National Council for Researches, Finalized Project on Parallel Computing, and by the Italian National Institute for Nuclear Science.

REFERENCES

[1] G.Castellini,P.L.Emiliani,L.Pierucci, F.Pirri, S.Rocchi "Multiprocessor structure based on commercial DSPs" ICASSP April 11/14 1988,New York

[2] G.Castellini,U.Camerini,A.Fort, A.Gabbanini, M.Tesi,V.Vignoli "The Vertex Detector Readout Electronic" I.R.O.E. Technical Report TR/ESI/89.21 October 1989

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

PARALLEL PROCESSING WITH A DATA FLOW ARCHITECTURE

P. ABELLARD - G. NOLIBE - N. RAZAFINDRAKOTO
Laboratoire d'Automatique et d'Informatique Appliquées de Toulon
Université de Toulon, BP 132, 83957 LA GARDE CEDEX, FRANCE

The design and the control of robotic arms require the elaboration of a mathematical model of the manipulator but the equation complexity pose a problem in practical use. An approach based on Data Flow Petri Nets is proposed because conventional multiprocessors built on Von Neuman's model have some important limitations and do not almost allow to obtain the performances expected, contrarily to Data Flow architecture which is structurally different.

1 - INTRODUCTION.

In classical teleoperation, we consider that man does everything : he knows the desired task, he has a certain knowledge of the state of progress of the task, and he continously works out an action strategy for the master components in order to give a correct progress to the task aimed at [1]. Moreover, man brings all the mechanical energy to the carrying of the task, including that lost by the shortcomings of the telemanipulator. In the teleoperation with bilateral enslavement, the ratio of feedback effort which can be lower than one, divides by the same amount the operator effort and so, his fatigue.

Ideally, if the machine was perfect, the operator would have the sensation, after a brief apprenticeship, in the case of teleoperation, of carrying out the task with his own hands and not by using distant and limited prehensile gears. Then, he would be in *telesymbiosis* with the environment in which the slave set is [2]. It will be difficult to reach this stage, but computer science allows us to getnearer this situation of ideal *transparency* by seriously releaving the operator during his physical and mental task : that is the aim of Computer-assisted Telesymbiotics.

The computer is here in order to improve the teleoperation performances and it is conceived as a help and support for all the units in the system including the human operator. This assistance which is the robotic aspect of teleoperation, has been brought to the fore in connection with OCEANO 6000 [3] system with which we have carried out our manipulations (figure 1) and whose control requires the real-time carrying out of numerous and bulky calculations that must be performed in parallel on adaptated architectures [4].

The problem of parallel calculations involves several different but often interconnected aspects which converge to the same purpose : increasing the performances and faciliting hardware and software implementations of parallel systems with concurrent evolutions. However, this problem lies on two principal points : parallel program modelling and their implementations, and the selection and design of a parallel architecture able to carry out operations with concurrent evolutions.

The degree of connection between these two elements plays an important part in the implementation of a parallel process because the architecture must be adaptated to the model of parallel calculation and not the other way round.

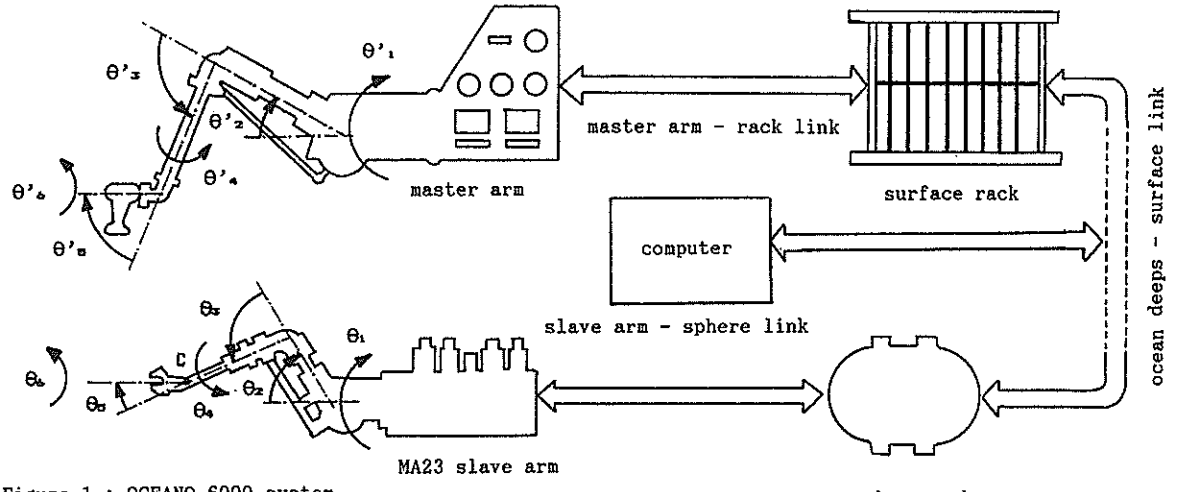


Figure 1 : OCEANO 6000 system.

In this aim, a *data flow multiprocessor* built with 14 data flow processors connected to an IBM host computer, and whose operation is modeled with Data Flow Petri Nets [5] has been studied and a specific software environment has been created.

2 - DATA FLOW PETRI NETS.

2.1 - Definition.

A Data Flow Petri Net (DFPN) is a 7 tuple $\langle R,\varphi,\xi,\psi,X,O,C\rangle$ in which :

- R is a comformable two-part places Petri net, P_v and P_o called set of places associated respectively to variables and operators,

  $\forall p \in P_v \rightarrow CARD(^\circ p) \leq 1$ and $CARD(p^\circ) \leq 1$

  $\forall p_i \in P_o$, $p_j \in P_o$ and $i \neq j \rightarrow {}^\circ p_i \cap {}^\circ p_j = \emptyset$, $p_i^\circ \cap p_j^\circ = \emptyset$

- φ is a surjective application $\varphi|_{P_v} : P \rightarrow X$, $\varphi|_{P_o} : P \rightarrow O$ as $\forall p_i \in P_o$, $p_j \in P_o$ and $\varphi(p_i) \equiv \varphi(p_j)$, $i \neq j \rightarrow \forall t_l \in {}^\circ p_i$, $t_k \in {}^\circ p_j \rightarrow \{\varphi({}^\circ t_l)\} \neq \{\varphi({}^\circ t_k)\}$ so 2 identical operators can't work on the same set of data,

- ξ is an injective application $\xi : X \rightarrow \mathbb{M} = \{ME_1, ME_2, \ldots ME_u\}$ as $\forall p \in P_v$, $ME \in \mathbb{M} \Rightarrow ME = \xi(\varphi(p))$, $\mathbb{M}$ is called set of memory areas.

- ψ is a surjective application, $\psi : T \rightarrow C$, (T is the set of the net transitions),

- $X = \{x_1, x_2, \ldots x_u\}$ is a set of variables (real, integer, logic...) with values in domains $D_1, D_2, \ldots D_u$,

- $O = \{o_1, o_2, \ldots o_t\}$ is a set of operators defined as internal applications of $D_1 x D_2 x \ldots D_u$,

- $C = \{c_1, c_2, \ldots c_r\}$ is a set of conditions (predicates) on X variables and transitions

2.2 - Representation.

Figure 2 shows the representation of an operation carried out with an operator and a set of variables. Software and hardware simulation of those nets have been realised with elementary studied modules connected together according to the net architecture [6].

- t_i and t_j are respectively called input and output transitions of the operator o_r associated to the place p_{ij}

- places ${}^\circ t_i = \{p_1, p_2, \ldots p_r\}$ and $t_j^\circ = \{p'_1, p'_2, \ldots p'_s\}$ respectively represent the data necessary for the operator o_r and the results obtained.

2.3 - Marking.

A mark put down in a *variable* place means that the value of the variable is written. A mark put down in an *operator* place means that the operator is activated. We assume that a place can't store more than one mark, so the nets are safe.

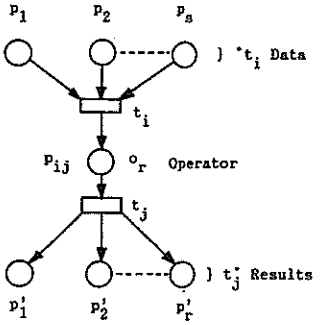


Figure 2 : Data Flow Petri Net.

3 - DATA FLOW PROCESSOR.

The choice of a NEC μPD 7281 Data Flow processor [7] has been kept because it reflects at component level the macro-structure of figure 3 defining the Data Flow architecture which is fundamentally different from conventionnal ones, because it has no central processor. It is replaced by a processing section composed with processing units, logic and arithmetic units, input/output processors... There is no RAM central memory. It is replaced by a section of memory modules including addresses, operation codes, operands... Neither is there any program counter. A decision making array enables connection of a memory module section output to the appropriate processing unit. A distribution array enables the connection of a processing unit output to the memory section concerned.

So, in this architecture, there is no conflict to access a bus or a data in the memory.

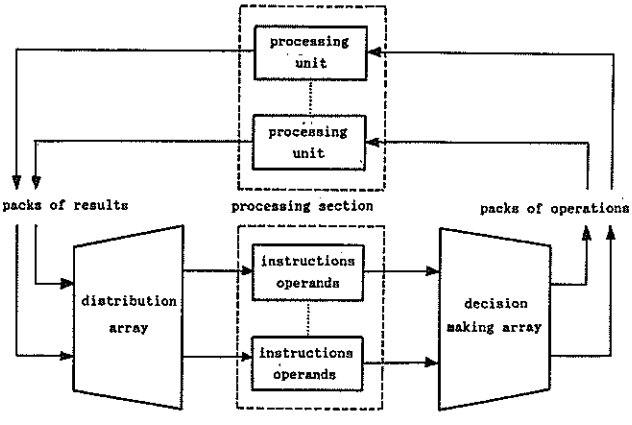


Figure 3 : Data Flow Architecture.

The data flow processor starts its function when receiving the data to be processed. These data are combined with the instructions received after initialization. When the instruction has received all the data necessary for the operation, it is introduced in an input queue and then in a processing unit. The result may be an input for a supplementary instruction or an output if it is a final result. The instructions are represented as nodes, while the data which circulate through the links, between the nodes, are called tokens.

The tokens have variable lenghts and subdivisions according to the blocks which are passed through. For example, input and output tokens are composed with 32 bits and are divided into four areas : chip address, identification area, command area and data area. The identification area tells the processor the calculation in which the token must be included. The command area indicates if the input token contains data to be processed, diagnosis information or if it is a part of an object-program loading.

After the acceptance of the token by the processor (address zone corresponding to the module number defined by Reset), the address area of the chip is not necessary and the 28 remaining bits constitute the token to be transmitted to the following internal block. During the processing, several areas are tied up and some others are suppressed according to the kind of specified operation and the place of the token in the internal pipeline. A token incorporated in the processing unit may have a 64 bit size.

4 - DATA FLOW MULTIPROCESSOR.

The 7281 component structure allows a simple realization of multiprocessor systems (figure 4) because two 16 bits buses (IDB0-IDB15 in input and ODB0-ODB15 in output) and request pins (IREQ,OREQ) and acknowledge pins (IACK,OACK) are available. The other pins are relative to the clock CLK, the Reset and the power. The studied multiprocessor is composed of fourteen 7281 processors. It is connected to a host IBM computer.

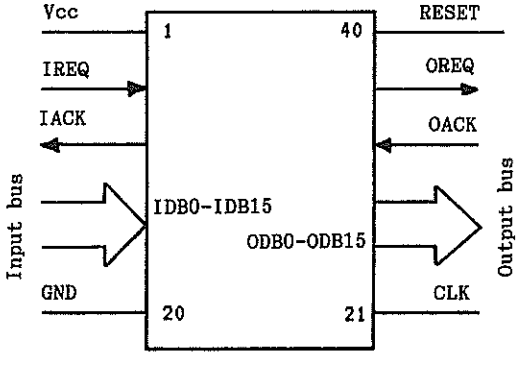


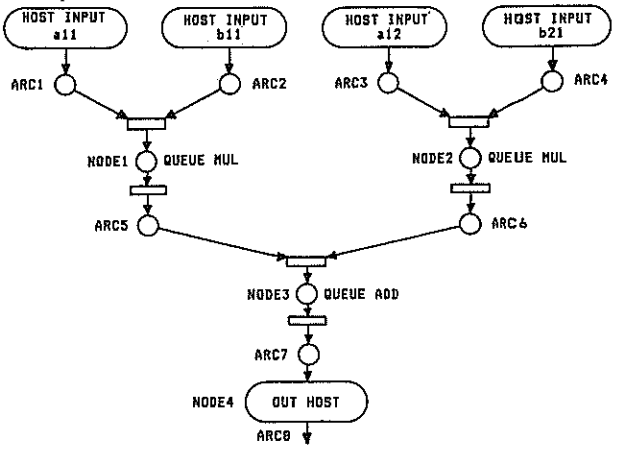
Figure 4 : μPD 7281 Data Flow Processor.

After the token arrival in the first 7281 of the chain, the component address area is compared to the 7281 module number. If this module number indicates that the token is to be processed in another 7281, the first one transmits it to the second and so on until the token reaches the concerned processor.

As regards the tokens which release by themselves the function to be processed, it is not necessary to take care of processing state in the following 7281. This allows the multiprocessor to be considered as a fast processing module whose performances can be increased by parallel attaching supplementary 7281 chips.

The Data Flow Petri Nets are used for the data flow multiprocessor operation. Figure 5 shows that they can advantageously replace flow graphs for validation, simulation and scheduling : the nodes correspond to the operator places, while the notion of arc is to be considered in the wide meaning of an inter-operator link with a variable place.

In the functional assembler language used, the source program describes the function to be processed and doesn't specify all the steps as classical assemblers.




```
1 . EQUATE   HOST=0;
2 . MODULE   EXONE=1;
3 . INPUT    ARC1,ARC2,ARC3,ARC4;
4 . OUTPUT   ARC8;
5 . LINK     ARC5=NODE1(ARC1,ARC2);
6 . LINK     ARC6=NODE2(ARC3,ARC4);
7 . LINK     ARC7=NODE3(ARC5,ARC6);
8 . LINK     ARC8=NODE4(ARC7,    );
9 . FUNCTION NODE1=MUL,QUEUE(QUE1,1);
10. FUNCTION NODE2=MUL,QUEUE(QUE2,1);
11. FUNCTION NODE3=ADD,QUEUE(QUE3,1);
12. FUNCTION NODE4=OUT1(HOST,0);
13. MEMORY   QUE1=AREA(1);
14. MEMORY   QUE2=AREA(1);
15. MEMORY   QUE3=AREA(1);
16. START;
17. DATA     EXEC(EXONE,ARC1);
18. DATA     EXEC(EXONE,ARC2);
19. DATA     EXEC(EXONE,ARC3);
20. DATA     EXEC(EXONE,ARC4);
21. END.
```

Figure 5 : Data Flow Petri Net and associated assembler language of 7281 processor for the calculation of a matrix multiplication term [C] = [A].[B].

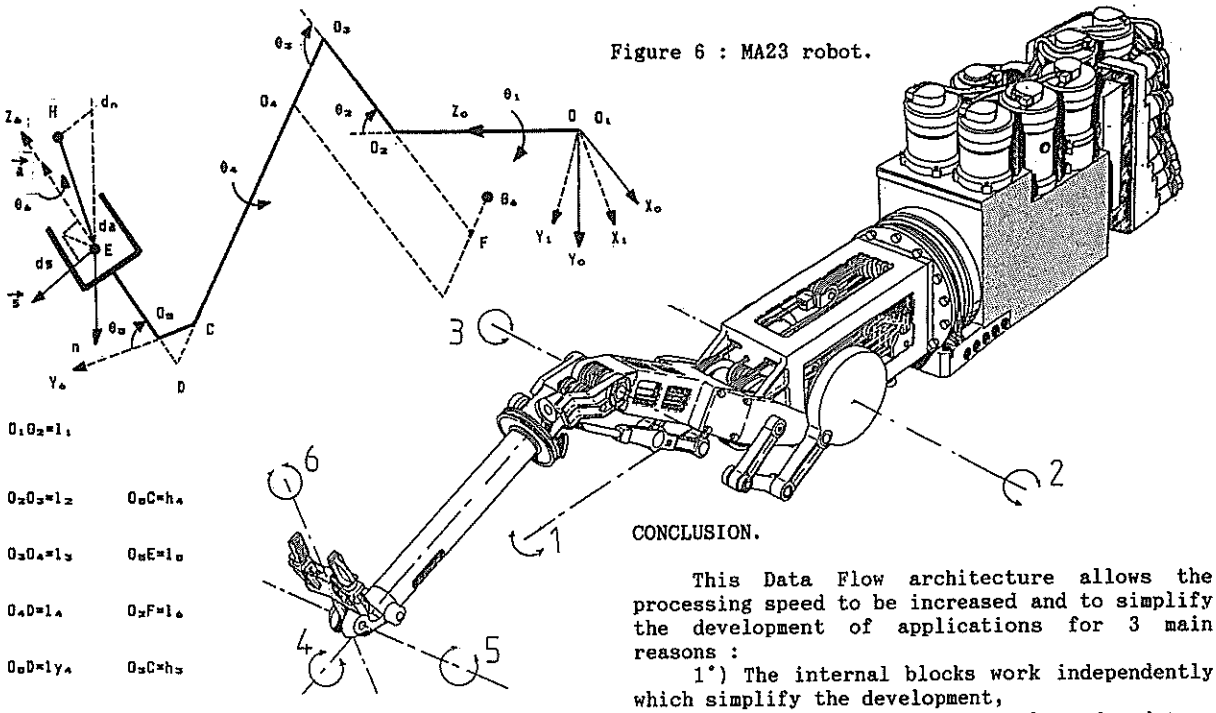


Figure 6 : MA23 robot.

5 - A ROBOTICS APPLICATION.

This example concerns a telemanipulator MA23 (fig 6). The following notations are used :

A : inertia matrix of the system (6*6) depending on the acceleration vector for the equation setup of the manipulator. Each term is a complex function of variable θ_i.

B_1 : position matrix of the system (6*6) depending on the position vector for the equation setup of the manipulator.

B_2 : friction matrix of the system (6*6) depending on the speed vector in the equation setup of the manipulator.

C_1 : weighting control matrix (6*6) of the command.

Q : vector (6*1) corresponding to the weight torques. Each term is a complex function of θ_i.

The manipulator modelling leads to represent its dynamical control by the following equation in which the time is a continous variable [8] :

$$[A]\ \underset{6*6}{\ddot{\underline{\theta}}}\ \underset{6*1}{} = \underset{6*1}{\underline{Q}} + [B_1]\ \underset{6*1}{\underline{\theta}} + [B_2]\ \underset{6*1}{\dot{\underline{\theta}}} + [C_1]\ \underset{6*1}{\underline{U}}$$

Matrix multiplications can be easily implemented on this fast Data Flow architecture (figure 5).

CONCLUSION.

This Data Flow architecture allows the processing speed to be increased and to simplify the development of applications for 3 main reasons :

1°) The internal blocks work independently which simplify the development,

2°) The command signals can be reduced to a minimum, because the tokens contain this information themselves,

3°) The block independence allows the individual test of each block.

KEYWORDS : Data Flow, Petri nets, Robotics.

BIBLIOGRAPHY.

[1] VERTUT, COIFFET : Les robots, tome 3 : la téléopération, Ed. Hermès 1984.
[2] VERTUT, CHARLES : La télésymbiotique. Colloque International sur l'exploitation des océans, Bx 156, pp 1-26, Bordeaux, Octobre 1974.
[3] VERTUT, CHARLES : Le MA23 - 6000 Télémanipulateur sous-marin à asservissement bilatéral et retour d'efforts. Colloque International sur l'exploitation des océans, Bx 155, pp 1-21, Bordeaux, Octobre 1974.
[4] ABELLARD, NOLIBE : Computation modelling with Data Flow Petri Nets. ICCI'89 Conference, Toronto, May 1989.
[5] ALMHANA : Modélisation par Réseaux de Petri à Flux de Données. Application à la synthèse de l'opérateur de Riccati rapide. Thèse d'Etat, Aix-Marseille, Juin 1983.
[6] ABELLARD, BARBAGELATA : Parallel processing modelling with Data Flow Petri Nets. First European Workshop on Parallel Processing Techniques for Simulation, UMIST, Manchester, October 1985.
[7] MESCHACH : Data flow IC makes short work of though processing chores. Electronic Design, pp 191-206, May 1984.
[8] DUMAS : Sur l'implémentation optimale de structures de commande sur calculateur en temps réel. Thèse d'Etat, Montpellier, Juin 1979.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

HIGHLY PARALLEL RADAR ARRAY SIGNAL PROCESSOR: WSI ARCHITECTURE

V. K. Jain **D. L. Landis**

Department of Electrical Engineering
University of South Florida
Tampa, Florida 33620, U.S.A.

A highly parallel WSI architecture is presented for a Radar Array processor. Its purpose is to perform real-time weight-vector computation for adaptive nulling. The major step involved in this computation is an L-U decomposition of a square matrix. Encouraged by the recent advances in wafer scale integration, we have mapped the algorithm to a systolic architecture for wafer implementation. As is well known, however, it is not possible to have all of the cells on wafer to be functional (with the current or forseeable future technology). Hence redundancy and reconfiguration have to be an integral part of a WSI design. Our design employs only two types of cells -- the MA and the R cells, thus facilitating restucturing through laser cutting and linking. The paper also discusses the timing and control of the internal and external switches to achieve the systolic flow of data.

I. INTRODUCTION

Considerable interest exists in massive parallel processing and its application to signal processing [1]-[6]. Typical applications include image filtering, phased-array radar, speech recognition and the like. A large subset of applications involves one common hurdle, namely the solution of a linear system of equations $\mathbf{A}\,\mathbf{x} = \mathbf{b}$ where $\mathbf{A}$ is a square matrix, $\mathbf{b}$ is a data vector, and $\mathbf{x}$ is the vector of unknowns. For a phased-array radar, adaptive nulling can be created by using a set of weights as given by the equation $\mathbf{R}\,\mathbf{w} = \mathbf{e}$ where $\mathbf{R}$ is the correlation matrix of the snap-shots of array observations, and $\mathbf{e}$ is some preference vector. Since the snap-shots are complex vectors, in general, the matrix $\mathbf{R}$ as well as the solution vector $\mathbf{w}$ are complex valued. However, the complex system of equations can be converted to an equivalent real system of equations. Specifically, consider (for N=2)

$$\begin{bmatrix} r_{11,R}+jr_{11,I} & r_{12,R}+jr_{12,I} \\ r_{21,R}+jr_{21,I} & r_{22,R}+jr_{22,I} \end{bmatrix} \begin{bmatrix} w_{1,R}+jw_{1,I} \\ w_{2,R}+jw_{2,I} \end{bmatrix}$$

$$= \begin{bmatrix} e_{1,R}+je_{1,I} \\ e_{2,R}+je_{2,I} \end{bmatrix}$$

Then it is readily shown that an equivalent system of equations is

$$\begin{bmatrix} r_{11,R} & -r_{11,I} & r_{12,R} & -r_{12,I} \\ r_{11,I} & r_{11,R} & r_{12,I} & r_{12,R} \\ r_{21,R} & -r_{21,I} & r_{22,R} & -r_{22,I} \\ r_{21,I} & r_{21,R} & r_{22,I} & r_{22,R} \end{bmatrix} \begin{bmatrix} w_{1,R} \\ w_{1,I} \\ w_{2,R} \\ w_{2,I} \end{bmatrix} = \begin{bmatrix} e_{1,R} \\ e_{1,I} \\ e_{2,R} \\ e_{2,I} \end{bmatrix}$$

We will denote this real system of equations as $\mathbf{A}\,\mathbf{x} = \mathbf{b}$, and its dimensionality as N. An efficient way of solving this set of equations is through forward and backward substitution steps, after performing an L-U decomposition on the system matrix: $\mathbf{A} = \mathbf{B}\,\mathbf{C}$, where $\mathbf{B}$ is lower triangular, and $\mathbf{C}$ is upper triangular.

The focus of the paper is a highly parallel architecture for L-U decomposition. The remainder of the paper has the following sections: II. Radar Processor and L-U Decomposition, III. L-U Decomposition Array Implementation, IV. Systolic Data Flow and Control, and V. Restructuring for WSI.

II. RADAR PROCESSOR AND L-U DECOMPOSITION

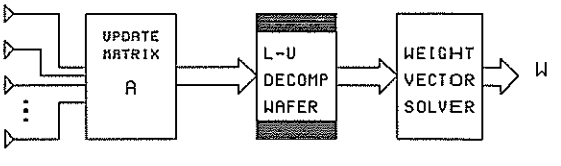
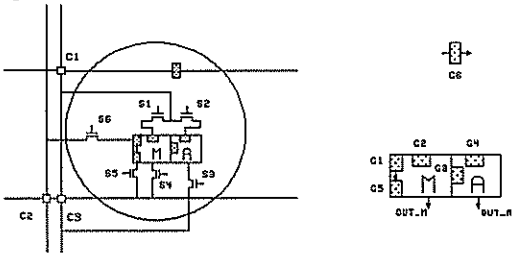


Fig. 1 Radar array processor

This research was supported by DARPA Grant No. MDA 972-88-J-1006

The paper presents a radar array processor (see Fig. 1), in particular a systolic architecture for L-U decomposition using a new reciprocal cell and a special MA cell. The MA cell shown in Fig. 2 is tailored for the L-U decomposition wafer so that it can perform any of the special functions required for the systolic architecture. The multiple functions are achieved by controlling six bus-switches S1 - S6 internal to the MA cell, and two bus-switches external to the cell C1 - C2. Thus, only a few control lines are needed to control the MA cells and the external switches. The external switch C3 is intended for defect tolerance and can be essentially disregarded for the purposes of this paper. The new reciprocal cell is not shown here due to lack of space but can be found in [7].



OUT_M = G1 x G2; G3 = Delay(G1 x G2);
OUT_A = G4- G3; G5 = Delay(G1)

Figure 2 MA cell with internal and external switches

III. L-U DECOMPOSITION ARRAY IMPLEMENTATION

The systolic array for the decomposition algorithm is given in Fig. 3. For simplicity we take N=3 in the figure as well as in the rest of the paper. However, the actual design is intended for N=8. The details of the data flow and the control of the switches are given in tabular form. The following notation is used

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ a_{21}/a_{11} & 1 & 0 \\ a_{31}/a_{11} & 0 & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}' & a_{23}' \\ 0 & a_{32}' & a_{33}' \end{bmatrix}$$

$$= \begin{bmatrix} 1 & 0 & 0 \\ a_{21}/a_{11} & 1 & 0 \\ a_{31}/a_{11} & a_{32}'/a_{22}' & 1 \end{bmatrix} \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ 0 & a_{22}' & a_{23}' \\ 0 & 0 & a_{33}'' \end{bmatrix}$$

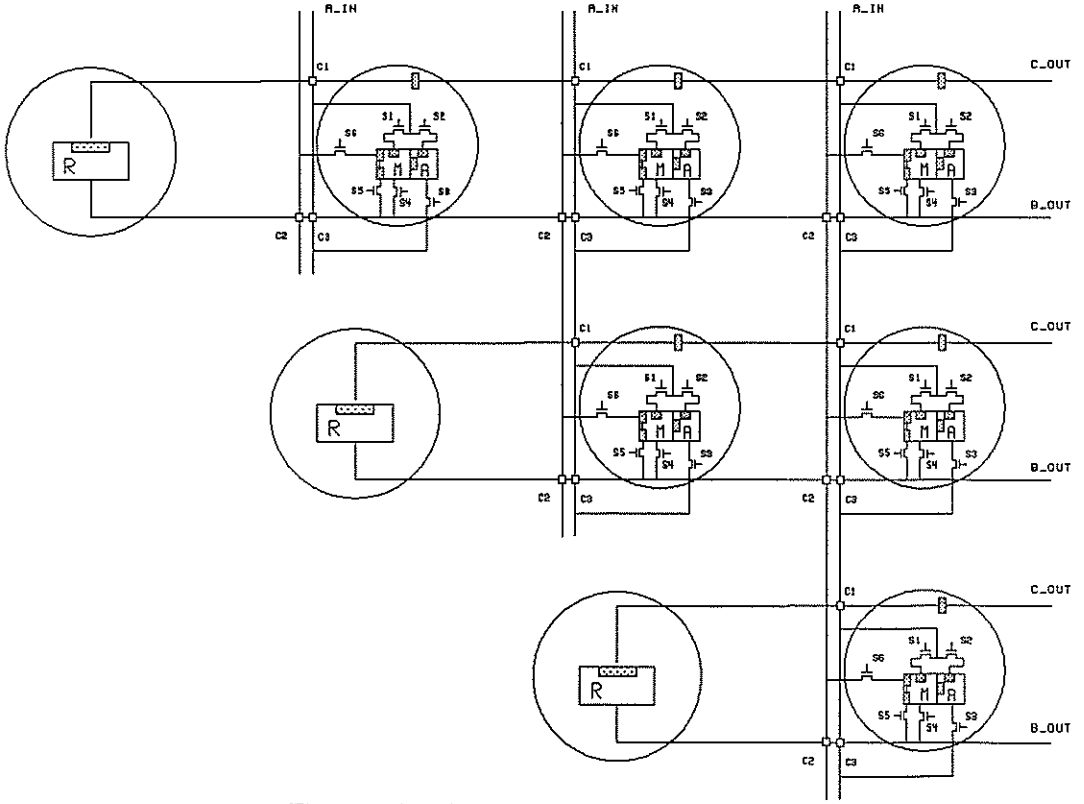


Figure 3 Systolic array for L-U decomposition (N=3 for simplicity)

The data of matrix **A** enters the array from the top through the A_IN buses. The results, matrix **B** and matrix **C**, are obtained from the right through the B_OUT and C_OUT buses, respectively.

IV. SYSTOLIC DATA FLOW AND CONTROL

The control logic of the external switches (C1, C2, and C3) used in the systolic array is given in Fig. 4.

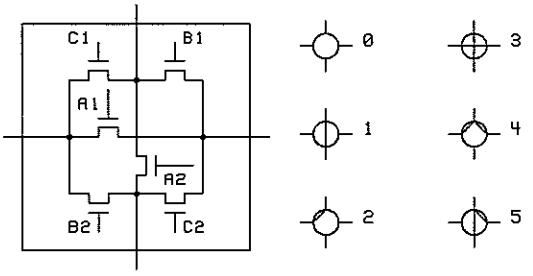


Figure 4 Control logic for external switches and settings used

Although the switch is capable of several possible settings, six specific settings are used to route the data. For these settings, also shown in Fig. 4., we will use the following short-hand notation:

O	(Setting 0) for 'no connect'
|	(Setting 1) for 'N-to-S' connection
/	(Setting 2) for 'N-to-W' connection
+	(Setting 3) for 'N-to-S' and 'W-to-E' connections
^	(Setting 4) for 'N-to-E' and 'N-to-E' connections
|\	(Setting 5) for 'N-to-S' and 'N-to-E' connections

Due to lack of space only the behavior and control of the upper two nodes, i.e., cells R_11+MA_11, and MA_12 are described. These are given in a tabular form in Tables 1, and 2. In interpreting these tables, the following comments are helpful.

- The dash '-' implies don't care (or data invalid)
- The equal-to sign '=' implies that the entry equals the entry in the previous row
- The column R in Table 1 pertains to the register in Cell R_11
- Register G5 is not shown in the tables because its entry always equals the contents of register G1 in the previous row
- The value $a_{22}' = a_{22} - (a_{21}/a_{11})a_{12}$ becomes available during the interval (4,5) and is latched into both R_22 and G6 of MA_22 at time instant 5. Similarly, a_{23}' is latched into both R_23 and G6 of MA_23 at time instant 7.

V. RESTRUCTURING FOR WSI

Only two types of cells are employed, R and MA. The R cell represents a new reciprocal design presented elsewhere [7]. The MA cell provides simple arithmetic multiply and add (actually, subtract) operation. The logical array provides for the L-U decomposition of an 8x8 real matrix, and therefore requires 8 R cells and 36 MA cells. The physical wafer will be provided with 14 R cells and 70 MA cells. The switches C1, C2 and C3, together with laser linking and cutting on wafer-level tracks, will be used for extracting the functional wafer from the physical wafer. Restructuring for defect tolerance by laser linking and cutting will be performed in house. The 'laser linking and cutting' facility has been installed at the University of South Florida as an integral part of DARPA wafer scale integration project.

VI. CONCLUSIONS

A systolic wafer scale processor has been described. The architecture is based on the processing of a real square matrix. The estimated time for such a decomposition is (20 clock cycles x 125 ns) = 2.5 microsec. Although we have described the L-U decomposition of only an 8x8 matrix two alternatives are available for larger matrices. First, it is possible to use this wafer processor as a work-horse and use external switches and memory to route appropriate segments of data. Such an approach has been taken in [8] for the FFT algorithm. This approach achieves lower cost but at the expense of speed. Alternatively, several wafers can be configured together to achieve extremely high speed.

ACKNOWLEDGEMENTS

The authors are deeply indebted to Dr. Earl E. Swartzlander, Jr. for his helpful comments on this research.

TABLE 1
Data flow and control for MA_11 and R_11

t	IN	G1	G2	G3	G4	G6	R	S1	S2	S3	S4	S5	S6	C1	C2	C3
0	a11	-	-	-	-	-	-									
								0	0	0	0	0	0	^	O	O
1	a21	-	-	-	-	a11	a11									
								1	0	0	0	0	1	|	/	O
2	a31	1/a11	a21	-	-	-	-									
								1	0	0	1	0	0	|	O	O
3	-	1/a11	a31	-	-	-	-									
								0	0	0	0	0	0	O	O	O
4	-	-	-	-	-	-	-									
								=	=	=	=	=	=	=	=	=
5	=	=	=	=	=	=	=									

TABLE 2
Data flow and control for MA_12

t	IN	G1	G2	G3	G4	G6	S1	S2	S3	S4	S5	S6	C1	C2	C3
0	-	-	-	-	-	-									
							0	0	0	0	0	0	O	O	O
1	-	-	-	-	-	-									
							0	0	0	0	0	0	O	O	O
2	a12	-	-	-	-	a11									
							1	0	0	0	0	1	|\	/	O
3	a22	a21/a11	a12	-	-	a12									
							0	1	0	0	0	1	|	/	O
4	a31	a31/a11	a12	(a21/a11)a12	a22	-									
							0	1	1	0	1	0	|	O	|
5	-	-	-	(a31/a11)a12	a32	-									
							0	0	1	0	1	0	O	O	|
6	-	-	-	-	-	-									
							0	0	0	0	0	0	O	O	O
7	=	=	=	=	=	=									
							=	=	=	=	=	=	=	=	=

REFERENCES

[1] Proceedings of the International Conference on Wafer Scale Integration, January 1990 (Published by IEEE Computer Society Press).

[2] J. I. Raffel, et. al., "A wafer scale digital integrator," Proc. IEEE Conference on Computer Design, p. 121, 1984.

[3] G. J. Li, and B. W. Wah, "The design of optimal systolic arrays," IEEE Transactions on Computers, vol. C-34, pp. 66-77, Jan. 1985.

[4] Partitioning and mapping algorithms into fixed size systolic arrays," IEEE Transactions on Computers, vol. C-35, pp. 1-12, Jan. 1986.

[5] J. P. Petrolli, "A systolic device for image transformation," in Wafer Scale Integration, Proc. IFIP Workshop on WSI, North Holland, 1986.

[6] K. Yamashita, A. Kanasugi, S. Hijiya, G. Goto, N. Matsumura, and T. Shirato, "A wafer scale 170,000 gate FFT processor with built in test circuits," IEEE Trans. on Solid State Circuits, Vol. SC-23, pp. 336-342, 1988.

[7] V. K. Jain, D. L. Landis, and C. E. Alvarez, "Wafer scale L-U decomposition array with a new reciprocal cell," Proc. International Conference on Computer Design, Oct. 1989.

[8] H. Hikawa, and V. K. Jain, "20 Million samples/s wafer processor FFT architecture," Proc. EUSIPCO-1990, Sept. 1990.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A MOTION ESTIMATOR REALIZED WITH VLSI-CHIPS SUITABLE FOR EXPERIMENTS ON A LOW BIT RATE PICTURE PHONE

J. Kraus, H. Wendt+, J. Sudheimer, G. Schuch

Deutsche Bundespost TELEKOM, Forschungsinstitut beim Fernmeldetechnischen Zentralamt, Postfach 100003, 6100 Darmstadt, FRG

Most picture codecs with a high data compression make use of motion estimation. Only few basic circuits are needed for the development of a VLSI-chip for motion vector calculation in a block matching algorithm for full search. Some of these chips, a picture storage and a few controlling elements form the entire displacement estimator. This configuration allows some variation of the block size and the search window within certain limits.

1. INTRODUCTION

For the transmission of moving pictures in a low bit rate channel (e.g. the 64 kbit ISDN) several complex coding algorithms working together are required to achieve the necessary picture quality. Besides cosine transformation and movement adapted interpolation most picture codec proposals make use of a displacement estimator for motion compensation as one of the basic key elements of the codec [1]. Given two successive pictures, you find out by a pixel matching algorithm where a coherent pixel quantity has moved. Then the according motion vector has to be computed. The only information that has to be transmitted consists of the several motion vectors and the difference between the actual and estimated picture, which is mainly the new visible background. If the pixel quantities under consideration are square blocks, the estimation method is called "block matching". If we take into account every possible motion of these blocks in a well defined search window, then we call this algorithm "full search".

Since an experimental, flexible system with variable block size and search window should be developed for the high computational effort of full search within a limited amount of time and manpower, a simple architecture with high parallel structures and only few, often applied basic circuits must be found. Furthermore the architecture of the motion estimator should be independent of the data organisation of the other codec elements.

2. ALGORITHMIC ASPECTS

As match criterion an error sum is defined. For every possible displacement v of a block B in a search window U the sum of the absolute difference between the luminance values of the block in the picture k and the displaced block in the picture k-1 is calculated (fig. 1). So we investigate where a block in the actual picture comes from.

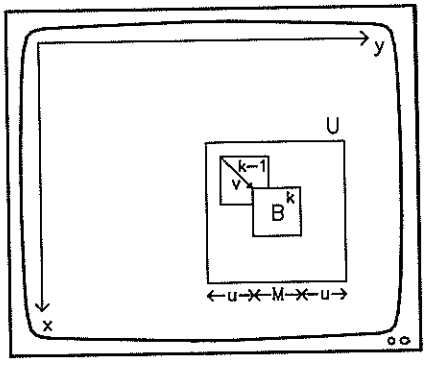


Fig. 1: Illustration of block B, search window U and displacement vector v

(1) $$G_k^B(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{M-1} |s_k(a+m,b+n) - s_{k-1}(a+m-i,b+n-j)|$$

$s_k(x,y)$: luminance of a pixel in the picture k, [0 - 255]

v=(i,j): examined displacement vector $|i|, |j| \leq u$, $(2u+1)^2$: size of the search window U

B(a,b) : examined block a = 0,M,2M, ...Z-M
b = 0,M,2M, ...L-M
M^2:block size
Z :number of lines
L :number of pixels per line

The minimum of this sum defines the motion vector of a block B. If there exists more than one minimum, the vector with the smallest length should be chosen.
The following real values are assumed for a first experimental system: M=8 u=7 L=352 Z=288

3. ARCHITECTURE

A special processor was developed for the calculation of G for one vector v=(i,j) (fig. 2). Corresponding to equation (1) substraction, absolute value and accumulation must be done respectively. Large values of G show that the according vector is not suitable for a motion estimation. Therefore, the values of G are limited to 12 bit.

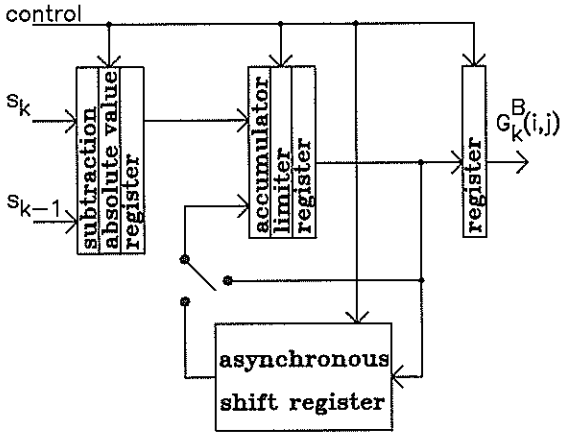


Fig. 2: Processor for the calculation of $G_k^B(i,j)$

While the camera presents the pictures line by line and not block by block and we want to avoid an extensive reorganization, which would cut the freedom for the architecture of the other components, the calculation for the next block starts after computing the inner sum of equation (1). So an intermediate result is to be stored, which is done in an asynchronous shift register (FIFO). When working on the last line of a block we get the final result of the error sum G.

For the computation of the values of G for all $|i|, |j| \leq u$ of a search window $(2u+1)^2$ processors are necessary. Since it is not possible to bring them all on one chip for $u > 2$, we realized one line of a search window with u=7, i.e. 15 processors. A shift register chain makes the necessary shift of the picture k-1 in line direction (fig.3). To determine the minimum of G in one line of the search window a comparator tree was built. In addition each chip contains one element of a pipelined comparator chain, which compares the internally calculated minimum of G of the line i with the external applied minimum of the lines u...i+1. Since the corresponding vectors accompany the values of G, the last chip of the chain delivers the minimum of G and the motion vector for all blocks.

Every chip must be supplied with another line of the search window. Therefore the circuit contains a line storage element, so that all lines of the search window can be generated by delaying from chip to chip.

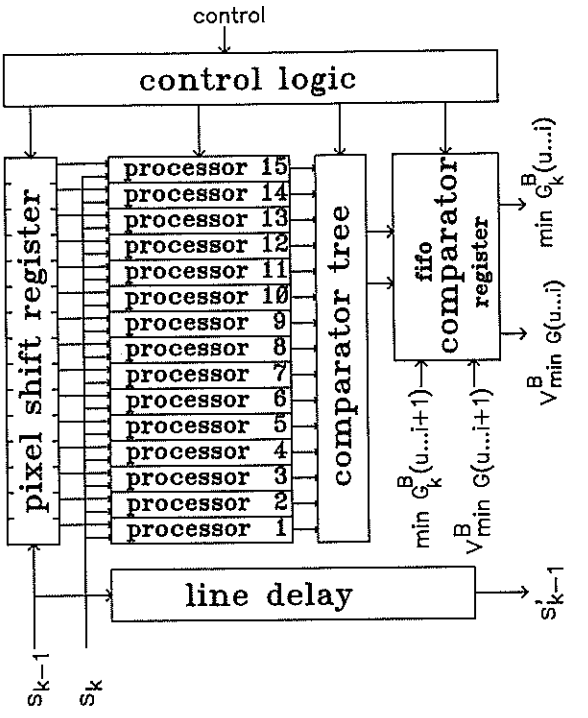


Fig. 3: Schematic of one chip

The schematic of a displacement estimator realization is shown in fig. 4. For experiments the blocksize can be changed by the controlling circuits from 8x8 up to 16x16. The search window can be manipulated by the addition of another chip per line and overlapping processing.

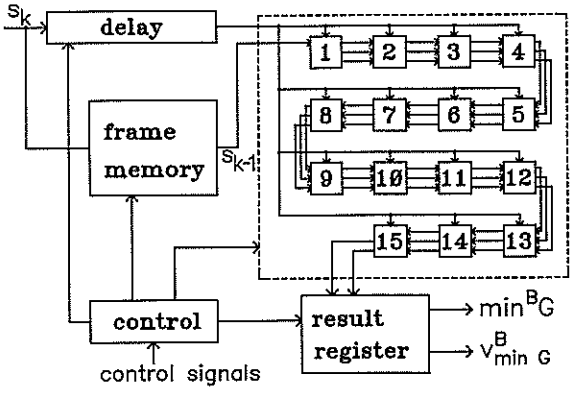


Fig. 4: Schematic of a displacement estimator realization (1...15: chip of fig.3)

4. IC-DESIGN

A 1.5 µ CMOS technology with two layer metal wiring was used. The subtraction and accumulation (fig. 1) are implemented by fast transmission gate adders [2]. To save time, carries and data are treated alternately negative. The asynchronous shift registers (FIFO) consist of a static D-Flip-Flop and a controlling cell for each storage column. Each controlling cell is connected to its neighbour cell by two handshake lines. In the cell the state (empty or full) of each column is fixed. As soon as a column is filled up, the controlling cell tells the next one that data are available. If the column of this cell is empty, the data are taken over and then the previous control cell is set to empty. The advantage of such a memory is that the number of data to be stored can be changed in certain limits allowing the variation of the blocksize.
The compare operations on the chip are performed as subtraction. Since only the sign of the result is of interest, the carry part of a standard adder chain is sufficient [2,3]. The line delay is fully static.
Figure 5 shows the layout of the chip. The circuit consists of 150 000 transistors and its size is 7.6 x 6.9 mm^2. For packaging a ceramic pin grid array with 132 pins was chosen, 96 pins being used. The power consumption is smaller than 500 mW. The chip is projected for a clock rate of 13.5 MHz and therefore the circuit can also be used for digital television. At this data rate one chip performs 400 million 8 bit calculating operations per second.
With a control signal each path in the comparator tree is selectable, which allows the test of each processing element.

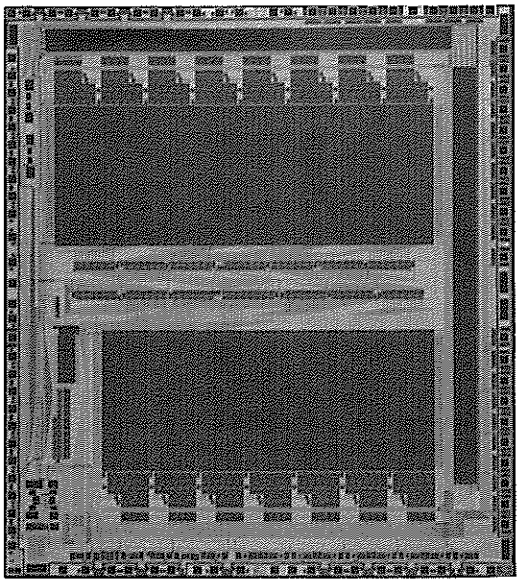


Fig. 5: Layout to the schematic of fig. 3

5. CONCLUSIONS

This contribution shows how by a simple and straight forward architecture and only few basic cells a VLSI circuit, flexible enough for experiments on motion estimation with full search, could be developed in a reasonable amount of time. If the parameters like block size and search window are fixed in a standardized system and if this system is restricted to a small data rate, then a displacement estimator can be implemented on a single chip [4,5].

ACKNOWLEDGEMENTS

The authors would like to thank K. Merzenich and A. Plasberg for the printed circuit board conception, B. Schneider for a lot of layout work, J. Schaaf and D. Dieluweit for the maintenance of the CAD System and INTERMETALL, Freiburg, for providing the CMOS technology.

REFERENCES

[1] May, P., Weniger Daten - bessere Bilder, Funkschau 10 (1987) 42.

[2] Weste N. and Eshraghian, K., Principles of CMOS VLSI Design (Addison-Wesley, Reading, Massachusetts, 1985).

[3] Allen, J., Introduction to VLSI Design, in: Randell, B. and Treleaven, P.C. (eds.), VLSI architecture (Prentice-Hall, Englewood Cliffs, New Jersey, 1983).

[4] Pirsch, P., Systemstrategien für die schnelle digitale Signalverarbeitung, in: Gallenkamp, W. (ed.), Nachrichtentechnik im Zeichen der Mikroelektronik (Professorenkonferenz im FTZ, Darmstadt, 1987).

[5] Schwerzel, W., Eine Signalprozessorarchitektur zur Bewegungsvektorsuche für die Bewegtbildcodierung im ISDN-Bildtelefon, in: Gallenkamp, W. (ed.), Nachrichtentechnik im Zeichen der Mikroelektronik (Professorenkonferenz im FTZ, Darmstadt, 1987).

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SYSTOLIC IMPLEMENTATION OF FIR FILTERS

F. El-Guibaly, S. Sunder, and A. Antoniou

Department of Electrical and Computer Engineering,
University of Victoria, P.O. Box 1700, Victoria, B.C., Canada, V8W 2Y2

Two approaches for the implementation of FIR filters are used to derive linear and triangular systolic arrays. The approaches yield structures with the maximum data rate possible, i.e., a new input sample is supplied and a new output sample is obtained every sampling period. The structures derived are simple, modular, expandable and can be readily cascaded for the implementation of higher-order filters.

1. INTRODUCTION

Systolic designs can be applied to any compute-bound problem that is based on regular recurrence relations [1]-[2]. Consequently, they have been used effectively in the areas of digital filtering, digital signal processing, matrix algebra [3]-[4]. Several types of systolic arrays have been proposed such as linear, triangular, and hexagonal [5]-[6].

In this paper, two systematic approaches are used to derive systolic architectures for finite impulse response (FIR) digital filters. One approach involves manipulating the flow graph representation and the other involves manipulating the z transform of the difference equation of the filter.

2. FIR EQUATIONS AND ARCHITECTURES

An FIR filter can be represented by [7]

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \tag{1}$$

where N is the length of the filter. Our computation model assumes that all processing elements (PEs) contain a storage element at their output. As a consequence, (1) is modified as

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-1-k) \tag{2}$$

where $x(-1) = 0$.

Systolic architectures can be derived using an algebraic approach. Equation (2) can be written in the z domain as

$$Y = \sum_{k=0}^{N-1} h(k)z^{-(k+1)}X \tag{3}$$

$$= z^{-1}\left(h(0)X + z^{-1}(h(1)X + \cdots + z^{-1}(h(N-1)X)\cdots)\right) \tag{4}$$

where $Y \equiv Y(z)$ and $X \equiv X(z)$.

The above equation can be mapped onto a systolic architecture as depicted in Fig. 1 for a filter of length four where dashed lines enclose PEs. As can be seen, each PE has two inputs and one output. One of the inputs is signal X and the other is the partial sum from an adjacent PE. The output signal is the partial sum produced in that PE delayed by one sampling period. The main disadvantage of this realization is that signal X is broadcast to the various PEs and, consequently, as the length of the filter increases signals may become skewed owing to the increased length of communication lines.

Signal broadcast can be eliminated by using an approach based on signal flow graphs. The dependence graph of (2) is shown in Fig. 2 where both the inputs and the weights of the filter are broadcast throughout the index space (n, k). The first step in the systolic implementation of (1) is to convert it to a single assignment code as

$$y_n^k = y_n^{k-1} + h(k)x(n-k) \tag{5}$$

$$y(n) = y_n^{N-1} \tag{6}$$

for $k = 0, 1, \cdots, N-1$ where $y_n^{-1} = 0$ for an initially relaxed filter. The dependence graph of (5) is shown in Fig. 3. If the arrow directions in Fig. 3 were reversed and the choice of the projection and scheduling vectors made to satisfy (8) and (9) below, one can obtain the semi-systolic linear array of Kung [3]. Figure 3 indicates that the input data $x(n)$ as well as the filter weights $h(n)$ are broadcast.

In order to eliminate the input signal broadcast, we modify (2) as

$$y_n^k = y_n^{k-1} + h_n^k x_n^k \tag{7}$$

where

$$x_n^k = x_{n-1}^{k-1},\ x_n^0 = x(n),\ h_n^k = h_{n-1}^k,\ h_0^k = h(k)$$

The modified dependence graph is shown in Fig. 4. The projection vector $\vec{d}$ determines the array configuration. All nodes lying along a straight line parallel to $\vec{d}$ are assigned to one PE. The scheduling vector $\vec{s}$ has to satisfy the two conditions

$$\vec{s}\cdot\vec{e} > 0 \quad \forall\, \vec{e} \tag{8}$$
$$\vec{s}\cdot\vec{d} > 0 \tag{9}$$

where $\vec{e}$ represents any arc in the signal flow graph of the algorithm. Equation (8) states that all dependence arcs flow in the same direction across the hyperplanes which ensures causality in any scheduling scheme. Equation (9) states that hyperplanes cannot be parallel to the projection vector which ensures that the nodes on a hyperplane will not be projected onto the same processor. The hyperplanes in Fig. 4 represent different time instants.

A linear systolic array based on the graph of Fig. 4 is shown in Fig. 5 for a filter of length four. In this array a new input sample is supplied and a new output sample is obtained for every sampling period. The I/O data rates are equal to the processing rate.

A realization of the systolic array of Fig. 5 can be obtained by writing (3) as

$$\begin{aligned} z^{-(N-1)}Y &= \sum_{k=0}^{N-1} z^{-(N-1-k)}h(k)Xz^{-(2k+1)} \qquad (10)\\ &= z^{-1}\Big(h(N-1)Xz^{-(2N-2)}\\ &\quad +z^{-1}\big(\cdots+z^{-1}\big(h(1)Xz^{-2}\\ &\quad +\ z^{-1}\big(h(0)X\big)\big)\cdots\big)\Big)\\ &= z^{-1}\Big(h(N-1)X_{N-1}+z^{-1}\big(\cdots\\ &\quad +z^{-1}\big(h(1)X_1\\ &\quad +\ z^{-1}\big(h(0)X_0\big)\big)\cdots\big)\Big) \qquad (11)\end{aligned}$$

where $X_k = z^{-2}X_{k-1}$, for $k = 1,\ 2,\ \cdots,\ N-1$ and $X_0 = X$. The mapping of (11) onto a systolic architecture is shown in Fig. 6. The PE in this realization, like that in Fig. 1, has two inputs and one output. However, the number of storage elements is increased.

Some systolic arrays and pipelined computing structures have local memory to store data for use in subsequent operations. These data could be input data samples or intermediate results. The local memory is usually in the form of first-in-first-out (FIFO) elements of linearly increasing sizes [8]. It would be interesting to see if such architectures can support digital filtering. Below we show that this indeed is the case.

Equation (2) can be expressed as

$$y_n^k = y_n^{k-1} + p_{n-k}^k \tag{12}$$

where

$$\begin{aligned} p_n^k &= p_{n-1}^k, \quad x_n^k = x_n^{k-1}, \quad x_n^0 = x(n),\\ h_n^k &= h_{n-1}^k, \quad h_0^k = h(k), \quad p_{n-k}^k = h_{n-k}^k\, x_{n-k}^k\end{aligned}$$

for all $n,\ k \geq 0$. The input samples are propagated vertically as shown in Fig. 7 and the partial products for a particular output are evaluated at the same instant. This can be seen by examining the node activities at each hyperplane. Choosing the projection vector $\vec{d} = [1\ 0]^T$, we get the systolic array illustrated in Fig. 8 for a filter of length four. The zeroth PE in Fig. 8 can be represented by

$$\begin{aligned} y_n^0 &= y_n^{-1} + p_{n-1}^0, \quad y_n^{-1} = 0,\\ p_{n-1}^0 &= h(0)x_{n-1}^0, \quad x_n^1 = x_n^0\end{aligned}$$

On the other hand, the kth PE for $k = 1, 2, \cdots, N-1$, can be represented by

$$\begin{aligned} y_n^k &= y_n^{k-1} + p_{n-k}^k, \quad p_{n-k}^k = p_{n-k+1}^k,\\ p_{n-k+1}^k &= p_{n-k+2}^k, \quad \cdots, \quad p_{n-2}^k = p_{n-1}^k,\\ p_{n-1}^k &= h(k)x_{n-1}^k, \quad x_n^{k+1} = x_n^k\end{aligned}$$

At any time instant, the PEs of the array produce all the partial products for a particular output sample and each PE stores one of the filter weights. We notice from Fig. 8 that each PE in the 1-D array performs two separate operations: multiplication and addition. The multiplication operation forms all the partial products for a particular output sample. On the other hand, the addition operation forms the summation of the relevant partial products to produce the outputs. In each PE, there is also a first-in-first-out (FIFO) memory whose size increases linearly with the position of the PE. Thus the topmost PE will be associated with a FIFO memory of length $N-1$. We can simplify the functionality of each PE and at the same time speed up the processing rate if the multiplication and addition operations are done by separate PEs. A triangular systolic architecture based on Fig. 8 is shown in Fig. 9.

The structure of Fig. 9 can also be obtained by writing (10) as

$$\begin{aligned} z^{-(N-1)}Y &= z^{-1}\Big(\cdots z^{-1}\Big(z^{-1}\big(h(0)X\big)\\ &\quad +z^{-1}\Big(h(1)Xz^{-1}\Big)\Big)+\cdots\\ &\quad +z^{-(N-1)}\Big(h(N-1)Xz^{-(N-1)}\Big)\Big)\end{aligned} \tag{13}$$

3. APPLICATIONS

The 1-D systolic arrays for FIR filters presented here can be used to derive systolic arrays for 1-D IIR filters as well as 2-D FIR and IIR filters [9]-[10].

4. CONCLUSIONS

Two systematic approaches for the hardware implementation of FIR filters have been proposed and have been used to derive linear or triangular structures. Because of the systematic approach used compared to the earlier heuristic approaches, structures with the maximum data rate possible have been obtained, i.e., a new input sample is supplied and a new output sample is obtained every sampling period. The structures derived are simple, modular, and expandable and can be readily cascaded for the implementation of higher-order filters.

ACKNOWLEDGEMENTS

The authors are grateful to the Natural Sciences and Engineering Research Council of Canada for supporting this research.

REFERENCES

1. S. K. Rao, *Regular Iterative Algorithms and their Implementations on Processor Arrays*, Ph.D. thesis, Stanford University, Stanford, California, 1985.

2. S. Y. Kung, *VLSI Array Processors*, Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1988.

3. H. T. Kung, "Why systolic architectures?," *IEEE Computer*, vol. 25, pp. 37-46, Jan. 1982.

4. M. A. Sid-Ahmed, "A systolic realization of 2-D filters," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-37, pp. 560-565, Apr. 1989.

5. P. S. Liu and T. Y. Young, "VLSI array architecture for picture processing," in *Picture Engineering*, K.S. Fu and T. Kunii, Eds. New York: Springer Verlag, pp. 171-186, 1982.

6. P. S. Liu and T. Y. Young, "VLSI array design under constraint of limited I/0 bandwidth," *IEEE Trans. Comput.*, vol. C-32, pp. 1160-1170, Dec. 1983.

7. A. Antoniou, *Digital Filters: Analysis and Design*, McGraw-Hill, New York, 1979.

8. B. C. McKinney and F. El-Guibaly, "A multiple-access pipeline architecture for digital signal processing," *IEEE Tran. Comput.*, vol. 37, pp. 283-290, Mar. 1988.

9. S. Sunder, F. El-Guibaly, and A. Antoniou, "Implementations of 2-D FIR filters using systolic arrays," *Canadian Conference on Electrical and Computer Engineering*, pp. 658-661, 1989.

10. S. Sunder, F. El-Guibaly, and A. Antoniou, "Systolic Implementation of two-dimensional recursive digital filters," to be presented at *Intl. Symp. Circuits Syst.*, New Orleans, 1990.

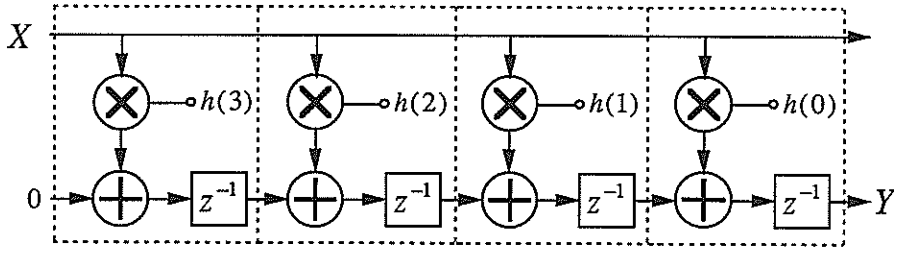


Fig. 1 Mapping of (4) onto a systolic architecture for a filter of length four.

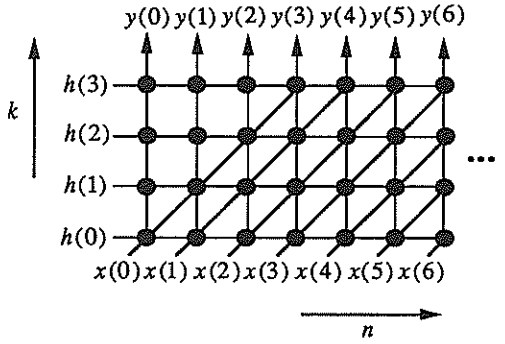


Fig. 2 The dependence graph for the FIR filter equation

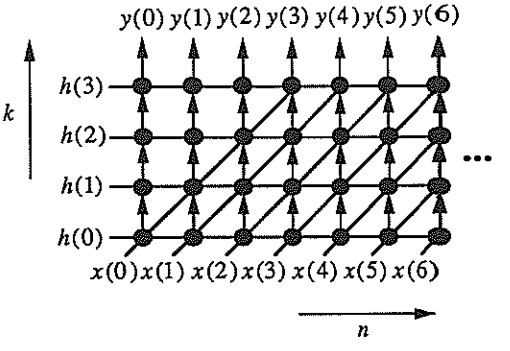


Fig. 3 The dependence graph for the modified FIR filter equation in which the filter outputs are propogated.

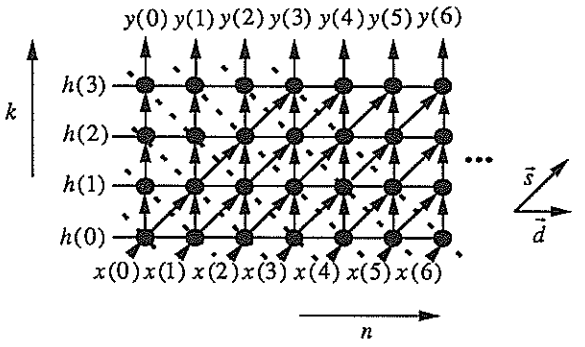


Fig. 4 The dependence graph for the modified FIR filter equation in which the inputs and the outputs are propogated.

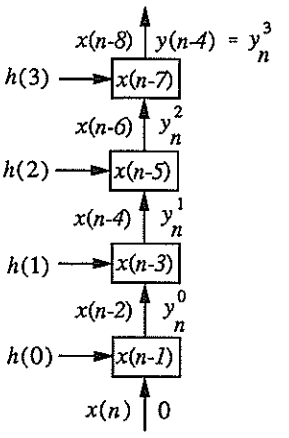


Fig. 5 A systolic array based on the signal flow graph of Fig. 4 for a filter of length four.

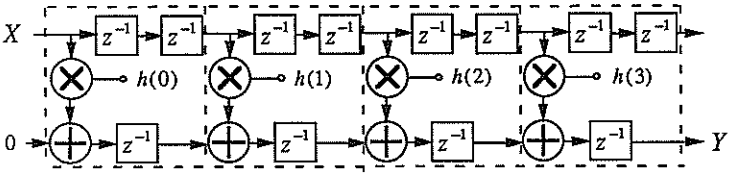


Fig. 6 Mapping of (11) onto a systolic architecture for a filter of length four.

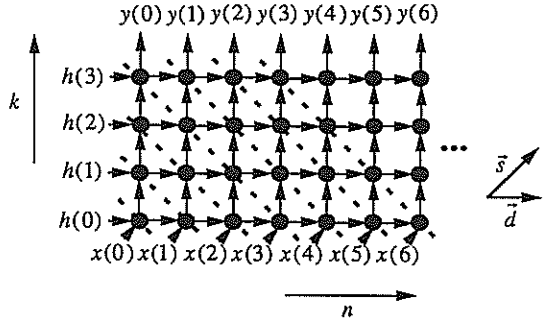


Fig. 7 Signal flow graph for the modified form of Fig. 4 showing the redirection of the filter inputs and and the formation of the partial products at the hyperplanes.

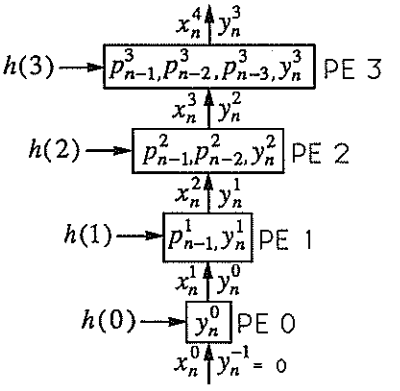
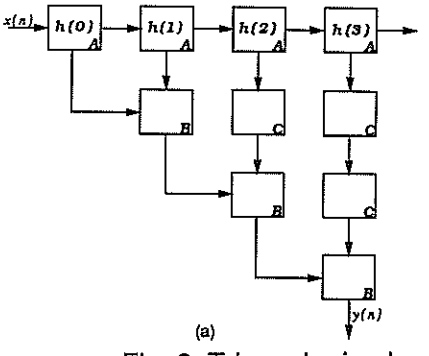


Fig. 8 Systolic array based on the signal flow graph of Fig. 7 for a filter of length four.



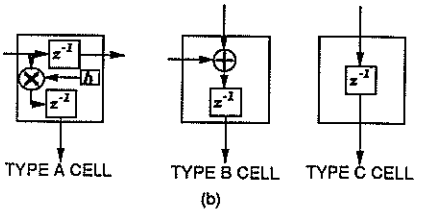


Fig. 9 Triangular implementation based on the array of Fig. 8 (a) Block diagram (b) Details of PEs

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

MAPPING DIFFERENT FIR FILTER BANKS ONTO A SYSTOLIC ARRAY OF FIXED SIZE AND FIXED STRUCTURE

Nikolay Petkov

University of Erlangen-Nürnberg, Institute of Mathematical Machines
and Data Processing (III), Martensstraße 3, 8520 Erlangen, West Germany

It is shown how different FIR filter banks can be mapped onto a systolic array of fixed size and fixed structure. The technique presented is based on the time-sharing properties of the *c*-slow circuits. It is proposed to use a *c*-slow systolic array for the flexible realization of filter banks of different size and structure. The model has been varified on an array of Transputers and holds high potential for VLSI-chip implementation.

1. INTRODUCTION

The systolic array implementation of the convolution and the related FIR filters is one of the problems in the area of systolic algorithm research which have been studied most exhaustively [1-15]. (The reader is referred to the monograph under [4] for an extensive introduction to systolic arrays, and for a collection of many systolic algorithms for different problems.)

The systolic convolvers given in the literature require processor arrays whose size depends on the size of the problem to be solved. Thus the convolution of a digital signal with a set of N coefficients would typically require a systolic array of $N/2$ or N processors. On the other hand, the system to be used for implementation - this might be a VLSI chip or a parallel computer - would typically have a fixed size and a fixed structure. The problem becomes even more complex when a whole filter bank with many channels having coefficient sets of different size has to be implemented in a processor array of fixed size and fixed structure.

In this paper, it is shown how systolic FIR filter banks of arbitrary size and structure can be efficiently implemented in a systolic processor array of fixed size and fixed structure. The approach used is based on the theory of the so called *c*-slow circuits and gives a discipline for mapping of FIR filter banks onto a linear systolic array of fixed size [12-14]. The paper is organized as follows: Elements of the theory of the *c*-slow circuits are presented in Section 2. In Section 3, the technique is illustrated on several examples of filter banks. A summary is given in Section 4.

2. C-SLOW CIRCUITS AND THEIR APPLICATION

The technique proposed here is actually a time-sharing method for the use of logic circuits. We illustrate it on a simple example, since this might be more illuminating than giving a general theory. Figure 1a shows a combinatorial circuit which, when connected to a register as shown in Figure 1b, is capable of computing the differential output signal $y_i = x_i - x_{i-1}$ for an input digital signal x_i, $i = \dots, 0, 1, \dots$. Let us change the circuit shown in Figure 1b by replacing the register by two (in general by c) chained registers, Figure 1c. The new circuit is capable of realizing the same function as the original one, if the time intervals between consecutive data units are increased twice (in general c times), Figure 1c. (The time intervals are measured in the number of cycles of the clock which controls the registers.) Since the new circuit is two (generally c) times slower than the original one, it is called a 2-slow (generally c-slow) version of the original circuit [16]. A c-slow circuit can be used for the concurrent execution of c independent computational problems. Figure 1d illustrates the concurrent differentiation of two input signals x_i and x'_i, $i = \dots 0, 1, 2 \dots$. The 2-slow circuits operates in turn on the two input signals computing the values of the respective output signals in the clock periods for which holds $t_{\text{mod } 2} = 0$ and $t_{\text{mod } 2} = 1$.

Figure 2 shows how the method described above can

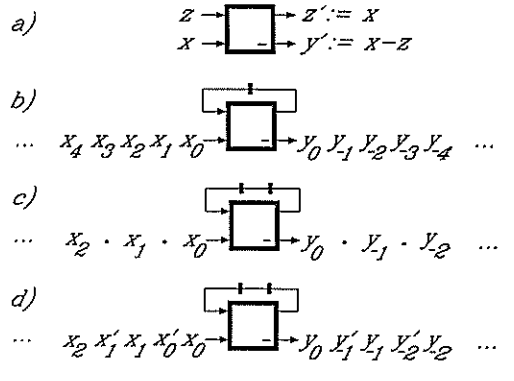


Figure 1 Construction and use of *c*-slow circuits

be applied to array structures. Three differentiating circuits forming an array are active on three independent processes: $y_i = x_i - x_{i-1}$, $y'_i = x'_i - x'_{i-1}$, $y''_i = x''_i - x''_{i-1}$, $i = \ldots, 0, 1, 2 \ldots$, Figure 2a. One 3-slow differentiating circuit can execute all three processes, Figure 2b. It is active on the first, second and third process in the clock periods for which holds $t_{\text{mod } 3} = 0$, $t_{\text{mod } 3} = 1$, and $t_{\text{mod } 3} = 2$, respectively. In this way, a c-slow version of an appropriate $(1/c)$-th part of a homogeneous array can execute the task of the whole array. The method can be generalized for the case in which the parts of the circuit are interconnected. Feedbacks and multiplexers have to be added to the model to take account of the interconnections between the parts. For this case and for further details, the reader is referred to [13].

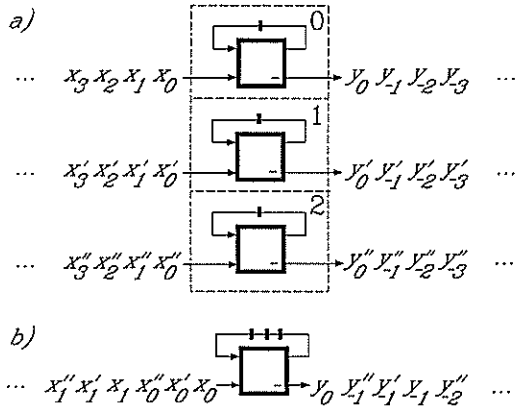


Figure 2 A c-slow version of a $(1/c)$-th part of a homogeneous array can do the work of the whole array

3. MAPPING FILTER BANKS

For simplicity, only small-size examples are considered in the following. Figure 3 shows a three-cell systolic array for implementing the convolution (FIR filter) $y_i = \sum_{j=0}^{N-1} a_j x_{i-j}$ of an input digital signal x_i, $i = \ldots -1, 0, 1, \ldots$, with a set $\{a_j | j = 0, 1, \ldots, N\text{-}1\}$ of three coefficients ($N = 3$). Figure 4 shows a 4-slow version ($c = 4$) of the array shown in Figure 3. Each register of the array in Figure 3 has been replaced by a chain of four registers. The number of clock periods between consecutive input/output operations is increased four times. The 4-slow systolic array shown in Figure 4 is capable of executing concurrently four different convolutions. It means that by time sharing, it can do the work of four systolic arrays like the one shown in Figure 3. This property will next be used for the realization of systolic filter banks.

The first example is a filter bank of four filter channels, each of them having three coefficients. Figure 5a shows a block diagram of such a filter bank; each row stands for a systolic array like the one shown in Figure 3, and these parts are enumerated from 0 to 3. Figure 5b illustrates how a 4-slow version of a 3-cell array can do the work of the bank by executing the operations of parts 0, 1, 2, and 3 in the clock periods for which holds $t_{\text{mod } 4} = 0$, $t_{\text{mod } 4} = 1$, $t_{\text{mod } 4} = 2$, and $t_{\text{mod } 4} = 3$, respectively. The middle input data flow consists of control bits for the multiplexers. A control bit with value 1 enables the passing of the input data to the array. A 0-bit enables the feedbacks. Since in this case only 1-bits are input into the control input of the multiplexers, the feedbacks and the multiplexers are not actually used. They are, however, needed for other filter banks.

The second example is a filter bank of just one channel. The number of coefficients of this channel is greater than the number of cells available in the system, so that a one-to-one mapping of coefficients onto processors is not possible. Figure 6a shows a block diagram of a 12-coefficient filter. This filter can be realized by a systolic array of the kind shown in Figure 3, but consisting of of 12 cells. Such a 12-cell array can be decomposed into four connected parts, each of which is identical with the array shown in Figure 3. Figure 6b illustrates how a 4-slow version of one 3-cell array can do the work of the 12-cell array by executing the operations of parts 0, 1, 2, and 3 in the clock pe-

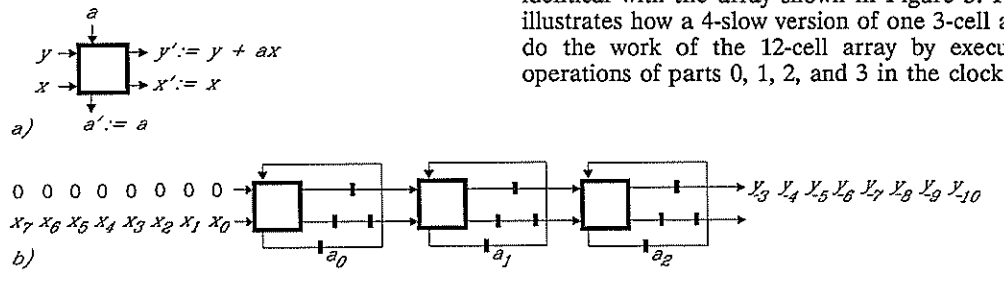


Figure 3 Systolic array for a convolution with 3 coefficients

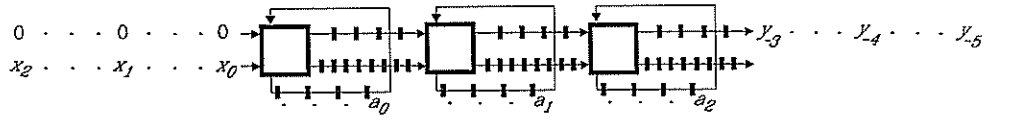


Figure 4 A 4-slow systolic convolver of three cells

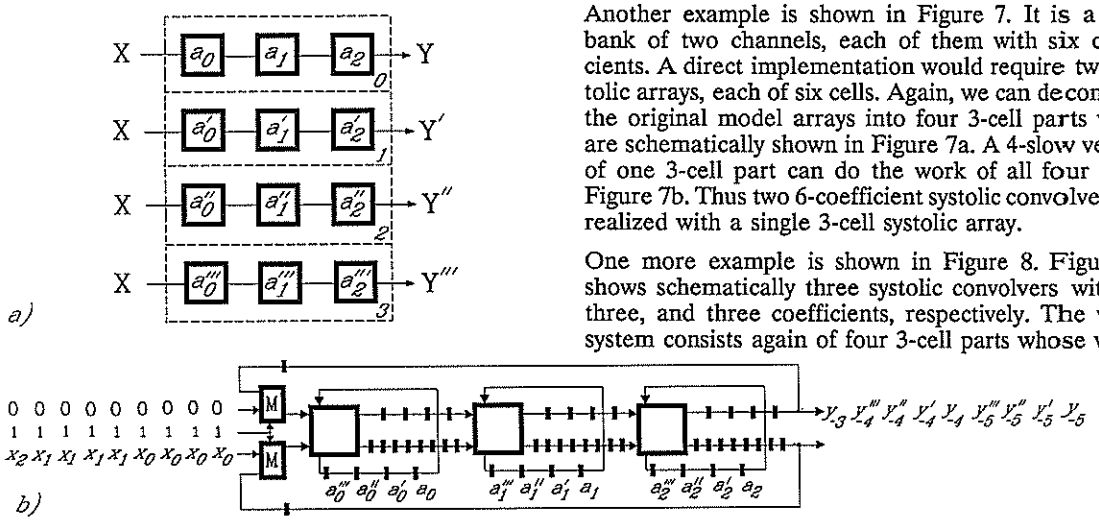


Figure 5 Using a 3-cell 4-slow systolic array for four 3-coefficient convolvers

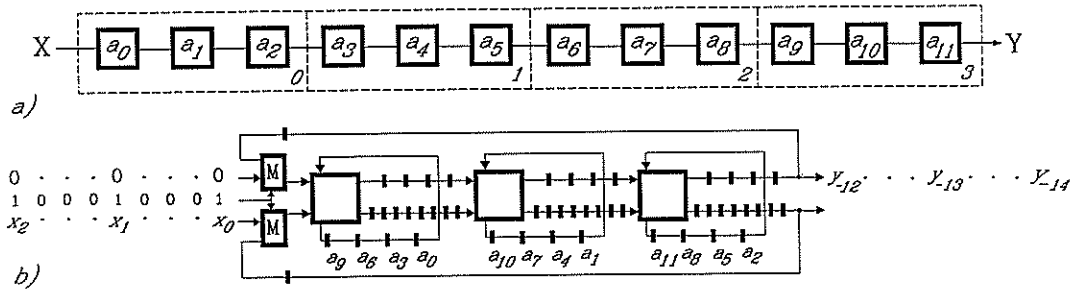


Figure 6 Using a 3-cell 4-slow systolic array for a 12-coefficient convolver

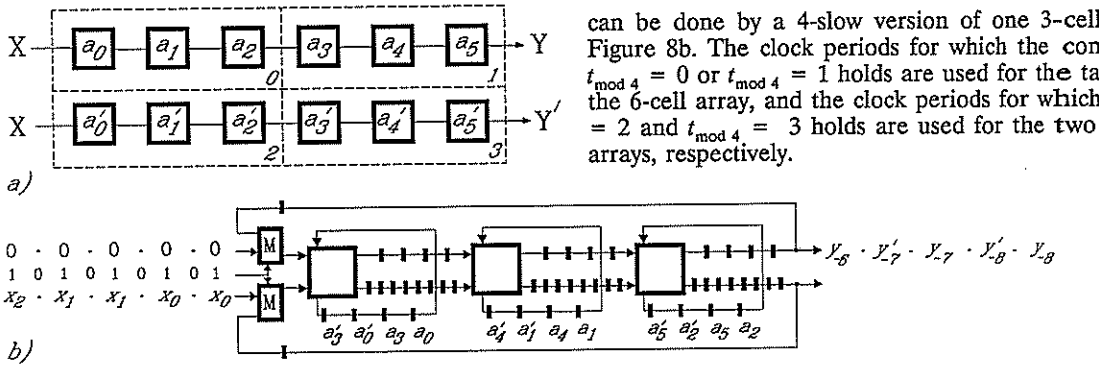


Figure 7 Using a 3-cell 4-slow systolic array for two 6-coeffifficient convolvers

Another example is shown in Figure 7. It is a filter bank of two channels, each of them with six coefficients. A direct implementation would require two systolic arrays, each of six cells. Again, we can decompose the original model arrays into four 3-cell parts which are schematically shown in Figure 7a. A 4-slow version of one 3-cell part can do the work of all four parts, Figure 7b. Thus two 6-coefficient systolic convolvers are realized with a single 3-cell systolic array.

One more example is shown in Figure 8. Figure 8a shows schematically three systolic convolvers with six, three, and three coefficients, respectively. The whole system consists again of four 3-cell parts whose work can be done by a 4-slow version of one 3-cell part, Figure 8b. The clock periods for which the condition $t_{\mathrm{mod}\,4} = 0$ or $t_{\mathrm{mod}\,4} = 1$ holds are used for the tasks of the 6-cell array, and the clock periods for which $t_{\mathrm{mod}\,4} = 2$ and $t_{\mathrm{mod}\,4} = 3$ holds are used for the two 3-cell arrays, respectively.

riods for which holds $t_{\mathrm{mod}\,4} = 0$, $t_{\mathrm{mod}\,4} = 1$, $t_{\mathrm{mod}\,4} = 2$, and $t_{\mathrm{mod}\,4} = 3$, respectively. In this way, a 12-coefficient systolic convolver is realized by a 3-cell 4-slow systolic array. A new sample is input every fourth clock period. In the other three out of every four consecuitve clock periods the feedbacks are enabled. This corresponds to the connections between neighbouring parts of the original 12-cell array.

In general, a c-slow version of an N-cell systolic convolution array can be used to concurrently execute the tasks of n systolic arrays (filter channels) with p_1N, p_2N, ... , p_nN coefficients, respectively, where

$$p_1 + p_2 + \ldots + p_n = c\,. \qquad (1)$$

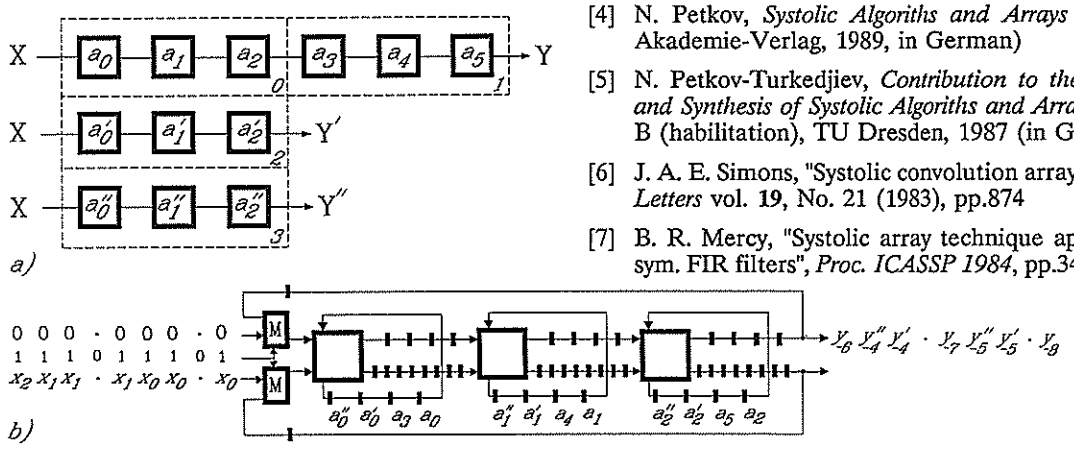


Figure 8 Using a 3-cell 4-slow systolic array for one 6-coefficient and two 3-coefficient convolvers

4. SUMMARY

Refering to Figures 5, 6, 7, and 8, we note that one and the same systolic array with fixed size and structure can be flexibly used for the realization of different filter banks. The additional structures represented in the model by feedbacks and multiplexers do not depend on the size of the filter bank to be implemented. The size of the bank and its particular structure are encoded in the control signals for the multiplexers.

The mapping technique presented above can be used to implement FIR filter banks

(i) in a dedicated VLSI chip and

(ii) in any multiprocessor system of appropriate structure (ring or linear array of processors).

The second possibility was varified on an array of Transputers as well as on the DIRMU and MPS multiprocessors operating at the University of Erlangen-Nürnberg and at the Central Institute of Cybernetics and Information Processing in Berlin, respectively.

ACKNOWLEDGEMENTS

The author gratefully appreciates the financial support of the Alexander von Humboldt Foundation, under whose research award this work has been carried out.

REFERENCES

[1] H. T. Kung, "Why systolic architectures", *Computer*, vol. **15**, No. 1 (Jan. 1982), pp. 37-46

[2] H. T. Kung, "Special-purpose devices for signal and image processing: ... " *SPIE*, vol. **241** (1980) *Real--Time Signal Processing III*, pp.76-84

[3] H. T. Kung, "The structure of parallel algorithms", *Advances in Computers*, vol.**19** (New York: Academic, 1980),pp.65-111

[4] N. Petkov, *Systolic Algoriths and Arrays* (Berlin: Akademie-Verlag, 1989, in German)

[5] N. Petkov-Turkedjiev, *Contribution to the Theory and Synthesis of Systolic Algoriths and Arrays*, Diss. B (habilitation), TU Dresden, 1987 (in German)

[6] J. A. E. Simons, "Systolic convolution array", *Electr. Letters* vol. **19**, No. 21 (1983), pp.874

[7] B. R. Mercy, "Systolic array technique applied to sym. FIR filters", *Proc. ICASSP 1984*, pp.34.A.1.1-4

[8] O. Ersoy, "Semisystolic array implementation of circular, skew circular, and linear convolutions", *IEEE Trans. Comp.*, vol. **C-34**, No. 2 (1985), pp.190-196

[9] N. Petkov-Turkedjiev, "Synthesis of systolic algorithms and processor arrays", *Lecture Notes on Computer Science*, vol. **237**: *Proc. CONPAR 86*, Aachen, Sept. 1986, ed. W. Händler et al. (Heidelberg: Springer-Verlag, 1986), pp.165-172

[10] N. Petkov and Fr. Sloboda, "Bit-level systolic array for digital curve smoothing", *Parallel Computing*, vol. **12**, No.3 (Dec. 1989), (Amsterdam: North-Holland, 1989) pp. 301-314

[11] N. Petkov, "A bidirectional systolic architecture with a single interface to the host", *Proc. of the Workshop on Languages and Systems for Parallel Processing*, Schmitten/Arnoldshein, West Germany, January 26-26, 1990 (Darmstadt, West Germany, 1990, in print)

[12] N. Petkov-Turkedjiev: "Systolic filterbanks for speech recognition", *Proc. of 20. Fachkolloquium Informationstechnik*, TU Dresden, Febr. 1987, S.69--72 (in German)

[13] N. Petkov, "Utilizing fixed-size systolic arrays for large computational processes", to be published in *Lecture Notes on Computer Science: "Recent Issues in Image Analysis and Processing"* (Heidelberg: Springer-Verlag, in print)

[14] N. Petkov: "Mapping systolic FIR filter banks onto fixed-size linear processor arrays", *Proc. 1990 Int. Symp. on Circuits and Systems*, May 1-3, 1990, New Orleans, Louisiana (New York: IEEE, 1990)

[15] H. Umeo "A design of time-optimum and register-number minimum systolic convolvers", *Parallel Computing*, vol. **12**, No.3 (Dec. 1989) pp.285-300

[16] C. E. Leiserson, F. M. Rose, and J.B. Saxe, "Optimizing synchronous circuitry by retiming", *Proc. of the 3rd Caltech Conf. on VLSI*, (Rockville MD: Computer Science Press, 1983), pp. 87-116

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

PARALLEL IMPLEMENTATION OF THE DISTANCE TRANSFORM ALGORITHM

Serge MIGUET

Laboratoire de l'Informatique du Parallélisme LIP-IMAG
Ecole Normale Supérieure de Lyon
46 allée d'Italie
69364 Lyon Cedex 07, France .

Abstract : We discuss in this paper the parallel implementation of the distance transform (DT) of a binary picture on a ring of general-purpose processors. We propose a version that updates segments of k positions within a step and allocates blocks of r consecutive rows of the picture to the processors in a wraparound fashion. We show how to analytically determine the optimal values of the parameters (k,r) which minimize the parallel execution time as a function of the number of processors p and of the problem size n. The theoretical results are nicely corroborated by numerical experiments on a ring of 32 Transputers. We obtain a speedup of 26 over the sequential algorithm.

1. INTRODUCTION

Let P be a binary picture consisting of a figure $F = \{\infty\}$, completely surrounded by its complement $\complement F = \{0\}$. The distance transform (DT) of F with respect to its complement $\complement F$ is a replica of F, where each pixel is labeled with its distance from $\complement F$, computed accordingly to a given metric. In the DT, each position (or pixel) p can be associated with a disc centered on the pixel itself : The radius of the disk depends on the label of the center, while the shape depends on the adopted metric [AS]. A disc is maximal if no other disc overlaps it completely. Since the union of the maximal discs coincides with F, the median axis, defined as the labeled centers of the maximal discs (the local maxima), plays an important role both for saving memory occupation and for shape description purposes [Lev, Mon, RK, YBR].
This paper is organized as follows : first of all, we present the classical sequential algorithm to compute the distance transform of F, which uses two scans of the image. These two scans are to be processed serially. Then we explain our parallel implementation of a single scan on a ring of processors. Finally, we present numerical experiments which nicely corroborate the theoretical results. These are obtained on an FPS-T40 hypercube (32 Transputers), configured as a ring.

2. SEQUENTIAL ALGORITHM

Picture P is an n x n grid of positions, for any positive integer n. Each of the eight neighbors of a position p are indicated by the corresponding cardinal point in the compass.

To compute the distance transform of P, there is a well known sequential dynamic programming algorithm which performs two scans on the image. The first one from top to bottom and from left to right (the forward scan, FS), and the second from bottom to top and from right to left (the backward scan, BS). Let t_1 and t_2 be respectively the two distance weights selected for the horizontal/vertical and for the diagonal neighbors of p. We have the following operators :

(FS) $p := \min(p, W + t_1, NW + t_2, N + t_1, NE + t_2)$
(BS) $p := \min(p, E + t_1, SE + t_2, S + t_1, SW + t_2)$

For $t_1 = 1$ and $t_2 = \infty$ we have the Manhattan distance d4. For $t_1 = t_2 = 1$ we have the chessboard distance d_8. For $t_1 = 3$ and $t_2 = 4$ we have the distance $d^* = 2d_8 + d_4$ of [AS], which is a good approximation of the Euclidian distance.

After the two scans each pixel is replaced by its distance from $\complement F$. Many applications of the distance transform for computing geometric properties such as the area, the contour and perimeter are illustrated in [YBR] for the distance d_8. Montanvert [Mon] shows how to process the local maxima for determining the median line (which is a connected extension of the median axis).

In the following, we concentrate upon the parallel implementation of a single scan (a forward scan) on a ring of processors.

3. PARALLEL IMPLEMENTATION

In what follows, we consider a ring of p processors numbered from 0 to p-1. Each processor has a private memory and can communicate by a message passing protocol with its two neighbors : processor P_i exchanges messages with P_{i-1} and P_{i+1} (throughout the paper, processor indexes are taken modulo p). We consider row oriented schemes, and allocate rows of the picture to the processor. The whole problem is to find an allocation strategy which distributes the workload to the processors well without leading to a communication overhead that is too important.

This work has been supported by the Research Project C3 of CNRS.

Bitz and Kung [BK] have recently studied the parallelization of a path planning algorithm, based on successive forward and backward scans of a digital map containing traversability costs of a region. They have mapped this algorithm onto the linear systolic array in the Warp machine [AAG], using a "greedy" algorithm. We first present their solution which minimizes the startup time and leads to a very well balanced workload among the processors. Then we explain why this implementation is not suited to a ring of general-purpose processors, due to the prohibitive cost of the communications as compared to the arithmetic. Finally, we derive a modified version that performs much better.

3.1. Greedy algorithm:

First assume that the number of processors p is equal to the problem size n. In this case processor i gets row i, $0 \le i < n$. For the forward scan, immediately after processor i has computed the value of a position, it will pass this value to processor i+1 so that it can compute the next position in its own row. Note that to begin a row, a processor needs two values of the previous row, and therefore can only begin two time-steps after its neighbor.

At time 2i+j, Processor P_i operates as follows (wherever indices make sense):

- it receives position (i-1,j+1) from P_{i-1}
- it updates position (i,j)
- it sends position (i,j) to P_{i+1}

When p is smaller than n, partitioning techniques must be considered. Assume for the sake of simplicity that p divides n. In order to let each processor begin as soon as possible, a solution is to assign the rows of the picture to the processors in a wraparound fashion : processor i gets rows j such that $i = j \bmod p$. The wrap mapping is a widely used technique to balance well the workload among the processors [GH, MR, MV, Saa]. Now P_0 needs to receive computed values from P_{p-1}. Note that P_0 receives the first value (p-1,0) from P_{p-1} at time 2p-1. At time 2p, P_0 receives the second value (p-1,1) and updates position (p,0). Hence we do not want P_0 to finish the updating of row 0 before time 2p, otherwise it would remain idle for a while. This implies that $n \ge 2p$. If $n > 2p$, P_0 will simply store the values it receives from P_{p-1} until it starts the updating of its second row.

We see that the latency between the startup times of two adjacent processors is small (two time-steps). The major drawback of the algorithm is that it involves many short communications between the processors. For current distributed memory machines, the time to transfer L words between two adjacent processors can be modelized by $\beta + L\tau$, and it turns out that β is significantly higher than τ ([GH, MV, Saa], see also the experiments reported in section 5). This renders the cost of small messages prohibitive.

Below we explain how to modify the greedy algorithm in order to decrease the communication overhead. We describe the new algorithm informally, and postpone its complexity analysis to next section.

3.2. Updating a segment of length k

The first way to decrease the communication overhead is to use longer messages. We use the same mapping strategy as before, but we update a segment of k consecutive positions at each step. The algorithm is illustrated figure 1. Note that k does not need to be a divisor of n. In figure 1, we let l_0 be the number of positions updated by P_0 at time 0 (l_0 can be any number between 1 and k). Each processor always updates k positions, except perhaps for the first and last updates. We start the update of the next row while finishing the update of the current row. Look at the example of figure 1 : at step 4, P_0 finishes the k-1 last updates of its first row and begins the first update of its second row (which is the fifth row of the picture). Note that this updating was possible because at the end of step 3, P_3 had transmitted its first two values to P_0. The condition for P_0 not to finish its first row before receiving data from P_{p-1} will be derived in the next section : we obtain the condition $n \ge (k+1)\,p$.

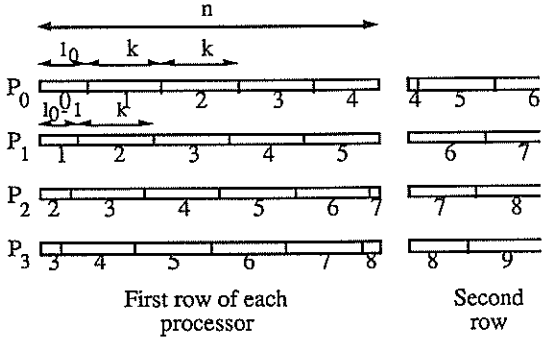


Figure 1 : Updating a segment of length k

The number of data items communicated between two neighbor processors is exactly the same as before, but the larger the k, the more efficiently the communications are performed. On the other hand, the larger the k, the greater the latency between the startup times of two adjacent processors is. We must be ready to find a compromise between the two contradictory exigences of mimimum startup delay (small k) and inexpensive communications (large k).

3.3. Moving to new mapping strategies

Another way to decrease the communication overhead is to communicate less data items between neighbor processors. We now consider more general allocation functions than the wrap mapping, and we assign blocks of r consecutive rows to the processors in a wraparound fashion. For instance with r = 3, n = 36 and p = 4 we have the following repartition:

	P_0	P_1	P_2	P_3
Rows	0,1,2	3,4,5	6,7,8,	9,10,11
	12,13,14	15,16,17	18,19,20,	21,22,23
	24,25,26	27,28,29	30,31,32	33,34,35

Analytically, processor i gets rows j such that $i = \lfloor j/r \rfloor \bmod p$, $0 \le j \le n-1$.

The time-steps are depicted in figure 2. At each step, except for the first and last ones, all the processors update r*k positions. Just as before for r = 1, we start the update of the next block while finishing the update of the current block (see figure 2).

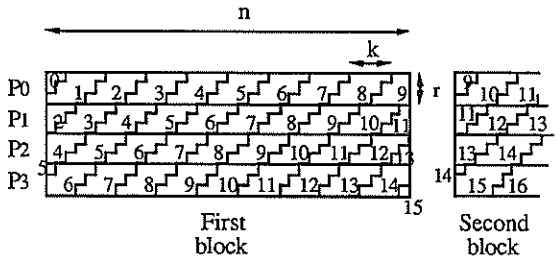


Figure 2 : Parallel algorithm, n=36, p=4, r=3 and k=4

The condition that k and r must meet to keep all processors activated is the following: $n \geq p\ (r+k)$ (see next section).

Now, the number of data items communicated between two neighbor processors is r times smaller than in the previous mapping strategy, because the processors only need to exchange information relative to the boundary rows of each block. Segments belonging to an internal row of a block do not require any inter-processor communication. The price to pay for such a dramatic reduction of the communication volume is again an increase in the latency between the startup times of two adjacent processors. Hence the best value of r will be the result of a compromise, just as the best value of k.

In the next section, we perform a complexity analysis. Given n and p, we analytically determine the values of k and r that minimize the parallel execution time.

4. PERFORMANCE EVALUATION

In this section, we analyse the performances of the parallel algorithm described above. For the arithmetic, we let τ_a be the elemental time needed for updating a position during the scan (formulae FS or BS).

4.1. Sequential execution time

Since there are n^2 positions to update during a scan, the sequential time for a problem of size n is $T_{seq} = n^2 \tau_a$.
As stated before, we consider an allocation by blocks of consecutive rows of size r in a wraparound fashion, where $1 \leq r \leq n/p$. For the sake of simplicity (without loss of generality), we assume that p*r divides n, so that each processor holds the same number of rows in its local memory.

4.2. Parallel execution time

Even though the implementation is asynchronous, we can view the parallel algorithm as a succession of time-steps, where each processor updates r segments of k positions at each time-step. Within a time-step, processor P_i receives a message of length k from processor P_{i-1}, updates r*k positions, and sends a message of length k to P_{i+1} (indices are taken modulo p). Note that the emission is non-blocking, whereas the reception is. P_i does not wait for its emission to be completed before moving to the next step. As a consequence, the communication within a time-step has a cost equal to $\beta + k\,\tau_c$. The total time needed to perform a time-step is $\tau_{step} = \beta + k\,\tau_c + r\,k\,\tau_a$

To evaluate the total number of time-steps in the algorithm, we first compute the time-step t_q at which the processor P_q initiates its computation, and then the number of time-steps after its initialisation. Recall that P_0 updates l_0 positions in its first row at time $t_0 = 0$. We see that P_1 updates $l_1 = (l_0 - r) \bmod k$ positions in its first row at time $t_1 = 1 + \lceil (r - l_0) / k \rceil$ and more generally, that P_q updates $l_q = (l_0 - q*r) \bmod k$ positions in its first row at time $t_q = q + \lceil (q*r - l_0) / k \rceil$
Now, we easily derive the total number of time-steps T_p, since P_{p-1} is the last processor to end its computation. After updating its first parallelogram, P_{p-1} has still

$$\lceil (n^2/(p*r) - l_q + r - 1) / k \rceil$$

parallelograms to update, so that

$$T_p = t_{p-1} + \lceil (n^2/(p*r) - l_q + r - 1) / k \rceil.$$

The parallel execution time of the algorithm is then

$$T_{//} = \tau_{step} * T_p$$

This evaluation is valid only if the processors are not kept idle waiting for some data they need from their predecessor. As explained in the previous section, this condition is equivalent to ensuring that P_0 has not finished the updating of its first block before receiving the data that it needs from P_{p-1} for its second block. P_0 performs its first reception at time t_p. At that time it has already updated $l_0 + k * (t_p - 1)$ positions in the first row of its first block. The condition is that the sum of the remaining positions in this row plus the number of positions that it might update in the first row of the second block is greater than or equal to k, so that it can update a whole parallelogram at time t_p. The condition amounts to :

$$n \geq p\ (r+k)$$

Neglecting low order terms and ceiling functions, we obtain the following analytical evaluation for the parallel execution time $T_{//}$:

Proposition : Given a problem of size n and a ring of p processors, the parallel execution time $T_{//}$ for a block-r allocation, $1 \leq r \leq n/p$, using segments of length k, $1 \leq k \leq n/p - r$, is

$$T_{//} = (\beta + k\,\tau_c + r\,k\,\tau_a)\left[(p-1)\left(1+\frac{r}{k}\right) + \frac{n^2}{p*r*k}\right]$$

Given n, p and r it is easy to find the value $k_{opt}(r)$ of k that miminizes the execution time $T_{//}$. We obtain the value

$$k_{opt}(r) = \min(\, k_{max}(r),\, k_{//}(r)\,)$$

where

$$k_{max}(r) = n/p - r$$

and

$k_{//}(r)$ is the optimal value obtained from the expression of $T_{//}$:

$$k_{//}(r) = \sqrt{\frac{\beta}{\tau_c + r\,\tau_a}\left(\frac{n^2}{p(p-1)r} + r\right)}$$

Given n, p and numerical values for the parameters β,τ_c,τ_a, it is easy to compute k_{opt} and to plug it into the expression of $T_{//}$ to determine the best value of r. We report numerical experiments in the next section.

5. NUMERICAL EXPERIMENTS

In this section,we report on numerical experiments on a ring of Inmos Transputers T414, using up to 32 processors.

In figure 3, we plot the speedups that we obtain with 32 processors when solving a problem of size n = 1920. Note that these speedups are computed according to Gustafson's recent proposal [Gus], in that they are normalized by the amount of arithmetic operations which they require (since it is impossible to solve such a large problem with a single processor). Using 32 processors, we report acceleration factors as high as 26.

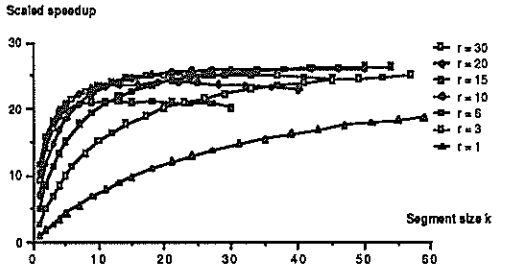


Figure 3: Speedup as a function of the segment size k ; problem size n = 1920 ; number of processors p = 32

In figure 4 we finally show a 3D-plot of the efficiency e(r,k) of the algorithm to better visualise the influence of the parameters on the execution time. The surface we show is the function e(r , k) for the following values of r and k: $1 \le r \le n/(2p)$, $1 \le k \le k_{max}(r)$. The optimal efficiency e = 0.83 is obtained for the highest point of this surface, with r = 6 and k = 54.

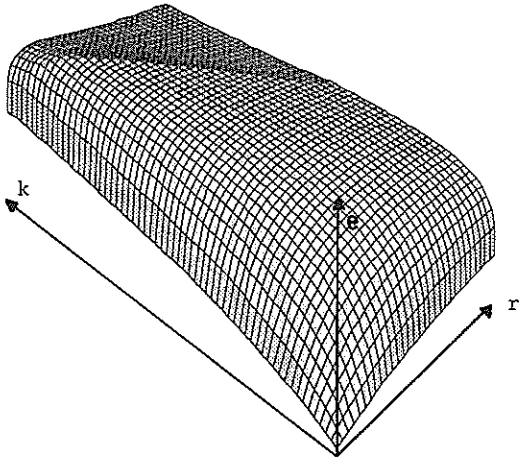


Figure 4: 3D-Plot of the efficiency e(r,k)

6. Conclusion

In this paper, we have discussed the implementation of the distance transform algorithm on a ring of general-purpose processors. We have designed a version that updates a segment of k positions within a step and allocates blocks of r consecutive rows of the map to the processors in a wraparound fashion. We have analytically determined the optimal values of the parameters (k,r) which minimize the parallel execution time as a function of the number of processors p and of the problem size n. The theoretical results are nicely corroborated by numerical experiments on a ring of 32 Transputers. We obtain a speedup of 26 over the sequential algorithm.

6.REFERENCES

[AAG] M. ANNARATONE, E. ARNOULD, T. GROSS, H.T. KUNG, M. LAM, O. MENZILCIOGLU, J.A. WEBB, The Warp computer: architecture, implementation and performance, IEEE Trans. Computers 36, 12 (1987), 1523-1538

[AS] C. ARCELLI, G. SANNITI DI BAJA, Finding Local Maxima in a Pseudo-Euclidian Distance Transform, Comput. Graphics Image process. 43, (1988) 361-367

[BK] F. BITZ, H.T. KUNG, Path planning on the Warp computer: using a linear systolic array in dynamic programming, Intern. J. Computer Math. 25 (1988), 173-188

[GH] G.A.GEIST, M.T.HEATH, Matrix Factorization on a hypercube multiprocessor, Hypercube Multiprocessors 1986, M.T. Heath ed., SIA M (1986), 161-180

[GHS] J.L. GUSTAFSON, S. HAWKINSON, K. SCOTT, The architecture of a homogeneous vector supercomputer, in Proceedings of ICCP 86, IEEE Computer Science Press (1986), 649-652

[Gus] J.L. GUSTAFSON, Reevaluating Amdahl's law, Communications of the A.C.M. 31, 5 (1988), 532-533

[Hwa] K. HWANG, Advanced parallel processing with supercomputer architectures, Proceedings of the IEEE 75, 10 (1987), 1348-1379

[Lev] M. D. LEVINE, Vision in man and machine, chap 10, McGraw-Hill (1985)

[Mon] A. MONTANVERT, Medial line : graph representation and shape description, 8th International Conference on Pattern Recognition, Paris, (1986) 430-432

[MR] S. MIGUET, Y. ROBERT, Dynamic programming on a ring of processors, First European Workshop on Hypercube and Distributed Computers (1989), to appear. Also available as Technical Report LIP-IMAG 89-01.

[MV] O.A. MAC BRYAN, E.F. VAN DE VELDE, Hypercube algorithms and implementations, SIAM J. Sci. Stat. Comput. 8, 2 (1987), s227-s287

[RK] A. ROSENFELD, A. C. KAK, Digital Picture Processing, Academic Press, New York, (1982), Vol. 2, Chap. 11

[Saa] Y. SAAD, Gaussian elimination on hypercubes, in Parallel Algorithms and Architectures, M. Cosnard et al. eds., North-Holland (1986), 5-18

[YBR] A. Y. WU, S. K. BHASKAR, A. ROSENFELD, Parallel Computation of Geometric Properties from the Medial Axis Transform, Comput. Graphics Image process. 41, (1988) 323-332

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A Systolic Array Implementation of the Fermat Number Transform

Jørgen Dall

Electromagnetics Institute
Technical University of Denmark
DK-2800 Lyngby, Denmark

This paper presents a systolic array architecture which is suitable for a VLSI implementation of the Fermat Number Transform (FNT). The FNT is a transform supporting fast digital convolution. General-purpose computers do not take full advantage of the merits of the FNT. However, being perfectly matched to the details of the FNT, the proposed architecture offers an efficient means of computing the FNT.

1. INTRODUCTION.

Fast convolution is crucial in many digital signal processing systems. Traditionally long filters have been implemented with the Fast Fourier Transform (FFT), which is an algorithm for the computation of the Discrete Fourier Transform (DFT). However, several other transforms support fast convolution, and although less well-known, some of them are superior to the FFT in certain respects [1]. What makes the Fermat Number Transform (FNT) particularly interesting is that in some cases it can be computed without multiplications because the multiplications can be implemented as simple shift operations. The FNT is based on modulo arithmetic where the modulus is a Fermat number.

Now, although many computers execute a shift operation faster than a multiplication, the merits of the FNT do not fully manifest themselves in case of implementations on general-purpose computers. These computers are not really geared to modulo arithmetic. Dedicated FNT processors, however, can be tailored to this arithmetic, and the fact that a shift register is a much less area-extensive device than a multiplier makes it possible to develop FNT chips with a higher parallelism than todays FFT chips.

Just like the DFT can be computed with a fast algorithm, e.g. the FFT, so can the FNT. In fact, since the definition of the FNT resembles that of the DFT, an algorithm similar to the FFT is applicable to the FNT. In turn this means that the FNT can be implemented with architectures that are equivalent to the architectures used for the FFT, the traditional pipeline FFT for instance. In the beginning of the 1980s, an FNT chip with this pipeline architecture was designed and fabricated [2]. However, due to the fast evolution of the VLSI technology it is now relevant to consider architectures offering still more parallelism and hence still higher throughputs [3]. This paper presents such an architecture.

2. THE FERMAT NUMBER TRANSFORM.

The definition of the Fermat Number Transform, G_m, of a signal, g_n, is similar to the definition of the DFT, but the complex number field, C, is replaced by the finite quotient ring, $Z/(F_t)$,

$$G_m = \sum_{n=0}^{N-1} g_n \omega^{nm} \pmod{F_t} \qquad (1)$$

The inverse transform is given by

$$g_n = \frac{1}{N} \sum_{n=0}^{N-1} G_m \omega^{-nm} \pmod{F_t} \qquad (2)$$

where n is the time index, m is the index in the transform domain, N is the block length and F_t is one of the Fermat numbers

$$F_t = 2^b+1 , \quad b = 2^t , \quad t = 1, 2,... \qquad (3)$$

F_t is a prime for $t \leq 4$, so according to the modern algebra $Z/(F_t)$ is a Galois field $GF(F_t)$ for $t \leq 4$ [1]. All numbers involved in the FNT, including the kernel ω, are integers in the range $[0; F_t-1]$, and all arithmetic operations are modulo operations. This means that in principle the result of an addition or multiplication is divided by F_t and only the remainder is retained.

No physical interpretation of G_m is known at present. Nevertheless, the FNT is valuable because it supports cyclic convolution in almost the same way as the DFT. What is computed is actually the convolution modulo F_t, but if it is known a priori

that the samples of the desired convolution are in the range [0; F_t-1], the modulo arithmetic has no importance at all. Also, if it is known that the desired convolution is confined to any other interval of length F_t, e.g. [-2^{b-1}; 2^{b-1}], it is simple to correct for the modulo arithmetic. F_t is simply subtracted from all samples exceeding 2^{b-1}.

Like the DFT kernel, the FNT kernel, ω, is a root of unity of order N, but ω is not uniquely specified for a given N and F_t. The principal advantage of the FNT is that it is possible to choose ω so that the FNT can be computed with simple operations like additions and shifts while multiplications are almost completely avoided. Obviously, if ω in Eq. (1) equals 2 or a power of 2, all multiplications degenerate into shift operations or more correctly cyclic shifts, see the appendix. These multiply-free transforms are quite short ($N \leq 2b$), but they are useful anyway as longer transforms can be decomposed into a serie of short transforms. The decomposition is efficiently implemented using the Double Level Decimation (DLD) algorithm [3].

3. THE BASIC ARCHITECTURE.

The architecture presented in this paper implements a 32-point FNT in the field GF(F_4). 32=2b is the order of ω=2 and hence the maximum length of a multiply-free FNT in this field. The choice of F_4 is favourable as it offers very long FNTs [1] and the 17-bit word length (effectively 16) resulting from the modulo-F_4 arithmetic also constitutes a reasonable compromise between dynamic range and chip area.

The fact that the definitions of the FNT and the DFT are so similar means that an algorithm equivalent to the Cooley-Tukey FFT is applicable for the computation of the short transforms.

The architecture known as the "FFT network" represents the ultimate degree of parallelism as it comprises one processor for each butterfly of the structure diagram. The FFT network can be laid out as shown in Fig. 1 where, for simplicity, N is smaller than 32. The FFT network of an N-point transform constitutes a regular two-dimensional array of $(N/2)\times(\log_2 N)$ identical butterfly processors. This makes it very suitable for VLSI implementations, but it has one major disadvantage: It calls for global communication. Long wires interconnect two successive stages, i.e. two successive columns of processors. Global communication costs chip area and it gives large wire delays, so in the VLSI technology, where wire delays tend to be comparable with switching delays, the effect of global communication is usually a significant reduction of the maximum clock frequency.

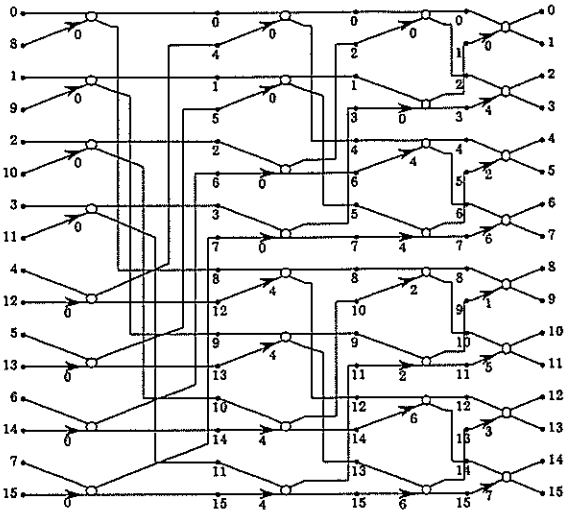


Fig. 1. FFT network architecture for N=16.

In order to eliminate this problem Willey et al. [4] have proposed a systolic elevator concept. The long wires that interconnect two successive stages are replaced by two vertical stacks of registers which respectively clock data up and down, one step per clock cycle. In this way local communication is substituted for global communication. The systolic elevators also tend to be favourable as chip area is concerned. Asymptotically they require an area proportional to N, whereas direct wiring calls for an area proportional to N^2 (both the length and the number of parallel wires are proportional to N, see Fig. 1). Indeed communication is slower. It takes several clock cycles for data to move up and down the elevators, but if the data migration and the butterfly computation are pipelined this does not necessarily introduce bottlenecks. Originally the systolic elevators were proposed for an FFT processor based on the CORDIC algorithm. This algorithm is iterative so a butterfly computation takes several clock cycles anyway.

4. ARCHITECTURAL REFINEMENTS.

The basis of the systolic FNT processor proposed in this paper is the FFT network architecture and the systolic elevator concept, but

- FFT butterflies are replaced by FNT butterflies
- the systolic elevator concept is improved
- the FNT butterfly computation time is matched to the improved elevators
- the FNT butterfly processors are merged with the elevators to form processing elements.

4.1. The FNT butterfly.

An FNT butterfly processor comprises a modulo-F_4 adder, a modulo-F_4 subtracter and a device

that can perform the cyclic shift operations described in the appendix. This device can be implemented as a modified barrel shifter [2]. In this case a word can be shifted any number of locations in one clock cycle. However, a barrel shifter is a complex device which takes up too much chip area considering that 16×5 butterfly processors are to be integrated on a single chip. Therefore a shift register approach has been adopted instead. The principle is illustrated in Fig. 2 where, for simplicity, the field GF(2^8+1) is assumed instead of GF(2^{16}+1). It is seen that the clock is gated by the complement of the most significant bit in order to inhibit the shift if the number is zero. Obviously a b-bit shift takes b clock cycles, but this is the price that is paid for the reduced chip area.

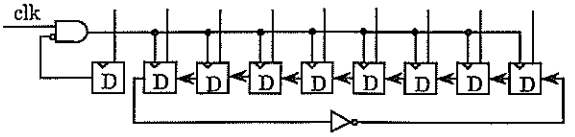


Fig. 2. Modulo-(2^8+1) multiplication by 2^p.

The original systolic elevators are based on an FFT signal graph where each butterfly is split into two halves [4]. This means that for each stage of an N-point transform both the upwards elevators and the downwards elevators include N registers. Also, it takes a maximum of N/2 cycles to clock data up and down.

For an N-point transform the highest twiddle factor power is N/2-1. This is also the maximum number of clock cycles required for a twiddle factor multiplication implemented with the shift register. Therefore, the data migration time and the twiddling time are well balanced and the elevators do not constitute bottlenecks even if the FNT butterfly processors do not involve any CORDIC phase rotator.

4.2. Elevator improvement.

The systolic elevator concept can advantageously be applied to the FFT network in Fig. 1 instead of a signal graph. The resulting architecture is illustrated in Fig. 3. The numbers to the left and to the right of the elevators are labels indicating how data move up and down. It is seen that for each stage of an N-point transform both the upwards elevators and the downwards elevators involve N/2 registers. So the sizes of the elevators are halved compared with those originally proposed in [4]. This in turn halves the data migration time to N/4, but since the twiddling time is still N/2-1, the butterfly computation now constitutes a bottleneck and the most obvious advantage of the improvement is a reduced chip area.

4.3. Timing matching.

With a little ingenuity it is possible to benefit from

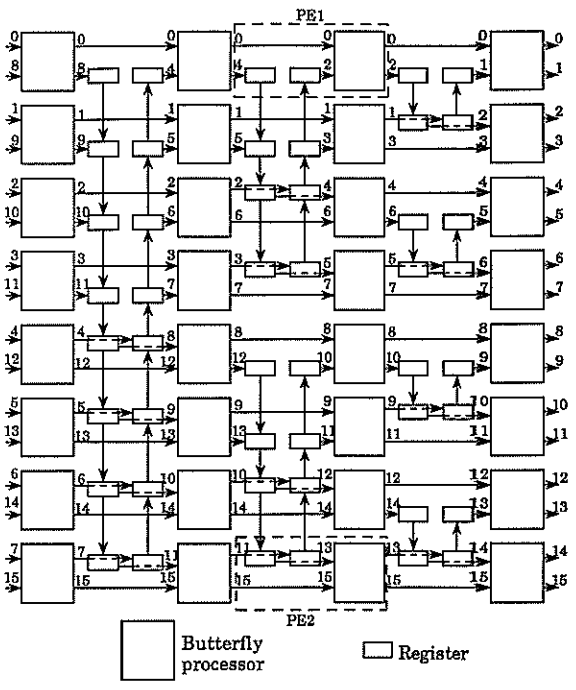


Fig. 3. FFT network with systolic elevators.

the fact that also the data migration time is halved. The point is that a multiplication by 2^p (mod 2^b+1) can be implemented as a multiplication by $2^b 2^{p-b} = -2^{p-b}$ (mod 2^b+1). In this way a b-bit left rotation is replaced by a (b-p)-bit right rotation followed by a negation (which can be absorbed by the following addition or subtraction). This is preferable if $p \geq b/2$ since the maximum twiddling time is then b-b/2 = N/4 cycles. Now the throughput is doubled and the computation time is again matched to the communication time.

4.4. Processing elements.

Two different processing elements are defined in Fig. 4. See also Fig. 3. A regular array of 8·5 PE1s and 8·5 PE2s makes up the systolic FNT processor. Each PE1 (PE2) consists of one of the butterfly processors that are fed from an upwards (a downwards) elevator and the two elevator registers on its left side. The DIT structure is assumed because, as it is argued subsequently, the DIT is preferable to the DIF. Also it is assumed that data are first moved and then shifted. This conveniently implies that left rotations are accomplished by PE1s and right rotations by PE2s.

In accordance with the discussion in Section 4.3. the data migration and the shift operations are taking place simultaneously. The feasibility of this concurrency is not evident since the cyclic shift in PE1 is undertaken by the upwards elevator register. However, in the DIT case the number of cycles needed for the upwards migration of

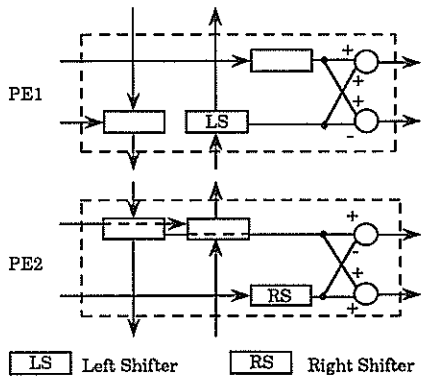


Fig. 4. Processing elements of type 1 and 2.

data and the number of cycles needed for the longest shift of these data add up to N/4 = b/2 = 8 in each stage. Therefore it is concluded that using elevator registers to implement shift operations does not prevent shifts and migrations to take place simultaneously and within the minimum possible time.

This is not the case with the DIF structure. The elevator following the stage with the longest shift is not the one with the shortest data migration and vice versa.

The adders/subtracters in the PEs are 4-bit parallel adders/subtracters. With 8 cycles allocated for additions/subtractions and the above-mentioned 8 cycles for data migration and shifts, the computational bandwidth is 32/(8+8)=2 samples per cycle. In order to match this with the I/O bandwidth, 2 input busses and 2 output busses are required, and an application of single cycle adders/subtracters would not change this requirement. Due to the carry subtraction discussed in the appendix, 2·16/8=4 bits must be processed in parallel. In turn this means that the horizontal interconnections in Fig. 4 are 5 bits wide (the MSB is needed throughout an operation).

5. CONCLUDING REMARKS.

A systolic array architecture which is taylored to the Fermat Number Trans-form (FNT) has been presented. A more complete description including I/O buffering, logic designs, control signals etc. is found in [3]. The systolic FNT processor is a mesh-connected array of 16×5 processing elements, each with a complexity of approximately 1000 transistors. The architecture, as it is presented in this paper, implements a 32-point FNT in $GF(F_4)$, but with minor modifications also shorter transforms and inverse transforms can be computed. Assuming a 40 MHz clock, a 32-point FNT is computed in 0.4 μs, i.e. four times less than a state-of-the-art FFT processor takes to compute a complex DFT of the same length.

APPENDIX: ARITHMETIC IN $Z/(F_t)$.

Since the quotient ring $Z/(2^b+1)$ includes 2^b+1 distinct integers, b+1 bits are needed. The fact that $2^b = -1 \pmod{2^b+1}$ means that arithmetic operations in $Z/(2^b+1)$ are basicly carried out as ordinary integer arithmetic with subsequent carry subtraction (bit number b is considered a carry bit unless the result of the operation is equal to 2^b).

Modulo-F_t arithmetic is advantageously implemented using the diminished-one number representation instead of the binary number representation [3]. The diminished-one representation of a non-negative number, A, is obtained by using the ordinary binary representation for A-1. In particular, 0 is representated as -1 but since $-1 = 2^b \pmod{2^b+1}$, the binary representation of 2^b is used, i.e. the most significant bit (MSB) of a number, A, is 1 if and only if $A = 0 \pmod{2^b+1}$.

One reason why the diminished-one representation is convenient is that the (b+1)-bit arithmetic is virtually replaced by b-bit arithmetic as an operation is trivial if the MSB of one of the numbers is 1. Another reason is that subtractions become simpler. As usual a subtraction is carried out by negating the subtrahend and adding it to the minuend. However, negation is simpler when the number is in the diminished-one representation since the negative of a nonzero number is just the complement of its b least significant bits (LSBs). (The negation is inhibited if the MSB is 1).

When this subtraction scheme is applied to the carry subtractions inherent in the modulo-F_t arithmetic the procedure for multiplications by 2^p, $p \leq b$ is: Inhibit the multiplication if the MSB of the multiplicant is 1. Otherwise, perform a p-bit cyclic left shift of the b LSBs and complement the bits shifted out to the left before they are shifted in from the right.

REFERENCES

[1] Blahut, R.E., Fast algorithms for digital signal processing (Addison Wesley, 1985).

[2] Truong, T.K. Reed, I.S., Yeh, C.-S. and Shao, H.M., A parallel VLSI architecture for a digital filter of arbitrary length using Fermat Number Transforms, IEEE Circuits and Computers Conf., pp. 574-578, 1982.

[3] Dall, J., Fast Transform and convolution Algorithms for Synthetic Aperture Radar Processing. Ph.D. dissertation, LD 70, Electromagnetics Institute, Technical University of Denmark, August 1988.

[4] Willey, T., Durrani, T.S. and Chapman, R., An FFT systolic processor and its applications, Proc. of ICASSP'84, Vol. 2, pp. 34A.4.1-4, 1984.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SOLUTION OF LEAST SQUARES PROBLEM ON DISTRIBUTED MEMORY PARALLEL PROCESSING ARRAYS

Kaushal K. Dhar
Institute for Communication Technology
Swiss Federal Research Institute, ETH Zentrum
CH 8092 Zurich, Switzerland

ABSTRACT

We study the execution of the solution of a set of least squares equations on distributed memory multi-processor system using various techniques. Apart from well known parallel programming methods, that of code and data partitioning, we also apply algorithmic partitioning approach. Required computational complexities for the concerned problem are calculated corresponding to several mapping approaches. The degradation in the achieved multi-processor efficiencies is brought about by the time spent for inter-processor communication needed to redistribute the intermediate results obtained after a certain part of the computation is finished and before the subsequent part of the computation can take place. The DMPP topology under consideration is the 1-D ring.

I. INTRODUCTION

The solution of a set of least squares (LS) equations is required in several applications like numerical analysis and adaptive filtering [1]-[5]. The process of solution is computation intensive requiring iterative calculations with the order of computational complexity $\geq O(n^3)$ where n is the dimension of the unknown vector to be determined. Accordingly, faster methods of the solution are required particularly if the solution is to be sought for real time signal processing applications. One of the alternatives is to execute the solution on a multi-processor architecture. A distributed memory parallel processing (DMPP) architecture is a suitable vehicle to perform this task. We have chosen a homogenious DMPP architecture where in several absolutely identical processing elements (PE)'s are configured together by connecting them in a suitably required fashion. Each of the PE's have their own local memories and there is no shared memory for joint access by more than one processors. The implementation of the barrier or synchronization primitives is done by message passing between the processors. Each of the PE's has a set of neighbouring processors, depending on the topology under consideration, to which it is linked by communication channels. The inter-processor communication is brought about along these channels. Both one- and two-dimension topologies are used in several applications. These kind of multiprocessor architectures have several advantages namely [8]-[15]:

(i). They have repetitive and regular planar structures.
(ii). They have scalability which implies that the size of the configuration can be expanded without unduely stretching the demand on the performance of any of the specific element of the architecture since there are no critical shared resources.
(iii). The communication links are local.

All the above mentioned features qualify the hardware architecture based on DMPP system to be a good candidate for developing a VLSI for a dedicated application.

Of the several available techniques like Householder transformation, Gram-Schimidt (GS) orthogonalization, Givens transformation and QR decomposition, we are using the GS factorization approach [4]-[7]. We show two types of mapping of the conventional GS factorization by using two different data partitioning approaches and an additional mapping using algorithmic partitioning.

II. MAPPING OF LEAST SQUARES SOLUTION ON A RING ARCHITECTURE

It is desired to solve the system of LS equations of the type

$$X * g = y \qquad (1)$$

where X is an (m x n) input matrix, g an (n x 1) un-

known vector, to be determined, y an (m x 1) output vector and the operator * represents matrix/matrix or matrix/vector multiplication. This is done by expressing the matrix $X = Q * R$, where Q is an orthonormal matrix and R an upper triangular matrix , by using GS orthogonalization. For the sake of simplicity and without loss of generality, we assume that the matrix X is a full rank matrix. As far as the notations in the present writeup are concerned, whenever a new matrix or a vector is introduced, its dimensions are indicated as parenthesized subscripts e.g., (m,n) for a matrix and (n,1) for a column vector.

The j^{th} unnormalized column vector $q_j \ \forall\ 2 \leq j \leq n$ of the matrix Q is given as in [4]

$$q_j = x_j - \sum_{k=1}^{(j-1)} (< x_j, q_k > / < q_k, q_k >) q_k \qquad (2)$$

where $< ., . >$ represents the inner product of two vectors and x_i's are the columns of the matrix X. The computation of q_j, by above equation, can be performed as

$$P_j^{(i)} = P_j^{(i-1)} - r_{ij} q_i \ , \quad 1 \leq i \leq (j-1) \qquad (3)$$

where $P_j^{(0)} = x_j$ and $q_j = P_j^{(j-1)}$ and the computation of r_{ij} given as

$$r_{ij} = (< q_i, x_j >) / < q_i, q_i > \qquad (4)$$

We organize the process of computing $P_j^{(i)}$ into computational units where each unit of computation consists of calculation of an r_{ij} and a $P_j^{(i)}$. The time taken for one unit of computation, T_{uc} is given as

$$T_{uc} = 2[mT_x + (m-1)T_+] + mT_x + mT_+ + T_d \qquad (5)$$

Here T_x, T_+ and T_d are the times taken to perform a multiplication, an addition and a division operation respectively. In case a single PE were used to perform orthogonalization, the corresponding computation time is given by T_{or} as

$$T_{or} = \sum_{k=2}^{n} (k-1) T_{uc} = (n/2)(n-1) T_{uc} \qquad (6)$$

The next process is the back substitution which takes a time which is less than 7% [6] of the orthogonalization time, even for m as low as 5 and even lesser for larger m, and hence we don't consider it for the time being.

We shall first propose a very simple mapping of the solution of LS equation on a DMPP ring architecture, depicted in the Fig. 1, consisting of $(n-1)$ PE's. The mapping is done such that the PE with the ID # j computes the vector $q_{j+2}\ \forall\ 0 \leq j \leq (n-2)$. After the vector q_j is computed by the PE # $(j-2)$, it is passed down rightward, along the communication channel connecting the two

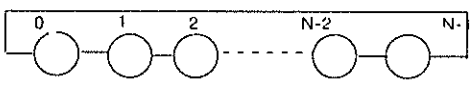


Fig. 1 A DMPP architecture of ring configuration with size N.

PE's, such that each of the PE's with the position such that $(j-1) \leq ID\# \leq (n-3)$ receives it and passes it also right. This kind of communication is referred to as for the intermediate data distribution in contrast to that used for initial data distribution, which is performed before the computation begins, to distribute the elements of input matrix X and the input vector y so that all the PE's have the appropriate data values on them. We organize the time taken for communication in terms of units wherein each such unit, defined as T_{cv}, represents the amount of time a PE takes to send a vector of m elements across to a neighbor connected by a communication link. If the communicating PE's are l links away, the time taken to pass an m length vector left or right will be lT_{cv} . The total time taken to parallelize the orthogonalization procedure on a ring architecture of (n-1) PE's is T_{orp1} given as

$$T_{orp1} = (n-1)T_{uc} + [\sum_{k=2}^{n} (k-1)T_{cv} = (n/2)(n-1)T_{cv}] \qquad (7)$$

The time T_{orp1} consists of two components, the first term due to computational requirement which comprises of calculation of q_n and the second term corresponding to the intermediate communications. The processor activity diagram is shown in the Fig. 2. The representation therein is shown in two dimensions; spatial one, which is displayed along horizantal axis and shows the spread of the different PE's forming the 1-D ring architecture and temporal one, which along the vertical axis shows the elapsed time. It must be mentioned here that the activity diagram, for a 2-D array of PE's, will need three dimensional representation consisting of two spatial dimensions and one temporal dimension and as such would not be so simple to depict. The solid vertical lines under a PE indicate that the corresponding PE is performing certain computations while the dotted vertical lines indicate that at the beginning of the line, a barrier or synchronization is needed. The PE under which it falls is waiting for the data, till the corresponding dotted line ends, which is either being computed by other PE or is in the process of transfer, before it can proceed with the further computation. The horizantal arrows indicate that intermediate results are being being passed over in the direction of the arrow head. Here, we have neglected the parallelization of the back substitution process because of the reasons forwarded in [6].

Now we offer another parallel mapping approach, shown in the Fig.3, wherein a different sort of data partitioning is resorted to. In this mapping, for n being even, the orthogonal vectors q_j and q_{n-j} are computed by the PE with the ID # $(j-2)$, where $2 \leq j \leq n/2$. The vector $q_{n/2}$ is computed by the PE with the ID # $(n/2)-1$. For the case of n being odd, the vectors q_j and q_{n+2-j} are computed by PE with the ID # $(j-2)$, $\forall\ 2 \leq j \leq (n-3)/2$. For n being even we need $n/2$ PE's and for n being odd we require $(n-1)/2$ PE's. However now unlike the approach of the Fig.2, each of the PE's has to wait for some time as shown by vertical dotted lines. The total time taken to orthogonalize T_{orp2} is given as

$$T_{orp2} = nT_{uc} + ((n-3)/2)T_{uc} + (3/4)(n-1)(n-3)T_{cv} \qquad (8)$$

The first term of the above expression corresponds to the time spent by the PE which does most of computing. The second and the third terms are the waiting times because of computational and communications delays respectively. Again, here we neglect the time attributed to the back substitution process which follows the orthogonalization.

III. RESULTS AND DISCUSSION

For any given application on hand and a specific multi-processor architecture to execute it on, there are two figures of merit which measure the achieved performance. These are the processor utilization p_u and the multi-processor efficiency η and both of these are expressed as percentage figures. We assume that the concerned task, when executed on a single processor machine sequentially, takes a time T_1 and the same task when executed on a multi-processor architecture consisting of N PE's takes the time T_N while the processor with the identification # p is busy over the time interval of $T_N^{(p)}$, then

$$\begin{aligned} \eta &= (T_1/NT_N)100 \\ p_u &= (T_N^{(p)}/T_N)100 \end{aligned} \tag{9}$$

We run our programs on a simulator K9 [13]-[15]. It is a useful tool either to compare two multi-processor architectures for executing a particular application task or to evaluate comparative suitability of a particular archi-

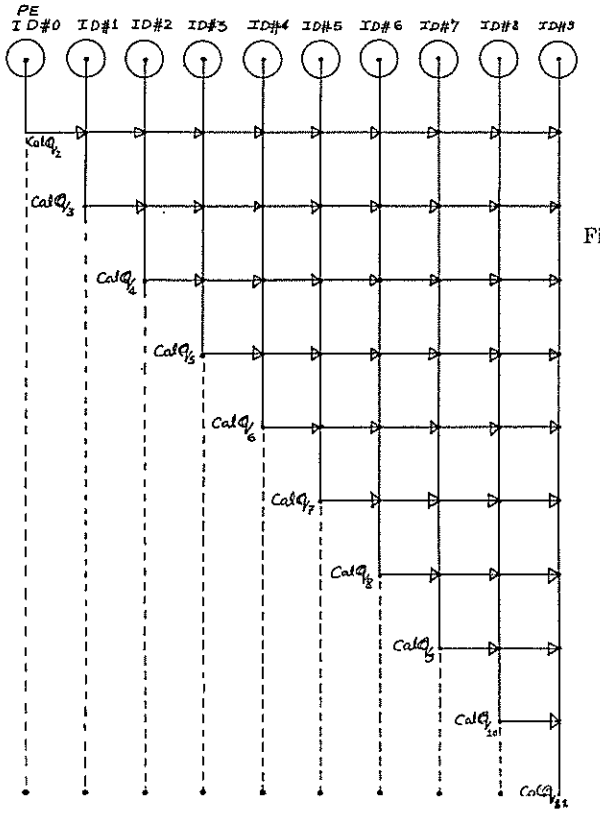


Fig. 2 Mapping of type I of (15x11) problem on a ring of 10 PE's.

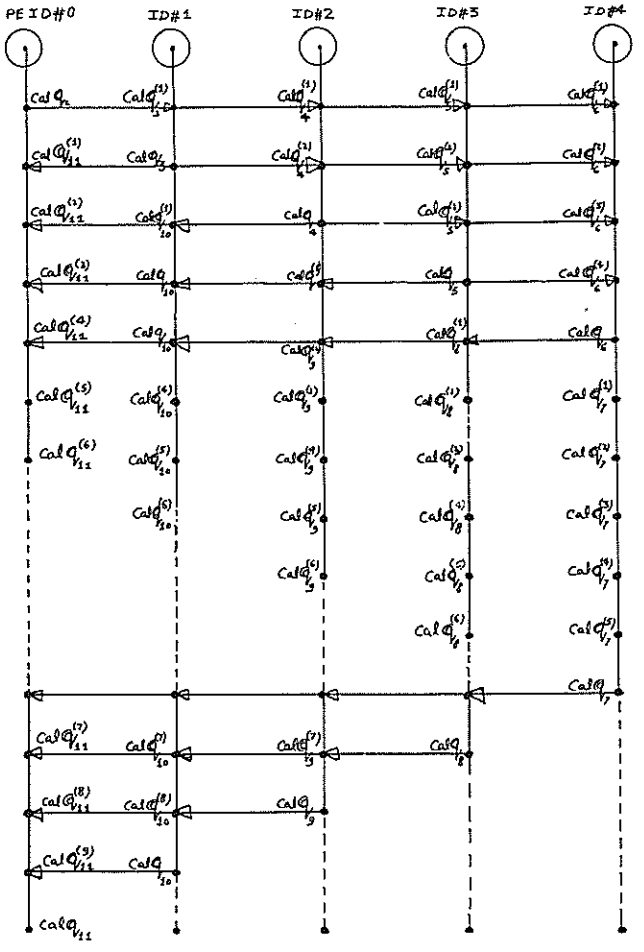


Fig. 3 Mapping of type II of (15x11) problem on a ring of 5 PE's.

tecture to more than one application tasks. The simulator provides statistics which evaluate the performance of a task on a multi-processor topology. For each of the PE's forming the topology, there is an ID#, the time taken to perform computations referred to as 'opdel', time spent performing inter-processor communication referred to as 'comdel' and the total time referred to as 'all-delay'. The total time is sum of the computation time and the communication time. The simulator also provides the processor utilization corresponding to each of the PE's. Table I shows the computed statistics for the GS orthogonalization of a (15x12) problem by the conventional method. It is clear that there is substantial computaional load imbalance with the PE # 0 busy less than 10% of the time and the PE #10 busy more than 75% of the time. These figures are typical of the representation of the Fig. 2 wherein the processor utilization increases as we move right along the array. This gives rise to poor overall achieved multi-processor efficiency.

Now we mention algorithmic partitioning approach [6] wherein the matrix X and the vector g are partitioned into s smaller matrices and vectors respectively such that:

$$X = [X_1 : X_2 : X_3 : \ldots : X_s];$$
$$g^T = [g_1^T : g_2^T : \ldots : g_s^T] \quad (10)$$

Here s is the partitioning factor. It is explained in [6] that such a partitioning results in a computational process, which when executed on a sequential SISD machine, requires a computational complexity that is nearly same as that of the conventional method. Further, [7] explaines that in addition the solution process has much more parallelism than the conventional approach. This parallelism can be exploited by executing several tasks concurrently which improves the overall achieved multi-processor efficiency. Table II presents the computed statistics corresponding to the GS orthogonalization for the partitioning factor s =2. There are ony 5 PE's needed now and processor utilizations are much more uniform and higher. Hence, the partitioning approach achieves much better multi-processor efficiency than that achieved by the figures of the Table I.

Table I. Computed statistics for s=1:

Runtime: 23824 cycles.

cell #	opdel	comdel	all_delay	util. %
(0.0)	2064.00	68.00	2132.00	8.95
(0.1)	3806.00	204.00	4010.00	16.83
(0.2)	5548.00	340.00	5888.00	24.71
(0.3)	7290.00	476.00	7766.00	32.60
(0.4)	9032.00	612.00	9644.00	40.48
(0.5)	10774.00	748.00	11522.00	48.36
(0.6)	12516.00	884.00	13400.00	56.25
(0.7)	14258.00	1020.00	15278.00	64.13
(0.8)	16000.00	1156.00	17156.00	72.01
(0.9)	17742.00	1292.00	19034.00	79.89
(0.10)	17806.00	680.00	18486.00	77.59

Table II. Computed statistics for s=2:

Runtime: 16826 cycles.

cell #	opdel	comdel	all_delay	util. %
(0.0)	9562.00	300.00	9862.00	58.61
(0.1)	10104.00	600.00	10704.00	63.62
(0.2)	10104.00	600.00	10704.00	63.62
(0.3)	10104.00	600.00	10704.00	63.62
(0.4)	9562.00	300.00	9862.00	58.61

ACKNOWLEDGEMENTS

The author thanks Prof. P. Leuthold, director Institute of Communications Technology, ETH Zurich for his support. Thanks are due to Roland Ruehl and Claude Pommerell for their time and patience to discuss several issues related to K9 simulator. I would also like to thank Marco Annaratone for discussions and advices concerning parallel programming/architecture.

REFERENCES

[1] D. Monolakis et. al., "Efficient time recursive algorith for finite memory adaptive filtering," IEEE Trans. on Circuits and Systems vol. CAS-34 No.6 pp.400-407,1987.

[2] S. Kalson and K. Yao, "Results in LS estimation algorithm with systollic array architectures," Proc. II Tirrenia Intl. Workshop on Digital Communications, Tirrenia, Italy pp 235-249, 1985.

[3] F.Ling and J. Proakis, "A recursive modified Gram-Schmid algorithm with applications to LS estimation and adaptive filtering," Proc. ISCAS-84 pp 781-784.

[4] C.L. Lawson and R.J. Hanson,"Solving least squares problems," Englewood Cliffs, NJ, Prentice Hall, 1974.

[5] A. Bjorck, "Solving linear least squares problem by Gram-Schmidt orthogonalization," BIT vol. 7 pp.1-21, Jan. 1967.

[6] Kaushal K. Dhar, "Fast least squares algorithms based on partitioning technique," Proceedings IEEE 1990 International conference on Acoustic, speech and signal processing, April 1990.

[7] Kaushal K. Dhar, "An improved least squares solution technique for multi-processor architecture," Proceedings IEEE 1990 International symposium on Circuits and systems, May 1990.

[8] M. Annaratone, "Gaussian elimination on one- and two-dimensional array of processors," Technical report No. 88/5, Swiss Federal Institute of Technology, Zurich.

[9] A. Bojanczyk, R.P. Brent, and H.T. Kung, "Numerically stable solution of dense system of linear equations using mesh-connected processors," SIAM J. Sci Stat. Comput., 5(1):95-104, March 1984.

10] G.H. Golub and C.F. van Loan,"Matrix computations," John Hopkins University press, 1983.

11] M. Annaratone, C. Pommerell and R. Ruehl,"Interprocessor communications speed and performance in distributed-memory parallel processors," Proc. 16th symposium on computer architecture, IEEE-ACM, June 1989.

[12] S. Borkar et al.,"iWarp: An integrated solution to high speed parallel computations,"Proc. Supercomputing 88, Nov. 1988.

[13] Peter Beadle, "The K9 simulator," Technical report No. 88/23, Swiss Federal Institute of Technology, Zurich.

[14] Peter Beadle, C. Pommerell and M. Annaratone,"K9: A simulator of distributed-memory parallel processors," Technical report No. 89/10, Swiss Federal Institute of Technology, Zurich.

[15] Peter Beadle, C. Pommerell and M. Annaratone,"K9: A simulator of distributed-memory parallel processors," Supercomputing '89, Reno, Nevada, November 13-17, 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

TRENDS AND PROSPECTS IN ARCHITECTURAL FEATURES OF DIGITAL SIGNAL PROCESSORS

P. LE SCAN . M.CAND
FRANCE TELECOM C.N.E.T.
chemin du vieux chêne
38243 MEYLAN cedex
FRANCE

INTRODUCTION

Since the beginning of this decade, there has been a continual growth in the market of digital signal processing. This is due to the increasing computational complexity of the DSP systems and algorithms. At the same time, the performances of the technology in terms of speed, consumption and size are continuously being improved. Consequently, more and more sophisticated chips are being designed by the architects of DSP integrated circuits. On the other hand, DSP customers require powerful, but tailored chips for a best performance versus price deal, so there is an increasing need for versatile ASICDSP facilities

After a brief description of the most important characteristics of the architecture of the presently available general purpose programmable DSPs, we show the trends and future prospects of these chips, designed using silicon compilers.

MAIN FEATURES OF AVAILABLE DSP's

Real time signal processing is fast and often complex. In fast chip architecture,two fundamental principles are applied by today's programmable DSP designers: parallelism and pipelining.

The Von Neumann structure, widely used in general purpose microprocessors, has almost never been used in DSPs due to the bottleneck resulting from the fact that there is only one internal bus which reduces the necessary computing speed.

Duplicating the internal buses between the memories (instruction/data) and computing units (address decoding and computing) represents one of the subliminal forms of parallelism : the Harvard architecture. Once again, the data memories are also duplicated, often with specific access like double access and with a stand-alone address computation unit (modulo arithmetic for convolution or bit reversed computing for FFT).

Furthermore the microinstruction field width has increased with the level of parallelism .

The fundamental functional block of a DSP is the data arithmetic and logical unit (DAU), with, inside this DAU, the most important block: the multiplier-accumulator. This operator includes some pipelining stages and its typical datapath is 16 bits wide, the accumulator being 32 or 40 bits wide. Pipelining is moerover used to optimize the "Fetch-Decode-Fetch-Data-and-Execute" sequence of instructions.

The main architectural features, which allow control of the application program running in the DSP, have often been taken from general purpose microprocessors. For example a count stack with a cache memory used for the Repeat instruction, which is more efficient than a standard loop with a test, or a program counter stack for subroutine nesting and interrupt managing [1].

Although having certain Input/Output characteristics such as DMA, it can be noticed that the I/O bandwidth of current DSPs is very narrow, which makes the multiprocessing with synchronism very difficult to implement, knowing that new DSP algorithms ask for more computing time and have to be dispatched in some DSPs.

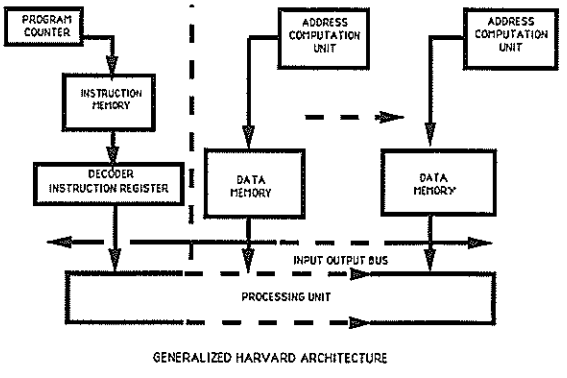


Fig. 1

TRENDS FOR TODAY'S DSP

The strongest trend at present is to integrate a whole digital processing system on a single chip. This is achieved by increasing the high level of concurrency in the data path, the number of buses, and the amount of embedded memory, thus generalizing the Harvard structure (see figure 1).

In parallel, a new generation of DSPs is available with on-chip floating point arithmetic, but they are below the performance of the fixed point DSP's in standard benchmarks. This is the reason for the coexistence with fixed point DSP's which have reached the limits of the technology for the arithmetic performance (33 Mips). Nevertheless, the important problem for these DSP's is the software for developing applications, as writing an efficient program is time consuming. To generate an efficient code, certain C-compilers have appeared, but they remain inefficient ; the new compilers with optimizer could well prove to be better.

To summarize theses trends, we can say that most available DSPs are able to execute the school cases such as FIR, IIR, FFT with a good time efficiency and they generally support high level language compilers (C) making them good tools for breadboarding. Also, although complex algorithms may be splitt into several processes with low communication rates, they are easy to implement on several DSPs. However some major disavantages still remain when complex algorithms have to be implemented with time and price limitations.In this case, C compilers are not usable and multiprocessing may be feasible but not acceptable. Generally speaking, if it is easy to provide good benchmarks for each DSP, it is not so obvious to find the best DSP for a good algorithm. In addition, the algorithms used for benchmarks are better hardwired on full custom or ASIC chips. So, manufacturers offer today a wide spectra of DSPs according to the market :

-General purpose DSPs for labs or low volume applications.

-Application Oriented DSPs, for graphic or speech processing for example.

-Application on Specific Processor, with embedded program ROM for stabilized applications.

-ASICs with a GP DSP core for experienced customers. Not to be confused with the utopian ASICDSP including a true full custom DSP .

-Full custom designs when possible and cheaper.

Notice that the ASIC techniques are equally used by silicon manufacturer and system designers, so that an ASIC design may become a standard product as well as remain a captive product.

PROSPECTS

Notwithstanding the current trends given above, the best general purpose DSP will never be the best for one specific application. The perfect application specific DSP still remains to be built.

The "building block approach" has already been successfully used in research laboratories as well as at the industrial level (bit-sliced) despite the inherent difficulty of microprogramming. Nevertheless, the hardware industrialization of these studies is nothing more than a reproduction of the laboratory model, with major problems such as high cost, low reliability, heavy weight, high consumption and large volume.

The current performance of the technology leads us to believe that this industrialization will result in the design of a monochip DSP optimally tailored to application requirements: an ASICDSP or a ASDSP. Thus the problem of weight and volume will disappear, and the reliability will improve. Only the problem of cost, directly connected to design time, will remain.

The latest design concept being introduced, i.e. the core of an available DSP surrounded by certain specific units (memory, address arithmetic unit ...), enhances the flexibility of the outcoming DSP but not completely because the core is not flexible.

The challenge is to find a design approach which maximizes the performance (speed and size) of the chip with regard to the specific application while minimizing the design time.

Let us examine the problem in terms of algorithm complexity, and speed of operation that represent the major constraints. Fig. 2 depicts the areas where there are one or several solutions, regardless of the multiprocessing management. Algorithm complexity reflects the difficulty in defining the specification rather than the computational power. The required speed of operation refers to the widely used Multiply and/or Addition operation and does not take into account the control operations or data management.

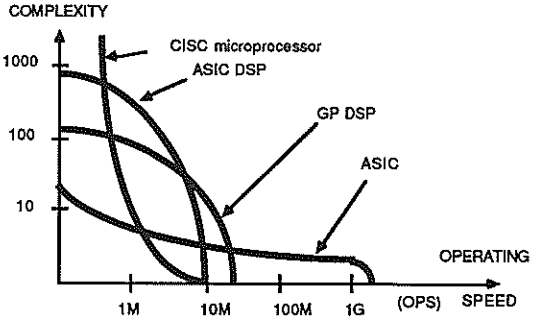


FIG. 2 COMPLEXITY VERSUS MOPS

For low complexity algorithms (this does not mean low complexity chip), it is legitimate to believe that actual layout generators will become real function compilers, with a formal entry and extensive parametrization. All the tedious design problems such as clock skew elimination and power line sizing will be solved by the CAD tool. In addition, the CAD system must offer various specification entries according to the type of function, for instance the template for filters or the mathematical expression for transforms. Also the best compromise between size, power and processing time for a given speed must be proposed leads to bit serial, systolic array or parallel/pipelined implementation. This results in systems in which specification preprocessors and evaluation postprocessors take more place than conventionnal layout generators, sometimes called silicon compilers [3].

For medium complexity algorithms,the use of GPDSP will offer an increasing diversity and alone or surrounded by specific blocks will be able to digest the most commonly used algorithms. But for high complexity algorithms, what can we do if there is too small size memory, or one level of nested subroutine is missing? You must make yourself your own ASDSP. For this, there are already many methodology solutions, but not real automated solutions [5]. Most authors emphasize the multiprocessing capability of their approach, but it seems that it is the problem of the system designer to know if his algorithm can be implemented on one or several ASDSPs, or has to wait for the next technological improvement. Among the architectural synthesis methods, CATHEDRAL_II [2] and SPAID [6] seem to be the most complete. An industrial version of the former will be appearing soon. The problem will be solved by the use of software tools, dedicated to DSP architectural studies, which can assemble basic cells or blocks allowing the designer to synthesize the architecture adapted to his application [4]. The first step is to describe the algorithm with a high level language or signal flow graph and to translate it into basic operations; the second step is to assemble basic blocks or cells using a method which minimizes such criteria as chip size and takes into account the possibilities of using parallelism, multiplexing and pipeline. The encouraging results of artificial intelligence research lead us to think that these new methods could be used for this architectural design tool.

A library will provide the flexible basic blocks and cells. For example, this library will contain a multiplier-accumulator (MAC) with parameters such as datapath width and pipeline level, many types of memory and registers, an I/O unit ...

Connected to other more specific design tools such as logical and electrical simulators, placement and routing tools and layout generator, they will join the family of silicon compilers, which are already available in other domains. The question is to know whether automated and interactive phases are really well adapted to producing an optimal result.

CONCLUSION

These ASDSP's are not in competition with either general purpose DSP's (there will always be a need for these) or nonprogrammable ASIC's, but are complementary because the type of implemented algorithm is different (irregular algorithm such as test with breaking of sequence) and would often involve a too high level of complexity on a non programmable ASIC design.

REFERENCES

[1] T. Tsuda, H. Gambe and R. Hoshikawa, "Digital Signal Processor" in Fujitsu Sci. J. , 25, 3, pp 171-193, September 1989.

[2] F. Catthoor, J. Rabaey, G. Goossens, J. L. van Meerbergen, R. Jain, H. de Man and J. Vandewalle, "Architectectural Strategies for an Application-Specific Synchronous Multiprocessor Environment" in IEEE Trans. on ASSP Vol. 36 N° 2, February 1988.

[3] G. Mirchandani, P. Twombly, "A Software Development Tool for Scheduling Signal Processing Algorithms on Multiprocessors with Arbitrary Connectivity"proceedings of ICASSP1989 pp 1146-1149.

[4] P. Le Scan, G. Privat, "Traitement Numérique du Signal : Architecture et intégration" in L'écho des Recherches N° 138, 4ième trimestre 1989, pp 19-31.

[5] Luis Bonet, T. A. Williams, "Digital signal processor IC's."VLSI SYSTEMS DESIGN december 1988

[6] B.S.Haroun, M.I.Elmasry, "SPAID: An architectural synthesis tool for DSPcustom applications" IEEE OF S.S.C. vol 24 no 2 april 89

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SYNTHESIS OF DEDICATED VLSI STRUCTURES FOR SIGNAL AND IMAGE PROCESSING

Stewart G. SMITH, Ralph W. MORGAN and Julian G. PAYNE

VLSI Technology e.u.r.l., Route des Dolines, Sophia Antipolis, 06560 Valbonne, FRANCE

We present the progress and innovations behind an industrial VLSI cell-compiler which is currently near to completion in its initial form. This design automation tool targets a generic pipeline architecture and companion synthesis technique at integrated high-performance signal and image processing applications. Algorithm partitioning isolates a fixed core processing task, specified by the user in a simple data-flow schematic of primitive functional icons. The user is able to specify the desired machine in terms of three high-level conceptual entities: function, throughput and accuracy.

1. Introduction

The field of Digital Signal Processing encompasses many different manifestations of numerically-intensive computation. While each must ultimately observe the constraints of real-time operation, the different throughput demands of individual applications can lead to greatly contrasting architectural solutions. Thus, in contrast to general-purpose computation, the eventual form of a machine depends not only on its required function but on its required throughput.

Historically, high-throughput (e.g. radar bandwidth) hardware has tended to be less general-purpose than low-throughput (e.g. speech bandwidth) hardware. There is no inherent reason for this; undoubtedly radar applications can stand to benefit from the sophisticated feature-extraction techniques used, for example, in speech recognition. However it is only really possible to achieve high throughput by 'brute-force' methods such as parallelism and pipelining, which impose structure on algorithms, reducing flexibility. There is little reason to doubt that, as technology progresses, this 'flexibility gap' will always exist to some extent.

Advances in IC CAD have been driven mostly by the computer industry, and it is natural to adapt well-tried and tested methods for new problems. High-level synthesis has inherited a sizeable knowledge-pool from the wider field of parallel processor and compiler design. Parallelism and pipelining are two well-known ways of boosting performance; the principal motivation of high-level synthesis research is to furnish datapath machines with this power, while optimising architectures for target applications. Broadly speaking, these techniques apply graph-theoretic analysis to data-flow algorithm specifications, assign resources for computation, communication and storage, then schedule operations on this application-specific machine to complete the algorithm in the shortest time [1].

However we would point out that the problems and requirements of high-performance computers and dedicated ASIC signal processors may sometimes differ. Our proposed approach (which is intended to complement high-level synthesis) hinges on several assumptions about target applications which would be hopelessly restrictive in the general case. We address applications which stand to benefit from a dedicated computational engine, preferably reasonably challenging in function, to alleviate some central computational bottleneck. Efficiency gains are achieved by hardwiring wherever possible, thereby losing many of the flexibility advantages of the datapath approach. At the level of this central task, programmability is impractical beyond simple mode control. Thus, while keeping a watchful eye on developments in high-level synthesis, we have decided to concentrate our efforts on providing CAD solutions for applications on the high-throughput side of the flexibility gap. The use of digit-serial techniques linkls our approach with that of the PARSIFAL project [2].

2. DSP Engine Compilation

We require target applications to be expressed in terms of the *operational* template of Fig. 1: a series of nested control loops, the innermost of which contains a fixed integer arithmetic 'task'. Word-level scheduling *within* this task is hardwired.

```
for .... {
    control
    for .... {
        control
        for .... {
            control
            task
        }
    }
}
```

Figure 1: Operational template

Operands are sequenced through the task processor, according to outer-loop control, to accomplish the overall processor function. The task processor is defined structurally, as a schematic network of icons representing commonly encountered DSP 'primitives' such as multiply,

add, rotate etc. Along with the structure of the task processor, the user must enter the rate at which the processor must execute tasks, and the internal 'working' integer arithmetic precision desired. We feel that throughput and accuracy should be part of the processor specification, not merely a by-product.

Machines are synthesised according to the *physical template* of Fig. 2. This consists of a hardwired, pipelined processor cell, and a set of register banks for communication [3]. Input data arrive in sequence on a bit-parallel bus, and load up the register bank. When this process is complete and the register bank is full, the data block is transferred into the 'serial domain' and transmitted in pipelined fashion (according to the compiler's choice of word structure) through the processor. The register bank immediately starts filling up with operands for the next computation. Results are captured in the output register bank, are transferred back into the 'parallel domain', and depart the register bank in sequential fashion. The machine is pipelined at several levels, and all hardware is 100% utilised under a fast clock.

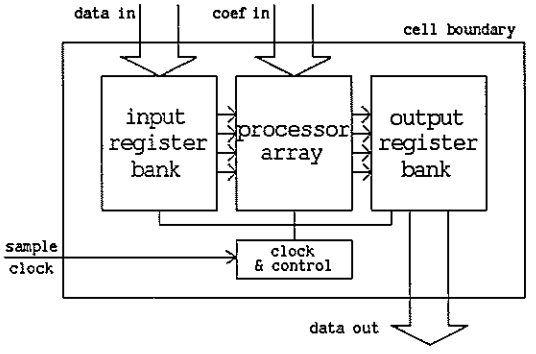


Figure 2: Physical template

3. Word Structure

The result of bit-level scheduling is to decompose the specified working precision into three factors [3]. Two of the factors control bit-parallelism; the other controls bit-serialism by allocating a fixed budget of internal clock cycles to each task. Thus two budgets (bits per operation, and operations per second) are reconciled by trading bit-parallelism with bit-serialism. The register banks of the physical template perform the translation, on data blocks, from bit-parallel format into the chosen word-structure, and vice-versa.

The three-way factorisation yields a potentially large space, or 'ball-park', of possible solutions. Fig. 3 illustrates the three extremes of the ball-park, for a 6-bit example. Here unit space interval (vertical) is a wire and unit time interval (horizontal) is a clock cycle - note that *elapsed* time reads left-to-right, consistent with a left-to-right flow of data. In each case the least-significant bit (LSB) is at the top-right corner.

The degree of bit-parallelism employed is equivalent to the spatial dimension, and the degree of pipelining to the temporal dimension. The three extremes are: (a) bit-serial (min. parallelism, max. pipelining), (b) parallel-combinatorial (max. parallelism, min. pipelining) and (c) parallel-pipeline (max. parallelism, max. pipelining). Note than minimal parallelism implies maximum pipelining, so the fourth combination (both minimum) cannot exist. We employ a concise notation for word structure; the hierarchical triplet (*bits,digits,subwords*). In Fig. 3, (a) = (6,1,1), (b) = (1,6,1) and (c) = (1,1,6).

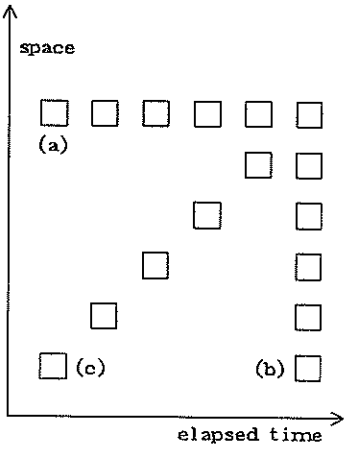


Figure 3: Three extremes, 6-bit word example

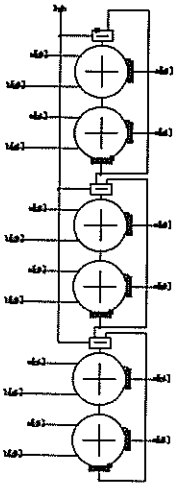


Figure 5: (2,5,3) adder

Fig. 4 shows a typical example from inside the ballpark; the (2,5,3) decomposition of a 30-bit word. The word comprises three 5-digit subwords, each digit consisting of 2 bits.

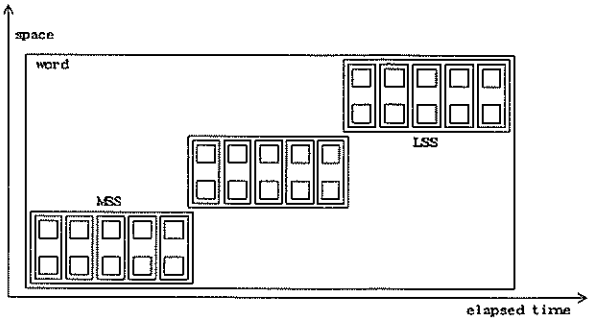


Figure 4: (2,5,3) word structure example

Fig. 5 shows the (2,5,3) adder which corresponds to the word structure of Fig. 4. It can be seen that the length of the ripple-carry path (and hence maximum internal clock rate) depends on digit-size. The two spatial dimensions *bits* and *subwords* are not equivalent; increasing *bits* at the expense of *subwords* leads to smaller, slower solutions.

4. Principal Software Components

The tool consists of three well-delineated software subsystems: a front-end (for communication with the user), a kernel (for data-base manipulations, including synthesis) and a back-end (for circuit generation). The following section, in which we list the main functions of the tool and highlight the major innovations, should be read with this in mind. Only components which are peculiar to the tool are listed, and synthesis is described in the next section.

Design capture is the process of specifying the core task, in the form of an interconnected network of icons which represent familiar DSP functional objects, e.g. multipliers, adders etc. This form of data flow specification is natural and intuitive in a DSP context [4]. The interface specification and the operational template are currently captured in textual format (high-level programming language). It should be noted that these specifications are purely functional. Issues of throughput and accuracy (which in DSP are no less important than function) are addressed in the parameter system.

The **actor set** is a highly-parameterised, multi-purpose set of routines which correspond to the icons seen by the user. As well as containing the tool's inherent digit-serial architectural knowledge, the actors' functions include the transformations on local parameters required by the software kernel, provision of estimation information and behavioural modelling and reporting. The actor set is modular and extensible.

The simplicity of functional specification belies the inherent descriptive power of the tool, much of which lies in the **parameter system.** The icon network is heavily parameterised at local level, and these parameters are synthesised from a few key global parameters supplied by the user. These include: task rate (throughput), input precision, working precision, and numerical growth factor. Local parameters include: word format parameters (position of binary point, growth above binary point etc.), basic parameters (e.g. multiplier coefficient resolution), and derived parameters (e.g. noisefloor, data). Although the parameter set is almost entirely synthetic, any parameter (other than derived) may be overridden by the user.

Estimation tracks the physical and behavioural consequences of synthesis and user-overrides, and is executed by the compiler on request. Physical quantities which are estimated, on a local as well as a global basis, include area and power consumption. Estimated numerical quantities include word growth above the binary point, and the intrusion of quantisation noise. These parameters are propagated by the kernel. As well as providing instantaneous feedback of design information, numerical estimation may often prove more useful than simulation [5]. Often the performance of a system is best analysed from a statistical point of view, rather than by chewing on large sets of test vectors.

Data may be considered as just another parameter to be propagated, which is the basis of the **verification** system. High-level simulation over the operational template (and beyond if necessary), producing bit-perfect results from user-supplied test vectors, may be achieved in run-times orders of magnitude faster than logic simulation. Performance is comparable to the execution of a high-level language model (which is, after all, more or less what transpires during data-propagation). Thus interactive algorithm exploration is possible, freeing the designer to spend more time in the creative phase, evaluating alternatives and selecting the appropriate configuration in the light of acquired wisdom.

5. Synthesis

Synthesis proceeds in four distinct stages: bit-level scheduling, parameter assignment, structural expansion and circuit generation. While the first two are highly interactive processes, the last two are mostly back-end implementation mechanisms. Synthesis may commence after the user has selected an operational template, and identified the throughput and working precision desired. He may or may not have entered an icon schematic.

Bit-level scheduling arrives at the best word-structure for the application. We saw earlier that *bits,* the number of bits in a digit, affects combinatorial depth (hence max. achievable clock rate in a given technology), which in turn affects the clocking budget. Using simple heuristics, the compiler is able to search for the most appropriate combination of the three factors which allows the working word to be transmitted/processed in the time allotted by the user's throughput requirements. Should this process fail, the use must re-partition the operational template so that word structure falls inside the performance range (ball-park) dictated by the chosen technology.

Our approach inherits the entire bit-serial designer's bag of tricks [6,7]. The difference is that the bit-serial designer must rely solely on these *ad hoc* methods to arrive at an efficient implementation. We require only that he arrive within the vicinity of the best solution - i.e. in the ball-park - by *ad hoc* algorithm partitioning; fine-tuning of performance comes through the methodology of bit-level scheduling (see below). Thus reliance on inherent algorithm properties is greatly reduced. Note that the size of the working word may be increased by the compiler if it results in more judicious factorisation, and that the user may override the decision of the compiler if desired.

Parameter assignment gives default values to the set of parameters which exist at each node in the icon network; thus all local parameters are initially synthesised from a few globals. While the objective of synthesis is to maximise use of the allotted numerical resources, the user may once again override any decision of the compiler if desired, with arithmetic consistency maintained automatically throughout the network at all times. During specification, estimation procedures may display the consequences on size, area, accuracy, behaviour etc. of each design move, on an instantaneous basis [5].

By this time the user should be satisfied with his processor specification, and 'pushes the button' to trigger

structural expansion Each icon on the schematic, together with surrounding parameters, finally produces an optimised circuit module which performs the function of the icon on data formatted according to the word structure imposed. Through its innate integer arithmetic module generation capability [3], the cell-compiler has the means to synthesise netlists across all possible bit-level scheduling schemes.

Circuit generation is the replacement of a functionally-correct low-level netlist (the expanded structure) with the physical netlist, including clock tree, buffers etc. Although this has little to do with synthesis, we include it for completeness. It is at this point where real technology binding occurs, although bit-level scheduling also requires knowledge of the target technology. Two forms of layout may be pursued; standard-cell place and route, and compiled or structured layout. The former is more tractable but less area-efficient. In both cases, some interaction with the user may be required before acceptably efficient layout is achieved.

6. Example Systems

Using mostly conventional CAD, we have already implemented a 20 MHz 16 x 16 Discrete Cosine Transform [8]. The dataflow specification of this machine, which performs some 110 million multiplies/sec., is shown in Fig. 6.

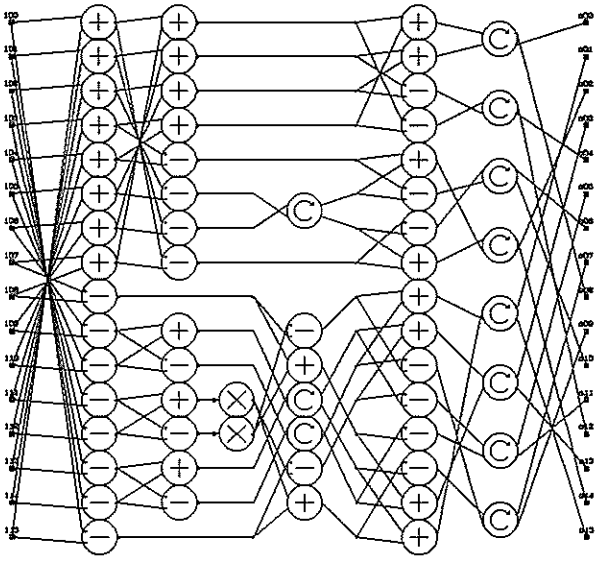


Figure 6: DCT flowgraph

This diagram, together with a few global parameters, is enough to enable the synthesis of a DCT engine. While the 1.25 MHz task rate plus the apparent irregular dataflow might pose a few challenges to a high-level synthesiser, we simply expand the structure as shown. In this particular example, one standard bit-serial design trick (2-way multiplexing) is required to hit the ball-park; bit-level scheduling does the rest.

More recently we investigated a 512-point transversal matched filter [5]. Sample rate was 100 kHz and sample resolution 12-bit. Statistical analysis was used to arrive at an internal 24-bit wordlength, with (6,1,4) word structure. The processor consisted of a single multiply-accumulator, stepped down the length of the filter.

7. Conclusions

We have discussed a VLSI CAD system capable of synthesising a restricted class of architectures from a very high-level description. A unique approach to design capture decoupling specification of processor function, throughput, and accuracy is introduced. Intelligent attribute propagation relieves the designer of repetitive well-defined tasks while preserving the flexibility to specify details of interest. The resulting system allows DSP designers to specify efficient processors without detailed knowledge of the underlying implementation. Estimation features such as noise tracking assist in synthesis as well as providing the designer with insight into system behaviour. High-level simulation allows rapid exercise of the engine in the intended architectural and operational circumstances. After the user has identified a central task and an operational template, the design process is rapid and automatic.

References

1. M. Potkonjak and J. Rabaey, "A Scheduling and Resource Allocation Algorithm for Hierarchical Signal Flow Graphs," *Proc. 26th IEEE/ACM DA Conf.* pp. 7 - 12 (Las Vegas, NV, June 1989)
2. R. I. Hartley et al., "A High Speed FIR Filter Designed by Compiler," *Proc. IEEE CICC'89* pp. 20.2.1 - 4 (San Diego, CA, May 1989)
3. S. G. Smith, R. W. Morgan, and J. G. Payne, "Generic ASIC Architecture for Digital Signal Processing," *Proc. IEEE ICCD'89* pp. 82 - 85 (Cambridge MA, October 1989)
4. L. Herlitz, P. Titchener, and J. Coles, "Automating the Design Process for DSP ASICs," *High Performance Systems* pp. 72 - 78 (December 1989)
5. S. G. Smith, R. W Morgan, and J. G. Payne, "A High-Level ASIC Design Tool for Digital Signal Processing," *Proc. IEEE ICASSP'90* pp. 52.V5.8.1-4 (Albuquerque NM, April 1990)
6. P. B. Denyer and D. Renshaw, *VLSI Signal Processing - A Bit-Serial Approach,* Addison-Wesley (1985)
7. S. G. Smith and P. B. Denyer, *Serial-Data Computation,* Kluwer Academic Publishers (1988)
8. S. G. Smith and J. M. Rischard, "20 MHz 16 x 16 Discrete Cosine Transform IC: CAD and Architectural Methodology," pp. 369 - 378 in *VLSI'89,* ed. G. Musgrave & U. Lauther, Elsevier Science Publishers (1990)

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

FORMALIZATION OF DSP ARCHITECTURES SYNTHESIS

Khaled M. Elleithy and Magdy A. Bayoumi

The Center for Advanced Computer Studies
University of Southwestern Louisiana
Lafayette, LA 70504, U.S.A.

The major drawback of reported high level synthesis techniques is their limited applicability to a specific class of algorithms without extendibility to general algorithms and the lack of a formal approach to prove the correctness of the such techniques. In this paper, we introduce a novel approach for high level synthesis for DSP architectures. Two features are provided by the approach: *completeness* and *correctness*. Completeness means the ability to use the approach for any general algorithm. Correctness is achieved by using a set of transformations that are proved to be correct. A formal framework for the synthesis procedure has been developed which can be easily automated. Simultaneous recursion is used to synthesize parallel architectures.

1. Introduction

The advances in VLSI fabrication and packaging technologies have led to having very complex systems of 50,000 to 600,000 transistors on a single chip. Designing, testing, and verifying these chips using ad-hoc design techniques can take a life-time. As a result, cost-effective VLSI design methodologies have become vital to produce correct chips which meet the corresponding specifications without exhaustive iterations and redesign. The difficulty of chip designs that it involves several levels of design; starting from chip level specifications to the geometrical layout. Automating the whole design spectrum has become of the main goals.

The scope of this paper is a formal high level synthesis DSP architectures. Those systems have special features to be considered such as:

(1) DSP architectures are based on fixed functions.

(2) Algorithms are data-independent, so control can be hard-wired. The most efficient hardware solutions directly reflect the data-flow inherent in the algorithm.

(3) Processor throughput is more important than processor latency. Throughput is achieved by pipelining and functional parallelism.

Several systems has been developed for high level synthesis of VLSI architectures from DSP algorithms[1-7]. Those approaches have concentrated on resolving and optimizing several architectural issues such as using a single fixed architecture with a parameterizable data path and control unit, selecting one from a set of these units, or trying to use kind of multiprocessor systems. Also, most of these systems are only applicable to special class of algorithms. Our new approach addresses two issues: *Completeness* and *Correctness*.

The proposed synthesis framework can be summarized as follows:

(1) The given algorithm is represented in the a new language termed ASL (Algorithm Specification Language) which is capable of representing any algorithm using a few number of constructs based on μ-recursive functions.

(2) A transformational method is given to transform an algorithm represented in ASL to a specific realization format termed ASL (Algorithm Specification Language). The RSL version specifies the components and connectivity of the digital architecture that realizes the algorithm. Every construct in ASL has an isomorphic representation in RSL, which is the basis of the automated transformation.

2. Overview of the System

A formal framework for high level synthesis has been introduced in [8-10]. The architectures produced by this framework can be classified as uniprocessor architectures. In this paper, we generalize the approach so that it can produce a DSP multi-processor architecture. The system is composed from two subsystems: synthesis subsystem and user-interface subsystem.

2.1 Synthesis Subsystem

(1) The given algorithm is specified in a new language, termed ASL, which is based on μ-recursive functions. ASL is capable of specifying any algorithm using a limited number of constructs. Although the constructs are very primitive, complex constructs can be used through a developed cell library. Units that have been designed through this design methodology can be added to the cell library.

(2) A transformation technique is developed to transform an algorithm represented in ASL to a specific realization language, termed RSL. The RSL version specifies the components and connectivity of the digital architecture that realizes the algorithm. Every construct in ASL has an isomorphic representation in RSL, which is the basis of the automated transformation. That transformation algorithm is proved to be correct. Everything in the system is built from certain primitives. The proofs of correctness are applied to these primitives. A hierarchical proof is used for ensuring correctness at different levels.

(3) A library of the basic functions is defined starting from the initial functions to be used in the definition of larger

functions. This approach is useful for building a cell library to support the automated synthesis system. All the basic functions that have been designed using the proposed approach can be used for specifying other functions. This technique supports a hierarchical design methodology in the sense that the specification can be stopped at any level as long as the lower levels were previously defined. When *Basic functions* are used to represent another functions, each *Basic function* is written in a box.

2.2 User Interface Subsystem

The user interface environment that is used for the synthesis process will be implemented as a logic programming environment. The logic programming environment supports specifying, simulating, and testing Digital Signal Processing (DSP) systems. Backtracking and pattern matching of Prolog are employed for simulation and testing, respectively. Prolog provides homogeneity to the developed system as it supports hierarchical development and mixing of description at various hierarchical levels. It can be employed for many DSP algorithms and applications development.

Two transformation algorithms are used to link the synthesis subsystem and the user interface subsystem. The purpose of these two algorithms is to allow the user to use the system through the logic programming environment without the need to know the details of ASL and RSL. The two algorithms are as follows:
(1) The first algorithm is used to transform between Prolog and ASL. This transformation algorithm allows the user to specify his algorithm in terms of Prolog.
(2) The second algorithm is used to transform between the RSL and prolog. This algorithm allows us to have the output circuit specified in Prolog.

3. Simultaneous Recursion

In this section we introduce simultaneous recursion that can be used in specifying many algorithms. We show how it can be transformed to the standered forms or implemented without transformation.

If x_i $(1 \leqslant i \leqslant r)$ are n place functions, z_i $(1 \leqslant i \leqslant r)$ are $n+1$ place functions and y_i $(1 \leqslant i \leqslant r)$ are $n+r+1$, then z_i $(1 \leqslant i \leqslant r)$ are defined by the following ASL:

$$z_1(arg_1, \cdots, arg_n, 0) = x_1(arg_1, \cdots, arg_n)$$

$$\cdots\cdots\cdots\cdots$$

$$z_r(arg_1, \cdots, arg_n, 0) = x_r(arg_1, \cdots, arg_n)$$

$$z_1(arg_1, \cdots, arg_n, m+1) =$$

$$y_1(arg_1, \cdots, arg_n, m, z_1, \ldots, z_r)$$

$$\cdots\cdots\cdots\cdots$$

$$z_r(arg_1, \cdots, arg_n, m+1) =$$

$$y_r(arg_1, \cdots, arg_n, m, z_1, \ldots, z_r)$$

3.1 Transformation to one equation:

Simultaneous recursion does not take us out of the class of recursion[11]. The previous system can be replaced by the following equation[11]:

$$\phi(arg_1, \ldots, arg_n, m) = \prod_{i=1}^{r} p_i^{z_i(arg_1, \cdots, arg_n, m)} \qquad (1)$$

such that:

P_i is the ith prime number {1, 2, 3, 5, ...}
z_i can be computed from:

$$z_i(n) = \exp_i(\phi(n))$$

such that:

$$\exp_i = \mu_j[j \leqslant n \ \& \ n \text{ is not divisible by } p_i^{j+1}]$$

Let us show the ASL representation of equation(1):

$$\phi(arg_1, \ldots, arg_n, m+1) =$$

$$y(arg_1, \cdots, arg_n, m, z_1, \ldots, z_r)$$

$$y(arg_1, \cdots, arg_n, m, z_1, \ldots, z_r) =$$

$$product(pow(p(i), z_1(arg_1, \cdots, arg_n, m)), \ldots$$

$$pow(p(i), z_r(arg_1, \cdots, arg_n, m)))$$

$$product(arg_1, \ldots, arg_r) =$$

$$pro(\ldots (pro(pro(arg_{n/2-1}, arg_{n/2}), pro(arg_{n/2+1}, arg_{n/2+2}))) \ldots)$$

3.2 Direct implementation:

Instead of transforming the r recursion equations to one equivalent equation we implement each equation using the same method described in section 5. Here is the RSL representation of the system:

$$Initp(0, m\ ;\ 1, arg_1\ ;\ 2, arg_2\ ;\ \cdots\ ;\ n, arg_n) \qquad (1)$$

$$suc_{control_1} = g_1(\overline{arg})^{ready} \qquad (2)$$

$$\cdots\cdots\cdots\cdots$$

$$suc_{control_r} = g_r(\overline{arg})^{ready}$$

$$I = \rho_1^{n+1}\, suc(I) \qquad (3)$$

$$Ready = eq?(I, m) \qquad (4)$$

$$Result_1 = comp(\overline{arg}, I, \overline{y}, \rho_{g(\overline{arg})}^{n+r+1}\ Result\ \#\ y_1) \qquad (5)$$

$$\cdots\cdots\cdots\cdots$$

$$Result_r = comp(\overline{arg}, I, \overline{y}, \rho_{g(\overline{arg})}^{n+r+1}\ Result\ \#\ y_r)$$

Equation 1 is used to indicate that we use n registers to be initialized with the arguments of the h component and a register for the constant m. Equation 2 means that the unit *Suc* which is a basic function has its inputs $control_i$ $1 \leqslant i \leqslant r$ connected to the $ready_i$ output of the unit computing $g_i(\overline{n})$ to be sure that I is not incremented until $g_i(\overline{n})$ is computed. Equation 3 is used to represent the fact that I is incremented every clock cycle using the *Suc* unit, and I is initialized to the value 1 using the register number $n+1$. Equation 4 determines the end of operation when I reaches the value m. Equation 5 means that h_i has n inputs from the n registers containing the input arguments, r inputs from r registers containing the values computed from $\overline{y}$, one output from register number $n+r+1$ which is initialized to the value $g(\overline{n})$, and an input I which is the output of the *Suc* unit.

4. A Design Example

Recently, The Residue Number System (RNS) has received increased attention due to its ability to support high-speed concurrent arithmetic. The example we introduce here is a modulo adder as an application from this area. The modulo addition represents the computational kernel for RNS-based architectures. Subtraction is performed by adders using the additive inverse property. Multiplication can be transformed

into addition by several techniques. Also, modulo addition is the basic element in the conversion from RNS to binary using the Chinese Remainder Theorem (CRT).

In[12] a $\theta(1)$ modulo addition algorithm has been developed. Using the design methodology introduced in the previous sections, the ASL and RSL representation of the modulo adder architecture are shown in Table 1 and Table 2 respectively. A schematic diagram of the architecture is shown in Figure 1.

5. Conclusions

A formal framework for DSP architectural synthesis has been introduced in this paper. In order to exploit the parallelism in the given algorithms simultaneous recursion is used to represent the algorithm. Techniques for transformation to standered forms or direct implementation are introduced. Applications from the finite field arithmetic theory are analyzed using the formal framework.

The framework enjoys the following advantages: it is suitable for large problems, algorithms of linear order complexity are used, no priori target architecture should be known, no restrictions on the input description, the technique is fully automated, the designer is not responsible for specifying the operations sequencing and communications among different unit, the approach is applicable to any general algorithm, a logic programming environment supports user interface, and parallel properties of the algorithms are explored.

References

[1] J. Rabaey, Ss. Pope, and R. Broderson, "An Integrated Automatic Layout Generation System," IEEE Trans. Computer-Aided Design, vol. CAD-4, July 1985, pp. 285-296.

[2] P. Rutez et al., "Computer Generation of Digital Filter Banks," IEEE Trans. Computer-Aided Design, vol. CAD-5, pp. 256-265, Apr. 1986.

(1) $modulo\_adder_{Sum}(A_S, A_C, B_S, B_C) = Temp_9(A_S, A_C, B_S, B_C)$
(2) $modulo\_adder_{carry}(A_S, A_C, B_S, B_C) = Temp_{10}(A_S, A_C, B_S, B_C)$

{Stage # 1}
(3) $Temp_1(A_S, A_C, B_S, B_C) = Sum(A_S, A_C, B_S)$
(4) $Temp_2(A_S, A_C, B_S, B_C) = Carry(A_S, A_C, B_S)$

{Stage # 2}
(5) $Temp_3(A_S, A_C, B_S, B_C) = Sum(Temp_1(A_S, A_C, B_S, B_C), Temp_2(A_S, A_C, B_S, B_C), B_C)$
(6) $Temp_4(A_S, A_C, B_S, B_C) = Carry(Temp_1(A_S, A_C, B_S, B_C), Temp_2(A_S, A_C, B_S, B_C), B_C)$

{Stage # 3}
(7) $Add\_bit(Var_1, Var_2) = \boxed{\text{Add}}(\boxed{\text{Add}}(Var_1[n+1], Var_2[n+1]), 1)$
(8) $Case(A_S, A_C, B_S, B_C) = add\_bit(Temp_2(A_S, A_C, B_S, B_C), Temp_4(A_S, A_C, B_S, B_C))$
(9) $Temp_5(A_S, A_C, B_S, B_C) = \eta^3_{Case(A_S, A_C, B_S, B_C)}(Temp_3(A_S, A_C, B_S, B_C), Sum(Temp_3(A_S, A_C, B_S, B_C), Temp_4(A_S, A_C, B_S, B_C), (2^n - m)), Sum(Temp_3(A_S, A_C, B_S, B_C), Temp_4(A_S, A_C, B_S, B_C), 2(2^n - m)))$
(10) $Temp_6(A_S, A_C, B_S, B_C) = \eta^3_{Case(A_S, A_C, B_S, B_C)}(Temp_3(A_S, A_C, B_S, B_C), Sum(Temp_3(A_S, A_C, B_S, B_C), Temp_4(A_S, A_C, B_S, B_C), (2^n - m)), Sum(Temp_3(A_S, A_C, B_S, B_C), Temp_4(A_S, A_C, B_S, B_C), 2(2^n - m)))$

{Stage # 4}
(11) $Test\_bit(Var) = \lambda(Var[n+1])$
(12) $Temp_7(A_S, A_C, B_S, B_C) = \eta^2_{Test\_bit(Temp_6(A_S, A_C, B_S, B_C))}(Temp_5(A_S, A_C, B_S, B_C), Sum(Temp_5(A_S, A_C, B_S, B_C), Temp_6(A_S, A_C, B_S, B_C), (2^n - m)))$
(13) $Temp_8(A_S, A_C, B_S, B_C) = \eta^2_{Test\_bit(Temp_6(A_S, A_C, B_S, B_C))}(Temp_5(A_S, A_C, B_S, B_C), Carry(Temp_5(A_S, A_C, B_S, B_C), Temp_6(A_S, A_C, B_S, B_C), (2^n - m)))$

{Stage # 5}
(14) $Temp_9(A_S, A_C, B_S, B_C) = \eta^2_{Test\_bit(Temp_8(A_S, A_C, B_S, B_C))}(Temp_7(A_S, A_C, B_S, B_C), Sum(Temp_7(A_S, A_C, B_S, B_C), Temp_8(A_S, A_C, B_S, B_C), (2^n - m)))$
(15) $Temp_{10}(A_S, A_C, B_S, B_C) = \eta^2_{Test\_bit(Temp_8(A_S, A_C, B_S, B_C))}(Temp_7(A_S, A_C, B_S, B_C), Carry(Temp_7(A_S, A_C, B_S, B_C), Temp_8(A_S, A_C, B_S, B_C), (2^n - m)))$

{Full Adder}
(16) $Sum(Var_1, Var_2, Var_3) = \boxed{\text{And}}(\boxed{\text{Or}}(Var_1, Var_2), \boxed{\text{Or}}(Var_1, Var_3), \boxed{\text{Or}}(Var_2, Var_3))$
(17) $Carry(Var_1, Var_2, Var_3) = \boxed{\text{Xor}}(Var_1, Var_2, Var_3)$

Table 1. Modulo Adder Algorithm (ASL)

[3] J. Schuck, M. Glesner, and M. Lacken, "First Results and Design Experience with Silicon Compiler ALGIC," VLSI Signal Processing II, IEEE Press, NY, Nov. 1986.

[4] B. Petersen, B. White, D. Solomon, and M. Elmasry, "SPIL: A Silicon Compiler with Performance Evaluation," Proc. ICCAD, pp. 500-503, Nov. 1986.

[5] G. Goesens et al., "A CAD Methodology for Mapping DSP algorithms onto MP custom Architectures," Proc. IEEEI CCAS 86, May 1986.

[6] J. Rabaey, H. De Man, J. Vanhoof, G. Goosens, and F. Catthoor, "CATHEDRAL-II: A Synthesis System for Multiprocessor DSP systems," in Silicon Compilation, D. Gajsk, Ed., Addison-Wesley, pp. 311-360, 1988.

[7] B. Haroun and M. I. Elmasry, "Architectural Synthesis for DSP Silicon Compilers," IEEE Trans. on Computer-Aided Design, Vol. 8, Apr. 1989, pp. 431-447.

[8] K. M. Elleithy and M. A. Bayoumi, "A Formal High Level Synthesis Approach for DSP Architectures," Accepted for inclusion in the 1990 International Conf. on Acoustics, Speech and Signal Processing, Albuquerque, New Mexico, April 1990.

[9] K. M. Elleithy and M. A. Bayoumi "A Frame-work for High Level Synthesis of Digital Architectures from μ-recursive Algorithms," Proc. of the ACM Eighteenth Annual Computer Science Conference, pp. 305-311, Feb. 1990.

[10] K. M. Elleithy and M. A. Bayoumi, "A Formal Framework for Synthesis of Parallel Architectures," Accepted for Inclusion in the Fourth Annual Symposium on Parallel Processing, April 1990.

[11] R. Péter, Recursive Functions, Academic Press, 1967.

[12] K. M. Elleithy and M. A. Bayoumi, "A $\theta(1)$ Algorithm for modulo Addition," Accepted for inclusion in the IEEE Transactions on Circuits and Systems.

(1) $modulo\_adder_{Sum} = Comp(A_S, A_C, B_S, B_C \,]\# \; Temp_9)$

(2) $modulo\_adder_{Carry} = Comp(A_S, A_C, B_S, B_C \# \; Temp_{10})$

{Stage # 1}

(3) $Temp_1 = Comp(A_S, A_C, B_S \# \; Sum)$

(4) $Temp_2 = Comp(A_S, A_C, B_S \# \; Carry)$

{Stage # 2}

(5) $Temp_3 = Comp(Temp_1, Temp_2 \# \; Sum)$
$Temp_3^{Ready} = And(Temp_1^{Ready}, Temp_2^{Ready})$

(6) $Temp_4 = Comp(Temp_1, Temp_2 \# \; Carry)$
$Temp_4^{Ready} = And(Temp_1^{Ready}, Temp_2^{Ready})$

{Stage # 3}

(7) $X_1 = Comp(Var_1[n+1], Var_2[n+1] \# \boxed{\text{Add}})$
$Add\_bit = Comp(X_1, \rho^1 \# \boxed{\text{Add}})$
$Add\_bit^{Ready} = And(X_1^{Ready}, \rho^{1Ready})$

(8) $Case = Comp(Temp_2, Temp_4 \# \; Add\_bit)$
$Case^{Ready} = And(Temp_2^{Ready}, Temp_4^{Ready})$

(9) $X_2 = Comp(Temp_3, Temp_4, \rho^2_{(2^n-m)} \# \; Sum)$
$X_2^{Ready} = And(Temp_3^{Ready}, Temp_4^{Ready}, \rho^{2Ready})$
$X_3 = Comp(Temp_3, Temp_4, \rho^3_{2(2^n-m)} \# \; Sum)$
$X_3^{Ready} = And(Temp_3^{Ready}, Temp_4^{Ready}, \rho^{3Ready})$
$Temp_5 = mux_1(Temp_3, X_2, X_3 \# \; Case)$
$mux_1^{Ready} = mux_{1Control}$

(10) $X_4 = Comp(Temp_3, Temp_4, \rho^2_{(2^n-m)} \# \; Carry)$
$X_4^{Ready} = And(Temp_3^{Ready}, Temp_4^{Ready}, \rho^{2Ready})$
$X_5 = Comp(Temp_3, Temp_4, \rho^3_{2(2^n-m)} \# \; Carry)$
$X_5^{Ready} = And(Temp_3^{Ready}, Temp_4^{Ready}, \rho^{3Ready})$
$Temp_6 = mux_2(Temp_3, X_2, X_3 \# \; Case)$

{Stage # 4}

(11) $Test\_bit = suc_1(Var[n+1])$
$suc_1^{Ready} = suc_{1Control}$

(12) $X_6 = Comp(Temp_6 \# \; Test\_bit)$
$X_6^{Ready} = And(Temp_6^{Ready})$
$X_7 = Comp(Temp_5, Temp_6, \rho^2 \# \; Sum)$
$X_7^{Ready} = And(Temp_5^{Ready}, Temp_6^{Ready}, \rho^{2Ready})$
$Temp_7 = mux_3(Temp_5, X_6 \# \; X_7)$
$mux_3^{ready} = mux_{3Control}$

(13) $X_8 = Comp(Temp_5, Temp_6, \rho^2 \# \; Carry)$
$X_8^{Ready} = And(Temp_5^{Ready}, Temp_6^{Ready}, \rho^{2Ready})$
$Temp_8 = mux_4(Temp_5, X_8 \# \; X_4)$
$mux_4^{ready} = mux_{4Control}$

{Stage # 5}

(14) $X_9 = Comp(Temp_8 \# \; Test\_bit)$
$X_9^{Ready} = And(Temp_8^{Ready})$
$X_{10} = Comp(Temp_7, Temp_8, \rho^2 \# \; Sum)$
$X_{10}^{Ready} = And(Temp_7^{Ready}, Temp_8^{Ready}, \rho^{2Ready})$
$Temp_9 = mux_5(Temp_7, X_{10} \# \; X_9)$
$mux_5^{ready} = mux_{5Control}$

(15) $X_{11} = Comp(Temp_7, Temp_8, \rho^1 \# \; Carry)$
$X_{11}^{Ready} = And(Temp_7^{Ready}, Temp_8^{Ready}, \rho^{2Ready})$
$Temp_{10} = mux_6(Temp_7, X_{11} \# \; X_9)$

{Full Adder}

(16) $X_{12} = Comp(Var_1, Var_2 \# \boxed{\text{OR}})$
$X_{13} = Comp(Var_1, Var_3 \# \boxed{\text{OR}})$
$X_{14} = Comp(Var_2, Var_3 \# \boxed{\text{OR}})$
$Sum = Comp(X_{12}, X_{13}, X_{14} \# \boxed{\text{And}})$
$Sum^{Ready} = And(X_{12}^{Ready}, X_{13}^{Ready}, X_{14}^{Ready})$

(17) $Carry = Comp(Var_1, Var_2, Var_3 \# \boxed{\text{Xor}})$

Table 2. Modulo Adder Algorithm (RSL)

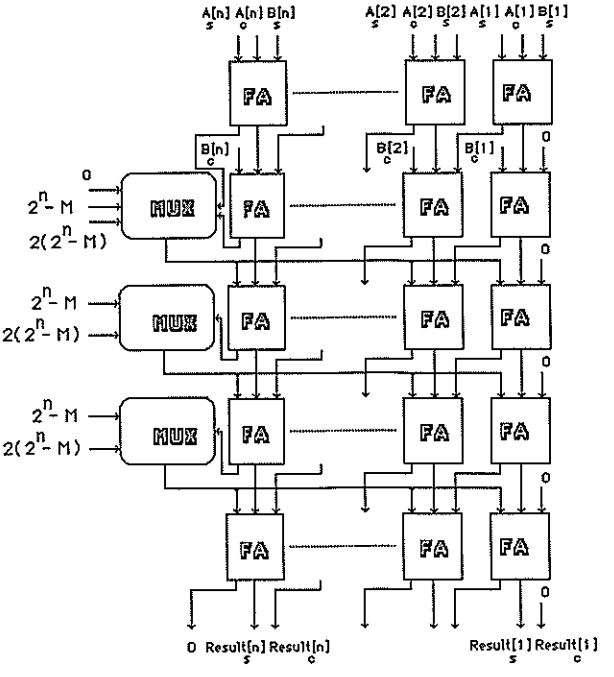


Fig. 1. Modulo Adder Architecture.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A Software Tool For DSP Systems Design and Implementation

M. Veiga, J. Parera and J. Santos

Departamento de Señales, Sistemas y Radiocomunicaciones
ETSI Telecomunicaciones U.P.M.
Ciudad Universitaria, 28040 Madrid (Spain)

1.INTRODUCTION

DSP systems of medium and large complexity demand, for their implementation, the use of hardware architectures exploiting parallelism. Unfortunately, the available programming tools have been, most often, inherited from uniprocessor programming techniques and, therefore, hardware programming can become a very heavy burden.

We introduce a programming environment oriented to specify, design, execute and debug real-time DSP applications running on multiprocessor (MIMD) architectures.

2. THE ENVIRONMENT

The environment can be considered as a collection of integrated services that covers, almost entirely, the designer's activities from system specification up to system maintenance. Integration [1] means that services are not independent tools; all the utilities share a common data base containing the information handled by the environment. This fact is a crucial point because it determines the capability of the environment to help the user, not only by providing several services, but by keeping track of the user's actions in order to free him of executing mechanical activities: coherency checks, control of versions, automatic linking, file dependencies, etc.

The environment resides in a host computer (a SUN-3 Workstation in the prototype) connected to a MIMD machine (a hypercube computer in the prototype) where the DSP applications are executed.

The host operating system (O.S.) is UNIX and the environment makes extensive use of its utilities. The habitual peripheral devices are required: graphic display, massive storage devices, plotter, etc.

For portability reasons the MIMD machine is seen as a virtual machine characterized by a reduced set of virtual O.S. services that must have its counterpart in the actual O.S. of the multiprocessor. Changing the multiprocessor only requires a new implementation of the virtual O.S. in the new machine.

3. THE LANGUAGE OF THE ENVIRONMENT

The language ties the environment to DSP applications. It belongs to the Large Grain Data Flow (LGDF) class of languages and is conceptually based on Lee's and Messerschmitt's works on Synchronous Data Flow Graphs (SDFG) [2],[3] having, accordingly, a block diagram like semantics. Parallelism is natural in these diagrams.

DSP System Specification is accomplished through three different stages: Algorithm, Partition and Distribution. This division separates what the System does from how it is done. Every stage is represented by a Graph and Graphs play the role of Source Programs.

3.1 Algorithm

The Algorithm is an implementation independent description of the system functional behaviour. Algorithms are mainly composed of Nodes (Program Units) and Arcs (Data interchanged between Program Units).

Nodes: Any abstract operation present in a program is called a node. By definition a node is a data transformation place that takes data from every input at the same time - before starting execution - and puts data on every output at the same time - after finishing execution. Every node is a sequential unit. This means that node execution will take place in one Von Neumann's processor.

For the algorithm graph two kind of nodes, corresponding to two hierachycal levels are

supplied: hand coded function and composite function.

Hand Coded Functions constitute the lowest definition level of an action and can be seen as the operators of traditional high level languages. These functions are routines written in any high (C, FORTRAN) or low (ASSEMBLERS) level language for which there exists an adequate compiler or assembler.

Composite Functions are the result of enclosing in a black box an interconnected net of functions which can be, in turn, hand coded or composite. Hierachy is introduced in the language via the composition mechanism.

Allowing the user to codify the language operators in another language decreases the cost associated to encapsulate small grain operations.

Data Abstractions: Block Diagram Simulation Languages, known as next-state-simulators, consider that the only data type interchanged between modules are signal samples (normally sent and received by A/D and D/A converters). Although this is the usual situation, in order toenhance the expresiveness of the language, typical data types found in C language are supported: simple types int, char, float, double, etc, and composite types of the form array of simple types.

The language contains two data abstractions: interface variables and arcs. They play the same role than formal and real parameters in high level languages routines.

Interface variables are characterized by type and dimension and two other attributes provided by the language: the first attribute only affects input variables, and determines if the data they carry constitute a firing condition of its node (a node fed only by input variables affected by the non-fire-condition attribute can always execute). The second attribute affects both input and output interface variables and is called synchronism.

An input variable will be named asynchronous if the number of data consumed during the execution of the node is only known after execution, synchronous if it is known in advance. An output variable will be named asynchronous if the number of data produced during the execution of the node is only known after execution, synchronous if it is known in advance.

Arcs connect output variables to input variables. Arc attributes depend on the attributes of the variables they connect saving one attribute affecting only external arcs (arcs connecting i/o devices to operator nodes): the clock attribute.

An external arc carrying a clock signal is a stronger fire condition than normal arcs. The arrival of each new signal forces the node to execute, although there are not enough data in the rest of the incoming arcs.

Control: The language follows some sort of Hatley's [4] extension to Structured Design in order to include a control discipline wider than pure data flow (a node would be executed any time the necessary data are present at its input). Basically, control structures are conditional statements affecting nodes in the sense active/inactive.

Composite functions or Algorithms can be accompanied by a control centre. A control centre can recieve as input any arc coming from an node (data arc). The outputs of the control centre are inhibitor arcs taking one of two values: active/inactive. Outputs from a control centre can affect any operator node.

The information contained in a control centre is twofold: declares explicitly what nodes become inactive for every value of an inhibitor arc and defines the user supplied C routine in charge of processing all the information coming into the control centre and producing a set of binary values on the inhibitor arcs.

3.2 Partition

When the user, or a program, decides what parts of the system will be executed simultaneously, he is actually making a partition of the system.

The partition breaks the previous algorithm graph into groups of nodes, each one of these groups is called a Task (the uppest level node). Tasks are very much similar to Composite Functions. In fact, the only difference stays in that while Composite Functions are merely procedural definitions of an action, Tasks execute the action concurrently.

3.3 Distribution

A Distribution map assigns Tasks to Processors, leaving the application on the machine ready for execution.

Both Partition and Distribution constitute the so called mapping problem that belongs to the NP-Hard class of computing complexity [5]. The environment does not supply a tool to go from the algorithm stage to a distribution map but allows the introduction of any mapping program for research purposes.

4. THE TOOLS OF THE ENVIRONMENT

4.1 The Graph Compiler

The Compiler translates data flow diagrams to an intermediate language (C Language) which in turn must be compiled to obtain object code for the target processor. We have choosen C due to the availability of C compilers for almost any DSP floating point processor.

Basically, the compiler is in charge of:
- Syntax and Semantic Analysis of Source Progams.
- Generating a C routine for every composite function.
- Generating a C routine for every task defined in the partition stage.

The principal job of the compiler is related to the generation of C routines for every composite function and every task defined in the system. The compiling process is divided in two phases: the first one consists, mainly, of finding a PASS [2] [3] (Periodic Admisible Sequential Schedule) for the block diagrams, while the second phase translates the sequency of shots of the nodes into C sequential code. The first phase is common to both diagrams, composite functions and tasks. The second phase is slightly different for each one.

Obtaining a PASS. The interested reader should consult [2] where it is proved the existance of a class of algorithms (Class-S Algorithms) which will find a PASS for a SDF (Synchronous Data Flow) graph, if a PASS exists; if the algorithm cannot find a PASS, no PASS will exist for the graph. An SDF graph implies that every arc in the graph is synchronous, that every arc is a fire condition of the node that feeds and that the control discipline is pure data flow. In [3] Lee & Messerschmitt accept a limited control structure if-then-else.

The kind of graphs supported by the language scapes the SDF graph category, so the compiler must cope with control structures other than pure data flow, the presence of a limited class of asynchronous arcs and nodes fed by arcs whose data are not necessary to fire them.

Control Centre. A diagram containing a control centre encloses several graphs (not simultaneously actives). The compiler obtains the PASS associated to every graph and the conditional structure that defines which of the possible behaviours correspond to every moment. Note that once a full set of values of the inhibitor arcs is known, the graph is completely determined.

Asynchronicity arises when the number of data carried by an arc is not defined at compile time. There are three common situations:
- The number of data produced by an operator node in some output variable is only known after the execution of the node. For example: a module implementing a Huffman's coding sheme, where the length of the generated codewords -the number of output data- depends on the values of the input data.
- The number of data consumed by an operator node from some input variable is only known after the execution of the node. For example: In a modem, due to slight differences in the sampling frequencies of the local and remote clocks, the receiver can consume more or less signal samples to recover the transmitted symbol.
- Both, the number of data produced and consumed on the same arc are unknown in advance, but there is a synchronous arc connecting the same nodes. For example: any connection between two high level routines such as one parameter (whose value may change from run to run) tells the called routine how many data have been produced on another variable and, therefore, must be processed.

The compiler only accepts two kinds of asynchronous arcs: those arcs belonging to the third situation or those arcs belonging to the first two situations, but such as if they are removed from the graph, this graph will appear divided in two subgraphs with no connections between them (condition of separability).

The reason why the compiler rejects the first and second case, when the arc does not observe the condition of separability, is located in the necessary condition of existance of a PASS imposed to the rank of the topology matrix of a connected graph [2]: any aditional arc must be a linear combination of the matrix files. This means that if the arc does not observe the condition of separability, the arc must be at every moment, during execution, a linear combination of the rest of the arcs. If only one of the variables connected by the graph is asynchronous, it will be very probable that, during execution, the linear combination cannot be hold. Then, comparing the difficulty of compiling a graph with no separable asynchronous arcs with the benefits achieved, we have decided to avoid these situations. When the condition of separability holds, a PASS can be found by extending the PASS concept in order to include conditional execution.

In the third scenary the compiler asumes that whatever the number of data carried by the arc are, the asynchronous relationship does not violate the synchronous rate. This choice obeys to the practical importance of the situation and to the fact that the probability of keeping the linear combination safe is greater in this case than in the two others, due to the existance of an additional degree of freedom (both interface variables are asynchronous, so changes in one of them can be easily absorbed by the other). This scenary is not treated by the compiler as an asynchronous one, because the asynchronous relationship is considered to be overrided by the synchronous one.

Read-only arcs modifies the Class-S algorithms by giving every node the oportunity to fire before allowing a node to fire continously and by removing the arc when testing if the node can start execution.

Translating a PASS into C sequential code. The only difference between the generated code for a composite function and for a task is located in the parameter passing mechanism. Composite functions are implemented as C functions and its interface variables appear in the header of the function. Tasks are implemented as C functions too, but interface variables are received and sent via the communication services of the O.S.

Note that asynchronicity between tasks does not constitute any problem because the data-passing mechanism is asynchronous too. It can be considered that the virtual machine supplies the appropiate mechanism to handle the data flow firing condition, yet in a very high level.

4.2 Main Program Generator

After the graph compiling phase each processor has been asigned a group of functions written in C and/or assembler languages. The environment creates a main program per processor in order to initialize state variables and create the data structure to handle task scheduling.

There are three different types of main programs depending on three different execution modes. The DSP application can be executed in Normal, Monitor or Debugger mode.

Normal mode: The observable behaviour of the system is constrained to input/output operations. The user can only start/stop execution.

Monitor mode: The application runs under the control of the monitor program which acts as some kind of performance meter. The monitor collects measures affecting:
- Time spent in sending messages between tasks.
- Load balancing between processors.
- Time spent by each processor in the following states: computing, inactive, sending/receiving messages.
- Memory utilization

All the information collected by the monitor is stored in the data base for futher processing.

Debugger mode: The application runs under the control of the debugger program. For each component of the multiprocessor the following operations are provided:
- Start, interrupt and stop the execution.
- Control task execution via:
  * Trace/Break points on selected events.
  * Step by step execution. The minimum step corresponds to the execution of a hand coded function.
- Full symbol table access.

For the multiprocessor:
- Suspend/Resume task execution
- Trace/Break points on events related with the message passing mechanism
- Inspect/Modify the state of the tasks.
- Data windows to observe the evolution of signals in the multiprocessor.

4.3 Other Tools

Other auxiliary tools that should be mentioned are the Update Service and the Statistics Service.

The Update Service remembers any operation carried out in the Environment, so posterior changes affecting source programs do not need the user's intervention to rebuild the application.

The Statistics Service is in charge of post processing the information collected by the monitor. It should be usefull for comparing performances of different implementations.

REFERENCES

[1] Lean J. Osterweil, "Toolpack: An Experimental Software Development Environment Research Proyect"
IEEE Transactions on Software Engineering, Vol SE-2, No. 6, November 1986.
[2] E.A. Lee & D.G. Messerschmitt, "Static Scheduling of Synchronous Data Flow Programs for Digital Signal Processing"
IEEE Transactions on Computers, vol C-36, no. 1, pp.24-34, Jan. 1987.[5]
[3]E.A. Lee & D.G. Messerschmitt, "Synchronous Data Flow"
Proceedings of the IEEE, Vol. 75, No. 9, September 1987
[4] D.J. Hatley & I.A. Pirbhai: *Strategies for real time System Specification*. New York: Dorset House Publishing, 1987.
[5]John E. Hopcroft, Jeffrey D. Ullman: *Introduction to Automata Theory, Languages and Computation*. Massachussets: Addison-Wesley Publishing Company, Inc. 1979..

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

RANGE-CHART-GUIDED RATE-OPTIMAL SCHEDULING TECHNIQUES FOR RECURSIVE DSP ALGORITHMS

Sonia M. Heemstra de Groot and Otto E. Herrmann

University of Twente, Faculty of Electrical Engineering
Laboratory for Network Theory
P.O.B. 217, 7500 AE Enschede, The Netherlands

This paper describes an alternative approach for rate-optimal scheduling of iterative data-flow graphs based on the scheduling-range chart. This chart contains the information of the scheduling range of each operation in the data flow graph. The scheduling algorithm looks for an optimal position within the scheduling range of each operation in such a way that some quality criteria are optimized at the same time that a specific sampling rate (rate-optimal as special case) is guaranteed. A scheduling algorithm for minimization of hardware resources, for acyclic as well as cyclic data-flow graphs, is presented. This technique solves optimally problems for which other proposed techniques fail.

1 Introduction

Parallel processing is unavoidable for the implementation of high speed and complex DSP algorithms. The speed of the implementation will not only depend on technology aspects, but also on the efficiency of the design tools. Among all of them, the scheduling and resource assignment tools, have a crucial role.

In this paper, the scheduling problem is modeled in a similar way as proposed by Schwartz in [12]. The DSP algorithm is mapped on a *data-flow graph* (DFG), where the nodes represent tasks or delay elements, and the directed arcs represent the precedence relations. Associated to each task T_i there is a processing time, tc_{T_i}. In this way, it is possible to model acyclic as well as cyclic graphs. Figure 1 shows a cyclic DFG. The DFG together with the set of hardware resources, that for reasons of simplicity will be considered as a set of identical processors, provide the model for the scheduling problem. Interprocessor communication delays are assumed to be neglectable with respect to the computation time. The (processor) scheduling of iterative DFG's consists in finding a suitable periodic ordering of the tasks and their assignment to the set of processors, such that the precedence relationships are not violated, and a certain objective function is optimized.

For a big number of DSP algorithms, with no data-dependent conditionals, the scheduling process can be performed statically, at compile time [8]. Besides, DSP algorithms are characterized by their infinite repetition, which allows the exploitation of not only the parallelism between operations of the same iteration (*inter-iteration parallelism*), but also the parallelism between operations of different iterations (*intra-iteration parallelism*).

In *acyclic DFG's* the sampling period, T_0, can be reduced arbitrarily by using pipelining and supplying enough computing power. The speed bound for *cyclic DFG's* depends not only on the processing time of the tasks, but also on the topology of the graph. This bound, called the *iteration period bound* $T_{0_{min}}$, is determined by the *critical loop* [11] of the graph. Schedules that can be executed at this sampling period are called *rate-optimal schedules.*

Basically, three scheduling techniques that provide such schedules have been reported: maximum spanning tree [11], search of cyclo-static schedules [12], and optimum unfolding [9].

The first has the disadvantage that it does not try to optimize the number of processors, leading in general to solutions with suboptimal processor utilization [5]. The second consists of a depth first search of cyclo-static solutions, by fixing both the sampling period and the number of processors. This method cannot guarantee a schedule since when both parameters are fixed a solution may not exist [7]. The third method consists of reducing the graph to an equivalent perfect-rate data-flow program that can always be scheduled rate-optimally. This is performed by unfolding with an optimal unfolding factor. The main disadvantage is that the

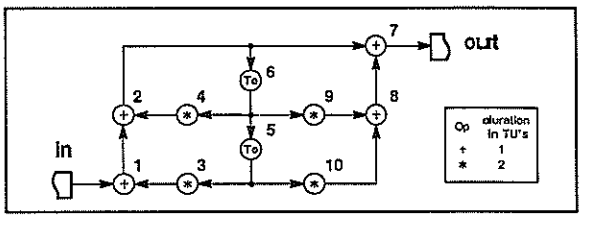


Figure 1: *DFG of a second order digital filter section. The duration is given in an abstract time unit, TU.*

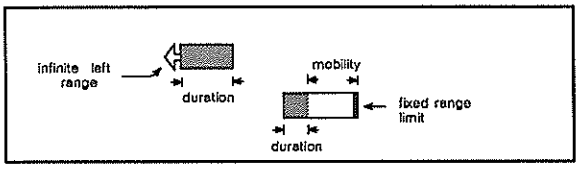


Figure 2: *Notation for the scheduling range.*

memory space necessary to store the program increases proportionally with the unfolding factor.

This paper presents an alternative approach for the rate-optimal scheduling of cyclic DFG's based on the determination of a *scheduling-range chart*. The information in this chart is used during scheduling in order to optimize some quality criteria (number of hardware resources, latency, register life time) at the same time that a rate optimal solution is guaranteed.

2 The Scheduling-Range Chart

The set of positions where a operation can be time scheduled (i.e. assigned to a specific repetitive time slot) is called its *scheduling range*. The *scheduling-range chart* [4] displays this information for every operation in the DFG. The scheduling range is relative to a *reference operation*, an operation that has been time scheduled. The scheduling range of an operation of an iterative DFG can be finite or infinite. The difference between the length of the scheduling range of an operation and its duration is called its *mobility*. When a limit of the scheduling range has a well-defined position, i.e. when all the predecessors or successors of the operation have been assigned to a specific repetitive time slot (time schedule), it is called a *fixed limit*. The notation used in this paper is shown in Figure 2.

To build the scheduling-range chart, an operation is selected as a reference. The scheduling range of the rest of the operations in the graph is determined by (the joined effects of) the following constraints (see Figure 3):

- *Forward precedence relation*
  Any pair of operations, T_i and T_j, interconnected by an arc, such that T_j is a predecessor of T_i, should be scheduled such that:

$$t_{T_i} \geq t_{T_j} + tc_{T_j}$$

  where
  t_{T_k} is the time when operation T_k starts its execution, and
  tc_{T_k} is the execution time of task T_k.
- *Backward precedence relation.*
  Operations which are in a common path, and separated by a node representing one or more delay elements should be scheduled such that the value stored in a register is not rewritten before the successor operation(s) has read it. Expressed in another way: let T_i be an operation whose result is written to node D_m, a node representing a number m of delay elements. Let T_j be an operation that reads a value from D_m. Then, T_i and T_j should be scheduled such that the following relation is respected:

$$t_{T_i} + tc_{T_i} \leq t_{T_j} + mT_0$$

  where
  T_0 is the iteration period.
- *Slack time.*
  This is a consequence of the backward precedence relation among operations that belong to the same loop [4]. The slack time of a loop $\mathcal{L}$ is defined by:

$$slack\ time\ =\ n_{\mathcal{L}}T_0 - \sum_{T_i \in \mathcal{L}} tc_{T_i}$$

  where:
  $n_{\mathcal{L}}$ is the number of delay elements in $\mathcal{L}$.

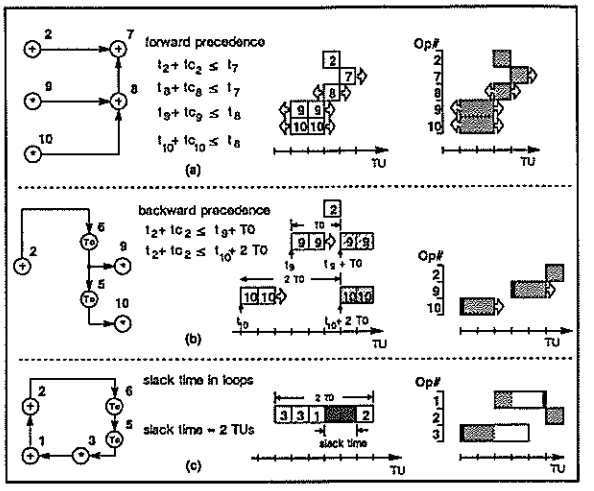


Figure 3: *Scheduling range of operations for the DFG of Figure 1, for $T_0 = T_{0_{min}} = 3$ TU's. Forward precedence (a), backward precedence (b), and slack time (c).*

Figure 4 shows the scheduling-range chart corresponding to the DFG of Figure 1, for $T_0 = T_{0_{min}}$. Operation 2 has been taken as reference. Operation 4 is also member of the critical loop and does not have mobility. Operations 1 and 3 are members of the non-critical loop 3–1–2–6–5; their mobility is equal to the slack time of that loop, i.e. 2 TU's. The rest of the operations have infinite scheduling range. Since operations 2 and 4 have a fixed position in the chart, i.e. they are time scheduled, the scheduling range has a fixed limit for operations 3, 9, and 10, (whose only predecessor is 2), and 1 (whose only successor is 2).

3 The Scheduling Algorithm

When both the sampling period and the number of re-

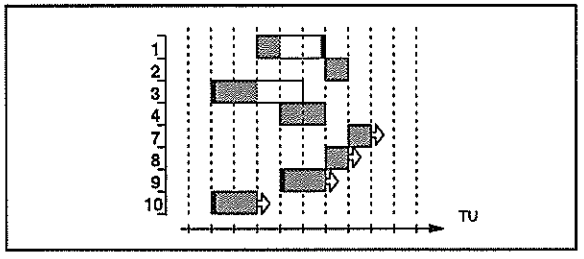


Figure 4: *Scheduling range chart for the DFG of Figure 1, for $T_0 = T_{0_{min}} = 3$ TU's.*

sources are fixed, as is done in [12], a solution for the scheduling problem may not exist [7]. In order to guarantee a solution either of the two parameters should be left free to be optimized by the algorithm. In [6] a scheduling algorithm is discussed that optimizes the sampling rate, for a fixed number of resources. Here an algorithm will be presented that minimizes the number of hardware resources, for a fixed sampling rate. The algorithm can be applied to cyclic as well as to acyclic DFG's.

Since the schedule needs to be periodic in T_0, we need to take into account the effect of the overlapping in time of the execution of operations of different iterations. This is performed by folding the execution of one iteration by modulo T_0 in the range $[0,\ T_0 - 1]$. In this way, the operations are grouped in T_0 *equivalence classes* [5,12]. The operations in each equivalence class are executed in parallel.

The iterative scheduling problem, as discussed in this paper, is NP-complete [2]. Only an informal sketch of a proof is given here. Since the operations in acyclic DFG's have an infinite scheduling range, the folding of the ranges on the interval $[0,\ T_0 - 1]$, will remove any constraint in their positioning. So, any instance of the bin-packing problem, which is NP-complete, can be reduced to an instance of the iterative scheduling problem.

The algorithm described here, schedules the operations by a priority given by their mobility in the scheduling-range chart. The T_0 equivalence classes are divided into levels; one processor per level is necessary. The algorithm assigns operations to equivalence classes and time slots by a priority determined by their scheduling range, and in such a way that the number of levels is optimized. Once this is performed for all the operations, the algorithm enters a new phase where the operations are assigned to processors.

A short description of the algorithm is given below:

1. For acyclic DFG provide T_0.
   For cyclic DFG determine the critical loop and $T_{0_{min}}$. Provide a sampling period such that $T_0 \geq T_{0_{min}}$(make $T_0 = T_{0_{min}}$ for rate-optimal schedules).
2. Select a reference operation and determine the scheduling-range chart for T_0.
3. Partition the scheduling-range chart in sectors that correspond to each equivalence class.
4. Maintain for each equivalence class a pointer to the *first available level*; initially this pointer has a value 1 for all equivalence classes.
5. Select the unscheduleded operation with the shortest scheduling range. In case of equal length, give priority to an operation with a fixed-limit range. Select by label, if multiple operations have a range with fixed limit.
6. Fix the position of the selected operation within its scheduling range, such that it can be assigned to those equivalence classes with the lowest level. Since the level of a class is not associated to any processor, operations that cover more than one class can be assigned to different levels. Update the first available level of the affected equivalence classes.
7. Update the scheduling-range chart to incorporate the fixed position of the operation just scheduled.
8. Repeat steps 5, 6, and 7 until all operations have been scheduled.
9. Assign operations to processors.
   (a) Sort operations according to their computational delay; the longest first; in case of equal duration sort by label.
   (b) Maintain for each equivalence class a pointer to its first available level which now is associated to a processor. This pointer should be initialized to 1 for all equivalence classes.
   (c) Remove the first operation from the list and assign it to the first processor level that has empty all the equivalence classes to which the operation has been assigned to. Update the first-available-level pointers.
   (d) Repeat the last two steps until all the operations have been assigned to processors.

Figure 5 sequentially shows the previous steps for the rate-optimal scheduling of the DFG of Figure 1. Operation 2, that belongs to the critical loop, is used as reference. The result of the processor assignment phase is shown in Figure 5(h); the processor utilization is 100%. Figure 5(i) visualizes the overlapping in the execution of three consecutive iterations.

4 Time-Complexity Issues

The time complexity of the algorithm is dominated by:

1. *The computation of* $T_{0_{min}}$. The computation of this value is necessary for any rate-optimal scheduling algorithm . In [3] an efficient algorithm is presented that does not enumerate all loops. It has a time complexity of $\mathcal{O}(de + d^4)$, where d is the number of delay elements and e is the number of edges in the DFG.
2. *The combination of all inequalities expressing the forward and backward precedence constraints*

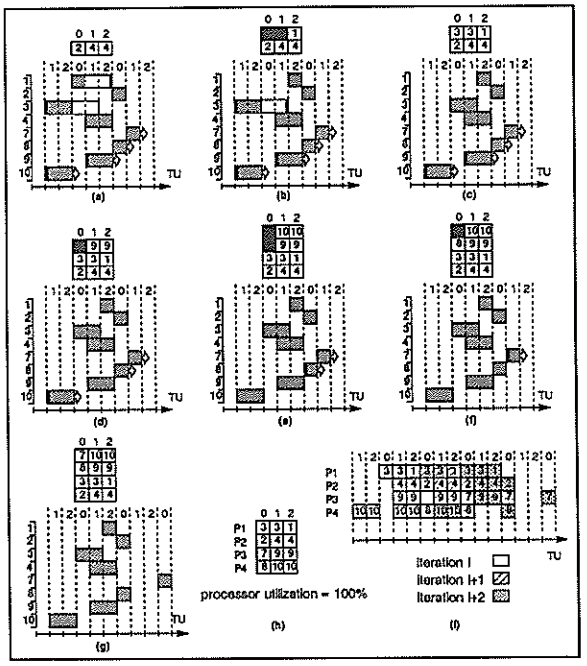


Figure 5: *Scheduling algorithm. Sequential steps of the time-scheduling and equivalence-class assignment, (a) to (g). Processor assignment (h). Processor execution of three consecutive iterations (i).*

to find the scheduling range for each operation with respect to a reference operation. In [3] it is explained how the Fourier-Motzkin elimination method [1] can be adapted to operate in occurring here. The result is an $\mathcal{O}(c^3)$ algorithm, where c is the number of tasks in the DFG.

5 Conclusions

A new approach for rate-optimal scheduling of recursive DSP algorithms has been introduced. It is based on using the information of the scheduling range of the operations, and is an intent to overcome the limitations of other proposed methods. The scheduling algorithm looks for an optimal position within the scheduling range of each operation, in such a way that some quality criteria are optimized, at the same time that a specific sampling rate is guaranteed.

More sophisticated strategies, that consider the global effect that the scheduling of an operation has on the total schedule, can be applied. In [7] a technique based on the forced-directed method [10] is mentioned. The algorithm, although simple, is powerful enough to solve many problems optimally.

Acknowledgement

The authors are very grateful to Sabih Gerez for his many valuable suggestions in the preparation of this paper and his contribution to the time-complexity issues.

References

[1] G.B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, Princeton, New Jersey, 1963.

[2] M.R. Garey and D.S. Johnson. *Computers and Intractability. A Guide to the Theory of NP-Completeness.* W.H. Freeman & Company, San Francisco, 1979.

[3] S.H. Gerez and S.M. Heemstra de Groot. *Implementation and Time-Complexity Issues of Range-Chart Scheduling.* Technical Report, University of Twente, March 1990. Int. Report EL-BSC-90N032.

[4] S.M. Heemstra de Groot. *Scheduling of Iterative Data Flow Graphs by Means of the Scheduling-Range Chart.* Technical Report, University of Twente, October 1989. Internal Report EL-BSC-89N181.

[5] S.M. Heemstra de Groot and O.E. Herrmann. Evaluation of some multiprocessor scheduling techniques of atomic operations for recursive DSP graphs. In *Proceedings of the European Conference on Circuit Theory and Design*, pages 400–404, September 1989.

[6] S.M. Heemstra de Groot and O.E. Herrmann. Maximum throughput scheduling with limited resources for iterative data flow graphs by means of the scheduling-range chart. In *Euromicro Workshop on Real Time Systems*, June 1990.

[7] S.M. Heemstra de Groot and O.E. Herrmann. Rate-optimal scheduling of recursive DSP algorithms based on the scheduling range chart. In *Proceedings of the IEEE International Symposium on Circuits and Systems*, May 1990. New Orleans.

[8] E.A. Lee and D.G. Messerschmitt. Static scheduling of synchronous data flow programs for digital signal processing. *IEEE Transactions of Computers*, C-36(1):24–35, January 1987.

[9] K.K. Parhi and D.G. Messerschmitt. Rate-optimal fully-static multiprocessor scheduling of data-flow signal processing programs. In *Proceedings of the 1989 IEEE International Symposium on Circuits and Systems*, pages 1923–1928, May 1989.

[10] P.G. Paulin and J.P. Knight. Force-directed scheduling for the behavioral synthesis of ASIC's. *IEEE Transactions on Computer-Aided Design*, 8(6):661–679, June 1989.

[11] M. Renfors and Y. Neuvo. The maximum sampling rate of digital filters under speed constraints. *IEEE Transactions on Circuits and Systems*, CAS-28(3):196–202, March 1981.

[12] D.A. Schwartz and T.P. Barnwell. Cyclo-static multiprocessor scheduling on the optimal realization on shift-invariant flow graphs. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1384 – 1387, 1985.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SPECIFICATION FOR DIGITAL SIGNAL PROCESSING: REQUIREMENTS AND SOLUTIONS

D.R. Genin
EDC Heverlee, Begium

Prof. J. Rabaey
Department of EECS UC Berkeley

Designers aim at better specification languages and more efficient implementations. These goals can be achieved if specific tools and specification languages are defined for specific domains. This paper presents a specification language for Digital Signal Processing, Silage, together with some of the tools designed around it.

1. INTRODUCTION

A specification language must provide the designer with the ***symbols*** he uses when thinking about the application domain. The ***graphical representation*** of DSP algorithms (fig. 1.) which implies a data-flow semantic is very popular among the designers. The design of a DSP language must capture the "flavor" of this sort of design in a linear, textual form. The language must not introduce other constraints than the ones implied by a data flow semantics, the ***order or concurrency*** of the operations may not be explicitly stated. Nevertheless modern days signal processing requires often more than a pure data flow semantic. Therefore the designer must also be provided with ***control expressions*** which superimpose a macro control flow on top of the data flow semantic.

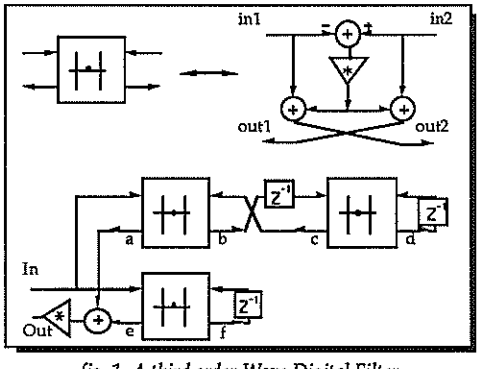


fig. 1. A third order Wave Digital Filter

The language must be able to express precise requirements on ***accuracy*** and ***data boundaries***, since noise, distortion and possible instabilities such as large and small scale limit cycles are extremely important characteristics of DSP algorithms.

Signals must have specific DSP types, overflow and saturation characteristics, they can be grouped in sets, vectors and matrices. High level operators must be available for these data.
The use of the data flow graph semantics has an additional advantage that optimizing transformations, an essential part of every synthesis tools, are more naturally and easily performed in such an environment. We do believe that the introduction of specific and well-defined specification languages drives and encourages the creation of new and efficient tools.

2. SILAGE

The language SILAGE has been developed in Berkeley by P. Hilfinger in 1984 [1] to specifically meet the expectations of the DSP designers. SILAGE is an applicative language, designed to capture the inherent parallelism of DSP algorithms in a linear textual form. The language has proven its suitability as a signal processing specification language and has been used as the input language for a number of design synthesis and code generation systems at different universities and within the industry [4], [5]. [7], [8]. However, experiences with the language revealed some of its deficiencies when describing complex operations such as matrix manipulations and multiple rate asynchronous systems. A number of extensions [2] to the language have been made which cope with the mentioned problems.

A one to one mapping between the signal flow graph (fig.1) and the Silage description (fig.2) can be observed. This example also demonstrates some other important features of the Silage language: applicative language, timing information and time domain operations, data typing, hierarchical description, pragmatic directives.

Silage Description

```
#define word fix<20,8>
#define coef fix<10,4>

func main (in: word) out: word =
begin
  b@@1 = 0;
  (a,b) = Adaptor (In, c, 0.1);
  (c,d) = Adaptor (b@1, d@1, 0.001);
  (e,f) = Adaptor (In, f@1, 0.01);
   out = word ((e - a) * 0.5);
end;

func Adaptor (In1, In2: fix; gamma: coef) Out1, Out2: fix =
begin
  Out1 = State + In2;
  State = (In2 - In1) * gamma;
  Out2 = State + In1;
end;

pragma (1 Alu, 1 Mult)
```

figure 2: Silage code corresponding to fig.1.

2.1 Applicative Language

A natural textual representation for data-flow semantics is an applicative language: a language whose fundamental operation is function application, and that has no variables or assignment operator. A program consists of an unordered set of definitions of functions and signals. The ordering or concurrency of operations can be determined by the compiler and the amount of parallelism is only limited by data dependencies. This allows to derive several different implementations from a single specification of an algorithm.
Functions are used to hierarchically describe an algorithm (*fig* 2.). Silage also provides the user with some predefined functions: absolute value, bit extraction, bit merging, integration, differentiation,.. vector operations like component-sum, sum, maximum, dot- and cross-product, vector delay, tapped-delay line vector,..

2.2 Timing Information and Time Domain Operations

Instead of deriving the timing of the operations from the ordering of the input description, it can be deduced from the signal dependencies (single assignment) and the presence of the delay operators (@). In the C languages the delay operator must be made implicit through difficult scheduling which make it impossible for the C compiler to use special delay instructions. Multi-rate functions are also available in Silage. The *interpolate* and *decimate* functions are specialization of the *switch* function which transforms N input signals into M output signals. The sampling rate of the output is N/M times the sampling rate of the input signals.

2.3 Data typing

All signals in the previous example are of the fixed point type: fix<20,8> means word-length of 20 bits, position of decimal point is 8. Integer, floating point and array types are also available in Silage.
Operations have default types. Data types are determined by deduction (and induction) from the input and output data types.

$$\text{fix}\langle w_1,d_1\rangle * \text{fix}\langle w_2,d_2\rangle \rightarrow \text{fix}\langle w_1+w_2, d_1+d_2\rangle$$

No declaration is thus needed for an intermediate signal. Nevertheless coercions can be used to enforce a data type on a certain signal (as is done for the "Out" signal in the main function).

$$c = \text{fix}\langle 16,15\rangle (a * b)$$

The user is able to select or write different rules that derive default types for standard operations. Generic types have been introduced, which allow for the definition of format free functions. The rigorous data type definitions makes it possible to model exactly the effects of truncation, rounding and saturation. Those effects are of crucial importance in signal processing, where the quality of an algorithm is measured in terms of the signal to noise distortion ratios. Because these types and characteristics are built-in, it is easy for a tool to identify them and to use specialized DSP knowledge in order to handle them efficiently.

2.4 Pragmatic Directives

A pragma is a compiler directive that supplies non-algorithmic information about a program. This information is useful for some compilers. In the above example, the pragma is used to direct the silicon compiler to map the algorithm on an architecture consisting of one ALU and one multiplier.

2.5 Loops and Conditionals

The above example only shows a few characteristics of Silage. Silage also provides powerful iterator constructs like definite and indefinite iteration, conditional expressions, logical expressions.
A loop in Silage is applicative, which means that any statement within the loop is still considered as a definition (not an assignment). This allows the compiler to implement a loop in any order, to expand it, to perform loop optimization such as loop folding or loop migration. Extending the use of the delay operator to loops allows the user to elegantly specify loops in an applicative way. In that

case the delay operator refers to the previous value of a signal or a variable. Control expressions have been introduced to superimpose a macro control flow on top of the data flow semantics (indefinite loop, if-then-else control construct).

3. SILAGE ENVIRONMENT

Designers can only profit from such a language if simulation and implementation environments are present around the specification language. The design of such tools is greatly facilitated when knowledge provided to the tool is formalized and articulated around the *same symbols or concepts* used by human designers. The SILAGE specification is translated into a Signal Flow Graph (SFG) which is manipulated by a set of tools. We do believe that the introduction of specific and well-defined specification languages drives and encourages the creation of new and efficient tools.

When starting the development of a new product the DSP system designer must analyse the characteristics of the different architectural styles. His decision will be influenced by many often conflicting factors. The architectural choices he will consider can be general-purpose signal processors (fixed and floating point) with different word lengths and from different vendors, customized signal processors, hardwired bit-serial and bit-parallel data paths, systolic array architectures,...
This section presents a set of tools designed around SILAGE. These tools originate from different universities and industries (UC Berkeley, KU Leuven, Imec, Philips, Tektronix, EDC). Most of these tools are integrated in the DSP Design Environment currently developed at EDC.

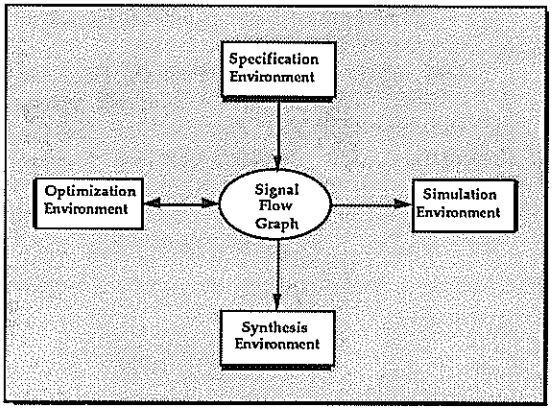


Figure 3.
Components of the DSP Design Environment

3.1 Specification

The *design flow* of a DSP system starts by the specification of an algorithm. This algorithm can either be specified textually by using SILAGE, by an equivalent graphical representation or be generated by one of the Filter Synthesis Tool.
The resulting algorithm is internally transformed and stored in the so-called ***Signal Flow Graph,*** (SFG) which is the central part of the system. In the SFG no commitment is made on the implementation of the algorithm. The maximum possible amount of parallelism in the specification is left unchanged and no assumption is made on the choice of a particular type of architecture or amount of memory. This has the great advantage that once an algorithm has been verified, it can be implemented in different ways without starting each time at the specification level.

3.2 Simulation

In order to verify the algorithm, a simulation phase is required. The result of this step may oblige the designer to partly modify his specification. Four types of simulation can be performed: DSP system simulation, DSP bit-true simulation, DSP simulation accelerator or a simulation of the complete system (DSP and non-DSP).

In the first type of simulation no assumption about the word-length (hardware) is made. The designer is then able to verify the ***functionality*** of his algorithm. When performing a bit-true simulation the system takes into account the word-length assigned to each signal, so that the designer can evaluate the ***quality*** of his algorithm. Because of the repetitive character of DSP algorithms compiled code simulation are much more efficient than event-driven simulation. Therefore the simulator is automatically produced by scheduling the SFG and by producing C code. When system and/or bit-true simulations must be performed on a large amount of input data (subjective tests in audio), it is mandatory to reduce the specification-simulation- analysis cycle by using a hardware accelerator. In this case assembly code is automatically and directly generated from the SFG [2].

A VHDL description of the DSP part of the system can also be produced in the same way and be combined with the description of the non-DSP part to simulate the complete system.

The simulation environment also provides a complete set of analysis tools together with a display package.

3.3 Optimization

The next phase of the design is not implementation independent any more. According to the type of hardware chosen for the implementation, different types of optimization can be performed on the Signal Flow Graph. The *optimization environment* consists of analysis and optimization tools which have access to the simulation environment. The following tools are actually available [6]: Direct Frequency Analysis for linear systems, Finite word length analysis, Noise Analysis, Sensitivity analysis, Template analysis, Characterization of parasitic oscillation (limit cycles), Optimization of canonical signed digit coded coefficients.

3.4 Implementation

The last step of the design cycle is the implementation phase. At this stage the user has also different tools at his disposal. Depending on the result of a cost evaluation analysis, the designer can chose for an implementation on: a commercially available DSP processor [5], a bit-serial ASIC or a bit-parallel ASIC [7], [8]. The silicon compilation consists of the selection and the scheduling of a set of building blocks which can implement the operations described in the signal flow graph. Rules for synthesizing particular architectures have been formulated based on practical design experience. These rules are coded in the knowledge base of a tool which automatically carries out the architecture synthesis.

The designer's choice will be influenced by many often conflicting factors such as computational complexity, word length (precision), regularity, input/output, memory bandwidth throughput range and cost. Selecting the right architecture for a specific task is a non-trivial job. Judging the importance of all influencing factors is difficult. The designer must be able to try different approaches: different word lengths to balance signal/noise ratio and cost, different general- purpose DSP processors, different styles of customized signal processors, different amounts of parallelism to balance speed versus cost and all this for different implementations of the original specification. In order to allow the designer to use this freedom, the design system must *guide* the user through the design process. Therefore a ***framework*** is made available to the designer where predefined script help the designer to navigate through specification- optimization- simulation- implementation cycle.

4. CONCLUSIONS

The language Silage has been presented together with a number of extensions to the original design which result in a powerful specification language and design environment of complex DSP systems. This has been very clearly demonstrated by the coding of many DSP specifications including a complete compact disk with error correction, speech and image recognition and coding (edge detection, ray tracing), adaptive filters for telecommunication, complex interpolation algorithm for digital audio, autocorrelation matrices and inversion,..

REFERENCES

[1] P. Hilfinger, "A High-Level Language and Silicon Compiler for DSP", Proceedings of the Custom Integrated Circuits Conference, May 1985.

[2] P. Hillfinger, J. Rabaey, D. Genin, C. Scheers, H. De Man: "DSP Specification Using the Silage Language" ICASSP 90 Albuquerque

[3] EDC /DSP Station Silage Reference Manual 1990

[4] EDC /DSP Station Basic DSP Manual: Overview of the DSP Design Environment 1990

[5] D.R. Genin, J. De Moortel, D. Desmet, E. Van de Velde, "System Design, Optimization and Intelligent Code Generation for Standard DSP", ISCAS 89, Portland.

[6] R. Jain, G. Goossens, L. Claesen, J. Vandewalle, H. De Man, L. Gaszi and A. Fettweis: "CAD Tools for the Optimized Design of Custom VLSI Wave Digital Filters", ICASSP Tempa, Florida, 1985.

[7] C. Chu, M. Potkonjak, M. Thaler and J. Rabaey, "HYPER: An Interactive Synthesis Environment for High Performance Real Time Applications", IEEE ICCD Conference, Boston, October 1989.

[8] J. Rabaey, H. De Man, J. Vanhoof, G. Goossens, F. Catthoor: "Cathedral II : A Synthesis System for Multiprocessor DSP Systems", in "Silicon Compilation", Ed. Gajski, Addison Wesley 1988.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

THE USE OF VLSI FLOORPLANNING TECHNIQUES TO ALLOCATE PROCESSES TO PROCESSORS IN A MASSIVELY PARALLEL ARRAY.

A.P. WILTON and G. F. CARPENTER

Department of Electrical and Electronic Engineering and Applied Physics,
Aston University, Birmingham, B4 7ET, UK.

Many DSP tasks are so demanding that very large arrays of concurrent processing elements are required. Communication costs then become as important as computation costs. Techniques are required for the optimal mapping of DSP computational processes to physical processors by embedding blocks of processors within massive arrays. By drawing on the analogy between this task and VLSI floorplanning and layout this paper suggests a number of approaches for adaptation in processor allocation.

1. INTRODUCTION

Many DSP applications are so computationally intensive that they cannot be carried out successfully using a conventional sequential architecture. Parallel processing techniques using sets of processes in concurrent execution on arrays of processing elements (PEs) offer the prospect of improved throughput as, at best, [1] a linear function of the number of PEs (n). Hence, to obtain a dramatic increase in performance, one has to compose software from a large set of processes and target this onto very many PEs (typically $n >> 100$). Such a processing system has been termed a Massively Parallel Processor (MPP) [2].

Recently, reconfigurable parallel architectures have been discussed [3] with the aim of reconfiguring all the PEs of a fixed-size array to perform a *single* task. This paper considers the more general case where a variety of disparate computational sub-tasks has to be performed, yet each sub-task is of sufficient complexity to require many PEs.

Present technological constraints dictate that PEs within large arrays are almost inevitably *physically* placed in a two-dimensional rectangular mesh though the *logical* interconnection may vary considerably. If, for each sub-task, an efficient parallel algorithm and appropriate architecture (N-cube, ring, mesh, shuffle-exchange, butterfly etc.) have been determined, then a rectangular block of processors may be assigned to each sub-task and the block reconfigured internally using established techniques [3].

This paper is concerned with methods of allocating such blocks within the total array so as to achieve the best possible overall efficiency. The task is restricted to that of determining the 'coordinates' of each block on the grid defined by the PE mesh.

2. DESIGN APPROACH AND VLSI ANALOGY

Digraphs (such as the Data Flow Graph [4] and Signal Flow Graph [5]) allow the computational task for an MPP to be depicted graphically. Each digraph node is a progressive sub-task and each edge is a data communication path between sub-tasks. A graph of this form (or its undirected equivalent), termed generically a *structure graph* , provides the starting point for the novel allocation technique described in this paper.

Each sub-task in the graph represents a processing requirement which must be achieved. Further decomposition is usually required to derive a hierarchical set of intercommunicating processes which will fulfil this processing requirement. For each sub-task it is then necessary to distribute the set of processes onto the processor array so as to minimise both computation and communication costs (within the sub-task and globally for the overall computational task). In an MPP communication is the key issue and this must be optimised even at the expense of under-utilised PEs.

The allocation problem is analogous to that faced by the designer of a VLSI device who has to lay out a given hardware architecture. During the design of a VLSI system early partitioning decisions are made which lead to the identification of the main architectural blocks. The first stage of layout is to determine the relative placement of these blocks so as to minimise inter-block wiring and reduce the total silicon area. This procedure is termed floorplanning [6]. The detailed layout within each block then proceeds independently.

VLSI designers have had considerable success at solving such problems. This paper examines floorplanning techniques to see if they may be applied usefully to processor allocation in multi-

processor arrays. Particular emphasis is given to DSP algorithms with a natural structure which maps onto rectangular arrays (low-level image processing, convolution etc.) since these allow the common VLSI design practice of using rectangular-bordered modules to be exploited.

3. FLOORPLANNING STRATEGIES

The many floorplanning strategies may be broadly categorised as: graph-based methods and iterative or force-based methods. In VLSI applications the choice between graph-based and iterative methods is largely dependent on the scale of the problem. The appropriate scale for processor block allocation is the number of blocks (m). The problem size for this paper ($10^2 < m < 10^4$) falls centrally between the two extremes (Fig. 1) so both approaches are examined.

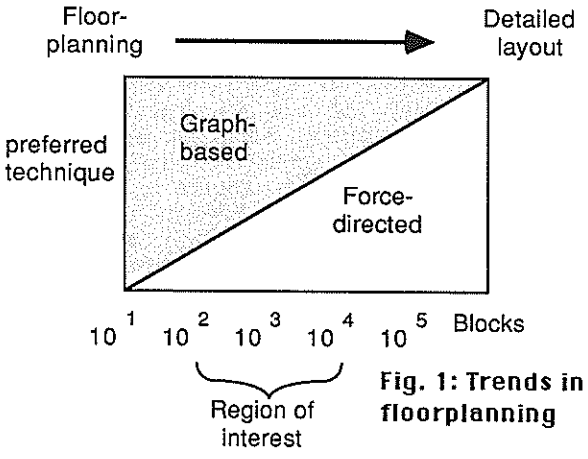


Fig. 1: Trends in floorplanning

3.1. Graph-based methods

Several successful graph-based VLSI floorplanning techniques employ graph duals, in particular the rectangular dual graph (RDG) [7, 8, 9]. It seems likely that the RDG may be of fundamental significance to this problem. Following the early work of Tutte [10], rectangular duals have found application in such diverse areas as building planning, packing of goods in containers and PCB layout as well as VLSI floorplanning.

A Rectangular Dual Graph (RDG) is defined as follows:

An n-vertex graph, $G = (V,E)$, has a rectangular dual D, if: each vertex i in V maps to a distinct rectangle i in D, and, for each edge (i,j) in V, rectangles i, j are adjacent in D.

In general floorplanning strategies using graph dualization consist of the following stages.

1. Planarization. Only planar graphs have duals [11]. Thus the structure graph must be tested for planarity. Efficient algorithms exist for this task [12]. If the structure graph is not already planar it must be made so. This task has been addressed in the context of VLSI floorplanning by Lokanathan and Kinnen [13] who concluded that the most appropriate method involves the introduction of additional (dummy) vertices at unavoidable edge intersections.

2. Triangulation. If an RDG exists, then all regions of the structure graph must be triangular and all complex triangles (cycles of length three which are not face boundaries) must be eliminated [8]. Hence, the (planarized) structure graph must be triangulated by the addition of dummy (zero-weight) edges [7, 8].

3. Edge marking. The edges of the planarized, triangulated graph (PTG) must be marked with appropriate orientations (ie. horizontal or vertical). This task can be reduced to a bipartite graph matching problem [7]. Each region of the PTG must be assigned to one of its adjacent vertices such that each vertex v has $d(v) - 4$ regions assigned to it (where $d(v)$ is the degree of vertex v). Such a matching problem is conveniently solved by variations of the Ford-Fulkerson method.

Strictly, the RDG is now fully specified. However, to obtain the RDG from the marked graph one still has to solve a constrained, linear minimisation problem which can be computationally expensive.

The approach outlined above has been described in detail by Lai and Leinwand [7]. However, there are problems if the procedure is applied to processor allocation. In particular the method uses the ordering of the edges adjacent to each node (e.g. clockwise) for the triangulation procedure. While this information may be available in some cases, it is more likely that the original structure graph will not have been explicitly drawn, so there is no unique edge adjacency ordering. Bhasker and Sahni [9] have reported a complex but efficient ($O(n)$) algorithm for the construction of an RDG directly from a PTG but this is unlikely to generate the smallest RDG and an optimisation scheme is still required.

The advantage of dualization is that it allows the topology of a circuit to be explored for 'interconnection via abutment' as well as conventional routing. This is clearly desirable for process allocation since it represents the tightest possible packing of PEs within a given area. The above method relies on the use of rectangular arrays but study of the technique should allow adaptation for other processing tasks requiring different topologies. The problem of transforming implementations of parallel algorithms from common array topologies (N-cube, shuffle-exchange, butterfly etc.) to rectangular and square arrays has been studied [3] and efficient solutions have been obtained. The special case of the embedding of a

rectangular array requirement into a square mesh has also been investigated [14].

3.2 Example of RDG approach

Consider the small structure graph in Fig. 2. There are 7 sub-tasks (a to g). Assume for simplicity in this example that each may be performed efficiently with a 4 by 4 array of PEs. The arrows represent external communications and are modelled by edges with one end at the infinity node [7]. The structure graph is planarized by the addition of node h. All regions (including the infinity region are then triangulated to obtain the PTG. shown in Fig. 3.

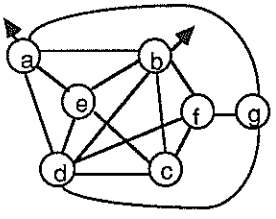


Fig. 2: Structure Graph

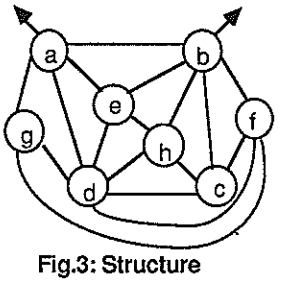


Fig.3: Structure Graph after Planarization and Triangulation.

The dual of the PTG is then obtained (Fig. 4) which gives a reasonable topology for processor allocation. The smallest isomorphic graph now has to be found. This general problem may be intractable [15]; however, a good solution may be obtained using an iterative series of suitable transformations from any initial RDG. Given such a transformation, an obvious candidate for the iteration algorithm is simulated annealing [16]. An optimal solution for the example is shown in Fig. 5.

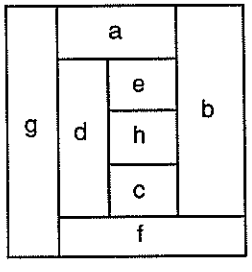


Fig. 4: Rectangular Dual of Structure Graph

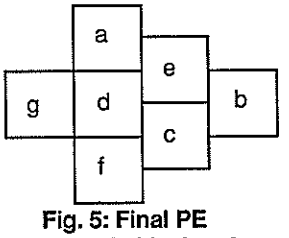


Fig. 5: Final PE allocation in blocks of 16 with minimum communication cost.

It is usual in the use of RDGs for floorplanning to specify that the whole RDG should fit within a rectangular enclosure. While this is a highly desirable feature for a hierarchical system (ie. each RDG block could be floorplanned using the same technique), it is not necessary if there is an ample supply of PEs and the wastage implied by a non-rectangular boundary is acceptable.

3.3 Force-based floorplanning

The problem of finding the most most appropriate placement of a set of interacting blocks may be solved by considering the analogous problem of finding a stable configuration for a set of interacting charged particles. A 'potential energy' model is developed to express the 'attractive force' between connected blocks. Blocks cannot overlap so it must also contain a 'repulsive force' between overlapping blocks. Since this is only empirical, there is considerable freedom in choosing the form of the potential energy. For example Ying and Wong [17] use a combination of hyperbolic repulsion and linear attraction to produce a potential as in Fig. 6(a).

The study of crystalisation has inspired considerable advances in computation techniques (Simulated Annealing [16], Zone-refining [18]). An energy model based on the conventional interatomic energy/spacing relationship [19] leads to the association of a repulsive force giving a positive potential varying with some inverse power, p, of the block separation and an attractive force contributing a negative potential with an inverse power, q. It is easy to show that the potential energy has a mimimum if $q > p$. This is shown in Fig. 6(b).

For a set of blocks: b_i, $i = 1,m$,
connected by a set of nets: n_i, $i = 1,n$, where:

x_i is the x-coordinate of the centre, y_i is the y-coordinate of the centre, w_i is the width and h_i is the height of block b_i,

then, for each pair of blocks b_i and b_j,

d_{ij} is the distance between the centres of blocks i and j. ie.

$$d_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

c_{ij} is the connectivity between blocks b_i and b_j ie.

$c_{ij} = 0$ if b_i and b_j are unconnected, else
c_{ij} = weight of connection between b_i and b_j.

Since rectangular blocks are difficult to manipulate mathematically, it is conventional [17] to model the blocks as circles. The radius of each circle is set to reflect the size of the corresponding block. Clearly, acceptable values will range between half the width and half the height of the block. A reasonable compromise is the arithmetic mean. Thus the radius, r_i, of the circle associated with block b_i is:

$$r_i = \frac{w_i + h_i}{4}$$

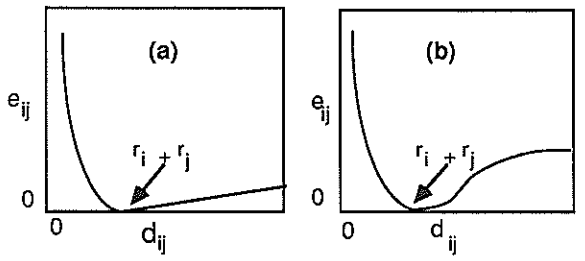


Fig. 6: Inter-block energies for different force models.

The relationships between the energy e_{ij} between blocks b_i and b_j according to the different force models are shown in Fig. 6 .

The objective is to minimise the total energy between all pairs of blocks. ie.

$$\text{Minimise } E = \sum_{i,j=1}^{m} e_{ij} c_{ij}$$

Energy models of this form retain the advantage of being amenable to analytic manipulation. In particular, all first and second partial derivatives exist so numerical methods involving linear search may be applicable. Minimisations of this form are critically dependent on the initial seeding but do converge fairly quickly. Once the relative placement has been found the circles must be replaced by the original rectangles. This will inevitably produce overlaps which must be removed by further perturbations of the floorplan.

4. CONCLUSION

Graph-based methods use more 'global' information throughout the floorplanning process and should lead to more optimal solutions but at the expense of a large computational cost. Force-based methods arrive at a solution must faster, particularly for larger problems, but the quality of the solution is highly dependent on the initial configuration.

A combination of the two approaches may prove attractive; a graph-based technique is used to obtain an initial, topologically acceptable plan and this is then optimised by a force-based iteration scheme. A set of programs has been written to derive processor allocations using this technique. Work is under way to simulate MPP systems and determine the efficiency of the solutions.

This approach is only likely to prove effective for static configuration as any iterative process for the final optimisation will be time-consuming. Suitable applications will require that the total running time of the system in any particular configuration be large compared to the reconfiguration time.

REFERENCES

[1] Quinn, M. J., Designing Efficient Algorithms for Parallel Computers, (McGraw-Hill, 1987).

[2] Batcher, K. E., MPP - a Massively Parallel Processor, Proc. Int. Conf. on Parallel Processing 1979, 249, IEEE.

[3] Yalamanchili, S., Aggarwal, J. K., Reconfiguration Strategies for Parallel Architectures, IEEE Comput, Dec. 1985, pp 44-61.

[4] Dennis, J. B., Data Flow Supercomputers, IEEE Computer, Nov. 1980, pp 48-56.

[5] Kung, S. Y., VLSI Array Processors, in: Moore, W., McCabe, A., Urquhart, R., (Eds), Systolic Arrays, (Adam Hilger, 1987), pp 7-24.

[6] Glasser, L. A ., Dobberpuhl, D. W., The Design and Analysis of VLSI Circuits, Addison-Wesley, 1985.

[7] Lai, Y., Leinwand, S. M., Algorithms for Floorplan Design Via Rectangular Dualization, IEEE Trans. CAD, Vol. 7, No. 12, Dec. 1988, pp 1278-1289.

[8] Tsukiyama, S., Koike, K., and Shirakawa, I., An algorithm to eliminate all complex triangles in a maximal planar graph for use in VLSI floor-plans, IEEE Int. Symp on Circuits and Systems, San Jose, 5-7 May 1986, Vol 1, pp 321-324.

[9] Bhasker, J., Sahni, S., Linear algorithm to find a rectangular dual of a PTG, Proc 23rd Design Automation Conf, 1986, IEEE, pp 108-114.

[10] Tutte, W., et al, Dissection of Rectangles into Squares, Duke Math. Journal, 1940, Vol 7, pp 312-340.

[11] Even, S., Graph Algorithms, (Pitman, 1979).

[12] Hopcroft, J., Tarjan, R., Efficient Planarity Testing, Journal of ACM., Vol. 21, No. 4, 1974, pp 549-568.

[13] Lokanathan, B, Kinnen, E., Planarization of a VLSI interconnectivity graph for floorplanning using rectangular dualization, Proc 7th Microelectronic Symp., IEEE, New York, 9-11 June 1987, pp 115-119.

[14] Aleniunas, Y., Rosenberg, A. L., On Embedding Rectangular Grids in Square Grids, IEEE Trans. Comp., Vol. 31, No. 9, Sep. 1982, pp 907-913.

[15] Fowler, R. et al., Optimal packing and covering in the plane are NP-complete, Inf. Proc. Lett., Vol. 12, No. 3, 1981, pp 133-137.

[16] Kirkpatrick, S., Gelatt, C. D., Vecchi, M. P., Optimization by simulated annealing, Science, Vol 220, No. 4598, pp 671-680.

[17] Ying, C., Wong, S., An Analytical Approach to Floorplanning for Hierarchical Building Block Layout, IEEE Trans. CAD, Vol 8, No 4, April 1989, pp 403-412.

[18] Shin, H. et al., Two-dimensional compaction by Zone-refining, Proc. 23rd Design Automation Conf., IEEE, 1986, pp 115-119.

[19] Rosenberg, H. M., The Solid State, Clarendon Press, Oxford, 1978, pp 11-12.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

4LP - LOW LEVEL LANGUAGE FOR LINE PROCESSOR - SYMPATI2 -

Pascal FERNANDEZ**, Pascal ADAM *, Didier JUVIN*, Jean-luc BASILLE**

*D.LETI/DEIN/SIR CEA SACLAY 91191 Gif/yvette France
** IRIT 118 route de Narbonne 31062 Toulouse France

We present the language we conceived for our line Processor. This structure is a good choice for Image Processing and more particulary for the iconic domain. It provides good performances and is cost effective. By means this structure, the language 4LP elimates those difficulties encountered when programming non-conventional structures.

INTRODUCTION

Image Processing gives rise to much research on machine architectures. Among the hundred or so specialized machines on the market or being studied three types may be distinguished: Pipe-Line like PIXAR , Processor Arrays like DAP or CLIP4 and Line Processors like AIS5000 or SYMPATI-2 . We are going to show the advantages of this last linear structure for pixel-level processing, particulary the easy way it provides for parallezing pixel-level algorithms, and the development facilities provided by the language available for this structure. This language makes it possible to program a line processor in a way that eliminates many of the problems usually found when using the non-conventionnal structures.

First, we shall briefly recall the line processor concept and give some details about the structure our two laboratories are developing in collaboration. Then we shall present the programming environment, then the language and illustrate these facilities with some examples. Finally, we shall give the execution times and the quality factor obtained when processing the Abingdon Cross benchmark.

1. THE LINE PROCESSOR STRUCTURE

Parallel structure may be considered as Von Neumann variations when duplicating some part of the so-called Von Neumann structure. Among the different structures that might be involved in Image Processing systems, the Line Processor concept provides a good solution, more particulary for pixel-level processing when considering the iconic domain, the domain concerned with all operations in the images in the true sense, that is extracting features from the images [1].

A Line Processor is a kind of Processor Array, a broader generic term that could include pyramids as well as the more traditional bidimensional Processor Arrays. In accordance with the two fundamental criteria, efficiency and cost, we have good reasons to believe that the Line Processor concept is a satisfying middle-road, in particular if we wish to design an Image Processing machine at a reasonable price, compatible with a work station configuration [2].

The system we have achieved present some characteristics providing more possibilities than a simply linear structure would offer [3]. The SYMPATI2 system is a mono-dimensional Processor Arrays with 2D addressing capabilities.

The principal features of this linear structure are the SIMD mode, the data organization scheme, the processor/processor interconnections. We have two data organization schemes. First, the helicoïdal scheme makes it possible to access without any conflict or border effect, either on a row or a column, to any M pixel segment, where M is the number of Processing Elements involved. Another data organization scheme, the tabulated addressing mode is also possible.

Processing Elements are linearly interconnected but each Processing Element is also connected to the memory banks of its adjoining Processing Elements. These interconnections make it possible to access to any point of the 3x3 window in only one cycle. Furthermore it speeds up access for larger neighbourhoods (Figure 1). The Processing Elements structure consists of two main parts : the processing part concerned with the operations performed on the pixel values, and the addressing part, allowing each PE to calculate the required pixel address in its own memory bank.

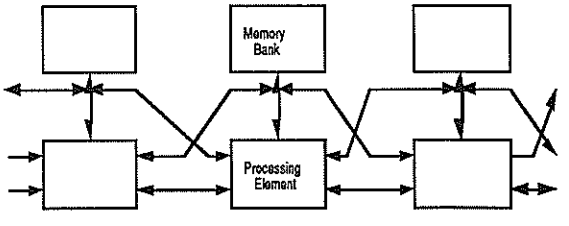


FIGURE 1 : Processor/processor interconnections

The processing part is rather classical with a 16 bit ALU for usual operations (logical, arithmetical and 8x8 bit multiplier) and a scratch pad of registers. But it also comprises specific components, in particular an interconnection network allowing the connection between the ALU input and the required output, and 4 work flags necessary for performing conditionnal sequences.

The addressing part includes two modules : The addressing calculation module makes it possible to address the memory either in helicoïdal mode or in tabulation mode. In helicoïdal mode it evaluates the considered pixel coordinates (i_r,j_r) from the current segment coordinates (i,j) sent by the command unit and then performs the helicoïdal transform :

$$@ = i_r.d+j/M \quad \text{where } d=N/M \text{ for an}$$

NxN image and M PEs.
This evaluation takes into account the horizontal scanning and vertical scanning.

2. DESCRIPTION OF THE PROGRAMMING ENVIRONMENT

One major difficulty for the user with a non-conventionnal structure is the way to program it. We present now the system features of SYMPATI2.
Four scanning modes are available as shown of figure 2. In any mode it can be noticed that the pixels of the current segment are processed in parallel but the different segments are processed in a sequential way. Thus it makes it possible to put the successive processed segments either in another image, and then emulate a bidimensionnal Processor Array, or in the same image, and then work in a recursive way as some neighbours may be the result of processing a previous segment.

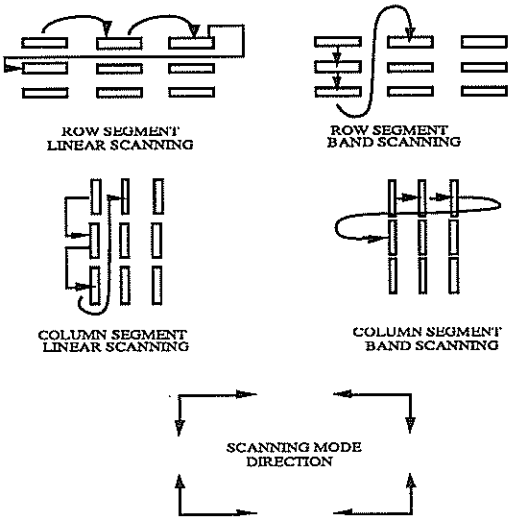
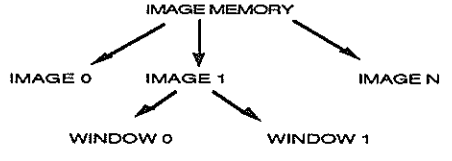


FIGURE 2 : DIFFERENT SCANNING MODES

We have an eight bit 1024x1024 image memory which can be split into sub-images according to a quadtree organization. Thus it is possible to access either a 1024x1024 image or 4 512x512 images and so on to 1024 32x32 images or any combination.



2.1. Works Image

The user may reference up to 16 different images in his program. These referenced images belong to one out of two types characterized by : the size (from 32 to 1024), the scanning mode, the incrementation step. Thus it is possible to process in the same program two images with different sizes and different scanning modes. It should be noticed that the successive iterations of the sequence to be processed, are managed indepentantly of the incrementation steps, as it will be explained when presenting the LEXT and LINT directives.

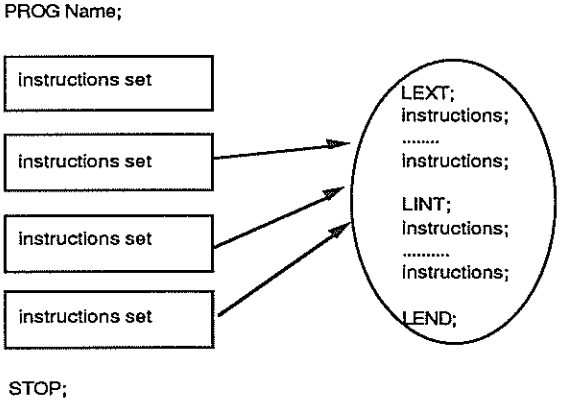
2.2. Processed Windows

The masking module compares the coordinates (i_r,j_r) with the window selected by the user and inhibits, if necessary, the calculation. If a selected point is outside the window, the considered processor will not store any result in memory or scratch pad register and will not set any work flag.
A window belongs to one of the two types characterized by their coordinates, top-left and bottom-right. Those coordinates are relative to an image type. A window is referenced independantly of the processed image. Thus it is allowed to process the same image with the two possible window types.

3. DESCRIPTION OF 4LP

3.1. The language features

Let us present now the language that we have conceived in order to eliminate those difficulties encountered when programming non- conventionnal structures. 4LP is a Low Level Language developed for the command unit with its own instruction set, some instructions are directly executable by the PE, others are executable by the command unit itself for the parallelism management. Due the SIMD structure 4LP is also a highly parallel language. The basic program structure of 4LP is the following one:

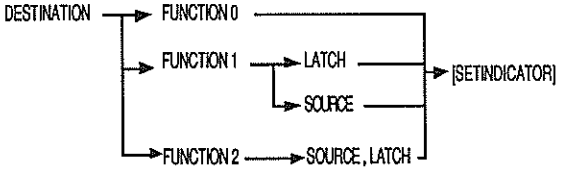


We wanted 4LP to be very easy to use and as independant as possible of the machine structure. So the user will be able to make full use of the machine without any knowledge of its structure, just by calling predefined functions. But the user also has the possibility to define

his own functions. In such a case the sequence of instructions is written for a virtual pixel whose neighbours are defined relatively to it with the distances Δi and Δj. This sequence is processed for all the pixels in accordance with one of the four scanning modes previously mentionned.

LEXT, LINT are the command unit directives for the external and internal scanning loop control. This double loop structure makes it possible to emulate a bidimensional Processor Array. This sequence of instructions is repeatedly processed according to two iterations numbers. These two numbers generally depend on the size of the processed image and on the incrementation step. Typically, in a row segment linear scanning for instance, the incrementation number M of Processing Elements and so the iteration number equals the size N of the processed image divide by M. In the same way the incrementation number for the external loop is one and the iteration number is N.

The elementary instruction has the following form :



Example :

S0.SP , L = AND M1.0 (-3,3) , L I2 = C;

Register S0.SP and the ALU Latch (L) receive both the result of the AND function between the Latch value and the neighbor (-3,3) of the current pixel in the window number 1 from the image number 0. The indicator I2 is set according to carry ALU Flag. This instruction is executed by all of the PE corresponding to M consecutive pixels.

3.1.1. Destinations - Sources

The principal destinations are : one PE's own scratch pad register, the PE's own memory, one PE's special register (addressing part for instance), the ALU latch. The principal sources are : one scratch pad register belonging to either the PE, or one of its neighboors at distance 1,2,3 right or left, the memory of PE at distance 0,1,2 , a common constant value given by its address in the command unit memory.

3.1.2. ALU Functions

We have three types of ALU functions. The function type 0 is the function whitout operand, the function type 1 is the function with one operand (source or latch), the function type 2 is the function with two operands (source and latch). We have an available set of 64 functions. These functions set up some ALU flags(a total of seven).

3.1.3. Memory description

Each PE may only write in its own memory. Depending on one of the two possible memory organizations, the syntax is different.

In the *helicoïdal* mode the syntax is the following one :

M x.y(Δi,Δj)

where x is the window type, y the number of image (0-15), Δi and Δj represent the relative deplacement(with Δi,Δj comprised in [-31,31])of the current segment from the virtual segment.

Due to the architecture, a direct memory transfer is possible between two PEs if the distance is 1 or 2. The compiler automatically generates the necessary micro-instructions to propagate a memory value in the destination PE if the distance is greather than 2.

In the *tabulated* mode , each PE has a special register named INDEX wich contains the address of the memory value in tabulated memory organization.

3.1.4. Branch instruction

We have also a branch instruction (GOTO) that can be either unconditionnal or conditionnal. In this case, we have two possibilities: either a branch when *all* corresponding flags are set up, or a branch when *any* corresponding flag is set up.

3.1.5. Conditionnal instruction

Finally, an elementary instruction can be conditionnal to a set up flag.

Example : S0.SP = ADD S0.SP ,L /I2

This instruction will be executed only by the PEs whose indicator I2 is set up, the other will execute a no operation.

3.2. Some examples

In order to illustrate the language use let us consisder a pixel-level algorithm : The dilation or the ORing of the 3X3 window pixels. Such an algorithm does not present any difficulty to be processed on a SIMD structure. One only needs to express it for a pixel. The whole image is scanned automatically by the command unit according to the parameters provided by the user. It leads to the following program :

```
PROG DILATION;
LEXT;      /* iteration on the line number
  LINT;    /* iteration on the segment number
  L = M0.0(0,0);
  L = OR M0.0(-1,-1),L;
  L = OR M0.0(-1,0),L;
  L = OR M0.0(0,-1),L;
  L = OR M0.0(0,1),L;
  L = OR M0.0(1,-1),L;
  L = OR M0.0(1,0),L;
  L = OR M0.0(1,1),L;
  M0.1(0,0) = L ; /* result is stored in image 1
                  /* type 0.
  LEND;
STOP;
```

In this program, the source image is scanned only once. But, as this algorithm may be expressed in a separable way, an intermediate image may be used in order to reduce the processing time. Then it comes :

```
PROG DILATION;
LEXT;
  LINT;
  L = M0.0(0,0);
  L = OR M0.0(0,-1),L;  (1)
  L = OR M0.0(0,1),L;
  M0.2(0,0) = L;
  LEND;

LEXT;
  LINT;
  L = M0.2(0,0);
  L = OR M0.2(-1,0),L;  (2)
  L = OR M0.2(1,0),L;
  M0.1(0,0) = L;
  LEND;
STOP;
```

(1) The three pixels on the row one OR'ed and the result stored in image 2.
(2) The three pixels on the column, from image 2 one OR'ed and the final result stored in image 1.

4. CONCLUSION

We already have the complete development environment including the compiler, desassembler, debugger and the simulator giving the exact computing time that will be required on the machine.Table 1 shows some of those computing times for classical image processing algorithms.The prototype of the machine is currently providing the first results in accordance with the simulator results previously obtained. We tried to show that 4LP is relevant fot the pixel level processing. it provides an easy way for programming any algorithm operating on a window up to 63X63. The results obtained up to now provide a very favorable quality factor as it can be shown with the Abingdon Cross benchmark [4]. Futhermore a $50K estimated commercial price woul lead to an excellent efficiency/cost ratio.

DILATION 3X3	2,58
DILATION 5X5	3,81
DILATION 7X7	5,89
LOCAL MAXIMUM	
FILTER	2,6
TUKEY	17,72
PREWITT	5,2
SOBEL	5,8
PSEUDO-TUKEY	6,5
DISTANCE TRANSFORM	9,54
ISEF	9,04
THRESHOLDING	1,2

TABLE 1: IMAGE 256X256 : 32 PE .Time in ms.

5. REFERENCES

[1]- BASILLE J-L, CASTAN S -Iconic and symbolic use of a line processor in Multilevel structures. In Duff MJB ed. Intermediate level image processing. London : Academic Press 1986.
[2]- DUFF M.J.B and LEVIALDI S. -Languages and Architectures for Image Processing, ed. by authors, Academic Press, 1981.
[3]- JUVIN D.,BASILLE J-L, ESSAFI H., LATIL J-Y -SYMPATI2, a 1.5D Processor Array for image applications, EUSIPCO 88, Grenoble 5-8 sept.1988
[4]- PRESTON K. Jr - The Abingdon Cross Benchmark Survey- computer july 1989 pp9-18.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

ECHO CANCELLERS FOR TELEPHONE APPLICATIONS BASED ON PROGRAMMABLE DIGITAL SIGNAL PROCESSORS

P. REUSENS, P. REYNDERS, P. GUEBELS

ALCATEL Bell Telephone
VLSI Design Department
F. Wellesplein 1
B-2018 Antwerp, Belgium

A 64 ms echo canceller for telephony applications is described. It is realised on a commercial Digital Signal Processor (DSP), which is shown to be a viable, cost-effective and flexible approach.

The firmware design is reusable as a module, and it can use the full arithmetic capacity of various DSP machines in many applications.

This makes the DSP programmed echo canceller an ideal intermediate step towards an integrated realisation which uses a signal processor as a cell inside a Very Large Scale Integrated (VLSI) circuit. The result is a flexible and low-risk monolithic realisation.

1 INTRODUCTION

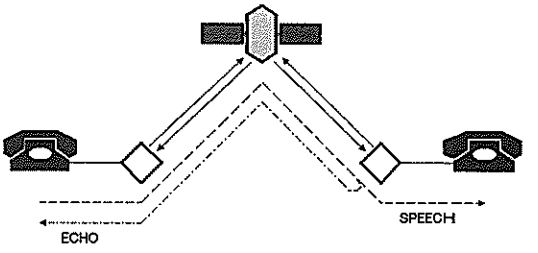
1.1 Echoes In Telephone Networks

Imperfections on analog telephone connections cause echo signals, which can be particularly annoying on lines with long roundtrip delay [1]. Therefore, echo control must be present, even if a part of the transmission goes via a digital long distance network. The echoes are generated by mismatches of line impedances and by hybrids converting from 2 to 4 wires.

In older analog networks, echoes were controlled by inserting attenuation. However, echo eliminating devices are indispensable on intercontinental or satellite connections, or in the new generation digital mobile cellular systems, where artificial delays are caused by coding and error protection [1, 2]; see figure 1. The technically outdated echo suppressors, which merely reduce a connection to a half duplex one, are replaced by Digital Echo Cancelling (DEC) devices, which give a very good echo control, especially in the cases of double talk [3]. Based on adaptive filters, they subtract the estimated echo from the returning signal.

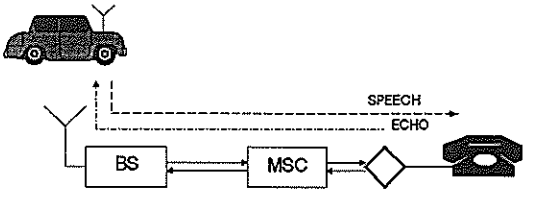
1.2 Acoustic Echo Cancellers

Echo control via cancelling is also needed in high quality handsfree telephone equipment, indispensable in e.g. mobile radio applications and conference calls. Here the echo is caused by the acoustic coupling between the loudspeaker and the microphone. Without echo control the quality is bad and instability is present. The often used echo suppressors give a poor quality compared to adaptive cancellers.



ECHOES IN LONG DISTANCE CONNECTIONS
A talker perceives an echo after 600 ms

Figure 1.a



ECHOES IN THE DIGITAL MOBILE NETWORK
A talker perceives an echo delay of 150 ms

Figure 1.b

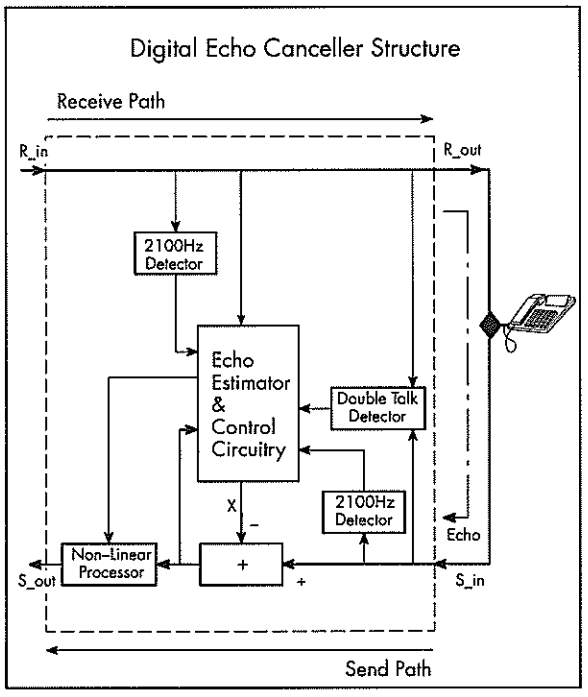
2 IMPLEMENTATION

Standard programmable DSP processors are very well suited to implement linear echo cancelling combined with non-linear clipping of residual echo. They can easily include control and maintenance functions, such as disabling tone detection. This allows the adaptability of the design through adjustable parameters, and enables the flexible reuse of the algorithms as firmware modules. Moreover, it reduces the hardware to a single device with a firmware based intelligence, which is superior to fixed adaptive filters, which only partially implement the necessary algorithms and must be assisted by other DSP machines.

Monolithic LSI realisation:

Finally, for a compact and low-cost realisation the integration of an echo canceller on a single VLSI circuit is needed. Here also the DSP programmed echo canceller is the ideal approach. Indeed, today a VLSI Integration can be based on DSP machines, which are used as a core or cell.

The DSP core based design results in a monolithic realisation with properties such as low power, high throughput, improved LSI design performance, lower risks, flexibility, and a better cost/performance ratio.



Signal Flow And Diagram Of An Echo Canceller

Figure 2

3 LINEAR ECHO CANCELLING WITH LMS ALGORITHM

The adapted block-update algorithm, described hereafter, is derived from the LMS (least mean square) algorithm [4, 5, 6].

A summary of the signals and dataflow is given in figure 2. The echo canceller will generate a replica X of the echo signal. Subtracting this replica from the corrupted near end signal at port S in, will result in a "clean" near end signal at port S out.

The echo is estimated by a transversal digital filter (FIR):

$$X(k) = A(0) * R(k) + A(1) * R(k-1) + \ldots + A(N-1) * R(k-N+1) \qquad (1)$$

with X: estimated echo sample,
A(i): echo canceller coefficients,
N: number of taps of the FIR filter,
R: reference signal, R in = R out.

The Widrow-Hoff LMS algorithm iteratively adjusts the echo canceller coefficients as follows:

$$A(i,k+1) = A(i,k) + BETA * E(k) * R(k-i) \qquad (2)$$

with: A(i,k+1): new value for coefficient A(i)
A(i,k): previous value of coef. A(i)
BETA: update loop gain
E(k): "error" signal or residu;
E(k) = S in(k) - X(k)

4 IMPLEMENTATION ON DSP: BLOCK UPDATES

Calculating the FIR filter, formula (1), on a standard DSP machine is straightforward. Indeed, most standard DSP processors have a pipelined architecture, optimized to calculate a sum of products in a minimal time.

However, the direct implementation of the LMS update, formula (2), does not exploit the internal arithmetic pipeline of the Arithmetic and Logic Unit. For each coefficient a single product must be calculated and time is lost to refill the pipeline.

In the block update algorithm (BLMS) as proposed by Clark, Mitra and Parker, the refilling losses of the pipeline are reduced [7]. We have changed the algorithm in such a way that it is possible to use only one TMS320C25 processor and still cancel echoes with a delay of 64 ms while satisfying CCITT G.165; see paragraph 5.2.

5 IMPLEMENTATION ON A TMS320C25 DSP

The TMS320C25 processor is a relatively inexpensive component, which is considered an industrial standard. Although it does not have the Harvard architecture of more powerful DSPs, it can do a multiply accumulate (MAC) operation in one pipelined machine cycle.

5.1 FIR Calculation

For a 64 ms echo at a sampling rate of 8 kHz, the FIR filter length is 512. One FIR result is calculated per 125 us. The necessary 4.1 million multiply-accumulate instructions per second require only 41% of the available 10 MIPs, with little overhead. The reference signal is stored in the internal data memory of the DSP, the adaptive coefficients are stored in the external program RAM.

5.2 Coefficient Update: Block Algorithm

As explained in paragraph 4, the coefficient update formula (2) can not be executed efficiently. To take maximal advantage of the DSP MAC instruction the coefficients are updated at a distance N in time:

$$\begin{aligned} A(i,k+N) = A(i,k) &+ BETA * E(k) * R(k-i) \\ &+ BETA * E(k+1) * R(k-i+1) \\ &+ \ldots \\ &+ BETA * E(k+N-1) * R(k-i+N-1) \end{aligned} \qquad (3)$$

Arranged in this way the formula can take advantage of the MAC instruction of the TMS320C25. The correlation between error and reference signal again requires little overhead, provided that N is sufficiently large.

5.3 Coefficient Update: Algorithm Stability

The stability of the update algorithm, and the related convergence speed are important issues. The actual gain multiplication is replaced by a set of fixed shifts, i.e. a multiplication with power of 2. The gain factor beta is given by:

$$\log(BETA) = [-2 * \log(LR/Pref) - 10.15] \qquad (4)$$

with LR: power of reference signal R.
Pref: reference power of a sine wave with maximal amplitude (3.14 dBm).

5.4 Data And Coefficient Precision

Another implementation problem is an efficient, but sufficiently accurate representation of the data and coefficients, to reduce noise and limit cycles. For the data representation 16 bits suffice.

The canceller coefficients must be represented by 24 bits: 16 bits are sufficient for an echo estimating FIR filter, efficiently exploiting the TMS320C25 processor as a 16-bits processor with a 32-bits accumulator. However, additional bits are needed for the updating. Only then the coefficients completely converge to the optimal values.

5.5 Power Estimation

Finally a simple and efficient power estimation of the input signals needs close attention. To avoid tedious double precision calculations, the mean value of the "rectified" samples is calculated instead of the mean value of the "squared" samples. The difference (in dB) between the "real" power estimation (i.e. mean square) and the mean absolute value lies between 1.7 dB and 3.2 dB for speech, and depends on the crest factor of the signals.

Stability and fast reaction times require a detailed study of the rise and fall time of the power estimating routines. It was found that asymetric rise and fall times should be avoided if accurate power estimation is necessary. However, different attack and decay times are important for stability and to limit divergence in case of double talk.

5.6 Code Implementation

The digital echo canceller on TMS320C25 was written completely in Assembler including A-law or u-law conversion resulting in only 1.3k words of code and 0.7k words of external data RAM, and 0.4k of data tables.

6 ADVANTAGES OF A PROGRAMMABLE DSP PROCESSOR

A programmable DSP processor allows to use the same HW for features like e.g. disabling tone detection, control of phase roll effects, control of update gain in case of double talk.

Moreover, a programmable DSP processor allows to include other applications like e.g. the reception of multifrequency signals on the same hardware.

The most important advantage is the possibility to integrate the echo cancelling function by using a DSP CORE or CELL as explained in the next paragraph.

7 ECHO CANCELLING WITH DSP CORES IN A VLSIC

Using the DSP CORE TECHNOLOGY a DSP based function can be realized in single components. The basic idea is to combine all the functional blocks of a DSP board into one chip. Each DSP CORE is surrounded by ROM, RAM, PLA logic, glue logic, input/output, transmission cells, and even Analog to Digital and Digital to Analog converters, if necessary.

It combines the complexity of advanced printed circuits with DSP processors and microcontrollers into one silicon chip, which allows proven design methods to be reused.

The DSP core technology is the major step towards the integration of complete systems on a single integrated circuit, while keeping a hierarchical modularity of blocks. Moreover, a component based on CORE technology obtains flexibility through programs in firmware, because a programmable DSP machine forms its heart.

The DSP core technology allows an echo canceller with:
- Recuperation of design efforts with a better LSI design productivity and turn-around time;
- An improvement of the component speed coupled with larger throughput or power reduction;
- Design flexibity and adaptability through changeable firmware in ROM or even RAM;
- Selftest integration;
- Size reduction and reliability increase;
- Test and production simplification.

8 SIMULATION AND TEST RESULTS

The BLMS algorithm gives no significant reduction in convergence time to reach an echo return loss of - 30 dB.

The implemented BLMS at 64 ms echo length is not updating the coefficients at the maximal theoretical rate. A duty cycle of 25% is used to update the coefficients. The convergence speed is sufficient to reach the requirements.

Phase-roll could be handled by the algorithm, provided that the non-linear processor (NLP) is active. The NLP removes the echoes after a silent period.

In the presence of near-end signals (double talk) the updating of the echo coefficients is stopped, and no significant divergence is seen. When background noise is added to the echo signal, the FIR convergence is reduced, but the NLP removes the residual echo.

A background noise is injected when the NLP is active. The subjective tests were positive.

The algorithm was used for acoustic echo cancelling. Then the update duty cycle was improved to reach the necessary higher adaptation speed.

9 CONCLUSION

Echo cancelling is an indispensable element in telephone communications, which can be realised efficiently on a commercial Digital Signal Processor.

It was shown that the use of a progammable DSP is a viable, cost-effective and flexible approach, which can not be offered by other implementations.

The design is reusable as a firmware module. It can be used on many DSPs, in all kinds of applications, especially for a compact and low-cost realisation. Indeed, when the integration on a single VLSI circuit is needed, a canceller based on a programmable DSP core is the ideal approach. The result is a flexible and low-risk monolithic realisation.

ACKNOWLEDGEMENTS

The authors wish to thank Daniel HOEFKENS, Eric VOS, Eddy BOEYKENS, Wim BOSIERS, Roland JELLY, Rudi VANKEIRSBILCK and others for the system study, the hardware realisation, the test setup, and characterisation of the digital echo canceller for the ALCATEL system 1240 exchange.

REFERENCES

[1] R.H. Moffett, "ECHO AND DELAY PROBLEMS IN SOME DIGITAL COMMUNICATION SYSTEMS", IEEE Communic. Magazine, Vol.25, No.8, Aug 1987.

[2] G.K. Helder, "CUSTOMER EVALUATION OF TELEPHONE CIRCUITS WITH DELAY", Bell System Technical Journal, September 1966.

[3] M.M. Sondhi & D.A. Berkley, "SILENCING ECHOES ON THE TELEPHONE NETWORK", Proc. of the IEEE, Vol.68, No.8, August 1980.

[4] C.W.K. Gritton & D.W. Lin, "ECHO CANCELLATION ALGORITHMS", IEEE ASSP Magazine, April 1984.

[5] D.G. Messerschmitt, "ECHO CANCELLATION IN SPEECH AND DATA TRANSMISSION", IEEE Journal on Selected Areas in Communications, Vol.SAC-2, No.2, March 1984.

[6] B. Widrow & S.D. Stearns, "ADAPTIVE SIGNAL PROCESSING", Prentice Hall Inc.

[7] G.A. Clark, S.K. Mitra, S.R. Parker, "BLOCK IMPLEMENTATION OF ADAPTIVE DIGITAL FILTERS" IEEE Transactions on Circuits and Systems, Vol.CAS-28, No.6, June 1981.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

IMAGE PROCESSOR FOR REAL TIME CONTOUR RECOVERY

Fèlix Ferrer and Josep Amat

FACULTAT D' INFORMATICA DE BARCELONA (U.P.C.)

c/ Pau Gargallo 5, 08028 Barcelona, Spain

1. INTRODUCTION.

Contours are one of the features of an image that are mainly used in computer vision systems. Analysis of contours is a widely used technique in pattern recognition, object classification and object inspection in general. The quality of the contour obtained from the scene which is analyzed affects the global performance of the system. Good contours are those that are one pixel wide, with their points as close as possible to the centre of the real edge, and without discontinuities. The connection of the points of the edge where an interruption has been detected is a step needed to provide the required data to the following levels of the system. Delays caused by this recovery can be an important penalization to the whole system processing time. In systems involved in industrial applications and robotics, the time required to achieve the output data has as much importance as its quality. This work tries to satisfy both requirements: obtaining closed contours in real time.

Different approaches [1][2][3][4][5][6][7] have been developed to locate the boundaries of objects and get thinned and closed contours within two-dimensional images:

- by applying a gradient edge detector in sequential steps, by relaxing the threshold level in the regions where there are discontinuities.

- by increasing selectively the spatial resolution and enhancing the grey level distribution in the regions where there are discontinuities.

- by matching edge fragments to lines and curves by using the Hough transform.

- by using several operators that maximize the signal to noise ratio.

- by recovering edges by linear or polynomical interpolation between the closest or selected end points.

If time is a constraint, the edge detectors usually used are the ones based in the local features of the image. Such edge detectors allow

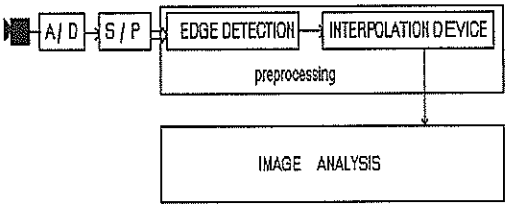


Figure 1.

parallel processing of the image, causing less delay than those based on global methods of image analysis.

Edges may be defined as local changes or discontinuities in image luminance. So, edges of homogenous objects in grey-level pictures are detected by looking for the boundaries between two regions of different gray level values. The decision of whether or not a pixel is on an edge does not depend on what other picture points lie on it. So, the edge operator may be applied simultaneously in the entire picture. But, for the same reason, such an operator is extremely sensitive to local noise, illumination and object characteristics. Contours obtained by these operators are easily

formed by thick, noisy and discontinuous edges.

In order to get closed contours, a local edge detector should not be an isolated or autonomous element in the system. Its outputs should be influenced by the results of other elements of the system: the edge detection process should be guided. But this guiding (ie: considering the characteristics of the scene, model of the object,...) takes so time that avoids a real time image processing.

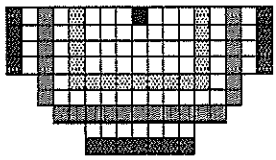


Figure 2.a.

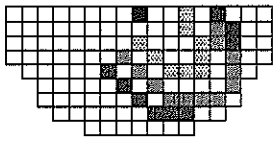


Figure 2.b

Our work focuses on a processor to rebuild discontinuous edges. This processor has been designed to operate in real time, allowing a global processing time of twenty milliseconds.

2. CONNECTIVITY CRITERION

The complete system is organized in two main sections (Figure 1). The interpolating device is included in the preprocessing level, following the acquisition unit and the edge detection and thinning step.

After detecting the discontinuity on the edge, the reconstruction process begins by selecting the two end points that are more likely to belong to the same edge fragment. This decision is not only taken by considering the minimum distance, but also looking for the points where connection will produce a minimum change in the direction of the contour. The approach that has been chosen to reduce the complexity of the system takes into account the distance between two end points, weighing it with just the direction of the edge on one of them.

With this procedure, the isotropic diagram of distances (Figure 2.a) is modified to a map of weighed distances that penalize the points not aligned with the direction of the edge on the selected end. As is shown in Figure 2.b, the effect is that at the same euclidian distance, the points that are considered closer to the selected end are those that are on the direction of the edge. The extreme point that is at the minimum weighed distance is chosen as the second end for the interpolation process. The orientation of the contour is not evaluated in both extremes, but this condition allows to implement an operator that can be inserted on the system without punishing its timing performance.

3.- STRUCTURE OF THE INTERPOLATING DEVICE

The device is structured into a two stage pipeline (Figure 3), corresponding to the main tasks to be carried out. In the first, the input serial data is mapped onto a matrix of 3 by 3 pixels, to detect discontinuities on the contour obtained from the edge detection unit.

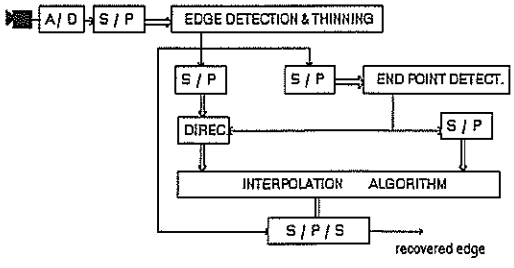


Figure 3.

Simultaneously, the direction of the contour is evaluated by means of a 7 by 4 pixels operator. This direction is only taken into account on the points where edge continuity has been missed.

In the second stage, it is selected the other end point that will be used in the connection process. Because of the characteristics of the image data, beginning at one end (OSP), the search of the other most suitable point may be restricted to the pixels inside a simetrical boundary of 127 pixels. For all the end points in the area, the distance to the OSP pixel is weighed

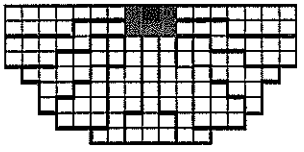


Figure 4.

with their relative position in the region plus the direction of the edge on the OSP. The effect of applying this function is that the search is done unisotropically, guided by the direction of the contour on the OSP.

Finally, the recovery process is completed by synchronously filling the pixels that connect the two selected extremes. At the end, the serial edge data is obtained with just the delay of twice the lines needed by the operator.

To implement this function, the contour analysis region is partitioned into eleven slices, corresponding to the main sectors that are going to be considered (Figure 4).

The device is structured in four logical levels. In the fisrt one, a ROM memory

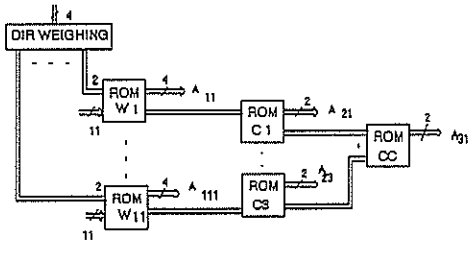


Figure 5.

per sector is driven by its associated pixels and two bits that are the result of weighing the direction of the edge with the position of the sector. The ROM is coded so that its outputs are the minimum weighed distance D_w from the origin to the end points in the sector, and the relative location A_{1i} of the chosen pixel.

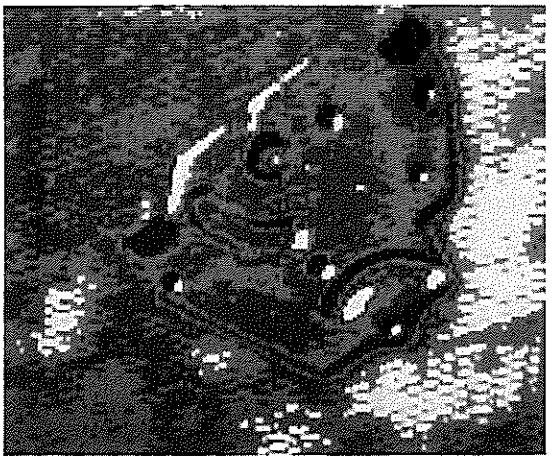


Figure 6.

The selection of the smallest distance provided by this step is obtained by means of the ROM memories distributed in the two following levels.

In a forth level, the address formed by A_{1i} A_{2i} A_{3i} drives to the selected end pixel (Figure 5).

A last conversion serial to parallel and

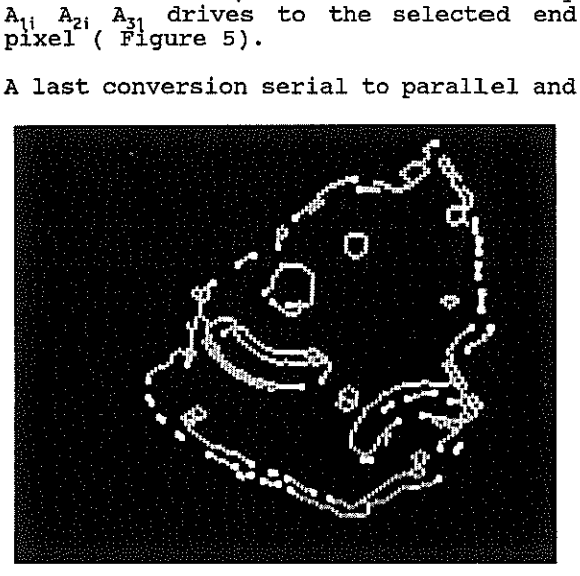


Figure 7.

serial allows to fill in parallel to the shift-registers matrix the bits needed to rebuild the processed contour fragment.

4. CONCLUSIONS.

The use of the direction of the edge on the point where a discontinuity is detected allows to select the corresponding end pixel that has more probability to fit in the connection process it has been discussed. Following

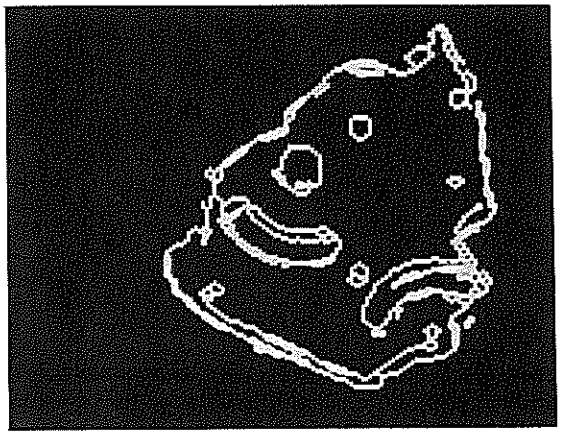


Figure 8.

figures show some results obtained by this method. In Figure 6, there is the grey level picture, where the characteristics of the object and illumination difficults the contour definition. In Figure 7, edges have been detected with some discontinuities, and the extremes or critical points of the contour have been marked. In Figure 8, the connection is done on the points that are at a distance lower than the range of the operator. As it has been discussed, these pictures illustrate that the range and the method are an attempt to satisfy the requirements of reducing the processing time and improving the quality of the final contour.

REFERENCES

[1] Canny, J., A Computational Approach to Edge Detection. IEEE Transactions on Pattern Recognition and Machine Inteligence. Nov 1986.

[2] Davis,L.S., A Survey of Edge Detection Techniques. Computer Graphics and Image Processing. Academic Press 1975.

[3] Gu,W.K., Huang,T.S., Connected Line Drawing Extraction from a Perspective View of a Polyhedron. IEEE Transactions on Pattern Analysis and Machine Intelligence. July 1985.

[4] Duda,R.O.,Hard P.E., Use of the Hough Transform to Detect Lines and Curves in Pictures. Comm. ACM. Computers Jan 1972.

[5] Pingle,K.K., Tenenbaum. An Accommodating Edge Follower. Proc 2nd IJCAI. Sept 1971

[6] Shou-Pyng Shu, J., One Pixel Wide Edge Detection. Pattern Recognition. Vol 22 1989.

[7] Lee, C.J., Catthoor, F., De Man, H., An efficient ASIC Architecture for Real Time Edge Detection. IMEC , Haverlee, Belgium. October 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

PARALLEL PROCESSING IN I/O MANAGEMENT

Technical Editor

Pascal PERNIN HORIZON TECHNOLOGIES 22, avenue de la Baltique Z.A de Courtaboeuf 91953 Les Ulis Cedex FRANCE

And Richard KROLL from APTEC GmbH.

1/ ABSTRACT

Data transfer remains a serious problem in most real-time systems based on traditional solutions, where I/O intensive applications can bring a powerful machine to its knees.

To address these problems, Aptec Computer Systems has developed an I/O processing system, which incorporates a highly parallel architecture in order to provide a large I/O bandwidth (up to 200 Megabytes per second) in a very partitionable and distributable fashion.

The engineers at Aptec, in their search fo new I/O architectures, decided to concentrate their efforts on VAX configurations where the pure computational capacity was adequate, but the I/O capability was not. The results of their development, the Aptec I/O Computer (IOC), is essentially a complete I/O system which relies heavily on a highly parallel architecture in order to provide a large I/O bandwidth (up to 200 Megabytes per second) in a very partitionable and distributable fashion.

The architecture of the Aptec I/O Computer has already demonstrated performance increases up to a factor of 40 in I/O intensive applications found in areas such as satellite or telemetry data processing, image and signal processing (seismic, radar, sonar vibration analysis), simulation, modelling, structural analysis, and general real time processing.

2/ THE APTEC I/O COMPUTE HARDWARE

The Aptec I/O Computer (IOC System 200) applies the principles of parallel processing (see figure 1) to the I/O problem by providing dedicated interface processors, called Programmable I/O Processors (IOPs) for each peripheral, a shared high-speed memory of up to 1 Gigabytes, and a high-speed (100-200 Megabytes/s) Data Interchange Bus (DIB). The Data-Interchange-Bus allows peripheral IOPs to perform high speed data transfers between each other and to and from locations in IOC memory. One of the IOPs in each configuration has the host as its "peripheral" in order to allow the host to access the IOC memory.

The IOP modules are currently available in four forms :
- one for connecting Unibus compatible processors and peripherals with transfer rates of up to 3 Megabytes/s
- one for connection of high speed devices with data rates of up to 12 Megabytes/s via an "OPENbus" that can be adapted to a wide variety of interfaces.
- one for connection of very high speed devices such as rotary head recorders with data rates of up to 50 Megabytes/s.
- one for connection of VMEbus devices to the Data-Interchange-Bus (DIB) with data rates of up 11 Megabytes/s..

The functions of the IOPs, regardless of type, within the I/O Computer are essentially the same. They are dedicated to the control of their respective peripherals and perform data transfers between their devices and the IOC memory or other IOC devices. In addition to I/O control, the on-board processors are able to perform some on-the-fly processing of the data they are transferring. One example of such processing might be the demultiplexing of data from PCM telemetry front-ends or multiple channel analog tapes. Another example of the use of the IOC on-board processing power is to read data from successive disk sectors and store it in IOC memory in the row and column arrangement most useful to an array processor. In addition to capabilities available through microprogramming, the IOPs have hardware to assist for such common functions as byte swapping on data passing through the module.
As hinted above, the IOPs are capable of communicating with each other, as well as with their attached peripherals and IOC memory . Each IOP has two independent addresses on the Data-Interchange-Bus, corresponding to the input FIFOs on the IOP board. These FIFOs accept command and data information from other IOPs even if their IOP is simultaneously processing a high speed data transfer from its peripheral to IOC memory. This mechanism is useful in a pipelining environment since it allows an IOP to stack requests from its neighbours. Specifically, in a situation where we are pipelining data blocks down a chain of IOPs, each IOP need only read its command FIFO to obtain the address of the next data block to work on, and, after completion of its task, pass the data address to the new IOP down the pipe.
Examination of the I/O Computer architecture diagram (figure 1) will reveal that each IOP is also connected to the host's IOP.
This transparent mode of operation not only allows the host to use the device if a dedicated I/O task is not in progress, it also simplifies debugging, and has the very practical use that all device diagnostics can be run without modification. All processors in the system send data across a Data-Interchange-Bus(DIB) built around 32-bit-wide address, write, and read paths that accommodates simultaneous bi-directional transmission of 8,16, 32,64 or 128 bit words to any combination of processors.

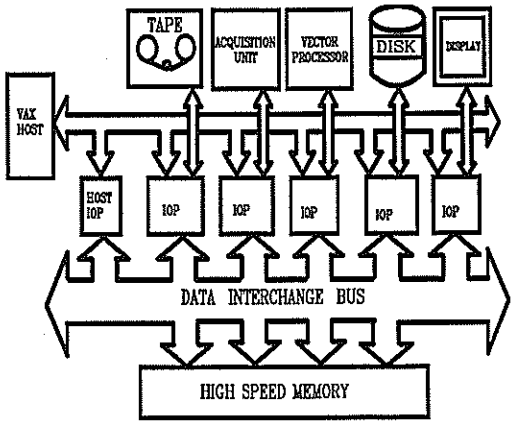


FIGURE 1

If the number of reads and writes on the bus are balanced, which they almost always are in typical pipelined applications, the bandwidth will be 200 Megabytes/s. In the worst case situations, the user is always guaranteed at least 100 Megabytes/s sustained bandwidth when using the read or write path alone.

The IOC high-speed memory , configured using 55 ns static random access memory, is implemented on 2,8,16,64 or 256 Megabytes boards. Each board, with a total board bandwidth of 50 Megabytes/s, has its own high performance memory controller to manage the board's four distinct memory banks at full speed.

3/ AN EXAMPLE APPLICATION

In order to see how this architecture is applied to a problem consider the following example. One thousand data blocks must be processed at high speeds one after another. The processing consists of reading a block in, sending it to an array processor for transformation and storing it on a high speed disk.

When solving the problem sequentially, the three steps (reading, transforming and writing) are performed for block 1, then for block 2, etc. Such a solution is represented in the simple timing diagram (figure 2). The times were selected for graphical representation only and are arbitrary. What we would like to do, however, is to create a data processing pipeline.

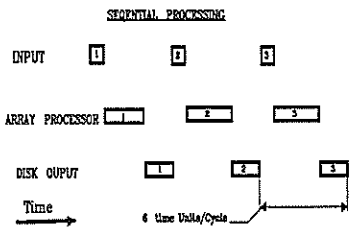


figure 2

In this solution the reading of data block n, the transformation of data block n-1, and the writing of data block n-2, proceed simultaneously. This solution is shown in figure 3.

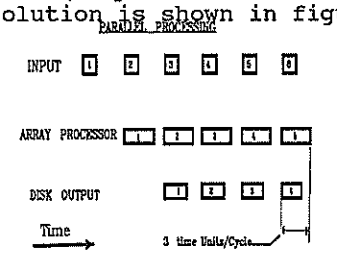


figure 3

With a conventional supermini, the second solution is only possible when the data rates are low. With the Aptec I/O Computer the realisation of the parallisation of the task is possible even with the data rates. The high available bandwidth of the IOC, the distributed intelligence and the elimination of monitor overhead allow one to archive data at 11 Megabytes/s (using the C51 system disk).

Utilization of such techniques on an Aptec IOC attached to a VAX system for simple problems as described in the example can result in many fold increases in performance. In more complicated systems, performance improvements up to a factor of 40 have been realized.

If a given IOP is programmed to process an input data block only when one is present in its request register and to pass a data block to its sequentially following neighbour only when space exists in the neighbour's request register, then the resulting chain of linked IOPs constitutes a classical pipeline. The IOP programming is essentially the same in every case, with the exceptions occurring, of course, at the boundaries of the pipeline where one needs to be concerned with memory allocation and de-allocation.

Maximum pipeline efficiency is achieved when all pipe steps are equal and as short as possible. In our example the array processor time is double that of the other two steps. If the mathematics allow it, the processes can be speeded up by another factor of two just by adding a second array processor. Note that with a convential supermini solution, the addition of a second array processor might be without benefit since it might not be possible to feed data to it fast enough.
Other variations on this scenario are just as easy to visualise.

4/ SOFTWARE ENVIRONMENT

Once an analysis of the problem has led to a decision on the hardware configuration, attention must turn to questions of processing task allocation and control, and programming. The objective is to set up the IOC so that a single command from the host starts the entire I/O subprocess. The host need not be disturbed again until the job is completed or the subsystem has intermediate results for the host. The general purpose software modules to do this reside in three phases on the host computer in each IOP and in the IOC high-speed memory.

Although the host plays little role during the execution of I/O subprocess in real-time, all program development and much debugging takes place there using VMS utilities and Aptec cross tools (assemblers, compilers, loaders and debuggers).

The host computer has access to the IOPs through a driver that enables the VMS user to down-load individual IOPs and send them requests for execution.

Host software is also available to allow the host to access the IOC high-speed memory as if it were a VMS files-11 device. Host programmers can create, open, close, read, write and delete data sets in the IOC memory.
The heart of the I/O subsystem software is naturally the microcode existing in the programm memory of the individual IOPs. Standard device oriented transfer routines for host and IOPs are provided by Aptec for a

wide range of industry standard array processors, disks and tapes.

In addition, an IOP runtime library consisting of both general and I/O primitives is useful in developing applications specific microcode.

For most applications, Aptec will have supplied all the bits and pieces needed to create the application without having to write additional IOP microcode. This allows the user to concentrate on combining these pieces into a program which coordinates the entire I/O process.

In order to simplify this process, Aptec created STAPLE, a high level, structured language similar to "C". STAPLE programs are written, compiled on the host into pseudocode and then loaded into a file area in the IOC memory. An IOP-resident STAPLE interpreter reads and executes the pseudocode when started by the host. Each IOP in the system can be executing a different STAPLE program if required.

Since, for most applications, the STAPLE program consists largely of a list of subroutine calls activating microcode routines in various IOPs in the system, the interpretive overhead is low. STAPLE is not intended to replace microcode for High-speed operations. Rather, it is designed as a complement to microcode, as a means of chaining together calls to library or user-written routines.

For those applications where the pipelining concept would be particularly effective, implementation totally in microcode is not that difficult. To achieve pipelined scheduling, existing code can be augmented by the use of the appropriate microcode to handle the various synchronising functions at each pipeline step in the process. This would include request acceptance from the outside world, passing a request of the new processor in the pipeline and perhaps informing the outside world when a request has completed pipeline traversal.

5/ COMPUTE POWER INSIDE THE IOC

The Vector/Scalar Processor (VSP-1) is a real-time computing solution for the IOC-24. The VSP-1 offers balanced scalar and vector processing and sustained high performance in an extremely reliable form factor and dramatic footprint reduction.
It consists of a 20 MFLOP vector processor, a scalar processor, a data formatter and a high-speed memory - all connected by a high-speed 80 Mbyte/s synchronous bus. Multiple VSP-1 boards may be configured in an IOC-24 (up to 24 in a dual chassis IOC), allowing compute power to be easily matched to an application's I/O volume.
The VSP-1 provides an alternative to attached array processors for many applications, as in the example given above. The software package includes development tools, high level languages and a comprehensive library of math subroutines.

6/ CONCLUSION

Traditional solutions do not easily support the integration of high-speed devices from multiple vendors and data transfers go through general purpose computers which have some limitations.
The Aptec solution is an open architecture which permits an easy integration of high-speed devices. The results are systems optimized for performance which respond to a lot of real time problems : telemetry, radar, sonar, image and signal processing, etc...
The bandwidth (up to 200 Mbytes/s) plus the parallelism and pipelining concepts support the interconnection of multiple peripherals such as: high-speed telemetry devices, A/D and D/A converters, computers,superminis, image processors, array processors, high-speed tapes,high-speed recorders, disks, VME devices...
For applications which require a high processing power, Aptec offers an off the shelf single vendor global solution by adjonction of its VSP-1 Vector/Scalar processors, which brings to the user 20 MFlops to 400 MFlpos of processing power. This solution resolves most of the real-time process.
The ease of use and power of the Aptec solution has been proven in more than 300 systems installed worlwide in the domains of Defense, Aerospace and Industries.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Adaptive IIR Echo Cancellers for Hybrids using the Motorola 56001

Markus Rupp†

Abstract

Because of bad adjusted hybrids electrical echo compensation is a desirable goal. Taking advantage of the special behavior of an electronic hybrid, an IIR filter structure for echo compensation is used. In comparison to a 32 order FIR Filter, only 12 coefficients are necessary to decrease the echo. Some special problems occuring in speech as excitation are solved. The speed of convergence reached is high enough for intelligible speech transmission.

1 Introduction

The disturbing phenomenon of hybrids in telephone networks is the occurence of inevitable electric echoes. On the one hand, these echoes may disturb the local speaker. On the other hand in long distance calls the energy of the echo may largely exceed the energy of the subscriber's signal and thus cause severe problems concerning quantization by the A/D converters after the hybrid. An echo canceller can considerably improve the quality of the received signal. Hereby, an improvement of 20dB is aspired to. Since every new connection changes the transfer function of the echo path, the echo canceller must be adaptive. Usually echo cancellers are designed as adaptive FIR filters [9], thereby employing the NLMS (Normalized Least Mean Square) algorithm [2]. In the last few years however, some new ideas arose to solve this problem by IIR filters [3] -[8]. In this paper two of the algorithms proposed [4,8] are discussed and their suitability for an implementation on the Motorola 56001 is outlined.

2 Motivation and Outline of the IIR Approach

Since the hybrid is an electronic circuit with discrete elements, the impulse response of the echo path is a linear combination of exponentials. This may be roughly seen in figure 1 which shows two typical impulse responses of an electronic hybrid. The first one has been measured with a very short path connection that is typical for an 'in house call', the other has been measured with a nine kilometer path connection.

Due to the exponential behavior, the required impulse response of the echo canceller can be modelled by a recursive rather than transversal structure. This fact motivates the application of an IIR filter, which may reduce computational complexity in contrast to the FIR filter approach. Figure 2 shows the hybrid with the compensator as it is used in this application. The filters BP1 to BP3, necessary for antialiasing and reconstruction, are equal bandpasses with cutoff frequencies at 300Hz and 3.3kHz. The influence of these bandpasses for measuring is discussed later.

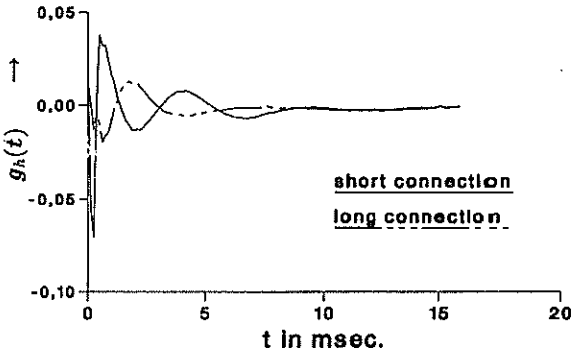


Figure 1: Measured impulse responses of an electronic hybrid

Employing the digital simulation theorem yields a time discrete description of the echo cancellation problem as shown in figure 3. Below the dashed line a first solution that is frequently used in parameter estimation [1] is depicted. The speech signal from the local speaker is directly filtered by a 'parallel part' as in former solutions. But the echo signal from the hybrid is also filtered by a 'series part'. Both the parallel part and the series part are transversal filters and the structure as a whole is no real IIR filter. Rather, it is

†Institut für Netzwerk und Signaltheorie, TH Darmstadt, Merckstr. 25, D-6100 Darmstadt, FRG

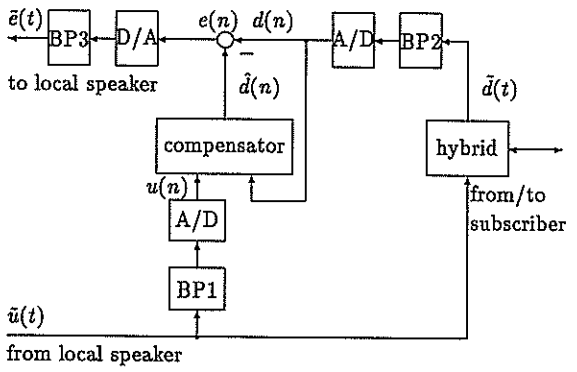


Figure 2: Compensator application

a special form of a linear combiner, which allows use of the well known NLMS algorithm for adaptation.

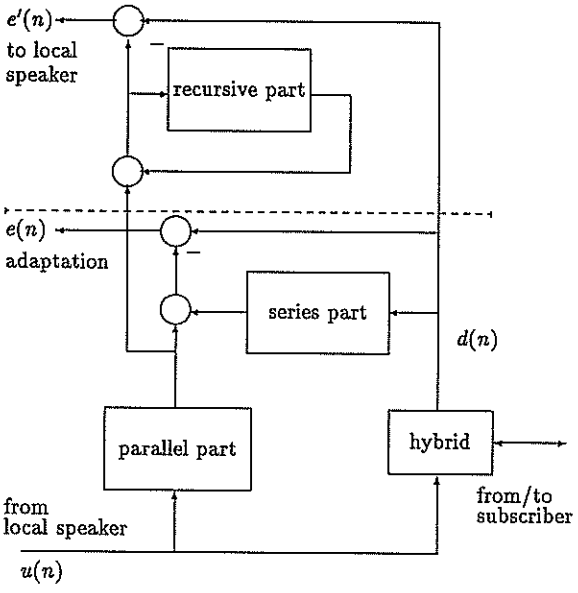


Figure 3: Echo cancellation of hybrids with additional recursive part

In this solution the error signal $e(n)$ is used for both the adaptation and the output signal for the local speaker. The error $e(n)$ is calculated by:

$$e(n) = d(n) - \sum_{k=1}^{M_a} a_k d(n-k) - \sum_{k=0}^{M_b} b_k u(n-k) \quad (1)$$

$$= d(n) - \underline{w}^T \underline{x}(n) \quad (2)$$

where a_k and b_k are the coefficients of the series and the parallel parts, respectively. The parameter vector $\underline{w}$ and the signal vector $\underline{x}(n)$ are as follows:

$$\underline{w}^T = (b_0, b_1, ..., b_{M_b}, a_1, ..., a_{M_a}), \quad (3)$$

$$\underline{x}^T(n) = (u(n), u(n-1), ..., u(n-M_b), d(n-1), ..., d(n-M_a)) . \quad (4)$$

Finally, the update equations are:

$$a_k(n+1) = a_k(n) + \frac{\alpha}{\underline{x}^T \underline{x}} e(n) d(n-k) \quad ; \text{ for } k = 1 \text{ to } M_a , \quad (5)$$

$$b_k(n+1) = b_k(n) + \frac{\alpha}{\underline{x}^T \underline{x}} e(n) u(n-k) \quad ; \text{ for } k = 0 \text{ to } M_b . \quad (6)$$

3 Improvements

Noise and the subscriber's speech signal considerably disturb the adaptation and consequently a large coefficient bias will arise. This phenomenon, however, is of principle nature since for this situation the Wiener solution is biased as well [8]. Because of its simple form the adaptation can be stopped without any problems and restarted, if desired. Therefore, a double talk detector can be applied to reduce the bias.

One major drawback of this structure is that the subscriber's speech signal is distorted by the series part. To avoid this, the local speaker signal is filtered by an additional recursive part (see Fig. 3). Its coefficients are a copy of the series part coefficients. The resulting signal $e'(n)$ contains the subscriber's speech without any distortion. The original error signal $e(n)$ is only used for the adaptation algorithm that remains the same. Therefore, the speed of convergence is the same as before. However, this IIR structure gives rise to another problem. A periodic signal like vowels in the subscriber's speech makes the series part investigate the LPC coefficients for this signal causing instability of the recursive part. Here again, the double talk detector and a moderate stepsize α improve the situation. Whenever the signal from the subscriber side exceeds a certain level, the adaptation is stopped and oscillations are prevented.

4 Measuring

Commonly the quality of the compensation is described by ERLE (=Echo Return Loss Enhancement), depending on the signals used. A better measure is the relative

system mismatch. It will be defined in the frequency domain, because the influence of the bandpasses BP1 and BP2 is easier to describe. Let $G_h(e^{j\Omega})$ and $G_c(e^{j\Omega})$ be the Fourier transforms of the impulse responses $g_h(i)$ and $g_c(i)$ of the hybrid and the compensator, respectively. The relative system mismatch in dB is:

$$S_{rel} = 10\lg \frac{\int_{-\pi}^{\pi} \left|B(e^{j\Omega})\left(G_h(e^{j\Omega}) - G_c(e^{j\Omega})\right)\right|^2 d\Omega}{\int_{-\pi}^{\pi} \left|B(e^{j\Omega})G_h(e^{j\Omega})\right|^2 d\Omega} \quad (7)$$

where $B(e^{j\Omega})$ is the Fourier transform of the bandpass impulse response. Figure 4 shows $\left|G_c(e^{j\Omega})\right|$ found by the NLMS in case of a transversal structure with length 32 and a recursive structure ($M_a = 3, M_b = 8$).

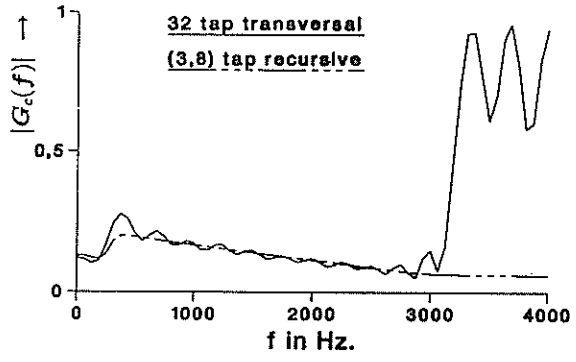


Figure 4: Magnitudes of $G_c(e^{j\Omega})$ employing a transversal and a recursive structure

In the passband both curves match very well in contrast to the stopband, where no excitation is present. With Eqn. 7 it is possible to investigate the achievable system mismatch assuming a certain number of poles and zeros. Here, the resulting system mismatch is 12dB in the transversal and 17dB in the recursive case. ERLE of the output error $e'(n)$ measured for these two realizations is depicted in figure 5.

ERLE measured with white noise as excitation is very close to the calculated system mismatch. Due to additional noise from the subscriber side, ERLE achievable is limited by approximately 28dB reached by a transversal filter of order 160. The power loss of a typical hybrid is at least 15dB. Therefore, together with the echo canceller a total echo attenuation of more than 30dB is sure. An important feature of the adaptation with the recursive structure is that its speed of convergence does not severely depend on the excitation signal. As depicted in Fig. 5 an improvement of circa 14dB is reached in a very short time (< 200ms). The

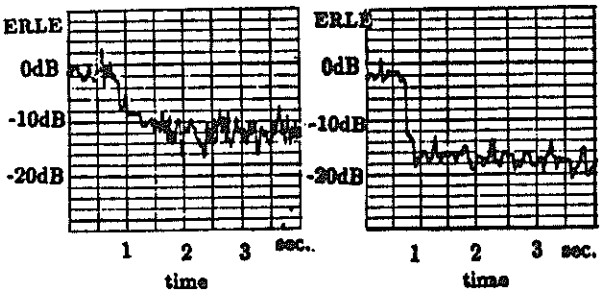


Figure 5: ERLE for a 32 tap transversal and a ($M_a = 3, M_b = 8$) tap recursive filter

remaining 2dB require a few seconds. This behavior is acceptable for typical speech applications.

A drawback of the additional recursive part is that after the adaptation has reached steady state the output error $e'(n)$ exceeds the adaptation error $e(n)$. The output error $e'(n)$ can be viewed as the adaptation error $e(n)$ filtered by the recursive part.

$$e(n) = e'(n) - \sum_{k=1}^{M_a} a_k\, e'(n-k) \quad (8)$$

With the measured coefficients the energy of $e'(n)$ exceeds $e(n)$ by 10dB. ERLE of 16dB reached with the recursive filter is a rather good value and can be improved only by augmenting the number of transversal coefficients. Using more series coefficients does not result in better compensation because the additional poles are compensated by the corresponding zeros. A FIR filter with the same computational complexity as the recursive filter achieves less than 10dB.

5 Fix-Point-Processor Limits

The last part of this article is engaged in investigating limits caused by quantization of coefficients and signals. From the fixed point analysis of the LMS [10] the following formula for the steady-state of the mean squared error $e(n)$ is obtained:

$$\begin{aligned} \zeta &= \zeta_{min} + \mu \frac{\zeta_{min}\, tr\mathbf{R}_{xx}}{2 - \mu\, tr\mathbf{R}_{xx}} \\ &+ \frac{1}{\mu} \frac{M\sigma_w^2}{2 - \mu\, tr\mathbf{R}_{xx}} \\ &+ \mu \frac{\left(1 + \underline{w}^T\underline{w}\right) tr\mathbf{R}_{xx} + \zeta_{min} M}{2 - \mu\, tr\mathbf{R}_{xx}} \sigma_q^2 . \end{aligned} \quad (9)$$

Hereby, the variances of the quantization errors of the signals $u(n)$ and $d(n)$ are assumed to equal σ_q^2. The order M is $M_a + M_b + 1$, the number of all coefficients. From the ACF matrix $\mathbf{R}_{xx}$ of the vector $\underline{x}(n)$ only the trace is used. It depends on the input energy σ_u^2 and the output energy σ_d^2 of the hybrid via:

$$\begin{aligned} tr\mathbf{R}_{xx} &= E[\underline{x}(n)^T \underline{x}(n)] && (10)\\ &= M_a\,\sigma_d^2 + (M_b+1)\,\sigma_u^2 && (11)\\ &\approx M\sigma_u^2\,. && (12)\end{aligned}$$

Here, persistent excitation on both speaker sides with $\sigma_u^2 = \sigma_d^2$ is assumed. Concerning the wordlength of the coefficients, σ_w^2 is the variance of the least significant bit changing. The stepsize μ corresponds to the stepsize α divided by the quadratic norm of the vector $\underline{x}(n)$. If the stepsize μ is sufficiently small the equation reduces to:

$$\begin{aligned} \zeta &= \zeta_{min} + \frac{\mu\zeta_{min}\,tr\mathbf{R}_{xx}}{2} + \frac{M\sigma_w^2}{2\mu}\\ &+ \frac{\mu}{2}\left(\left(1+\underline{w}^T\underline{w}\right)tr\mathbf{R}_{xx} + \zeta_{min}M\right)\sigma_q^2 && (13)\\ &\approx \zeta_{min} + \frac{\alpha\zeta_{min}}{2} + \frac{M^2\sigma_w^2\sigma_u^2}{2\alpha}\\ &+ \frac{\alpha}{2}\left(\left(1+\underline{w}^T\underline{w}\right) + \frac{\zeta_{min}}{\sigma_u^2}\right)\sigma_q^2\,. && (14)\end{aligned}$$

Because of the big wordlength of 24 bits of the Motorola 56001 the error power is limited by the signal quantization error. The biggest term that determines the reachable error is:

$$\zeta \approx \frac{\alpha}{2}\left(1+\underline{w}^T\underline{w}\right)\sigma_q^2\,. \qquad (15)$$

Therefore, approximately 60−70dB can be achieved assuming that the incoming signals $d(n)$ and $u(n)$ are quantized with 12 bits. A floating point processor would not improve this value.

6 Summary

It has been shown that general purpose fixpoint signalprocessors, like the Motorola 56001, are suitable for realizing algorithms which can sufficiently compensate electrical hybrid echoes. The results concerning a real time implementation of a recursive adaptive filter with 12 coefficients have been discussed. The compensation quality exceeds that of a transversal compensator of order 32, although requiring only 42% of the computational effort. Typical problems caused by the subscriber's speech like harmonics of the speech signal have sufficiently been solved. With the recursive algorithm described and normal operating environments, the desired improvement of 20 dB is closely reached.

Acknowledgement

The author wishes to thank Jürgen Cezanne for his helpful discussions.

References

[1] Pieter Eykhoff, System Identification, John Wiley & Sons, 1979.

[2] Symon Haykin, Adaptive Filter Theory, Prentice Hall, 1986.

[3] Maurice G. Bellanger, Adaptive Digital Filters and Signal Analysis, Marcel Dekker, Inc./ New York Basel, 1986.

[4] Michael G. Larimore, John R. Treichler, C. Richard Johnson, SHARF: An Algorithm for Adapting IIR Digital Filters, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-28, No. 4, August 1980, pp. 428-440.

[5] Hong Fan, A New Adaptive IIR Filter, IEEE Transactions on Circuits and Systems, Vol. Cas-33, No. 10, October 1986, pp. 939-947.

[6] Hong Fan, An Investigation of an Adaptive IIR Echo Canceller, Advantages and Problems, IEEE Transactions on Acoustics,Speech, and Signal Processing, Vol. 36, No. 12, December 1988, pp. 1819-1833.

[7] C. Richard Johnson, Adaptive IIR Filtering, Current Results and Open Issues, IEEE Transactions on Information Theory, Vol. IT-30, No. 2, March 1984.

[8] John J. Shynk, Adaptive IIR Filtering, IEEE ASSP Magazine, April 1989.

[9] Man Mohan Sondi, David A. Berkley, Silencing Echoes on the Telephone Network, Proceedings of the IEEE, Vol. 68, No. 8, August 1980, pp. 948-963.

[10] Christos Caraiscos, Bede Liu, A Roundoff Error Analysis of the LMS Adaptive Algorithm, IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. ASSP-32, No. 1, February 1984, pp. 34-39.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

The ESPRIT Algorithm on a Transputer array

J.S. McGarrity, J.J. Soraghan, T.S. Durrani

Signal Processing Division, Dept. Electronic and Electrical Engineering,
University of Strathclyde, Glasgow G1, Scotland.

Abstract- This paper describes the implementation of the TLS-ESPRIT algorithm on a Transputer array. TLS-ESPRIT is parameter estimation algorithm, using subspace rotations, which is used for obtaining high resolution, unbiased estimates of the frequencies and powers of complex sinusoids in noise. The algorithm involves complex decompositions which require extensive computation. In order to use the algorithm in close to real-time situations, it is altered to facilitate faster computation and the use of the parallel architecture. The altered algorithm is decomposed onto an array of Inmos T800 transputers and its performance compared with the standard algorithm.

1. INTRODUCTION

Problem Formulation and Standard Notation

Consider a uniform linear array of m sensors, which is receiving signals from *d* sources. The *d* sources are assumed to be *farfield* of the array, and their radiation is therefore incident on the array as the superposition of planewaves. The signals are also assumed to be *narrow-band* processes with the same known center frequency ω_0. and therefore a time delay τ incurred between a source and the receiving sensors can be expressed as a phase shift, i.e. for a source signal s(t), the delayed signal at a sensor is $s(t-\tau)=s(t)e^{-j\omega_0\tau}$.

For each sensor , k, there is a relative propagation delay $\tau_k(\theta_i)$ for a signal coming from angle θ_i , therefore the composite output of sensor k can be written as

$$x_k = \sum_{i=1}^{d} a_k(\theta_i)\ s(t-\tau_k(\theta_k))$$

$$= \sum_{i=1}^{d} a_k(\theta_i)\ s(t)\ e^{-j\omega_0\tau k(\theta i)}$$

where $a_k(\theta_i)$ is the response (gain and phase) at frequency ω_0 of the kth sensor to a signal coming from direction θ_i , or in vector notation

$$\mathbf{x}(t)= \sum_{i=1}^{d} \mathbf{a}\ (\theta_i)\ s_i\ (t), \qquad (1)$$

where

$$\mathbf{a}\ (\theta_i) = [\ a_1(\theta_i)\ e^{-j\omega_0\tau 1(\theta i)}, \ldots, a_m\ (\theta_i)\ e^{-j\omega_0\tau m(\theta i)}\ ]^T$$

$\mathbf{a}(\theta_i)$ is known as the *array response* or the *steering vector* as it contains the delays needed to be inserted in the array in order to electronically steer it in direction θ_i.

Now letting

$$\mathbf{A}(\theta)=[\ \mathbf{a}\ (\theta_1), \ldots, \mathbf{a}\ (\theta_d)\ ]$$

$$\mathbf{s}(t)=[\ s_1\ (t), \ldots, s_d\ (t)]^T$$

and letting $\mathbf{n}(t)$ be a vector of additive noise at each sensor, then

equation (1) can be written more concisely as

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) + \mathbf{n}(t)$$

where

$$\mathbf{x}(t), \mathbf{n}(t) \in C^m, \mathbf{s}(t) \in C^d \text{ and } \mathbf{A}(\theta) \in C^{m\times d}$$

Signal subspace and array manifold

Firstly consider the noise free case. The vectors $\mathbf{a}\ (\theta_i) \in C^m$, the steering vectors and the columns of the matrix $\mathbf{A}(\theta)$ are elements of a set called the *array manifold* which is the set of steering vectors for all possible angles θ_i. As $\mathbf{x}(t) = \mathbf{a}(\theta_i)s_i(t)$, for each source direction this means that $\mathbf{x}(t)$ is a linear combination of the d steering vectors $\mathbf{a}(\theta_i)$ and is therefore limited to the *d*-dimensional subspace spanned by the *d* steering vectors $\mathbf{a}\ (\theta_i)$, this is termed the *signal subspace*, denoted S_X.

ESPRIT algorithm

ESPRIT treats the samples in time series analysis as the outputs of two arrays of *m* sensors, X and Y, or one array of *m* sensor *doublets*. The displacement between all pairs of sensors must be identical. The signal subspaces for the X and Y arrays, S_x and S_y, are generated by finding two sets of *d* linearly independent vectors, $\mathbf{E}_x$ and $\mathbf{E}_y$, that span the spaces. The displacement of the arrays causes a *rotation* between the subspaces; this rotation operator when calculated, gives information about the signal parameters. In the first version of the ESPRIT algorithm [1] the least squared criterion was applied to the problem of finding

the rotation operator Ψ

$$\mathbf{E}_x\Psi=\mathbf{E}_y$$

or $$\Psi=[\mathbf{E}_x^*\mathbf{E}_x]^{-1}\mathbf{E}_x^*\mathbf{E}_y$$

But as noise is present in $\mathbf{E}_x$ and $\mathbf{E}_y$ a criterion which takes this into account is more appropriate i.e. the total least squared criterion. Hence the method is called TLS-ESPRIT[2].See [1][2] and [3] for summary of algorithm.

2. REDUCED COMPUTATION ALGORITHM

Algorithm Simplification

A) Firstly and most simply, the covariance matrix for the noise is set as the identity matrix.

B)In obtaining an estimate for the signal subspace, *any set of d linearly independent* vectors are required that span the same space as the data vectors and hence the direction vectors i.e. $S_x=\Re\{\mathbf{A}\}=\Re\{\mathbf{E}_s\}$. This set of linearly independent vectors can be obtained in a number of ways:

1)One such set is the d eigenvectors corresponding to the d largest eigenvalues of the covariance matrix.

2)Now only *d linearly independent* vectors are required, i.e. they do not have to be orthogonal. The column vectors in the covariance matrix will never be *linearly dependent* because of the noise in the data. They also span the signal subspace and the noise subspace (averaging is used to reduce noise) and therefore we can use d of these vectors to get an estimate of the signal subspace. This is obviously beneficial as it reduces the computation by eliminating the need to generate orthogonal basis.

C)In computing the eigenvectors of $\mathbf{E}_{xy}^*\mathbf{E}_{xy}$, the vectors of the **Q** matrix in the **QU** factorisation of $\mathbf{E}_{xy}^*\mathbf{E}_{xy}$ are a good enough approximation. So now the most complicated operation in the algorithm is only a d x d eigenvalue problem in obtaining the rotation operator.

Summary of Reduced Computation algorithm

1)Estimate the covariance matrix of the Z array, $\mathbf{R}_{zz}$ by outer product of snapshot data matrix

2)Estimate basis for signal subspace. by firstly averaging $\mathbf{R}_{zz}$, and then taking d column vectors

3)Decompose vectors into basis for the signal subspaces for both the **X** and **Y** arrays, $\mathbf{E}_x$ and $\mathbf{E}_y$ respectively

4)Compute QU factorisation of $\mathbf{E}_{xy}^*\mathbf{E}_{xy}$, where $\mathbf{E}_{xy}$ is as defined earlier,

$\mathbf{E}_{xy}^*\mathbf{E}_{xy}=\mathbf{EU}$

where **E** is the **Q** matrix in the QU factorisation

5)Partition E into four d x d submatrices

$$\mathbf{E} \stackrel{\text{def}}{=} \begin{bmatrix} \mathbf{E}_{11} & \mathbf{E}_{12} \\ \mathbf{E}_{21} & \mathbf{E}_{22} \end{bmatrix}$$

6)Calculate the eigenvalues of Ψ

$$\Psi=-\mathbf{E}_{12}[\,\mathbf{E}_{22}\,]^{-1}$$

7)Eigenvalues for complex sinusoids are on the unit circle from 0 to π Radians which represent *angles of arrival* in the Direction Finding case or just *frequencies* from 0 to .5 of the sampling frequency in the time series case. If the number of sources are over estimated, eigenvalues off the unit circle are generated. Problems arise if the number of sources are underestimated.

DOA results

Simulations were carried out to compare the performance of the standard algorithm and the reduced computation algorithm.

(I) Two complex sinusoids at 0.1 and 0.2 of the sampling frequency, in 10db noise were simulated with 20 samples, results are shown in figures 1(a) and (b).

(II)Three complex sinusoids were simulated at 0.1(a), 0.2(b) and 0.3(c) . Arrays of 10 sensors and 9 snapshots were used. 500 simulations were carried out and the variances and rms errors in the ESPRIT estimations for the frequencies at different SNR's calculated. The simulation results are shown in figure 2(a) for the standard algorithm and 2(b) for the reduced algorithm. From the two experiments it is clear that the variance and rms error of the estimates is similar for the two methods. However the way the vectors are averaged to form the signal subspace estimate and the way the sinusoids interact, strongly effect the accuracy of the algorithm. In the example these effects have caused one frequency estimate to be biased.

Timings

To illustrate the increase in speed the second method gives, timings for the two algorithms are shown below. These results were obtained using MATLAB on a SUN 3 without a floating point accelerator.

For d =2, m=10 and N=9	
Standard algorithm	10.66 s
Reduced computation algorithm	00.52 s
Speedup	~20

3. TRANSPUTER HARDWARE

The Inmos transputer family of microprocessors are designed so as to implement exactly in hardware the parallel programming language OCCAM. This language is based on a *dataflow process model* where all tasks are composed of a number of processes which either run sequentially or in parallel. Communication between parallel processes is solely via one way *channels*, two being required for two way communication; there is no concept of shared data or variables. Concurrent processes are timesliced to run on a single transputer, but when decomposed onto multiple processors, *channels* are placed on bidirectional *links* - 20Mb/s serial lines between transputers of which there are four. Separate DMA controllers for each link transfer data independent of the CPU. Interconnection of the serial links allow any complex parallel architecture to be constructed with the transputer plus some memory as a node.

The T800 transputer is a 32-bit processor with 4 links plus a 64-bit floating point unit for IEEE 754-1985 standard arithmetic, which also works independent of the CPU. Industry standard Transputer modules (Trams) on a mother board are used in the system. Each Tram comprises a 17.5 MHz T800 with 1Mb DRAM.

As the transputer is a complex microprocessor with serial communication it is intended as a general purpose node for use in *large grain-size* problems and not in special purpose, *small grain-size,* systolic array systems. In this problem the matrices are small (~10-20 elements) square, and hence maximum speedup is obtained from using a small number of processing elements to minimise communication time.

4. SERIAL IMPLEMENTATION

A program was written in OCCAM for serial computation of the algorithm on a single T800. The timings for the steps in calculations were used to determine the most computationally intensive section of the algorithm, with reference to the flop count in MATLAB. Complex data was used with 32-bit precision arithmetic. The matrix arithmetic involved in the algorithm is straightforward with the exception of the following routines:

QU factorisation by Householder rotations[4]
A fast and efficient method of QU factorisation (used in step 4) is Householder transformations. Here a series of elements below the diagonal are annihilated by one transformation. Successive columns are zeroed below the diagonal, resulting in the upper triangular matrix U. The Q is obtained from the product of the transformations.

Eigenvalue decomposition by the QU algorithm[4]
This task is still required in the reduced computation algorithm in the estimation of the rotation operator (step 6). The most common way to find all the eigenvalues of a matrix and the associated eigenvectors is the *QU algorithm.* This method repeatedly applies *QU factorisation* to converge to the eigenvalues.

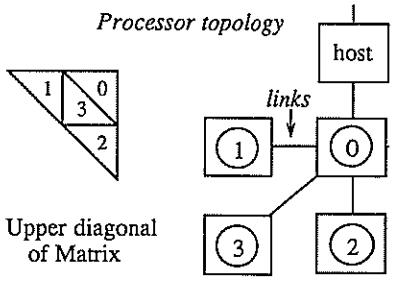
Timings: For d =2, m=10 and N=9

1) Form covariance matrix - 2mx2m multiplication	52.83 ms
2) Estimate signal subspace basis	05.34 ms
3) 2*dx*2*d* QU factorisation	13.32 ms
4) *dxd* eigenvalue decomposition and *dxd* matrix inversion	06.49 ms
total time	77.98 ms

5. PARALLEL IMPLEMENTATION

It is seen, from the above times, that for d small and m large the most computationally intensive section of the algorithm is infact the outer-product of the snapshot-data matrix. This operation is highly parallelisable.

Parallel matrix multiplication for small matrix
The multiplication here is a matrix times its complex conjugate transpose, therefore the result is Hermitian Toeplitz. Only the upper triangle of the result need be calculated with the lower triangle being easily obtained. Here the result is a square matrix so partitioning of the data is easy. Each processor has the task of computing the certain elements of the result matrix, as shown below. Processor 0 distributes the data to the others, collects the results, and communicates with the host..



The computation time for the multiplication was reduced to 15.1 ms. Speed up in multiplication is ~3.5.
The **total time** is therefore reduced to 40.25 ms.

Parallel Language

An example of how the OCCAM language decomposes an algorithm onto multiple transputers is shown below.

1)Declare channel types
2)Place channels on links
3)Place code on transputers, passing channels as parameters.

```
CHAN OF Datain into1, into2 :
CHAN OF Dataout outof1, outof2 :
CHAN OF ANY toHost, fromHost, Bootme1, Bootme3 :
PLACED PAR
  PROCESSOR 0 T8
    PLACE into1 AT link0.out :
    PLACE outof1 AT link0.in :
    PLACE into2 AT link1.out :
    PLACE outof2 AT link1.in :
    PLACE into3 AT link3.out :
    PLACE outof3 AT link3.in :
    PLACE toHost AT link2.out:
    PLACE fromHost AT link2.in:
    proc0(toHost,fromHost, into1,into2,into3,outof1,outof2,outof3)
  PROCESSOR 1 T8
    PLACE into1 AT link1.in :
    PLACE outof1 AT link1.out :
    PLACE Bootme1 AT link2.out :
    proc1(into1,outof1)
  PROCESSOR 2 T8
    PLACE into2 AT link3.in :
    PLACE outof2 AT link3.out :
    PLACE Bootme3 AT link2.in :
    proc2(into2,outof2)
  PROCESSOR 3 T8
    PLACE into3 AT link3.in :
    PLACE outof3 AT link3.out :
    PLACE Bootme1 AT link1.in :
    PLACE Bootme3 AT link1.out :
    proc3(into3,outof3)
```

6. CONCLUSIONS

As can be seen the computation of the TLS-ESPRIT Algorithm can be significantly reduced with a resulting reduction in performance. The experiments were carried out with m=10 and d=2 or 3. For m much larger, speed up in the parallel multiplication will be greater, due to processing-time/communication-time trade-offs.. Future investigation is still required in the tradeoff between performance and speed. In the future a vector processor will be used to speed up the algorithm.

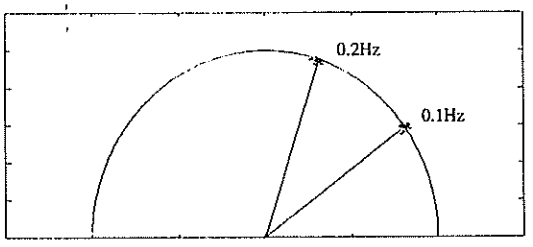


Figure 1(a) Standard Algorithm

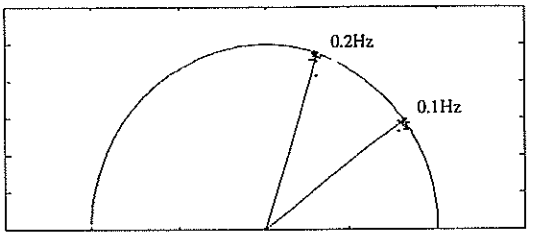


Figure 1(b) Reduced Computation Algorithm

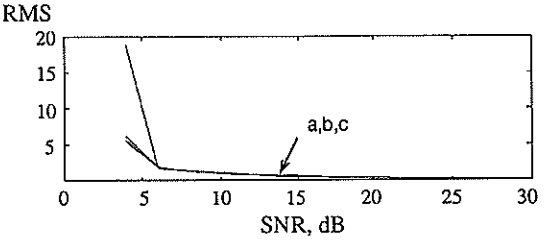


Figure 2(a) Standard Algorithm
Three Sources, m=10, N=9

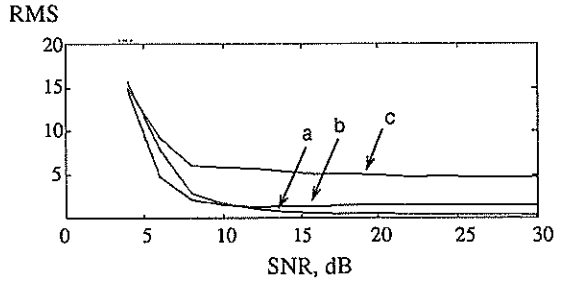


Figure 2(b) Reduced Computation Algorithm
Three Sources, m=10, N=9

ACKNOWLEDGMENT

The work was carried out at the Parallel Signal Processing Centre of the University of Strathclyde, with support from the Science and Engineering Research Council under contract number SERC GR/F/07217.

REFERENCES

[1]R. Roy, A. Paulraj, T. Kailath, "ESPRIT - A Subspace Rotation Approach to Estimation of Parameters of Cisoids in Noise", IEEE Transactions on ASSP, vol ASSP - 34, No.5, October 1986.

[2]R. Roy, T. Kailath, "ESPRIT - Estimation of Signal Parameters Via Rotational Invariance Techniques", IEEE Transactions on ASSP, vol 37, No.7, July 1989.

[3]J.S.McGarrity, J.J.Soraghan, T.S.Durrani, "Fast Implemenation of The ESPRIT Algorithm", ICASSP '90, Albuquerque, New Mexico.

[4]G.H. Golub, C.F. Van Loan, "Matrix computations", North Oxford Academic.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

DSP BASED TECHNOLOGY FOR EUROPEAN MOBILE RADIO

LUC MARY - CORPORATE STRATEGIC MARKETING
SGS-THOMSON MICROELECTRONICS
7, Av. GALLIENI - 94253 GENTILLY CEDEX - FRANCE

Digital Signal Processing has been extensively used in high speed telephone data modems such as 9600 bit/second full duplex V.32 modems with echo cancellation and Viterbi decoder. As Digital Mobile Radio Communications extend now rapidly, a big potential new area opens to Digital Signal Processing for low bit rate high quality speech coder, multipath channel equalization and Viterbi decoder. Additional cost and power consumption requirements push for semi-custom DSP based technology.

1. INTRODUCTION

In 1991, the new Pan-European cellular Digital Mobile Radio Network will start operations. The same recommendation, issued by ETSI (European Telecommunications Standards Institute) working "Groupe Special Mobile" (GSM) is supported by 17 countries (1) - the GSM network will be the most advanced mobile telephone system in the world offering international roaming (full service capabilities outside the home country), fully automatic traffic handling, high quality transmission due to digital implementation and other than speech communication (2) - The GSM recommendation includes sophisticated digital signal processing such-as 13 kbps (bit per second) RPE-LTP (Regular Pulse Excitation with Long Term Prediction) linear predictive voice coding, channel half-rate convolutional coding, diagonal interleaving, frequency hopping in the 900 MHz range, automatic adaptive channel equalization and Viterbi decoding for error correction. At the same time the potential market size is so huge (17 western european countries that may well extend eastward) that there is a formidable push for technology breakthrough from equipment suppliers on semiconductor manufacturers to reduce the Mobile Station (MS) to a few chips hence giving way to a 300 ECU handheld GSM phone in 1995.
This paper will first focus on the MS architecture and basic technology requirements to show that the best compromise, from a mixed economical, technical and industrial point of view is a DSP (Digital Signal Processor) based ASIC (Application Specific Integrated Circuit) approach together with digital technology compatible analog to digital and digital to analog converters.
Future trends examination will show that a Bi CMOS analog and digital technology will permit further integration down to a pocket sized MS.

2. BASIC TECHNOLOGY REQUIREMENTS

The mobile station (MS) can be first analyzed functionally then organically.
The functional analysis is examplified by Fig.1 and does not need here to be detailed.

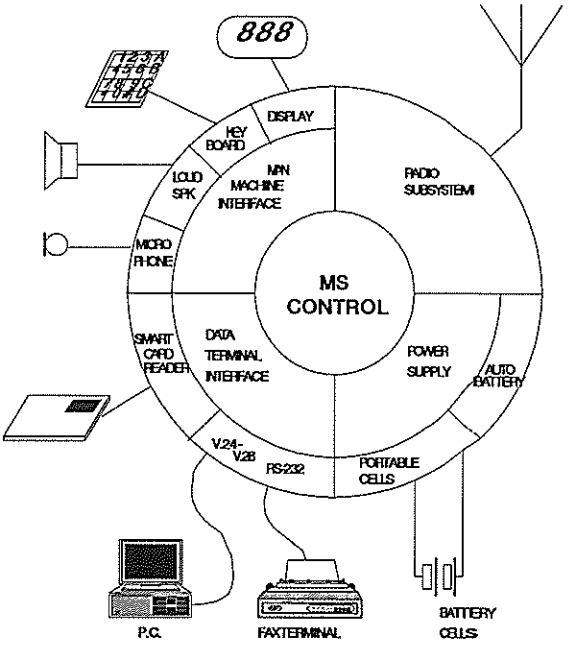
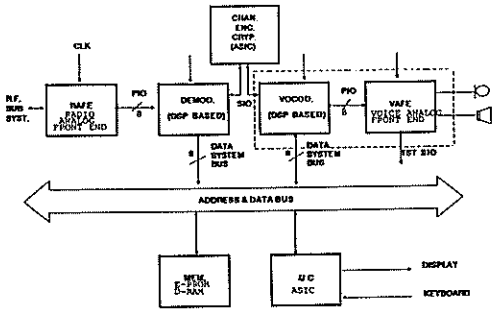


FIG.1 - MS FUNCTIONAL PARTIONING

However it may be informative for the reader to know, for instance, that the man machine interface may include such functions as voice dialing and hands free communication which mean voice recognition and synthesis, echo cancellation and noise reduction. Additionally it may also interface remote human calls by offering static telephone answering with voice messaging services. The data terminal interface may handle, besides a personal computer (PC), a fax terminal to offer a cheap solution to local road map and driver routing service without necessitating expensive and inadequate Compact Disk read and display machine.
The organic partitioning, as expected, does not map directly onto the functional analysis as a programmable device (microprocessor or digital signal processor) can be reused efficiently, thru program and context switching, to implement different functions which do not need to be supported simultaneously. The simplified organic diagram shown on Fig. 2 summarizes the main devices constituting the MS together with their interconnection scheme.
The radio subsystem built around a fixed point 16 bit DSP machine is connected to the bus of a general purpose 8 or 16 bit microprocessor (uP) in only one point.
It is here timely to emphasize that data exchange between the uP and the DSP must be designed in order to minimize the overhead for both machines.



GSM MOBILE RADIO BLOCK DIAGRAM
(EXCLUDING R.F.)

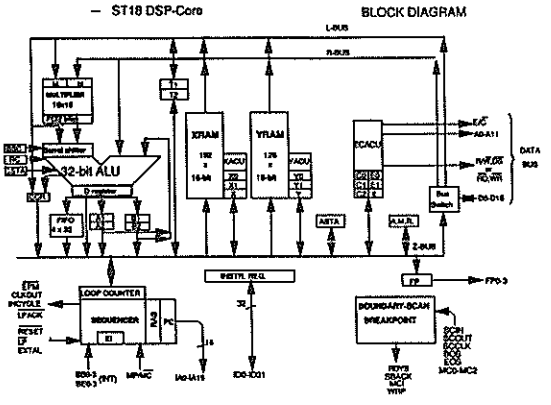
- FIGURE 2 -

A DSP can easily run today at 77 nS (derived from the 13 MHz GSM master clock) and is a highly parallelized and somewhat pipelined machine which is not intended to be interrupted frequently, in particular by a relatively slow and very sequential uP.
An adequate solution is to use a dual port set of registers that can be read and written by both the uP and DSP as provided by the system mailbox of ST18 DSPs (3). Other important considerations about data exchange between uPs and DSPs inside a MS are power consumption and cost.
It is out of the question on a battery fed and cost sensitive MS to have high speed buses laid down on Printed Circuit Board (PCB) running at over 10 MHz and linking expensive fast access time (under 70 ns) memory and peripheral chips.
More preferably, only the slow uP bus will be laid down (on 8 bit) on the PCB to take advantage of cheap, high volume, chips such as 1 M bit D-RAM or E-PROM. The DSP machine for obvious reasons will rely exclusively on on-chip high speed 16 bit buses making mandatory the DSP based ASIC approach which offer additional side benefits such as lower cost (highter integration level and lower package pin count) and better industrial property protection (the handheld MS will be a hot mass consumer market).
Moreover a DSP based ASIC design offer a cost effective yet flexible mixed solution to the diverse requirements of the GSM signal processing. For instance the automatic adaptive channel equalizer, the 13 kbps RPE-LTP vocoder and speech processing optional functions can be advantageously implemented on a programmable DSP offering basic algorithms support and room for fine tuning or improvements. However bit handling, which may be found in channel coder and decoder, interleaver and deinterleaver or even in the modulator, may be preferably implemented in dedicated wired logic. Indeed these functions would make poor use of the DSP's 32 bit arithmetic and Logic Unit (ALU) and moreover do not need any flexibility as they are completely defined by the GSM recommendation and can be designed in a single step.
The radio subsystem needs also several analogue converters which can be divided in two groups depending on the sampling frequency range. Voice analogue interface needs 8 kHz sampling frequency converters with a dynamic range requiring 13 bit and a differential linearity of about 10 bit. State of the art delta-sigma converters offer good performance with 14 bit linearity requiring only 7 successive half-band decimation (or interpolation) filters. In GSM this calls for a 128 x 8 = 1024 kHz clock which cannot be derived easily from the 13 MHz clock. A digital phase locked loop synchronized on the 8 kHz clock can solve this problem.
Radio Frenquency analogue interface needs much faster converters, in the range of 540 kHz for the I and Q receive analogue to digital converters and twice that frequency for the I and Q transmit digital to analogue converters.Fortunately, 10 bit are only required which allow to use the classic successive approximation technics implemented with a slightly modified digital C-MOS technology such as the HC-MOS 3A from SGS-Thomson and LETI which offers 1.2 um design rules for switched capacitor (sampled) analog designs.
It is quite possible, from a technology point of view, to put all the analogue converters on a single chip RAFE (Radio Analogue Front End)

using the HC-MOS3A process.

3. DSP BASED ASIC APPROACH

As digital C-MOS technology design rules are now under the micron level in volume production it becomes feasible to design an ASIC chip including a fixed point 16 bit DSP core.
The ST18932 DSP core from SGS-Thomson Microelectronics (Fig 3) is a good ASIC candidate as it compares favourably in terms of performances to any other fixed point 16 bit DSP yet has one of the smallest area leaving more than the equivalent core area available for Program ROM (3K x 32 bit), Data RAM (1k x 16 bit) and a few thousands of standard cells to accommodate dedicated Digital Signal Processing functions as well as interfaces.
The ST 18932 architecture is quite similar to the standard ST18931 (ROM less version of ST18930) but offers several distinctive enhancements to ease GSM signal processing implementation (4).
First it allows to run at 77 ns (32 bit) instruction cycle time which is adequate with the 13 MHz GSM clock.



- FIGURE 3 -

It also offers a full 32 bit ALU with floating point support operational codes (normalization in 2 cycles) which are welcome to design easily a GSM Rec 06.10 13 kbps RPE-LTP vocoder.
Of importance for a handheld telephone is the full static design which permits a very low power standby mode.

All necessary tools are provided to describe and simulate the complete chip in a high level language : an IEEE 1076-1987 VHDL (VHSIC Hardware Description Language) model of the standard cell library, including the DSP core, is available on standard Engineering Workstation (EWS) like Mentor.
Of course all other tools, macrogenerators, layout compilers and automatic place and route in particular, are also available to go up to the P.G. Tape.
Fig. 4 shows a typical DSP based ASIC chip which can be designed in a 0.8 um digital C-MOS technology such as the HCMOS4 from SGS-Thomson using a standard cell library.

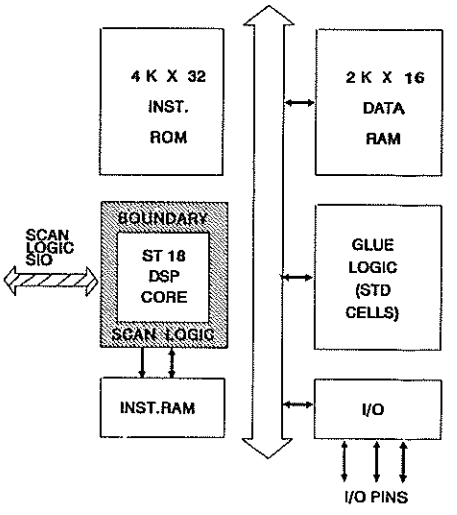


FIG.4 - TYPICAL DSP BASED ASIC

4. TESTABILITY ISSUES

The usual way to validate a standard DSP (or microprocessor) is to build an"emulation chip" which allows to emulate the future application where the software will be cast in silicon (firmware) and hence hidden in the package (instruction bus not available) at full speed and without any difference. For ASIC design the problem needs to be addressed by the customer who will design his own emulation chip, with a higher pin count package to access external program memory in order to be able to debug the software in real time environment. However, this problem has also been addressed by the supplier of the ST18932 which provides a powerful boundary scan logic which surrounds the DSP core.
All the accesses of the DSP megacell (buses, clocks,I/O) are "scanned".
This allows the user to record all these signals and to scan them "out" of the chip serially using a few extra pins.
Alternatively the user may scan in, serially again , any bit configuration allowing, for instance, to write the instruction register or to set up a breakpoint address.
Several emulation modes are available to ease the validation of the final chip and even to find out possible remaining problem.
For instance a step by step mode allows the user to look at the state of any internal register

and find precisely where, when and why a fault occurs.

5. FUTURE TRENDS

Today, as was detailed in paragraph 2, different technologies support different functions of the radio sub-system. However we can foresee a need to integrate the complete subsystem on one chip, hence using a single technology.
It appears that a bi CMOS technology is a good candidate to become the "wideband" technology that will allow to integrate together a 20 ns instruction cycle DSP, several analogue converters sampling signals at up to 2 Mhz with 14 bit resolution and 2 GHz frequency range radio transmit and receive filters and modulators by 1995 time frame.

At that time PCN (Personal Communication Network) and DECT (Digital European Cordless Telephone or CT3) will be in operation in the 1.8 MHz band and the bi-CMOS technology must also cover this range.
It also shows the way to go towards increased integration. By the end of the millenium, chips including about 50 millions transistors will be hardly standard. As complete systems will be integrated on a single chip, equipment manufacturers will need to be able to design their own proprietary solutions at reasonable cost. There is no other way than to provide them with powerful design tools together with rich libraries including complex Megacells.
Most important, sound methodology and consistent tools and cells must also be provided to ease validation and test.
From a strategic point of view this anticipated evolution shows that equipment manufacturers will face critical choices : they cannot go "shopping" Megacell libraries with the inherent associated high level of service required as they do for commodities or memories. They must select a "preferred" semiconductor ASIC supplier on the following criteria : perenniality, technologies and products range and ASIC offer (Megacells library and service level).

6. CONCLUSION

This paper has shown that all functions of a GSM mobile station can be supported with a few distinct chips.
Three available technologies support these chips : a.8 um CMOS digital C-MOS technology for uPs and DSPS, a 1.2 um analogue and digital C-MOS technology (such as ST's HCMOS3A) and a bi-CMOS technology with a FT of 12 MHz for the radio frequency analogue front End.
The wide european acceptance of the GSM recommandation makes a huge potential market reachable in 1991 and put strong pressure on equipment suppliers to reduce cost,size and power consumption to a minimum.
There is no other way for GSM suppliers to meet that challenge than the ASIC route. Fortunately, powerful CAD tools are here and semiconductor suppliers, as SGS-Thomson Microelectronics, provide the necessary libraries with appropriate megacells such as DSP core and uP core.

REFERENCES

(1) GSM Recommendation - ETSI

(2) G. GHILLEBAERT, P. COMBESCURE, A. MALOBERTI "Le système cellulaire numérique européen de communication avec les mobiles" - l'Echo des Recherches - n° 131 - 1er trimestre 88 - pp 5-16

(3) ST18930/1 - DATA SHEET - SGS-THOMSON MICROELECTRONICS.

(4) ST18932 - DATA SHEET - SGS-THOMSON MICROELECTRONICS

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

ON THE PARALLELISM IN SPEECH RECOGNITION

S. Alexandres, J. Morán, J. Carazo, A. Santos

Dpto. Ingeniería Electrónica. E.T.S.I.Telecomunicación (U.P.M.)
Ciudad Universitaria, 28040. Madrid (Spain)

ABSTRACT

This paper presents a multiprocessor modular architecture that achieves real-time performance on isolated-word recognition using large-vocabularies with a class of hidden Markov model algorithms. The system implements a methodology that deals efficiently with the concurrency and parallelism inherent in the application. Different parallel architectures are examined and their parameters are optimized for this task.

1.INTRODUCTION

The potential of parallel processing nowadays offers improved performance in most investigation areas. Speech processing has been a research issue with a large growth in last decade where conventional serial computers present several difficulties in order to obtain quick response. Multiprocessor structures have been successfully used to increase the performance requiered by applications with large amount of computations and real-time needs. This paper describes the implementation of an isolated word recognition algorithm based on hidden Markov models in a multiprocessor architecture.

Several examples of real-time speech recognition systems using multiprocessor architectures have been recently published. Among them, we could mention: the BEAM system [1] developed by Carnegie Mellon Univ. (CMU) that uses commercial processors, and SRI system [2] with custom circuits. Both of them work on HMM continuous speech speaker-independent recognition and medium vocabularies. Another system to be mentioned is Olivetti's [3] that uses custom and general purpose processors and works on isolated word speaker-dependent and large vocabularies (around 10,000 in a new version 60,000 words) with Dynamic Time Warping algorithm.

The use of general purpose processors with good capabilities for parallel processing like Transputers provides efficient systems for speech recognition. Our implementation is based on an array of INMOS T800 Transputers and three different configurations will be described showing their possibilities and performance. Two principal advantages of the use of processors like Transputers are analyzed: firstly the architecture is modular and scalable (the number of processor elements can be increased as the computation increases); and second, minimum software changes are requiered when the number of processors is changed. In this way different strategies to decompose and partition the speech recognition algorithms will be described.

2.RECOGNITION ALGORITHM

The use for HMM's for speech recognition can be represented as a time-varying sequence of spectral events (10-100 msecs. stationary) and this can be computed by a statical distribution over a observation sequence $O=(O_1,...,O_T)$. The probability computation can be performed using a Viterbi search algorithm to compute the maximum likelihood path, that is $P(O|S^v)$ for the each word v in the vocabulary, or $1 \leq v \leq V$ for V word vocabulary. Iniatilly we used a Spanish vocabulary [3] of 1,000 words, being each word a concatenation of allophone models (45 models that include the allophones plus two silence models). Each model has three states, in this way on average the words in the vocabulary have around 8 allophones (30 states including the silence models). The preprocessing is performed by a digital signal processor (DSP) that implements a vector quantization algorithm so that the observation vector is represented by 11 coefficients. This feature analysis or front-end generates an 8-bit vector that represents the reference sound in the codebook. In order to do isolated word recognition, for each unknown word to be recognized the likelihoods for all possible models are calculated, $v^* = MAX\ [\ P(O|S^v)\ ]$ for $1 \leq v \leq V$. fig 1. Finally the word with maximum likelihood is selected.

In order to reduce the computational load, pruning techniques have been added: nonpromising alternatives are discarded reducing the search space.

The following sections deal with the real-time computation, and the implementation of these algorithms in a pipelined parallel architecture. In order to support the computational load, all the states are updated by word model every frame (6.25 msec). This task is carried out using a software method that balances the load and reduces the

interprocessor communications in the architecture.

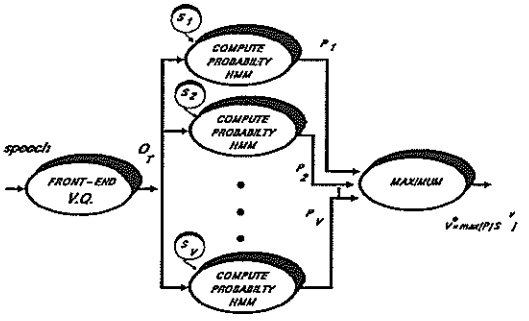


Figure 1. Basic word recognition system based on HMM's.

3.OVERVIEW OF ARCHITECTURE

The architecture is based on Transputer INMOS T800. These processors are able to execute 10 MIPS and have local memory with four links to communicate with the neighbour processors; the internal hardware is a 32-bit path with special features for concurrency; they have a floating point unit and use message passing methods through fast serial links. The architecture is programmed in Occam and it is a peripheral of a host processor that serves as an interface with the rest of the world and provides data and programs for bootstrapping and downloading.

To distribute the task on the system several support routines are needed. The basic idea is to distribute efficiently the word models among the processors so the system performance is increased as much as possible. So that the algorithm is partitioned into modules, each of them is broadcasted to the processor elements (PE); every PE contains in its local memory the data structures required to process the modules. The communication is performed passing messages through the architecture, and no shared memory is needed. The goal is to partition the task in such a way that allows to balances the load and minimize communications.

3.1 *System Communication Scheme*

The principal bottleneck for real-time computation on recognition algorithms is the memory bandwith (the high rate access to data memory) and the system response time. In our architecture the processor elements have their private memory to reduce memory bandwith requirements and contention for each PE, translating the principal problem to the interprocessor communications over the architecture. The basic approach is to distribute word models among the *N* processors so that they all work simultaneously on their own models. The communication environment is a structures called a farm model. A farm model is a system with a central processor (a host) and a pool of interconnected processors in a network with different topologies. The host sends an information packet in a fixed format that flows over the network. Each PE has a resident communication software, or communication internal control, to deal with this flow of data. If the PE is busy (the processor is computing another information packet at the moment) the communication kernel passes the data to the next processor. Otherwise the kernel would not pass the data and this processor would deal with them. When the computation is finished the output data is passed through the network back to the host (figure 2). The main characteristics for this scheme are the following ones:

1.- It is completely modular so that the network can be scaled up without any penalty.
2.- The architecture is fully regular and every PE executes the same program (on the different data).
3.- The software remains unchanged (or minimum changes are needed) as the computation increases or the number of PE changes.

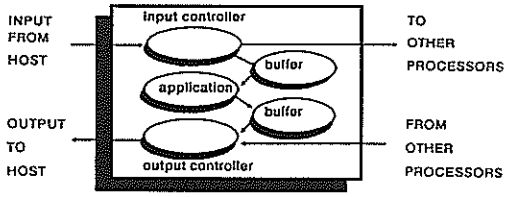


Figure 2. Processor communication environment (Farm model).

The model can be evaluated considering the time each processor dedicates to communication tasks and the time actually used on computation. An efficient system should keep the communication to a minimum so that an increase in the number of processors in the network increases linearly the global processing speed. In this scheme the communication cost is small.

For the recognition task the computation requirements grow linearly with the vocabulary size, while the communication bandwith per processor is fixed. We can say that the bottleneck of the system is the communication with the host. This factor gives a maximum efficiency that is independent on the actual number of processors. Once this maximum is reached an increase in the number of processors does not increase the system processing capabilities.

Three different topologies have been evaluated: linear array, tree and ring (figure 3). We predict and measure the performance of the recognition algorithms on these architectures. The diagram 4 shows their efficiency using four processors in terms of speed-up versus communication load. The elapsed time was measured by using the transputer internal clock in high priority (1 μsec. ticks) while observation sequence

O_T was being computed.

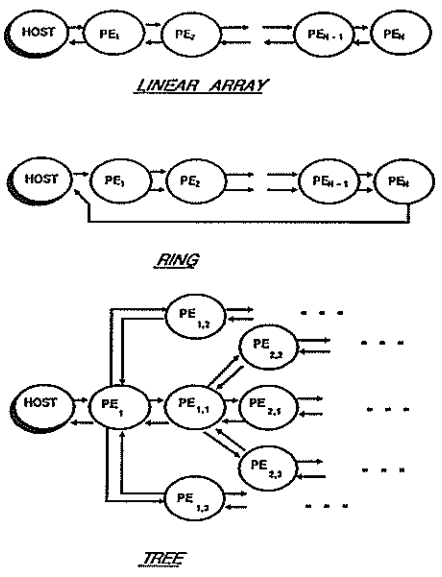


Figure 3. Different architectural topologies implemented.

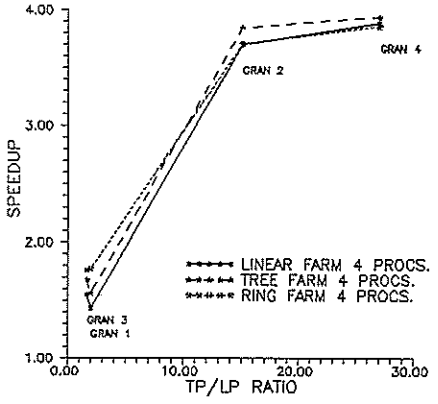


Figure 4. Performance measured

3.2 *Speech Recognition Performance*

The architectural requirements of speech recognition depend on the kind of recognition used. The farm model just described has been used to successfully different topologies. All the implementations have in common that they are distributed memory structures. Every PE has its own copy of the allophone transition and observation probability matrixes (around 150 Kbytes on the present implementation with 45 allophones). Each PE has also in its memory the communication kernel. That means on the whole 256 kbytes of RAM memory per PE. The following process describes the method for recognition computing:

1.- The host receives the input vector sequence.
2.- Initial probability scores for each state word matrix are defined.
3.- The state matrix is broadcasted and new values for all words (Viterbi algorithm) are computed concurrently using the farm model.
4.- The host arranges the data to broadcast the subsequent information to the network. If pruning is used the following operations are also needed:
   4.1 Evaluation step or time synchronization for the active word models.
   4.2 For every active word model a coefficient is obtained.
   4.3 The decision threshold is fixed depending on the best coefficient.
   4.4 The word models below the threshold are discarded.
5.- At the end the host computes the final scores and takes the *MAX*-probabilities.

Using this farm model the information packet is assigned to a free processor that updates and returns it to the host.

The message format for data structures interprocessor communication on the network is a packet, as shown in the figure 5.

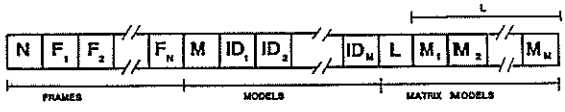


Figure 5. Message format.

The fields in the message format are as follows:

N	:Number of frames to be transmited in the packet.
F_1,F_N	:Index for the codeword vector corresponding to each frame.
M	:Number of index word-model references to be transmited.
ID_1,ID_M	:Index for the word-models to update.
L	:Number of the state matrix entrix to compute.
M_1,M_M	:State matrix to be updated in the processor.

Different packet sizes have been tried to obtain the granularity of the system that gives the best performance (figure 6). In every case, the processor that accepts the packet updates the matrix vector for the information referenced to in the packet. These results have been evaluated in an example with 1,000 word-models and a 200 observation sequence (frames). The best efficiency is shown in column 4.

The performance can be significantly improved if pruning is used. That means that some word models are eliminated during the recognition process. If pruning is used the host sends only the active word-model tags. This pruning algorithm represents a load that can be neglected compared with the processing and communication loads.

GRANULARITY	1	2	3	4
PACKETS SENT	200000	10000	10525	526
FRAMES/PACKET	1	20	1	20
WORD MODEL/PACK	1	1	19	19
PACKET SIZE BYTES (PS)	116	192	1916	1992
PROCESS/PACKET μSEC. (PT)	239	2906	3301	53920
PT/PS RATIO	2.0	15.1	1.7	27.06

Figure 6. Different grain types for recognition task.

The figure 7 gives the improvement in the time needed by four processors compared with the time needed by one single processor for the same recognition task. It also gives the speed-up with the differents topologies. The number of processors needed to obtain real-time is around 19 (without pruning). The use of a prune algorithm has improved the processing speed by a factor between 3 and 4 without any penalty in the recognition itself (the same recognition rate is achieved). The number of processors can then be reduced to one third or one quarter of the original figure.

GRANULARITY	1	2	3	4
TREE	1.56	3.85	1.68	3.94
RING	1.77	3.71	1.76	3.86
LINEAL	1.43	3.70	1.55	3.89

Figure 7. Performance for different architectures (speed-up).

4.CONCLUSIONS

Parallel processing with an architecture based on Transputers and its application to speech recognition has been evaluated. Furthermore, real-time performance viability has been proved with an effective processing environment that allows the development and evaluation of recognition algorithms using different structures and topologies.

Pruning algorithms have proved to reduce significantly the computational load and consequently the number of processors needed. We are presently experimenting with different architecture organizations, a more efficient communication environment and the integration of the system with a front-end, aimed at developing an autonomous speech recognition system.

ACKNOWLEDGMENT

S. Alexandres is supported by a scholarship from the Consejo Nacional de Ciencia y Tecnología (México). The work is partly supported by an Acción Integrada from the U.P.M..

The authors wish to thank J.M.Pardo and H.Hasan (from the Speech Technology Group in this Department) for their generous help and for providing us with the application database, and J.Meneses and E.Muñoz for their support.

REFERENCES

[1] R.Bisiani, T.Anantharaman and L.Butcher. *BEAM: An Accelerator for Speech Recognition.* in Proc. IEEE ICASSP-89. pp. 782-784. May. 89.

[2] H.Murveit, J.Mankoski, J.Rabaey, R.Brodersen, T.Stoelzle, D.Chen, S.Narayanaswamy, R.Yu, P.Schrupp, R.Schwartz, A.Santos. *A Large-Vocabulary Real-Time Continuous-Speech Recognition System.* in Proc. IEEE ICASSP-89. pp. 779-783. May. 89.

[3] P.Buttafava, R.Billi, W.Digiampietro, G.Massia, V.Vittorelli. *Architecture and Implementation of the Olivetti PC-based Very Large Vocabulary Isolated Word Recognition System.* Eurospeech 89. Sep. 89.

[4] J.M.Pardo, H.Hassan. *Large vocabulary speaker independent isolated word speech recognition using hidden Markov models: Status Report and Planned Research.* Eurospeech 89. Sep. 89.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A PERSONAL COMPUTER BASED CONTINUOUS SPEECH RECOGNIZER FOR LARGE VOCABULARY APPLICATIONS (*)

Alberto CIARAMELLA, Davide CLEMENTINO, Roberto PACIFICI

CSELT - Via G. Reiss Romoli 274 - 10148 Torino (Italy)

We implemented a PC housed continuous speech recognition system for a thousand words vocabulary, which extracts from the input utterance a lattice of hypothesized words: this lattice can feed an understanding stage implemented on a separate workstation.

The PC housed recognition system is an adaptation of a previous VME based recognition system; it is based on commercial DSP PC boards using the TMS320C25, but we added it a local intelligent extension memory for storing Discrete Hidden Markov Models emission matrix.

Possible limitations due to the PC cabinet have been counterbalanced by the DSP technology improvements: we had however to change a little the system control philosophy, since in the PC based implementationall the DSP boards are bus slaves, while in the VME case all the DSP boards could behave as master also.

Notwithstanding the reduced size and cost, this PC based implementation is at least as satisfactory in accuracy and speed than the previous more costly VME based implementation.

1. INTRODUCTION

We implemented a speaker-dependent continuous speech recognition system for a thousand words vocabulary [1]: it extracts from the input utterance a lattice of hypothesized words and sends it to an understanding stage implemented on a separate workstation.

This recognition system is easily adaptable both to a new application, by changing the vocabulary, and to a new speaker: in this case speaker training isperformed by an application independent sequence of words, which contains all language diphones in a statistically significant number of cases.

The system in fact is based on discrete densities hidden Markov diphones models and its general architecture (Fig 1) can be distinguished into a synchronous and an asynchronous section: the synchronous section at each time frame (i.e. each 10 ms) from the input utterance computes DCT and energy based codevectors and writes them into a circular buffer. The following section performs dynamic programming on the tree of diphones describing the application vocabulary: this computation is asynchronous since its duration not only depends on the specific input utterance, but changes from frame to frame in the same utterance. The asynchronous section is split into two levels both in the algorithm and in the data structure [4]: a high level, which implements dynamic programming on the whole diphone tree, and a low level, which speeds up dynamicprogramming on specific diphones.

A first implementation of this recognition system, produced for the P.26 Esprit project and terminated in 1988 within an Italian geography questions answering demo, was housed in a VME bus cabinet and used 3 custom DSP boards based on the TMS32020 and a 68020 uP as the system master [1,2].

We have now reimplemented this recognition system in a more compact Personal Computer environment, updating also the DSPs used, which are now the TMS320C25.

The main differences between the VME and PC bus are that the VME bus is an order of magnitude faster than the PC bus and allows a quite larger memory addressing space (16 Mbytes versus 1 Mbytes): this last difference is even more important, since, whilst in our VME based recognizer we could implement a global distributed memory scheme, in the PC based we have been forced to use as much as possible local memories for data structures, which is of course less efficient [5]. Besides in our VME based system all the computational boards could become bus masters, whilst in the PC based system all DSP boards are slave only: hence the last one has a simpler architecture, but lost some performance and generality.

Of course we faced some problems in downsizing our system to the PC environment, but this way we obtained a more transportable and cheaper system , and besides we reduced the level of customization of the system: in fact in the PC environment we need now only a custom memory board, whilst all the DSP section and the acquisition part is now implemented usingstandard vendor boards (**).

The technology evolution however compensated in some sort the reduction of architectural features since we use now the TMS320C25 , which is twice faster than the previously used TMS32020.

(*) this work has been partially supported by the ESPRIT SUNDIAL project.

(**) AU-21 boards, produced by OROS.

2. HARDWARE DESCRIPTION

Fig. 2 presents the hardware block diagram of the recognizer: to the PC/AT bus we connect:

- a DSP board for feature extraction (DSP1),
- one DSP (DSP2) boards for dynamic

programming; each of these DSP is interfaced to a local extension memory.

The DSP board is a vendor provided board (**), equipped with a TMS320C25, 64 Kwords of program and data memory, 1 A/D and 1 D/A (not used in this implementation) ; it is seen by the PC/AT bus through 8 registers only: some of these are used as command and status words, whilst other registers contain the data and the address in the TMS320C25 space usedfor reading or writing from the PC/AT bus: this is in fact not only a slave board for the PC/AT bus, but requires a sequence of instructions for accessing the DSP memory from the bus.

The DSP board admits an external extension of the i/o lines, which specifies the 16 bits data, the i/o address (one of 12 possible) and the read or write command: we used this connection for interfacing a private memory expansion, which is required for storing the spectral emission matrix B of discrete hidden Markov models.

The dimensions of this matrix are K * S, let K be the dimensions of the codebook used and S the number of states; S is about three times the number of diphones, since each diphone has 3 states in average in the representation we used [4]. Just to give an idea, we have K=256 and S=1024 as typical values, and in this case the matrix B contains 256K words: these constitute by far the larger section in the data and program areas of the DSP implementation of dynamic programming, and besidescan not fit in the whole data area of the TMS320C25, whose maximum extension is of 64 Kwords .

We sized the extension memory to 1 Mwords for allowing possible future extensions also; in this case 16 bits of data and 20 bits of address have to be multiplexed through the 16 bits of data provided by the i/o connector: hence we implemented these primary functions, which are differentiated by the i/o address:

- write 10 most significant bits of address,
- write 10 less significant bits of address,
- read or write data in the preselected address

This provides minimal functionality, but it is not too efficient, since for reading or writing we will need three i/o operations, the first two for defining the address and the third for really performing the data transfer: since in the TMS320C25 architecture an i/o instruction lasts twice than an ordinary one, the access to the external memory is 6 times the access to the internal one.

Hence we added i/o functions which read or write data in thepreselected address and increment it: in this way we reduced to 1/3 the read or write times for a sequence of consecutive accesses to the extension memory; we will see in the following the B matrix really is accessed in this way.

3. FIRMWARE IMPLEMENTATION

3.1. Generalities

The DSP boards used are provided by a library of high level (Turbo Pascal or C) callable driver functions in the PC plus a kernel in the DSP board which together implement file and memory transfers from/to the DSP board and allow the activation of different DSP functions from the master, providing also synchronizations at the begin and at the end of the DSP functions.

Hence, in adapting our previous VME based implementation, we restructured the firmware taking also into account the resident vendor provided kernel. The DSP firmware is now structured into 3 levels:

- a resident vendor provided kernel, with a general purpose interface to DSP functions,
- a DSP control program, providing specific interface to the implemented algorithm,
- specific subroutines called by the DSP control program, as for example the FFT.

3.2. Synchronous section

The synchronous section is implemented by a first DSP board (DSP1),which each 10 ms performs features extraction , from 12 KHz. sampling of the input speech to the FFT computation, band grouping, DCT computation of logarithms of band grouping, DCT vector quantization and energy scalar quantization ; for these computations we used the same assembler optimized routines [3] already implemented for the previous VME system .

The control program has been restructured and generalized in such a way that by suitably loading data areas of the TMS320C25 it is possible to change algorithm related and control related run time conditions. Algorithm related conditions are:

- the A/D converter gain,
- the initial and final rows in the FFT spectrum,
- the initial and final energy bands,
- the number of codevectors and the number of parameters in the codebook,
- the codebook itself.

By changing these initialization constants, it is easy to adapt the DSP application program to a new channel and to a new speaker.

Control related functions are even more general: it is possible in fact to configurate the DSP for performing feature extraction forever or for a definite number of frames, and to define the input and output enabled channels, with their specific allocation in the DSP space.

Computations in the same frame of the DSP1 are pipelined and results from one subroutine to the following are exchanged through the fast data memory allocated inside the TMS320C25 chip, which is used as a common area.

Different calls to the input and output routine are inserted between different computations; each call is characterized by three parameters, i.e. the input or output identifier, the first address in the common DSP area to read or write and the number of words.

Hence by suitably configuring an input and output control area it is possible to enable different inputs and outputs and to define for each the initial and final address of the corresponding circular buffer in the DSP area .

This i/o virtualization has been found quite effective both in debugging subsections of the application program, both in the configuration of the DSP in different applications: in the training phase in fact only DCTs are required, whilst in the recognition phase only codevectors are extracted.

An overhead of course arises by using slave boards with data areas not directly mapped to the PC bus , since in this case a double transfer of parameters is required: from the common area to the output buffer inside the DSP board first, under the control of the DSP program, then from this area to a master CPU buffer under the control of the master CPU.

A double transfer of data is required also at the initialization phase, in which the DSP control parameters are first written in a PC table and then transferred in DSP board using library functions .

When the DSP starts , and this is done using a specific library function, it begins the computation and at every frame it dowloads the results in the buffer specified at the initialization , and then increases the "frame counter" which is readable by PC.

The DSP program terminates or because the PC writes a stop configuration in the "status word", or because the "frame counter" matches the maximum number of frame preprogramed by the PC in the initialization phase : in everycase the DSP notifies to the PC its state via the "status word" and also by returning a particular function value .

It is hence possible to use this board in two different ways.

The first is simpler (we named it "cascaded"): the DSP1 starts and stops under PC control and at the end the PC reads the buffer of results by using the function library ; this implementation is used in the training phase.

The second mode (we named it "overlapped") is used during the recognition task :in this case when the PC needs new datas,having processed old frames, it reads the actual "frame counter" value and compares it with old value : if it is greater the PC can read the circular buffer results of new frames , instead, if the "frame counter" is equal to the old value , the PC has nothing to do and can wait until the "frame counter" has been incremented by the DSP1.

3.3. Asynchronous section

The recognition module has been implemented in a master-slave architecture,according to the previous VME system [4], the master being the task running on the PC and the slave the program on the DSP2 board.

The low level is charged with the more computational demanding, but more regular in program control and in memory accesses, recognition of the subwords selected by the high level which is less computational demanding, though less regular.

The master processes a precompiled subwords tree describing the whole vocabulary, and sends messages ,named "pushes", to the slave for starting the recognition of new subwords.

On a frame basis, the slave receives from the master the "pushes" and the observed emission symbols , performs the dynamic programming on all active subwords, updates the active subwords list by discarding ones for which all the state probabilities are below the best path by a given threshold value and gives back messages to the master (named "pops"),notifying the possible end of a subword, that will determine the messages generating by the master the next frame.

The recognition works in parallel with the sentence utterance and at the end of the utterance the master outputs the lattice of most likely words.

The slave firmware is organized as already implemented in the previous VME system [4] with the supervision of the resident kernel provided with the DSP board. Moreover all data structures used by the slave are allocated into on-board memory while in the previous system most data structures were accessed off-board through the VME and VMX busses: in this way now the memory is accessed at a faster speed.

The spectral emission matrix B, containing for each state of each subword the emission probability for codebook symbols, is allocated into the extension memory described above : low and high part of its address have been used as two indexes of the matrix.

The first index is the given codebook symbol and it will be set only once at the beginning of each frame;the second one is the current state and it will be set at the begin of each subword.Since dynamic programming on an active subword is performed in vector form for all states from the first to the last active , the autoincrement address capability is appropriate for speeding memory accesses.

It has to be pointed out that some speed is gained by processing push messages with a double buffer mechanism , in such a way that while the slave is performing dynamic programming for the subwords of frame n in the first buffer, the master can generate in the second buffer the messages for frame n+1 whenever a new pop message is received back from the slave. Relative improvements of recognition time will be outlined in the next paragraph together to the system speed-up results obtained also by using more DSP boards as slaves .

In fact, the master-slave architecture makes the recognition system flexible and expandable: the computational load of dynamic programming can be uniformly distributed among several DSPs working in parallel, and also larger vocabularies could be supported because by using DHMM of subwords the model memory size is independent from the vocabulary size.

4. SYSTEM CHARACTERIZATION

In this system PC based we have the same results of the VME based system [2] in terms of accuracy. The AT bus is well suited for this system, in fact master and slave tasks exchange 900 messages as an average for each frame : each message contains 4 words hence the load of the bus is about 720 KBytes/ sec.

Our system has been characterized by using two 386 based PC: one of this has a 20 MHz CPU and the second has a 33 MHz CPU; the results obtained with a standard beam search threshold (20) are shown in Table1; the utterance duration has been considered of net speech without starting and ending silence.

	CPU 20 MHz one DSP2		CPU 33 MHz one DSP2	
	single buffer	double buffer	single buffer	double buffer
Avg. speech utterance dur. [sec]	3.14		3.14	
Avg. recognition time [sec]	17.5	15	11.7	9.2
Recognition / Utterance time	5.5	4.8	3.7	2.9

TABLE 1 – Throughput results of PC based continuous speech recognition system on 100 phrases of words into 1008 words vocabulary

The final throughput results, seems adequate since the user perceives the system is nearly real time; in fact the recognition works also during the speech utterance and the speaker adds to the net utterance time the starting and ending silences.

5. CONCLUSIONS AND FUTURE DIRECTIONS

The new PC based implementation of our continuous speech recognition is space and cost effective without degradation of accuracy and speed; moreover there are some spare computational resources, especially in DSP1 , which can be used for further system improvements.

We are in fact now updating this system with delta cepstra extraction for a multicodebook implementation, capable to demonstrate a speaker independent telephone input continuous speech task with a thousand words vocabulary, using of course speaker independent hidden Markov models.

REFERENCES

1] G. Perucca, T. de Couasnon, S. Giorcelli, E. Hirsh, H. Mangold
Advanced Algorithms and Architecture for Speech and Image Processing 5th annual Esprit Conference
(Brussel, november 1988), pp. 543–561

2] A. Ciaramella, D. Clementino, R. Pacifici
Characterization of a Large Vocabulary Isolated Words and Continuous Speech Recognizer.
European Conference on Speech Communication and Technology
Paris, 26–28 september 1989, pp. 437–440

3] A. Ciaramella, G. Venuti
Vector Quantization Firmware for an Acoustical Front–End Using the TMS32020
ICASSP–87 (Dallas, april 1987), pp. 44.1.1–4

4] A. Ciaramella, G. Venuti
Dynamic Programming with Hidden Markov Models on a TMS32020 Digital Signal Processor
EUSIPCO–88, Grenoble 5–8 september 1988, pp. 751–754

5] M. Aimone Marsan, G. Balbo, G. Conte
Performance Models of Multiprocessor System
the MIT Press,1986

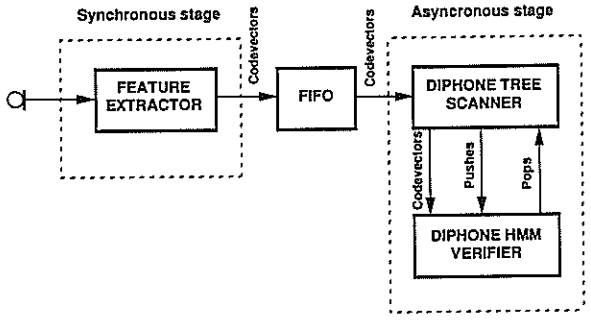


Fig. 1 – System interplay

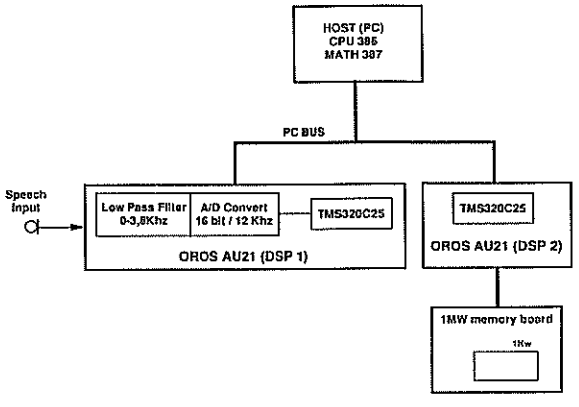


Fig. 2 – System block diagram

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

An Interactive Adaptive Digital Filter Software for Multichannel Signal

Hassane MIMOUN and Michel CIAZYNSKI
I.S.E.P, 21 rue d'Assas F-75006 Paris
Maurice BELLANGER
T.R.T, 5 Av. Réaumur, F- 92350 Plessis Robinson

Abstract:
An interactive software is presented for the simulation of adaptive digital filters with multichannel input signals. The transversal structure is considered with the Least Mean Squares (LMS) and Fast Least Squares (FLS) algorithms. For both algorithms, the design options are specified interactively by the user. The output error and the coefficients are available as the simulation results.
The software presented can be included into the MONARCH system, which is a PC software environment for Digital Signal Processing. It can be exploited for the design of adaptive systems. It can also serve as a teaching aid to illustrate a course.

I - INTRODUCTION:

The case of multichannel input signals is often encountered in applications of adaptive digital filters. As examples, multisensor systems like adaptive antenna arrays and pole-zero modelling can be mentioned [1]. For the practitioner, the design task becomes more complex and a simulation tool can be very helpful to select the parameter values as well as the type of algorithm and assess the performance.
A software module has been designed to cope with the case of multichannel or multidimensional input signals and simulate adaptive filters in transversal structure and using gradient and least squares algorithms. The methodology used in this interactive software, called ADFMD (Adaptive Digital Filter with MultiDimensional inputs) is presented through a description of the options.

II - The LMS options in the ADFMD software:

The principle of an adaptive filter in transversal structure with K inputs is shown in figure.1. The output error at time n+1 is given by:

$$e(n+1) = y(n+1) - \sum_{i=1}^{K} H_i^t(n) X_i(n+1)$$

Where:

$$H_i^t(n) = [\, h_{0i}(n),\ h_{1i}(n),\ .....,h_{Ni}(n)\ ]$$

is the N_i dimensional coefficient vector at time n. The N_i most recent data samples for input with index i are denoted:

$$X_i(n+1) = \begin{bmatrix} x_i(n+1) \\ ..... \\ ..... \\ x_i(n+2-N_i) \end{bmatrix}$$

Following the least mean squares (LMS) or gradient algorithm, the coefficients are updated by:

$$H_i(n+1) = H_i(n) + \delta_i e(n+1) X_i(n+1)$$

Where δ_i is the adaptation step size in subfilter i.

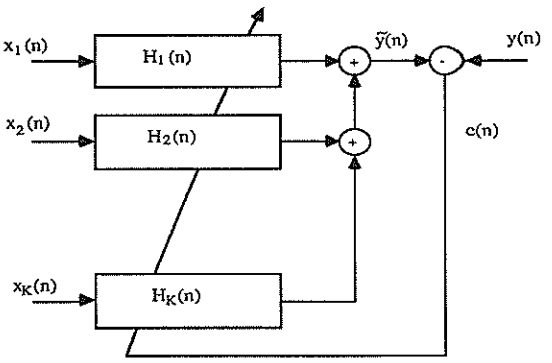


Fig.1.- Principles of a transversal adaptive filter with K inputs.

With this algorithm, the system can in fact be viewed as a set of K different filters, which coefficients are updated

using the same error sequence. The correlation between inputs is not taken into account with that approach, which can severely limit the performance.

The inputs to the ADFMD software are shown in table.1 for the LMS type of algorithm and for the filter named TEST. If a measurement noise is added to the reference signal, the signal-to-noise ratio expressed in decibels has to be provided.

The sofware computes the error signal and the coefficients for the number of iterations indicated. The noise, reference and input signals must be made available in separate files. The simulation results, error and coefficient sequences, are stored in files named TEST.ERR and TESTi.COF respectively. For the subfilter $H_1(n)$ the coefficient index is taken in the range [0,9]; for $H_2(n)$ it is taken in the range [10,19] and so on.

MULTIDIMENSIONAL ADAPTIVE DIGITAL FILTERS	
Filter File : TEST.MD	
Algorithm Type : LMS	Fixed Point Simulation : NO
Number of Inputs (1...10): 2	Add Noise to Ref. File : YES
Number of Iterations : 1001	Signal-to-Noise-Ratio (dB) : 20
Noise File : NOISE4	
Ref. File : REF11	
Input Files	Number of Coefficients
Input 1 : NOISE1	Filter 1 : 3
Input 2 : NOISE3	Filter 2 : 3

Table.1.- Inputs to the ADFMD software for the LMS algorithm.

The number of coefficients in each subfilter is limited to 10, and the number of inputs is also limited to K = 10. But. only 10 coefficients can be stored altogether and the selection is made by illuminating the relevant points in the matrix shown in table.2. This limitation might look rather restrictive, but if one wants to examine the evolution of more than 10 coefficients, the program can be run again with a different selection.

COEFFICIENTS TO SAVE	
Filter	Coefficients
1	0 1 2
2	0 1 2
CONTINUE	

Table.2.- The selection of the stored coefficients.

From a user's point of view, the ADFMD software has to be associated with a friendly environment, able to provide desired signal files and appropriate means to visualize the results. Such an environment is provided by the MONARCH system wich offers a module called SIGLAB to generate synthetic signals or prepare natural signals like speech and image files. The simulations described in a later section use the MONARCH system [2,3].

The performance of the adaptive filter critically depends on the adaptation step size. Here a step size value δ_i is chosen for every subfilter ($1 <= i <= K$). A stability limit is provided to guide the user's choice; in accordance with conventional results it is taken as:

$$\delta_i < 2 / N_i \sigma_{xi}^2$$

Where σ_{xi}^2 is the measured power of the input signal with index i.

A fixed point simulation option is also offered. The procedure is as follows: the maximum magnitude is detected in the files of the floating point variables and it is used to determine the minimum number of integer bits needed in the fixed point representation. The user first specifies the total number of bits and then, in view of the upper limit (total number of bits minus the minimum number of integer bits and minus the sign bit) he gives the number of fractional bits. Thus, he can control the scaling of the variables.The error and coefficient sequences obtained in fixed-point simulation are stored in files with the names TESTQ.ERR and TESTQi.COF respectively.

III - The FLS option in the ADFMD software:

If a least squares algorithm is used, the adaptive filter output error is written as:

$$e(n+1) = y(n+1) - H^t(n)X(n+1)$$

and now, $H^t(n)$ denotes the KN dimensional coefficient vector:

$$H^t(n) = [\, h_{10}(n), \ldots, h_{K0}(n), h_{11}(n), \ldots\ldots, h_{KN}(n) \,]$$

assuming all subfilters have the same order N.

The KN most recent data samples are arranged in the vector X(n+1) as follows:

$$X(n+1) = \begin{bmatrix} x_1(n+1) \\ \cdot\cdot \\ x_K(n+1) \\ x_1(n) \\ \cdot\cdot \\ x_K(n+2-N) \end{bmatrix} = \begin{bmatrix} X\,(n+1) \\ \cdot\cdot\cdot\cdot \\ X\,(n) \\ \cdot\cdot\cdot\cdot \\ X\,(n+2-N) \end{bmatrix}$$

According to least squares algorithm, the coefficients are updated by:

$$H(n+1) = H(n) + G_K(n+1)e(n+1)$$

The adaptation gain $G_K(n+1)$ is defined by:

$$G_K(n+1) = R_{KN}^{-1}(n+1)X(n+1)$$

Where:

$$R_{KN}(n) = \sum_{p=1}^{n} W^{n-p} X(p)X^t(p)$$

is the input cross-correlation matrix estimation. The adaptation gain vector is updated using forward and backward linear prediction, through the following set of equations, wich corresponds to the fast least squares (FLS) algorithm:

$$e_{Ka}(n+1) = X(n+1) - A_K^t(n)X(n)$$

$$A_K(n+1) = A_K(n) + G_K(n)e_{Ka}^t(n+1)$$

$$\varepsilon_{Ka}(n+1) = X(n+1) - A_K^t(n+1)X(n)$$

$$E_{Ka}(n+1) = WE_{Ka}(n) + e_{Ka}(n+1)\varepsilon_{Ka}^t(n+1)$$

$$G_{K1}(n+1) = \begin{bmatrix} 0 \\ G_K(n+1) \end{bmatrix} + \begin{bmatrix} 1_K \\ -A_K(n+1) \end{bmatrix} E_{Ka}^{-1}(n+1)\ \varepsilon_{Ka}(n+1) = \begin{bmatrix} M_K(n+1) \\ m_K(n+1) \end{bmatrix}$$

$$e_{Kb}(n+1) = X(n+1-N) - B_K^t(n)X(n+1)$$

$$G_{K1}(n+1) = \frac{1}{1 - e_{Kb}^t(n+1)m_K(n+1)}(M_K(n+1)+B_K(n)m_K(n+1))$$

$$B_K(n+1) = B_K(n) + G_K(n+1)e_{Kb}^t(n+1)$$

The initial prediction error energies control the initial speed of adaptation. Typical values are smaller or equal to the powers of the input signals.

The weighting factor is related to the filter observation time window T by T = 1/(1 - W). A value of W too small leads to numerical instability. Extending the bound known for the 1-D signal case, a lower bound is taken as:

$$\frac{1}{N + 4(2N)^{1/2}} < W <= 1$$

Finally, the inputs for the FLS algorithm are shown in table.3.

MULTIDIMENSIONAL ADAPTIVE DIGITAL FILTERS

Filter File : FLS.MD
Algorithm Type : FLS
Number of Inputs (1...10): 2
Number of Iterations : 501
Filters Order (1..10) : 2
Fixed Point Simulation : YES
Add Noise to Ref. File : YES
Signal-to-Noise-Ratio (dB) : 20

Noise File : NOISE4
Ref. File : REF11
Input Files
Input 1 : NOISE1
Input 2 : NOISE3

Weighting Factor > 0.90000 = 0.99

PREDICTION ENERGY SPECIFICATION

Filter	Max.In.Power	Initial Energy
1	< 1.02725	1.02725
2	< 0.9481	0.9481

CONTINUE

FIXED POINT SIMULATION

Wordlength in Bits [<= 32] = 16
Fractional Bits [<= 13] = 10

CONTINUE

Table.3.- Inputs to the ADFMD software for FLS algorithm.

IV - Simulations

The application of the ADFMD software will be illustrated by several examples.

Let the reference signal be:

$y(n) = x_1(n) + 2x_1(n-1) + 0.5x_2(n) + 1.5x_2(n-1) + b(n)$

where $x_1(n)$, $x_2(n)$ and b(n) are independent white noise sequences. With the following adaptive filter parameters for the LMS algorithm:

- K = 2 inputs : $x_1(n)$, $x_2(n)$
- Subfilter orders : N1 = N2 = 3
- Reference signal-to-noise ratio : 20dB
- Adaptation step sizes : $\delta_1 = \delta_2 = 0.01$

The 6 filter coefficients are given in figure.2 as a function of time. The time constant can experimentally be estimated to be $\tau = 120$, wich is in agreement with the theorical value in that case, namely $1/\delta\sigma_X^2 = 100$.

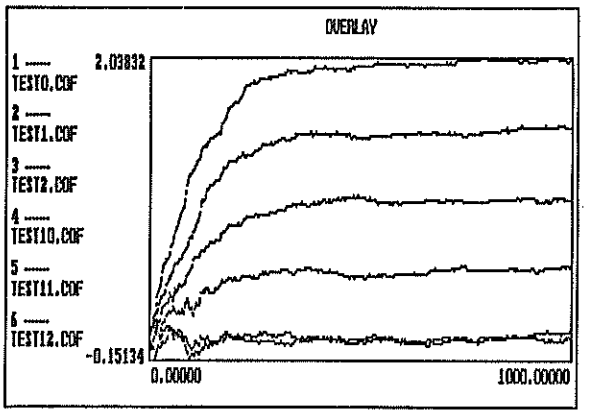


Fig.2.- Filter coefficients as function of time with LMS algorithm.

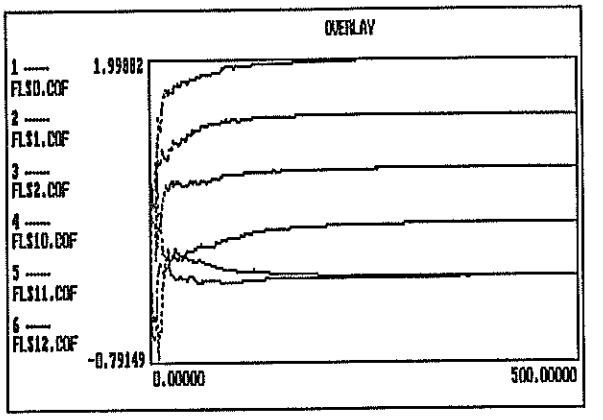


Fig.3.- Filter coefficients as function of with FLS algorithm.

As concerns the estimation of residual output error power E_R for multidimensional input signals, the derivations worked out for the 1-D gradient algorithm [1] can be repeated, leading to the following variance estimation for the coefficients:

$$E\,[\Delta H_i(n)\Delta H_i^t(n)\,] = (1/2)\,\delta_i E_R I_N$$

where:

$$\Delta H_i(n) = H_{iopt} - H_i(n)$$

Now:

$$E_R = E_{min} + \sum_{i=1}^{K} E[\Delta H_i^t(n) X_i(n+1) X_i^t(n+1) \Delta H_i(n)]$$

wich, considering the usual independance assumptions, leads to:

$$E_R = \frac{E_{min}}{1 - (1/2) \sum_{i=1}^{K} \delta_i N_i \sigma_{Xi}^2}$$

With the above parameters, $E_R = 0.0772$ and the value averaged over the time span [500, 1001] turns out to be $E_R = 0.0747$.

With the same signals the FLS algorithm yields the curves given in figure.3. The weighting factors is W = 0.99 and the initial error energies are taken equal to the powers of the input signals.

V - Conclusion:

The ADFMD software can be a valuable help for a system designer, who can experiment with a number of options and accurately determine the major parameters involved in multidimensional adaptive filters. Another application is to illustrate a course, for example by pointing out the correspondences between theorical and practical results.

In order for the software to be run on current PC computers, a number of limitations have been introduced in the options. Indeed, they can be removed with more powerful machines, in order to extend the application field.

References:

[1] M.BELLANGER, "Adaptive Digital Filters and Signal Analysis", Marcel Dekker Inc., New York, 1987.

[2] F.TAYLOR and T.STOURAITIS, "Digital Filter Design software for the IBM-PC", Marcel Dekker Inc., New York, 1987.

[3] The Athena Group Inc., "MONARCH - User's Manual", 3424 N.W. 31st street, GAINSVILLE, FL. 32605, USA, 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

COMPUTER AIDED DESIGN AND REALIZATION OF ROM/ACC DIGITAL FILTER BANK

Ljubomir D. Jovanović, Slobodan T. Jovičić

Department of Telecommunication,
Faculty of Electrical Engineering,
University of Belgrade, 11000 Belgrade, Yugoslavia

This paper is intended to provide a brief insight into design and realization procedure of ROM/ACC Digital Filter Bank with maximal usage of computer support. This automation gives simple, flexible and efficient realization. The particular advantage of proposed procedure is the possibility of easy error analysis in such digital filters and also to reach a compromise between hardware complexity (finite word length) and necessary accuracy (to avoid large and fatigue manual job). After a short description of design method, the software (computer simulation) and hardware realization is presented. In addition, some experimental results and application possibilities in speech analysis and automatic recognition are given.

1. INTRODUCTION

In digital signal processing, especially for short-time spectral analysis, digital filter bank (because of it's advantages: design flexibility in changing the filter characteristics, perfect reproducibility and efficient realization with time sharing the same hardware subsystem) is very attractive solution alternative to analog filter bank.

Digital filter bank had to performs linear or nonlinear frequency decomposition of total band in some number of channels, band-pass (BP) filters, and our basic demands were:

-to work in a real-time conditions (speed problem),
-to enable easily and fast changing of filter characteristics and complete arrangement (flexibility design problem and programmability), and
-to enable computer access to real speech samples (phonems, words, short phrases...) for the next processing steps. Namely, digital bank connected to the computer had to made spectrograms used in Automatic Speech Recognition (ASR) experiments.

Since former demands, we adopted an implementation of ROM/ACC digital filters which A.Peled and B.Liu [5,1] had proposed. In a short, this is a special architecture of digital filters with high speed work without usage of complex and expensive multipliers.

The complete work consists of two main parts: adequate mathematic modelling by means of computer simulation and hardware implementation as practical model confirmation. In that job, computer was of great usefulness.

2. DESIGN PROCEDURE

In opposite to clasical realization of digital filters (with VLSI multipliers) ROM/ACC (or Memory/Accumulator) architecture uses the possibility of exchange arithmetic operation "Multiply" with some number of "Add/Subtract and Shift" operations. This possibility results from corresponding binary notation of input/output samples, i.e. from interpretation of binary word in a bit-by-bit position. This exchange of parallel into specific combinatorial parallel-serial work doesn't work with digital filter coefficients but with memorised table which includes calculated all possible combination multiplicands of input/output samples (in a bit's notation) and coefficients.

From the point of hardware realization the most interesting is cascade implementation of 2-nd order IIR filter cell with its difference equation:

$$Y_n = A_0X_n + A_1X_{n-1} + A_2X_{n-2} - B_1Y_{n-1} - B_2Y_{n-2} \quad (1)$$

where {Xn} is the input sequence, {Yn} is the output sequence and {Ak},{Bk} are coefficients that determine the filter characteristics.

When the data values are scaled so that $|Xn|<1$ and used 2's complement code with B bits accuracy, X can be written as:

$$X = -X^0 + \sum_{j=1}^{B-1} X^j 2^{-j} \qquad (2)$$

where X^j is j-th bit in a binary notation of X. Then, the maximal number is $1-2^{-B+1}$ (0111...1) while the minimal number is -1 (1000...0).

With equation (2), form (1) can be rewritten as:

$$Y_n = -F\left(X_n^0, X_{n-1}^0, X_{n-2}^0, Y_{n-1}^0, Y_{n-2}^0\right) + \sum_{j=1}^{B-1} 2^{-j} F\left(X_n^j, X_{n-1}^j, X_{n-2}^j, Y_{n-1}^j, Y_{n-2}^j\right) \qquad (3)$$

where function F(*) is:

$$F\left(X_n^j, X_{n-1}^j, X_{n-2}^j, Y_{n-1}^j, Y_{n-2}^j\right) = A_0 X_n^j + A_1 X_{n-1}^j + A_2 X_{n-2}^j - B_1 Y_{n-1}^j - B_2 Y_{n-2}^j \qquad (4)$$

Thus, we see that there are 32 possible values of F(*), for the 2-nd order filter cell, and we can in advance all calculated values of F(*) memorise in ROM of 32*B bits dimension. That's the point of ROM/ACC architecture idea. According to (3) with reading some memory addresses (with function arguments as ROM address) and adding, we can calculate one output sample Yn.

Finaly, for the 16-bits accuracy (B=16) equation (3) can be rewritten as:

$$Y_n = 2^{-1}\left\{F\left(X_n^1, X_{n-1}^1, X_{n-2}^1, Y_{n-1}^1, Y_{n-2}^1\right) + \cdots + 2^{-1}\left[F\left(X_n^{14}, X_{n-1}^{14}, X_{n-2}^{14}, Y_{n-1}^{14}, Y_{n-2}^{14}\right) + 2^{-1}F\left(X_n^{15}, X_{n-1}^{15}, X_{n-2}^{15}, Y_{n-1}^{15}, Y_{n-2}^{15}\right)\right]\right\} - F\left(X_n^0, X_{n-1}^0, X_{n-2}^0, Y_{n-1}^0, Y_{n-2}^0\right) \qquad (5)$$

Block diagram of hardware implementation procedure in (5) is shown at Figure 1. Every cycle, in which one output sample Yn is calculating, has 16 subcycles and during each subcycle one memory location is reading from ROM and writing into input buffer while, at the same time, former content of input buffer is adding to the content of ACC buffer (except for j=0 when subtracting is accomplished).

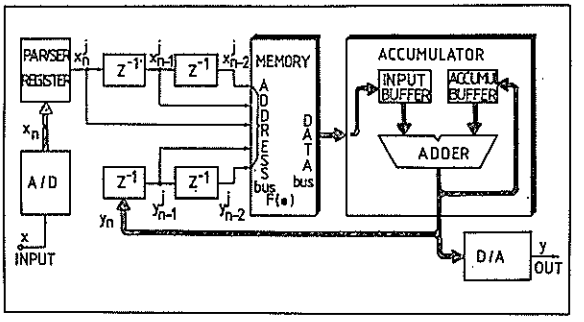


Figure 1.

The ADDER result is returned into ACC buffer with shift 2^(-1) operation. At the beginning of every cycle, ACC buffer should be reseted and Yn converted in a serial sequence again (this time for the next cycles preparation).

3. COMPUTER SIMULATION PROCEDURE

In computer design procedure, there are three programs. The first one (PROGRAM1) computes the recursive IIR digital BP filter, the second one (PROGRAM2) computes Memory contents - function F(*) (in a HEX code for direct EPROM writing), while the third one (PROGRAM3) gives impuls response and Transfer characteristics of BP filter. PROGRAM3 performs simulation of hardware in finite word length B=16 bit accuracy.

All of these three programs give one complete computer system dedicated to the hardware realization especially for simulation, measuring and fault finding works. PROGRAM2 is direct linked to PROGRAM1 (by using coefficients of digital BP filter it can compute EPROM contents) while PROGRAM3 is linked to the PROGRAM2 in order to be able to define B-bits accuracy simulation and very useful and important error analysis. For designing of BP filter it was used Billinear transformation. Transfer function of digital BP filter was determined on the base of starting analog filter specifications [4,9]. In general, the same program can be used for designing of low-pass and high-pass digital filters.

BP filter of 4-th order has two 2-nd order cascaded filter cells, but Function F(*) is normalized with scale factor 2 and 4 for 1-st and 2-nd filter cell, respectively. This fact is necessary because of overflow possibility in the ACCUMULATOR part of the hardware and it has been shown that these two scale factors were satisfactory in all cases of different bank arrangement.

PROGRAM3 calculates impulse response in two ways: by simulation of hardware realization (by means of adequate B-bits model or near D-decimal accuracy) and by the most accuracy (12-decimal) "Direct method" [1]. After extensive computer simulation [6,7] and from the point of eror analysis, limit cycles and complete hardware complexity, as compromise decision we adopted 16-bits accuracy model. By means of that simulation model and calculated Impulse response (in 256 samples) after the FFT we get Amplitude and Phase characteristics for each BP filter. In Impulse response we watched limit cycles while at Amplitude characteristic we've had some definite noise level (about -40dB for each BP filter)

For illustration, Figures 2 and 3 show amplitude characteristics of complete

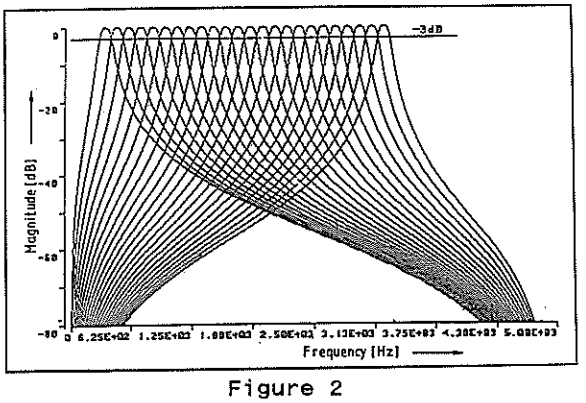


Figure 2

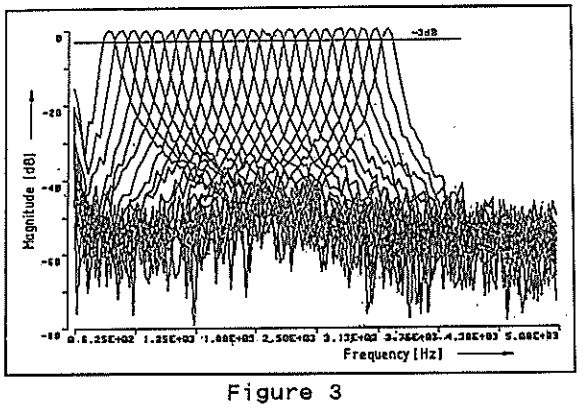


Figure 3

digital filter bank with 24 BP filters, each 125Hz wide and uniform distributed within 300-3300Hz. Figure 2 shows results with the most accuracy (12-decimal) model, while Figure 3 shows simulation results of 16-bits or near 5-decimal accuracy model. The results of hardware simulation show what we can expect from hardware realization especially in IIR filter stability and the level of noise floor.

4. DESCRIPTION OF HARDWARE REALIZATION

We've had realized programmable digital bank with maximal 24 channels (BP filters) inside of 300-3300Hz. Block diagram of that realization is shown at Figure 4.

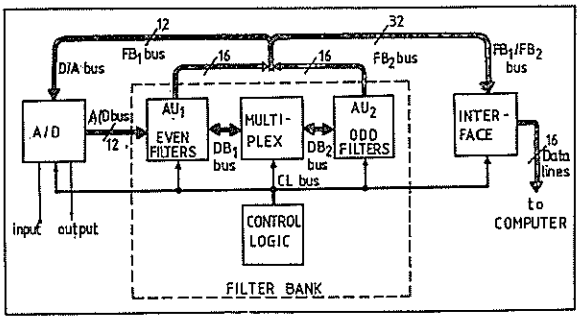


Figure 4

Complete realization works in a real-time and consists of two banks, each with 12 channels (in parallel working). All BP filters are Butterworth type 4-th order with two 2-nd order cascaded cell and overlapping at -3dB point. One Arithmetic Unit (AU) is multiplexed for even order (2,4..24) or odd order (1,3,..23) channels and accomplishes all calculating necessary for ROM/ACC procedure. Complete bank is connected through interface to host computer.

Because of high speed of CMOS chips and continuous multiplexing of arithmetic units in specific loop conditions (as IIR filter) it has been developped a special Program for step-by-step method of hardware testing. This program gives a detailled control of ROM/ACC calculating procedure of output samples Yn (especially usefull is AU adder outcomes). By means of that program we've had a direct connectivity between hardware and computer simulation.

5. EXPERIMENTAL RESULTS

In this paper we present only a few the most interesting experimental results. For example, Figure 5 shows nonuniform 16-channel digital filter bank adopted in accordance to one of possible Channel Vocoder specifications. Obtained real-time results are good enough for many applications.

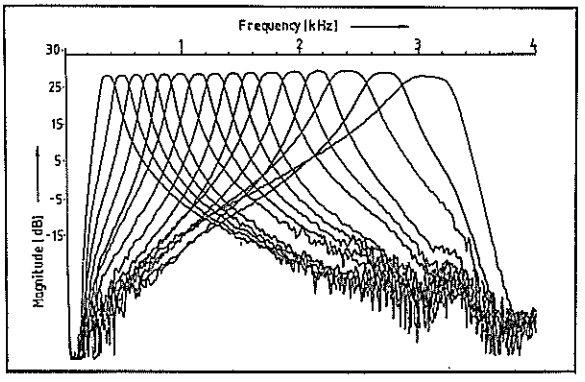


Figure 5

After interfacing digital filter bank with computer we worked on creation of referent speech spectrograms convinient for Isolated speech recognition experiments. Figure 6 shows the original sonogram allong with our measured spectrogram for the same pronunciation in near parallel position. From such spectrogram we can see descrete frequency resolution on 16 channels, time resolution on every 12 msec and intensity in maximal 10 possible shapes.

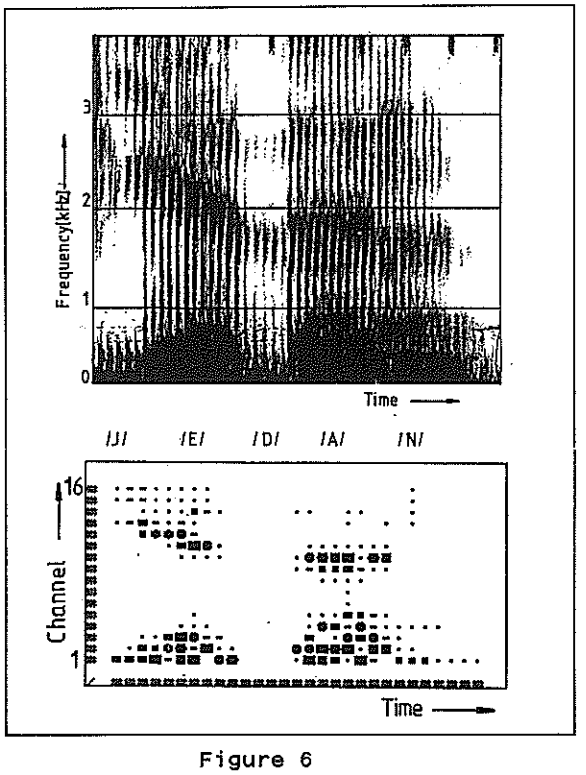


Figure 6

CONCLUSION

With proposed method of very useful computer upgrading of hardware realization we may conclude with next advantages:

-digital filter bank design is simpler and more reliable without fatigue, hard and large calculations (and errors also),
-flexibility in designing and exchanging of complete digital bank (simpler in a software and very simple in a hardware i.e. reprograming of only 8 EPROM),
-possibility of hardware complexity assuming on base of error analysis,
-simpler and more reliable hardware realization and reproduction of same digital bank performances, and
-possibillity of ASIC realization.

REFERENCES

[1] A.Peled,B.Liu, Digital Signal Processing, John Wiley&Sons, NY, 1976.
[2] R.Saal, Handbook of Filter Design, AEG Telefunken, 1979.
[3] Harry Y-F Lam, Analog and Digital Filters: Design and Realization, Prentice-Hall, Englewood Cliffs, New Jersey, 1979.
[4] Lj.Milić,M.Durić, Recursive digital filters (in Serbo-croation), Institut Mihailo Pupin, Naucna Knjiga, Beograd, 1982.
[5] A.Peled,B.Liu, A new hardware realization of Digital Filters, IEEE Trans. on ASSP, Vol.ASSP-22, No.6 (Dec.1974), pp.456-462.
[6] LJ.Jovanović, A Simulation of hardware realization of digital filters with bit serial implementation (in Serbo-croation), XXVI Konf. ETAN, Subotica 1982.
[7] Lj.Jovanović, A realization of programmable digital filters with bit serial implementation (in Serbo -croation), XXVII ETAN,Struga 1983.
[8] B.Liu, Effect of finite word length on the accuracy of digital filters-A review, IEEE Trans.on CT, Vol.CT-18, No.6 (Nov.1971),pp.670-677.
[9] Lj.Jovanović, A realization of Digital filter bank for speech resurch purposes, Master Thesis, University of Belgrade 1985.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

FAST PROTOTYPING OF SOFTWARE LIBRARIES FOR MULTIDIMENSIONAL SIGNAL PROCESSING

F. Russo, S. Broili, G. Ramponi

Dipartimento di Elettrotecnica Elettronica Informatica
Universita` di Trieste, via A. Valerio 10, 34127 Trieste - Italy

A Computer-Aided Library Prototyper (CALP), capable of easily generating, managing and testing libraries of multidimensional (M-D) signal processing routines is presented. User-friendliness minimizes the required knowledge of the tool; a portable code is generated.

1. INTRODUCTION

The growing research involved in M-D digital signal processing (DSP) today demands adequate software supports offering to the user a more versatile way of developing software than the direct use of programming languages can yield.

The strategy of classical libraries of DSP routines is to provide the user with the largest possible collection of routines, hoping that his own applications will be satisfied by the available software. More and more frequently, however, the end user (in particular if he is involved in research and development work) needs to quickly modify libraries for particular applications or, more concerning, to implement and test new algorithms.

Researchers, in particular, would require more suitable tools capable of increasing the productivity of their work by minimizing the crude programming effort and thus allowing them to completely concentrate on the algorithmic aspects. In this case, an approach which deals with the fundamental meaning of a DSP function, rather than with the specific implementation aspects of the corresponding routine, may be very useful. Such an approach should privilege rapidity and flexibility during the iterative process of creating, evaluating and adapting DSP functions to their particular goals: moreover, it should easily support further changes in requirements and exchanges of results among different researchers. Resorting to software engineering techniques such as the rapid prototyping methodology [1] appears to be very suitable for this purpose.

In this work we present the main characteristics of an experimental software tool, which we have developed in order to trying to actually cope with many of the particular aspects and problems of M-D DSP. This tool, which we address in the following as Computer-Aided Library Prototyper (CALP), has been designed according to the approach mentioned above, to give the researcher a user-friendly resource during the prototyping of signal processing functions.
Particular care about the human-computer interaction aspects has been taken in the design of CALP: for this purpose we have adopted a prototype description language which is very close to the universally known mathematical language. This approach turns upside-down (to the man's advantage) the classical terms of the man-machine interaction by making the system capable of understanding the natural way the user expresses himself.

In order to allow the users to easily exchange their results, a fast prototyping environment should also produce portable results. This is seldom provided by conventional development tools, which in general yield a specific machine code. The system we propose, on the contrary, generates a standard C source code for each developed routine; the choice of the C Language appeared in fact the most suitable to assure wide portability on different systems.

The peculiarity of the proposed tool with respect to M-D signal processing applications is the capability of both exploiting classical algorithms related to this field and easily creating or modifying new ones. Indeed, a general purpose library encompassing a set of conventional techniques can be included in the software; such a library can span from 2-D image processing algorithms (such as linear filters, median-type operators, simple algorithms for feature extraction) to the compression

and coding of sequences of images (3-D predictors and discrete transform algorithms). At the same time, the research-oriented user is able both to modify the existing algorithms and to use them as a basis for new and more complex operations.

2. THE FUNDAMENTAL STRUCTURE OF THE COMPUTER-AIDED LIBRARY PROTOTYPER

CALP is an integrated environment which makes available to the user a specific set of computer-aided tools: the library descriptor (LD), the library generator (LG) and the library experimenter (LE) (Fig.1).

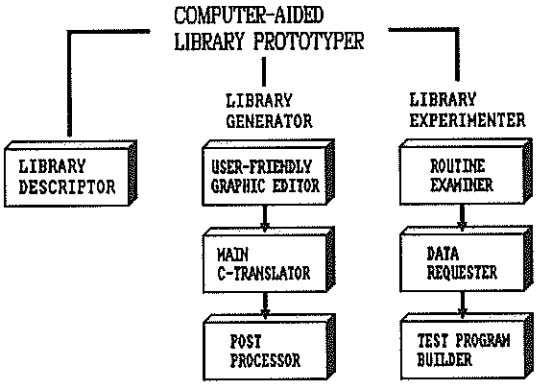


Fig. 1

The Library Descriptor has been designed to allow the user to acquire all the information needed to manage the routines available in the libraries. The LD extracts such information from the C-text of a routine (see the following paragraph) to highlight its purposes, the characteristics of its parameters, calls to other functions and, eventually, special directives.

The Library Generator is the core of the entire prototyping system: it resorts to a powerful user interface which exploits the istinctive mathematical language as the description language for DSP functions. For each created function the LG produces and makes available to the user two different kinds of code:
- the F-Code, which contains the formula representation of the function in encoded form;
- the C source code of the corresponding routine.

The LG also is able to acquire from the designer any useful information about the implemented function and to add this information to the C text in a proper form (comments).

In order to attain the proposed aims the LG includes a User-friendly Graphic Editor, a Main Translator and a Post-Processor. The editor provides the developer with a set of graphic symbols and conditional commands which allow him to write formula expressions on the screen. The editing section produces the F-Code mentioned above, which can be stored in the system memory for further processing. The importance of the F-Code relies on the fact that it renders the tool able to exactly re-create on the screen a previously designed DSP function. For this reason it represents an ideal medium to support software evolution among different owners of the tool. Once the F-Code has been produced, it is fed to a main C-translator which generates the standard C source text of the routine. A peculiar characteristic of such translator, which is very relevant to our applications, is its ability to automatically implement M-D arrays making them recognizable by the created routines. After the translation, the C text is suitably modified by the post-processing section in order to satisfy some particular requirements of second-step processing (i.e. insertion of comments, data types adjustment for specific purpose, etc.).

The Library Experimenter has been developed to render the evaluating phase of a routine easier. Differently from a simple application software which uses a fixed set of library routines, CALP is a development environment able to create a highly variable typology of routines: in fact, the LE plays the role of taking care of specific routine-dependent requirements such as data initializing, parameter passing, and displaying of results. For this purpose, a specific program section (the Examiner block in Fig.1) analyzes the structure of the routine selected by the user to determine the input/output data flow: a following section (the Requester) asks consequently the user to specify what data are to be processed by the routine. This information is used finally to build a complete executable test program of the routine, i.e. a C-language "main" which calls the routine itself.

3. THE USER INTERFACE

As previously mentioned, a key aspect in the design of the tool is its particular attention to the human-computer interaction: the problem of the user interface is arising in fact as a very important and compelling issue, especially when a development environment increases in performance and complexity.

As we pointed out in [2], we must carefully consider two related but very different aspects which concern the human-computer interaction in a computer-aided development tool:
A) level of operating interactivity (i.e. the way the user accesses all the available features of the tool);
B) level of specific knowledge (i.e. the knowledge required to the user to exploit entirely the features mentioned above).
The knowledge we consider here does not refer to an intrinsic user expertise of signal processing algorithms that are to be implemented (which we could define "cultural knowledge of the problem"); it deals rather with the training required to use the particular tool. These aspects must be considered , of course, on the basis of the specific purpose of a development environment.

The most advanced tools in the field of DSP generally privilege aspect (A). They are generally oriented to build the desired function by assembling a variable number of more or less complex ready-to-use "primitives"; examples of development environments of this kind, reported in the scientific literature, resort to graphic approaches including visual languages and block diagram data flow representations [3]. In our case, however, the target of the tool is exactly to build a primitive, and thus block diagram graphics alone appears to be unable to satisfy aspect (B) mentioned above, except for a very poor number of highly intuitive operations.

Therefore, the approach we adopted to optimize both aspects (A) and (B) in the Library Generator has been quite different. The choice of a formula-based approach has allowed us to make reference in general to a type of language which is rather user-related than machine-related. In particular, in order to minimize the required additional knowledge we have tried to:
i) free the user from learning the names of the commands: a limited but adequate menu of some 25 commands and symbols is readily seen in the editor screen;
ii) free the user from learning the syntax of the commands: an on-line syntax control deletes any erroneous input and, at the same time, prompts the user to insert particular symbols such as indexes and parentheses;
iii) free the user from defining the required function parameters: the editing section, when an unknown variable is met, adds its name in the parameter list.

4. PROTOTYPING OF FUNCTIONS FOR M-D SIGNAL PROCESSING

In this section, the particular characteristics the fast prototyping system must comply with in order to be used in M-D DSP are to put to evidence through some sample procedures.
We can begin with an operation which is very often performed on an image because it yields useful informations for the subsequent processing of the image itself: histogram evaluation. Fig.2a shows the formula representation of the code which measures the mean luminance value of an image b(x,y) and calculates the histogram. As it can be observed, the DEFINE statement is used to set an internal variable the value of which is then RETURNed to the main program. On the contrary, the evaluated histogram is returned to the main in the form of a parameter which is automatically inserted in the list of the function parameters. Notice also the presence of the "for any" operator which strongly simplifies the implementation.

Focussing our attention to specific algorithms for image processing, let us examine a stretching operator which acts directly on the gray-level value of each pixel, allowing contrast enhancement. A possible realization is shown in Fig.2b; there, F and G are the gray levels before and after stretching, while A, B and C, D are the minimum and maximum levels required before and after stretching. Proper values for such parameters must be provided by the user on the basis of the image histogram.
A slightly more complicate procedure is required in order to locally process an image, i.e. alter the value of a pixel according with the properties of its neighbours. Such processing can be of disparate types, being linear or not and oriented to the enhancement of the image or to the extraction of some features. Fig.2c shows a 3 X 3 highpass linear filter the coefficients of which are defined inside the function. On the other side, Fig.2d shows the realization of a noise-smoother (nonlinear) median filter which demonstrates a call to a previously defined function library. The edges of an image can be extracted by the well-known Sobel operator shown in Fig.2e; notice that its straightforward realization requires the nonlinear combination of two linear operators acting on the image.
The proposed tool can also cope with sequences of images, i.e. 3-D applications: an interframe predictor usable in a low bit-rate DPCM image transmission

system is presented in Fig.2f. The predictor exploits pixels on two subsequent frames a_i, a_{i-1} ; the prediction errors are overlaid on a_{i-1}. In the predictor realization the use of the conditional statements (WHILE, WEND) is demonstrated; notice also the indentation automatically performed by the editor to enhance the readability; finally, it should be observed that mulitple-indexed arrays are allowed (up to 8 indices); this simplifies the management of the image sequence.

REFERENCES

[1] Luqi: "Software evolution through rapid prototyping", IEEE Computer, vol.22, no.5, May 1989, pp.13-25.
[2] F.Russo and S.Broili: "A User-friendly Environment for the Generation of Highly Portable Software in Computer-Based Instrumentation", Proceedings of IEEE IMTC/90, San Jose, California, 13-15 February 1990, pp. 320-324.
[3] J.Santos, M.Veiga, J.Parera: "Automatic compilation of digital signal processing algorithms", Proceedings of EUSIPCO/88, Grenoble, September 1988, pp.763-766.

```
HISTOGR(N,b,h) :
 DEFINE L
```

$$L = \frac{\sum_{0}^{N-1}\sum_{0}^{N-1} b_{i,j}}{N^2}$$

$$h_{b_{x,y}} = h_{b_{x,y}} + 1 \qquad ,\forall x\in[0,N-1],\forall y\in[0,N-1]$$

```
 RETURN L
```

a

```
STRETCH(b,a,A,D,C,B,N) :
```

$$b_{x,y} = \left(a_{x,y} - A\right)\frac{D-C}{B-A} + C \qquad ,\forall x\in[0,N-1],\forall y\in[0,N-1]$$

b

```
HIPASS(a,b,N) :
 DEFINE K:3,3
```

$$K_{i,j} = -1 \qquad ,\forall i\in[0,2],\forall j\in[0,2]$$

$$K_{1,1} = 9$$

$$a_{x,y} = \sum_{0}^{2}\sum_{0}^{2} K_{m,n}\cdot b_{x+m,y+n} \qquad ,\forall x\in[0,N-3],\forall y\in[0,N-3]$$

c

```
MEDFILT(a,b,N) :
```

$$a_{x,y} = MED\left(b_{E,J},x,y,5,5\right) \qquad ,\forall x\in[0,N-1],\forall y\in[0,N-1]$$

d

```
SOBEL(b,a) :
 DEFINE C:3,3
 DEFINE D:3,3
 DEFINE E:128,128
 DEFINE F:128,128
```

$$C_{0,0} = -1$$

$$C_{1,0} = -2$$

$$C_{2,0} = -1$$

$$C_{h,1} = 0 \qquad ,\forall h\in[0,2]$$

$$C_{k,2} = -C_{k,0} \qquad ,\forall k\in[0,2]$$

$$D_{p,q} = -C_{q,p} \qquad ,\forall p\in[0,2],\forall q\in[0,2]$$

$$E_{i,j} = \sum_{0}^{2}\sum_{0}^{2} C_{m,n}\cdot b_{i+m,j+n} \qquad ,\forall i\in[0,125],\forall j\in[0,125]$$

$$F_{u,v} = \sum_{0}^{2}\sum_{0}^{2} D_{s,t}\cdot b_{u+s,v+t} \qquad ,\forall u\in[0,125],\forall v\in[0,125]$$

$$a_{x,y} = \sqrt{E_{x,y}^2 + F_{x,y}^2} \qquad ,\forall x\in[0,125],\forall y\in[0,125]$$

e

```
DPCM(M,N,a,h) :
 DEFINE z:4
 DEFINE i
 DEFINE x
 DEFINE y
 i = 2
 WHILE i ≤ M
  x = 1
  WHILE x ≤ N
   y = 1
   WHILE y ≤ N-1
    z0 = ai,x,y
    z1 = ai,x,y-1
    z2 = ai,x-1,y
    z3 = ai-1,x,y
```

$$a_{i-1,x,y} = z_0 - \sum_{1}^{3} h_k\cdot z_k$$

```
    y = y+1
   WEND
   x = x+1
  WEND
  i = i+1
 WEND
```

f

Fig.2 (a,b,c,d,e,f): Some examples of DSP functions (Hardcopy from editor screens)

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Realization and Optimization of a Speaker Independent Speech Recognizer for Isolated Words on a TMS 320C25

J. Zinke, S. Euler, A. Buch, and N. Jeck

TELENORMA, Zentrale Entwicklung
Mainzer Landstr. 128-146, D-6000 Frankfurt

In this paper we present a cost effective realization of a speaker independent automatic speech recognition system for isolated words using a single TMS 320C25 DSP. The recognition scheme is based on vector quantization of LPC coefficients and discrete density hidden Markov models. We discuss in detail the necessary steps to adapt the recognition algorithms and the parameter representation to the fixed point arithmetic DSP system. In simulation experiments with a representative data base we examined the influence of varying recording conditions such as different microphones and different AD quantization resolutions.

1 INTRODUCTION

In this paper a speaker independent isolated word recognizer designed for applications in telecommunication systems is described. The system should provide good recognition results within the future PABX environment with minimal memory and computation requirements. The realization is based on a flexible software architecture in order to allow further extensions.

Our recognition system is based on the use of discrete density hidden Markov models (HMM) for whole words [1]. Hidden Markov models have proved to yield a good representation of a large variety of different speakers while allowing a cost effective implementation of the recognition scheme with respect both to memory and computation time requirements.

The training and optimization of the recognizer, however, requires time consuming simulation experiments based on a reasonable amount of speech data. Therefore, we used a minicomputer environment for numerous simulations in order to investigate the properties of the HMM scheme and to test extensions of the basic HMM scheme. Furthermore, we adapted the simulation system to the projected DSP implementation and finally trained the models for the realtime system.

The programming of the DSP system is done on two levels. First, the real time part, namely word boundary detection and feature extraction, is programmed in DSP assembler in order to achieve maximum efficiency. The evaluation of the model probabilities by means of the Viterbi algorithm, however, is programmed in C, allowing larger flexibility and simple transfer of new modules from the minicomputer to the DSP system.

The paper is organized as follows. In section 2 we give a brief review of the recognition system [2]. The modifications of the recognition scheme required for the DSP implementation are described in section 3. Details of the software architecture and the memory and computation time requirements are discussed in section 4

2 SYSTEM CONCEPT

2.1 Data base

The vocabulary consists of the 23 German words shown in Table 1. The performance of the system was measured in terms of the recognition rates R23 for the 23 words, R10 for the digits with ZWO for '2', and RC for the 12 command words. A total of 200 speakers were included in the simulation experiments. One utterance of each word was recorded from each speaker. All speech signals were lowpass filtered with a bandwidth of 3400 Hz and sampled at 8 kHz and linearly quantized with 16 bit resolution. The recordings were done in a sound-proof cabin using a head-mounted microphone. Special care was taken in the recording sessions to avoid systematic effects caused by e.g. tiredness of the speakers or special emphasis on the first and last words. Therefore, the words were recorded in random order and additional non-vocabulary words were used at the beginning and end of each session.

Table 1: Vocabulary used in simulation experiments

Digits		Command words
NULL	(0)	ENDE
EINS	(1)	JA
ZWEI	(2)	NEIN
ZWO	(2)	HILFE
DREI	(3)	PAUSE
VIER	(4)	WIEDERHOLEN
FÜNF	(5)	WEITER
SECHS	(6)	KONFERENZ
SIEBEN	(7)	WAHLWIEDERHOLUNG
ACHT	(8)	RUFUMLEITUNG
NEUN	(9)	STORNIEREN
		RÜCKRUF

The set of 200 speakers was divided into two subsets of 100 speakers each, with one set used for training and the other for testing the system. The speakers were selected from a larger database such that each set represented a good approximation of the German population with respect to gender and age [3].

2.2 Preprocessing and word modeling

The endpoints of each utterance were determined automatically. Then the speech signals were divided into frames of 256 samples with an overlap of 64 samples between consecutive frames. For each speech frame an 8th-order LPC-analysis was performed and the prediction coefficients were used as components of the feature vectors. A vector quantization with the Itakura distance measure was applied to the feature vectors.

In the training of the system at first the codebooks were generated by means of the LBG-algorithm [4]. For each word in the vocabulary we estimated the parameters of a hidden Markov model, namely the transition and symbol probability matrices **A** and **B**, respectively. In the recognition scheme the Viterbi algorithm was used together with an a posteriori calculation of the state duration probabilities by Poisson distributions.

The optimization of the system parameters was done in simulation experiments with the database described above. A summary of results can be found in [5]. A good compromise between recognition accuracy, computation time, and required memory was obtained with a codebook of 64 entries and 5 model states. All results given in the following sections were obtained with this system configuration.

3 ADAPTIONS TO THE DSP SYSTEM

3.1 Preparation of the speech data

As described above, the speech signals were recorded at 8 kHz after a 3.4 kHz lowpass filtering with a 16 bit resolution of the A/D converter. An electret studio microphone with a flat frequency response was used in the recording sessions. In the projected application, however, distortions of the lower frequencies up to approximately 300 Hz have to be expected. In particular, we observed large variations in the frequency responses of different telephone handset microphones within the general specifications.

Best recognition results can be obtained if the frequency response is known and considered in the training of the models. Alternatively, in applications with varying microphones of a priori unknown properties only the frequency range from 300 Hz to 3400 Hz is used, where specifications of the telephone system allow differences of not more than 10 dB. Therefore, in our simulation experiments we used either the frequency response of specific microphones or the speech data was highpass filtered with a decline of 40 dB between 300 and 200 Hz.

In order to study the influence of the different frequency characteristics we examined the performance of the system for four combination of training and test data. In these experiments we used the 300 Hz highpass filter and the measured frequency response of one handset microphone. In case of a mismatch between training and testing data the recognition rates decrease by about 4%. The results show furthermore, that the recognition rates improve significantly with the increasing bandwidth.

In all simulations described so far we had used floating point arithmetic. Next we implemented a 16 bit integer arithmetic in the calculation of the autocorrelation coefficients, the vector quantization, and the Viterbi algorithm. Using proper scaling within the algorithms in order to prevent overflow while maintaining a optimum resolution all modules proved to be insensitive to the reduced accuracy.

Table 2: Recognition rates with different frequency characteristics of training and test data (HP: 300 Hz highpass, MIC: microphone frequency response)

training	test	R23	R10	RC
HP	HP	91.8	93.1	97.3
HP	MIC	88.0	89.8	93.3
MIC	HP	87.2	91.1	93.5
MIC	MIC	93.5	95.0	98.3

Next, we simulated the effects of the reduced resolution of the speech samples coming from the special AD converter. The speech signals were scaled in the range of 13 bits and then quantized according to the 8 bit A-law table. No significant changes in the recognition rates were found as long as the quantization was consistent in training and testing. Applying the A-law quantization to the test data only, however, gave an decrease in the rates of about 1%.

3.2 Parameter resolution

An important factor for the calculation time as well as the memory size is the required resolution of the stored parameters. In the vector quantization a representation with 16 bits for each component of the codebook vectors was sufficient.

The major portion of the memory is needed for the parameters of the hidden Markov models. In particular the amount of memory would increase linearly with the number of words in further applications using an larger vocabulary. The probabilities were first set to a minimum of 10^{-4} and then stored as scaled logarithms. In Table 3 the recognition rates are given as a function of the number of bits used. In our case a resolution of 8 bits is sufficient. It is interesting to note that even with a binary coding the recognition rate for the command words is still more than 80%.

4 DSP IMPLEMENTATION

4.1 Software architecture

The previously described algorithms were implemented on a PC-DSP-board with a single TMS 320C25 DSP. The speech IO is done via an extension board with a standard telecommunication filter/A-law codec combination (cofi) and a microphone preamplifier. Speech analysis software from recording of the speech samples to the computation of the vector indices as well as the word boundary detection was written in DSP assembler. The Viterbi algorithm was implemented in C code and translated into DSP-assembler by the TI C-crosscompiler.

Speech samples coming from the cofi every 125 μs are stored in the external memory by the interrupt service routine until a block of 128 samples is completed. Then a blockwise linearization, preemphasis and Hanning windowing is executed. The resulting data block is transferred into the internal RAM of the DSP. The main computation part of the speech analysis is the calculation of the autocorrelation coefficients. Using the MACD instruction together with the repeat counter facilities of the DSP the coefficients are computed in a loop very efficiently [6] [7].

Table 3: Recognition rates with different resolution of the model parameters A and B

Resolution	R23	R10	RC
1 bit	71.4	74.9	81.2
2 bit	76.0	78.0	85.7
3 bit	80.7	83.3	89.3
4 bit	84.4	86.1	92.4
5 bit	88.1	89.6	94.3
6 bit	91.2	91.8	96.5
7 bit	93.4	94.2	97.6
8 bit	94.2	95.3	97.9
Float	94.2	95.5	97.8

Internal facilities like pipelining and shifting are also used to implement the computation of the Itakura distance for vector quantization. The 64 codebook vectors with the LPC-parameters are stored in the external DSP-memory in a special format, well suited for the efficient evaluation of the Itakura distance.

Parallel to the computation of the vector indices a word endpoint detection based on energy features and zero crossing rates is done. The vector indices belonging to the two frames before and after the detected word boundaries are added to the automatically detected sequence in order to avoid truncation of the speech signal.

After filling a buffer with the sequence of the vector indices and the word length the cross compiled Viterbi algorithm is called. Due to the lack of long integer variables in the TI C-crosscompiler the 32 bit accumulator of the DSP can not be used effectively. Therefore, the Viterbi probabilities have to be downscaled within the DSP-computation to prevent overflow or saturation within the range of 16 bits.

4.2 Memory and computation time requirements

As discussed in the previous chapters we use only a single hidden Markov model for each word of our vocabulary. Using 64 symbols and 5 states per model we need 320 parameters in the B matrix and a maximum of 25 parameters in the A matrix. A resolution of 8 bits for these parameters results in a total memory requirement of 7935 Byte for all 23 word models. Including the state duration information into the models improved the recognition rate R23 by 3.1%. For this extension additional 345 Bytes of information about medium duration times per model state and 256 Bytes for an factorial look-up table are necessary. Further improvements of up to .5% were achieved by smoothing the symbol probabilities. The necessary addition of logarithmic values can be done efficiently by use of table look-up. Table 4 includes a summary of all used parameter and their memory requirements.

Table 4: Required memory for a configuration with 23 words, 64 codebook vectors of dimension 9, and 5 model states

Parameter	total size	resolution
Hanning–window	256 Byte	16 bit
VQ–codebook	1152 Byte	16 bit
HMM models (A, B)	7935 Byte	8 bit
Durations	345 Byte	8/16 bit
Look-up table factorial	256 Byte	16 bit
Look-up table log-addition	1024 Byte	16 bit
total:	10958 Byte	

Program code written in DSP-assembler needs 1214 words of memory, the crosscompiled HMM C-code needs 1102 words. There is still enough memory space left in the internal 4 kword of ROM in the TMS320C25 to implement additional modules.

The part written in assembler code needs a processing time of $\approx$ 19000 cycles for each frame of 16 ms. Using the repeat counter facilities to compute the autocorrelation coefficients results in 256 cycles wherein no interrupt service can be processed. At the sampling rate of 8 kHz an interrupt is issued from the AD only every 125 μs. Therefore it was possible to implement the feature extraction in a straight forward and effective use of DSP instructions. The vector quantizer requires 3400 cycles to compute the nearest vector out of 64. At a cycle time of 100 ns one 16 ms frame of data is processed in the feature extraction part in about 1.8 ms. There is also enough processing time to use greater codebooks, other features or more than one codebook.

The HMM C-code part of the software needs a processing time of $\approx$ 45000 cycles or 4.5 ms for each frame. Further effort will be spent to reduce this processing time by implementing assembler routines for the inner loops of the algorithm. Despite of the inefficient crosscompiler the C-code implementation of the Viterbi algorithms seems to be adequate in this stage because it simplifies adding new algorithm parts like computation of duration scores and smoothing of the symbol probabilities.

Table 5: Size of program code and processing time in cycles per frame

	code size	cycles/frame
Real-time Assembler part	1214 words	19000
HMM C-code part	1102 words	45000

5 CONCLUSION

The described recognition scheme based on vector quantization and discrete density hidden Markov modeling has been shown to yield good recognition results for the projected applications of speaker independent recognition of isolated words. With the optimized system using duration modeling and symbol probability smoothing, the best recognition rates were for all 23 words R23=94.4%, for the digits R10=95.4%, and for the command words RC=98.4%. Requirements of memory for parameters of the word models have been reduced with only minor decrease of the recognition performance. The implementation with a single TMS320C25 also has enough processing capabilities left to allow further extensions in the recognition scheme or to implement multichannel applications.

References

[1] L.R. Rabiner and B.-H. Juang. An introduction to hidden markov models. *IEEE ASSP Magazine*, 3:4–16, 1986.

[2] S. Euler, M. Falkhausen, D. Wolf, and J. Zinke. Sprecherunabhängige Einzelworterkennung mit Vektorquantisierung und stochastischen Wortmodellen. In *ITG-Fachbericht 105*, pages 93–98, Frankfurt, 1988.

[3] F. Englert, S. Euler, and D. Wolf. Zur Variabilität sprachlicher Äußerungen. *Informationstechnik it*, 31:407–413, 1989.

[4] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantization. *IEEE Trans. Comm.*, COM-28:84–95, 1980.

[5] M. Falkhausen, S. A. Euler, and D. Wolf. Improved training and recognition algorithms with VQ–based hidden markov models. In *ICASSP-90*, 1990.

[6] *TMS320C25 user's guide.* Texas Instruments, 1987.

[7] *Digital signal processing applications with the TMS320 family.* Texas Instruments, 1988.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A SMALLTALK-BASED ENVIRONMENT FOR DEVELOPING SIGNAL-PROCESSING PROGRAMS

Fuminori KOBAYASHI*, Kazuyoshi WARITA**, and Hiroshi AIMURA**

* Faculty of Computer Science & Systems Eng., Kyushu Institute of Technology
Iizuka 820, Japan
** Nagaoka University of Technology, Japan

An interactive, graphics-oriented computer system is implemented to assist trial-and-error development of signal-processing programs. The system employs the Smalltalk language, providing ample classes (data types) for various programs and a flexible multi-window environment maneuverable by a mouse. The prototype is implemented on 8086-based personal computers including a 'notebook' model. It is also useful as an instruction aid in signal theory education.

1 Introduction

Signal processing is a vital tool for a variety of applications. However, it includes many operations, and the most suitable one is dependent on each case. Thus, some trial-and-error procedure is usually required to find the best solution.

Though such tedious development processes will be resolved by expert systems, it will take time before they are practically available. Then, at least so far, some convenient tools are required [1-10]. The proposed system is intended to help engineers develop processing algorithms with an easy-to-use, friendly computer system.

Since these systems tend to become very large because they require sophisticated functions especially for improved human interface, they have sometimes been implemented with the emerging scheme, object-oriented programming (abbreviated as OOP). OOP have been used for such systems [1, 7, 8], and Smalltalk [11] is among the solutions [8]. Smalltalk best supports the OOP scheme as well as provides for flexible mechanism with graphics, which is suitable for implementing sophisticated user interface.

However, usual OOP languages, Lisp or Smalltalk-80, are heavy and require powerful workstations. The proposed system, on the other hand, incorporates a light-weight software which runs on a personal computer (PC). Several means including machine-language routines are devised to complement the limited power of PC's, yielding them as a reasonable tool for small- to medium-scale problems.

In this paper, general requirements to signal-processing environments and advantages of Smalltalk are briefly presented, followed by the description on functions provided by the prototype, then by some implementation details.

2 Signal Processing Environments and Smalltalk

2.1 Requirements to Signal Processing Environments

User-friendly tools for developing signal-processing programs require the following properties:

Visual presentation
Waveforms should be presented in graphical forms instead of numerals. This is a must.

Flexible multiple graphics
To ease comparative study in trial-and-error procedures, several results should be simultaneously presented. The number, size, and position of displayed waveforms are better not be fixed.

High-level abstraction
Frequently used operations, such as FFT or windowing, should be possible to apply with macros, without the need for specifying operation details.

User-customization
However, such built-in functions alone are not enough to achieve the best result. Arbitrary operations specific to each case should be able to be programmed.

2.2 Effectiveness of Object-Oriented Programming and Smalltalk

Usual programming languages, such as Fortran or C, treat a program as a combination of data and procedures to operate on them (Figure 1 (a)). Data are shared among procedures, thus are accessible to anywhere else.

In contrast, OOP considers a program as a collection of objects, which unify data and methods, the associated procedures (Figure 1 (b)). This hides details of data from external interface, emphasizing program modularity. Thus, OOP can satisfy some of the above properties as:

1. High-level abstraction is possible by hiding details, such as the number of data for FFT, and
2. Sophisticated operations such as multi-window manipulation leads to a large program, which are easily implemented with modular programming.

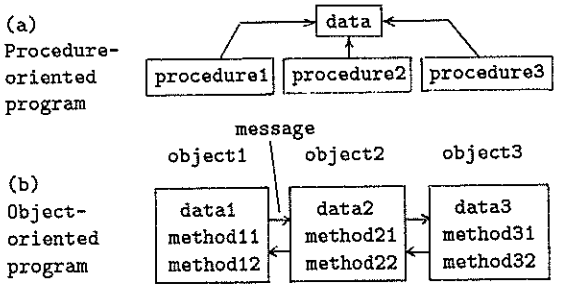


Figure 1: Data and procedures in a program

Smalltalk is a typical language for OOP. It supports not only the above two features but the following characteristics:

1. Graphics, capable of realizing sophisticated interactive manipulation,
2. Multiple windows, with any size and position, even overlappingly displayed,
3. Mixed execution of macros and user-written Smalltalk programs,
4. Blackboards for program texts, especially easy for partial execution and rewriting,
5. Easy maneuvering with menus and a mouse, and
6. Inheritance concept, to reduce program size.

3 Functions of the Prototype

Some sample operations on the prototype are shown below. Programs are executed by specifying the desired portion of a text (input or displayed) with a mouse and choosing *do it* from the menu (Fig. 2).

3.1 Signal Generation or Acquisition

Cos w:3
generates *cos* waveform with nominal frequency of 3 (w is for resemblance to ω).

Noise max:0.5
generates Gaussian noise in a range of -0.5 to +0.5.

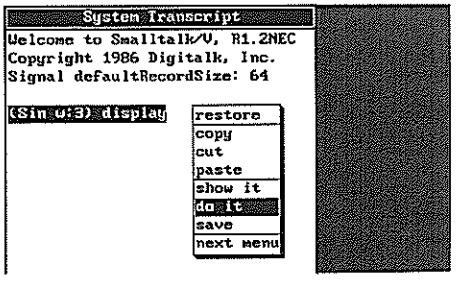


Figure 2: Program execution with mouse

(Sin w:2)*2+(Cos w:5)
[In-fix, single-character operators (such as '+' and '*') for vector operation greatly enhances program readability.]

Signal sampleWith: 100
acquires external data with a sample interval of 100ms.

The results of these operations are all vector data, which can be further processed, stored in a variable, or displayed.

3.2 Processing

3.2.1 Windowing

s1 hanning
applies Hanning window to a signal *s1*.

(Sin w:10) bartlett

3.2.2 FFT and IFFT

((Cos w:1)+((Cos w:5)*0.5)) fft

s2 ifft
[*s2* is assumed a signal in the frequency domain.]

3.2.3 Convolution

s3 convolve: s4

The results of these operations are also vector data, to be later processed and so on. Operation cascading is one of the features.

3.3 Assignment to Variables

Any signal generated/acquired or processed can be assigned to a variable. For example, after executing
s1:=(Sin w:3) fft,
s1 will hold the Fourier transform of $\sin 3\omega t$. Since variables in Smalltalk are typeless,
s1:=Sin w:5
is also valid.

3.4 Display Manipulation

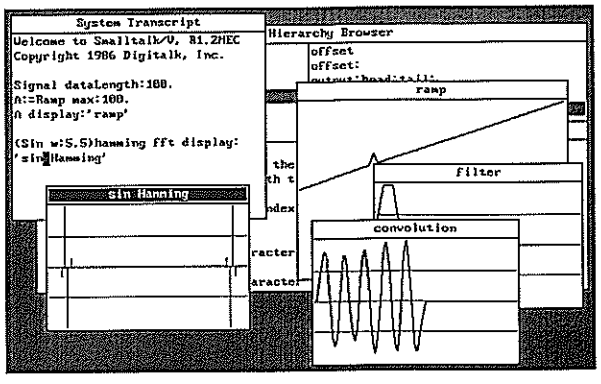


Figure 3: Sample screen hardcopy

When the statement

s1 display

is executed, the user will be prompted to specify the position of the upper-left and lower-right corners of the new window, in which s1 is to be displayed. The position and size of the once-appeared window can be modified by popping up and selecting a menu. Figure 3 shows a typical screen layout after 4 waveforms are displayed.

Signal value can be numerically read and be replaced by a new value, after which the waveform will be redrawn.

3.5 Mixed Operation of the Provided Functions and Smalltalk

In Figure 3, *System Transcript* and *Class Hierarchy Browser* of the Smalltalk system can be seen together with displayed waveforms. They indicates that the system can be also used as a usual Smalltalk environment. The user can arbitrarily program his own requirement, other than the operations the system provides. For example, signal values can be read by using an inspector, since signals are objects in Smalltalk.

4 Implementation

Hardware of the prototype is an NEC PC-9801 (comparable to IBM PC) with 640KB memory and an 8087[1]. A 10-bit A/D converter with a programmable sampling timer is included to acquire external signals.

Software is Digitalk's Smalltalk/V running under MS-DOS. Program text is approximately 1,000 lines long[2], 27KB..

4.1 Class hierarchy

The prime task of programming in Smalltalk is the definition of object *classes*. Of course, signals are the main objects of the system, and various *methods* are defined for the signal classes.

Ten classes are developed anew as shown in Figure 4. In the figure, the leftmost 4 classes in parentheses are superclasses in the original Smalltalk system.

4.2 Data Structures of Signals

Usual signal-processing programs define one data structure,

[1]An advantage of using PC's is that portable models are available. A version on a *notebook* PC (Toshiba J-3100SS) will be demonstrated at the Conference site.

[2]Though the system is functionally smaller, this amount of program is interesting to compare to 50,000 lines of Fortran in [5].

```
(Array) ------------┬-- Signal ------┬-- Sin
                    └-- Frequency    ├-- Cos
                                     ├-- Noise
(Object) ----------┬-- Complex      └-- Window
                   └-- ComplexSignal
(GraphPane) -------- SignalPane
(GraphDispatcher) -- SignalDispatcher
```

Figure 4: Class hierarchy of the prototype

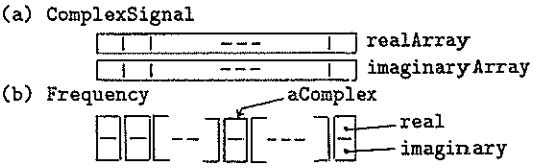


Figure 5: Data structure of ComplexSignal and Frequency

array of complex, for different signal domains: time and frequency. This system employs two dedicated types, *ComplexSignal* (for the time domain) and *Frequency*, in order to protect data from erroneous operations.

In the time domain the whole array of real or imaginary part is significant rather than a complex pair at a particular time instant, and two instance variables, realArray and imaginaryArray, hold data of real array (of class *Signal*) (Figure 5 (a)).

In the frequency domain, on the other hand, magnitude and phase of a particular element (frequency) in array are of interest rather than the profile of each part. Then, the primary structure is array, in which complex data are stored. (Figure 5 (b)). Operations on a complex data can be dispatched to class *Complex*.

These two structures provide for safety, since IDFT operation to a time-domain signal, for example, can be flagged error.

4.3 Mechanism to Implement the Desired Functions

4.3.1 generation

When a message, such as "Sin w:3", is issued, the 'Sin' class passes the appropriate function (in this case "sin") to class *Signal*. *Signal* generates an array of some length, stores appropriate values in it, and returns the array, as the following definition shows:

```
generateBy: aBlock
  |aSignal dt|
  dt:=2*Float pi/DataLength.
  aSignal:=self new:DataLength.
  1 to: DataLength do:[:i|
    aSignal at:i
            put:(aBlock value:i- 1*dt)
  ].
  ^aSignal
```

4.3.2 processing

Though "(Sin w:3)+0.5" and "(Sin w:3)+(Cos w:5)" seem to include the same addition, different operations are needed according to the argument type. While the former adds 0.5 uniformly to every element of "Sin w:3", the latter adds "Sin w:3" and "Cos w:5", element by element.

Since objects in Smalltalk carry information on what they are, the operation "+" includes two different processings for the two type of addend. The definition of "+" is shown below:

```
+ addend
  |size aSignal|
  (addend isKindOf:Number) ifTrue:[
    ^self collect:[:elem| elem+addend]
  ].
  ifFlase:[
    size:=self size min:addend size.
    aSignal:=Signal new:size.
    1 to: size do:[:i|
      aSignal at:i
              put:(self at:i)+(addend at:i)
    ]
    ^aSignal
  ].
```

4.3.3 displaying

Smalltalk employs the "MVC" (Model-View-Controller) concept to manage displaying on windows, menu driving, and status of the displayed object. This system assigns signals (Signal, ComplexSignal, Frequency) as "M", SignalPane as "V", and SignalDispatcher as "C".

When a signal object receives the message *display*, it generates a window (TopPane) and pastes a SignalPane (V) on it. Then, it places the pane under the control of SignalDispatcher (C) and prompts the user to specify the position and size of the window. Waveforms are automatically scaled so that they are to be displayed full in a window.

4.4 Mechanisms for Fast Execution

A disadvantage of Smalltalk is its slow execution, caused by its inherently (intermediate-code) interpretive operation and dynamical memory management. In the proposed system, limited power of PC's makes the situation worse. Then, the following two means are devised:

direct graphics drawing to the screen
: Smalltalk usually draws graphics by first drawing a *Form* and then pasting it on the real screen. With this method, however, nothing is seen until the form is completed. The prototype directly draws on the screen after clearing the window portion, which is then cut and stored as a form.

assembly-language FFT
: The slow speed of Smalltalk is emphasized when DFT/IDFT is executed. The prototype then employs a machine-language routine and the procedure shown in Figure 6 is used to achieve DFT or IDFT.[3]

```
DFT
  ComplexSignal → InterfaceArray → Frequency
                        ↓↑
                assembly FFT (FFT mode)
IDFT
  Frequency → InterfaceArray → ComplexSignal
                        ↓↑
              assembly FFT (IFFT mode)
```

Figure 6: DFT/IDFT implementation with a machine-language routine

5 Conclusion

A computer system with multi-window graphics capability for developing signal-processing programs is constructed on an 8086-based personal computer. It demonstrates its effectiveness in the development of small- to medium-scale program.

The system can be served as a tool in CAI (Computer-Assisted Instruction) of signal theory [3, 4, 12], as well. One of the authors has used it in a lecture for graduate students for a year and found it excellently effective.

References

[1] Kopec, G.E., The Integrated Signal Processing System ISP, IEEE Trans. Acoust., Speech, Signal Process. ASSP-32 (1984) 842.

[2] Chang, S. et al., An Image Processing Language with Icon-Assisted Navigation, IEEE Trans. Software Eng. SE-11 (1985) 811.

[3] Hanrahan, H.E., An Open-Ended Computer Package for Signal Processing Teaching and Design, IEEE Trans. Educat. E-28 (1985) 155.

[4] Frew, J. and Dozier, J., The Image Processing Workbench – Portable Software for Remote Sensing Instruction and Research, Proc. IGARSS 86 Symp. (1986) 271.

[5] Lager, D.L. and Azevedo, S.G., SIG – A General-Purpose Signal Processing Program, Proc. IEEE 75 (1987) 1322.

[6] Covington, C.D., Graphic Oriented Signal Processing Language – GOSPL, Proc. SPEECH TECH '87 (1987) 81.

[7] Karjalainen, M., Altosaar, T. and Alku, P., QuickSig – An Object-Oriented Signal Processing Environment, Proc. ICASSP '88 (1988) 42.D8.4.

[8] Hebel, K.J., Javelina: An Environment for Digital Signal Processing Software Development, Comput. Music J. 13 (1989) 39.

[9] Nieberle, R.C. and Knoll, A.C., CADiSP – A Graphical Compiler for the Programming of DSP in a Completely Symbolic Way, Proc. ICASSP '90 (1990) 52.V5.7.

[10] Covell, M., An Algorithm Design Environment for Signal Processing, Proc. ICASSP '90 (1990) 54.D13.7.

[11] Goldberg, A. and Robson, D., Smalltalk-80. The Language and its Implementation (Addison-Wesley, 1983)

[12] Gawthrop, P.J., Computer-Aided Learning of Signal Theory, Trans. Inst. Measure. Control 7 (1985) 61.

[3] A/D converter is also controlled with a machine-language routine, to cope with both speed and interruption by garbage collection.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

PicPEn – A Programming Environment for Picot, a Real-Time Image Processing System

Maximilian Ott* Kazumasa Enami** Mitsutoshi Hatori* Kiyoharu Aizawa*

*University of Tokyo
Dept. of Electrical Engineering
7-3-1 Hongo, Bunkyo-ku
Tokyo 113, Japan.

**NHK Science & Technical
Research Laboratories
1-10-11 Kinuta, Setagaya-ku
Tokyo 157, Japan.

Abstract

In an effort to improve the efficiency of transforming new algorithm concepts into code executable on a multiprocessor architecture, this paper describes the development of PicPEn, a programming environment supporting input in graphic form.

The user can paint a signal flowgraph on the screen just as if using a notepad. PicPEn will then automatically turn the desired application into a form which can be executed on the Picot system, a multiprocessor system for processing video signals in real-time.

1 Introduction

Processing dynamic images has only recently become practical due to new computer architectures and advances in semiconductor technology. These improvements make it possible to replace many dedicated hardware implementations with flexible, programmable systems. This will mean altering the working environment of many engineers, however by replacing the old tools with new, powerful and easy-to-use ones. The transition should be smooth and effective.

To be able to compute color images, like composite video signals in real-time (about 15 MHz pixel rate) NHK designed and built the Picot system using a novel multiprocessor architecture. Various identical processing units can be arbitrarily connected via a programmable network. It is possible to almost linearly increase processing power by adding extra clusters of processors. A comprehensive discussion of the hardware can be found in [1].

Unfortunately multiprocessor systems increase the complexity of the software design considerably. We will not only be faced with the obvious problems of partitioning and synchronizing, but with many more problems which will require a deep understanding of the actual system. To program such a complex machine efficiently it is necessary to provide tools which successfully hide the underlying complexity.

We are currently trying to develop a convenient programming environment where the user enters the desired algorithm in the form of a flowgraph. With this abstraction the designer does not need to take any special precautions regarding the multiprocessor structure. In fact the same input can be used for a software simulator implemented on conventional computers. This allows the user to fully debug her software independent of the target hardware.

2 Programming Applications

Researchers very often use data flowgraphs to present their algorithm concepts. The flowgraphs show the relation between different signals and how they get changed in a clear and concise way.

For digital signal processing, a number of software systems with a graphical input have appeared in the last few years, for instance ILS[2], I*S*P[3], ISP[4], Gabriel[5] and GOSPL[6], to name only a few. However, these they are mainly targeted for audio signal processing. In the area of image processing the only systems we know of with graphical input are a system supporting software development for a VSP developed by Philips[7] and the Princeton Engine[8]. The latter uses different representation forms for different levels of abstraction with a *graphical* assembler language at the lowest level.

Our motivation from the beginning of this project was to stay as close as possible to the way one draws a data flowgraph. For instance Fig.1 shows a simple adaptive noise canceller. The camera on the left denotes the sig-

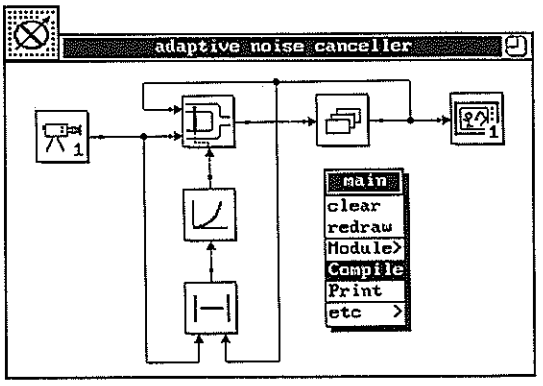


Figure 1: A screen copy showing a module window. The flowgraph inside the window represents the structure of the module. The icon used for this module at a higher level, together with its name are displayed in the header of the window.

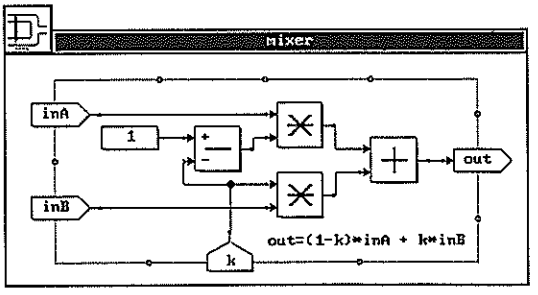


Figure 2: Part of a screen copy showing the internal structure of the mixer module in Fig.1.

nal source, followed by a 2-signal mixer on the right. The symbol at the utmost right symbolizes a monitor, preceded by a symbol for a frame buffer. The mixing factor is a non-linear function of the absolute pixel value difference. The non-linear function is implemented as a lookup table, with its icon representation shown below the mixer. Fig.1 is actually a copy of the screen image of our prototype system.

Any high-level programming language provides constructs to encapsulate and therefore hide details like subroutines, functions or procedures. We use hierarchically ordered *modules*, where each module includes a network of other modules and atomic operations (1 cycle operations provided by Picot, like alu or multiplier), and is represented as a single icon at a higher abstraction level. As an example, Fig.2 shows the mixer module from Fig.1 in an "exploded" form performing $out = (1 - k) * inA + k * inB$.

Any practical system will contain many modules. To ease the task of managing them we use hierarchically ordered libraries which contain modules as well as other

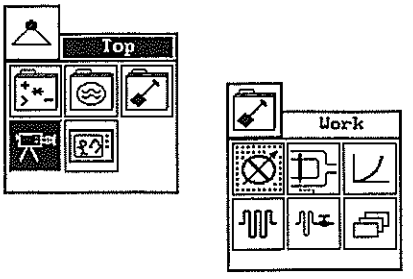


Figure 3: Two libraries, where *work* is included in its parent library *top* as work's icon is also shown in top.

libraries. Without a natural categorization of the modules a strict tree structure would be a severe and unnecessary restriction in many cases. It is therefore possible to include the same module or sub-library in different libraries to allow grouping depending on function, frequency of use, or any other categorization.

A window based browser allows one to skip through the libraries quickly. It is also very easy to view the internal structure of a module as well. Selected libraries can be kept on the screen and a personal library to collect modules and libraries of interest may be opened. Fig.3 shows two libraries where *work* is included in its parent library *top* and can be viewed with a simple click of a mouse button.

Parameters (like filter coefficients or scaling factors) defined at a lower level migrate upwards and can be set or derived from other parameters at any higher level. For interactive modules, parameters can also be linked to manipulators, like faders, on a controlling console.

In our notation everything is a module, from a simple adder, to such sophisticated modules as filters or complex geometrical mapping functions. If one programs an application, it will usually consist of various layers of module networks, but there will always be a single module at the highest level. Modules which only need to be connected to signal sources and output mediums (eg. monitors) in order to become a stand alone application, are called *application* modules.

PicPEn can therefore easily be configured as an operating system by only providing libraries containing application modules and I/O modules. For instance, to apply a certain transformation to a signal source, one just connects the corresponding module to the icon representing the desired signal source and the output to a monitor icon. After compiling and downloading, the result of the transformation, as performed by the real-time system can be viewed immediately on a real monitor represented by the above mentioned monitor icon.

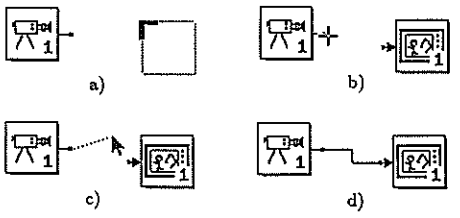


Figure 4: Sample editing session: a) positioning new module; b) cursor shape changes to a cross as it is positioned on a pin; c) starting to connect the output stream to an input pin; d) auto-router draws connection line.

3 System Implementation

PicPEn is currently implemented on a PC. Being a highly interactive system, we chose a commercially available Smalltalk implementation as it already includes an easily configurable window system and easy access to graphical primitives. It further allows integrating programs written in different computer languages.

3.1 Graphics Editor

The Picot system is also well suited for applications in broadcasting. However, since it is one of the first programmable image processing systems, many engineers will be confronted with programming for the first time. Therefore the man-machine interface is of vital importance.

One of the main problems with graphics editors is that they usually require the user to switch between different modes, like *place*, *move* and *connect*. We did not abolish the modes, but instead the editor automatically switches between them depending on the position of the cursor. For instance the *move* mode is entered as soon as the cursor is inside a icon, as icons are not allowed to overlap. The state the editor has switched to is further indicated by the cursor shape, which changes accordingly. This is very effective as the cursor location is normally also the center of attention for the user.

Fig.4 shows a simple editing session. By pressing the *action* button on the mouse the outline of a new module appears on the screen and can be positioned (fig.4a) by moving the mouse. Releasing the mouse button places the module at the selected position. Moving the cursor on top of an output pin automatically changes the cursor to a cross, as shown in fig.4b. By pressing the action button now, a rubberband is drawn between the pin and the cursor (fig.4c). Releasing the button when the cursor is positioned above an input pin will connect the two pins, and an auto-router will draw the connection on a rectangular grid without intersecting any already placed modules or text blocks (fig.4d). Note, that the whole sequence a–d could be achieved by only pressing the action button twice.

3.2 Compiler

When the user has finished designing a flowgraph and requested compiling, the editor sends a description of the flowgraph to the compiler. In order to create executable code for the the desired platform, the following steps are necessary:

1. Decomposition of the graph
2. Scheduling nodes on processing elements
3. Mapping the delay-free network on a synchronous network
4. Creating code

(1) Decomposition of the Graph

As the modules in the graph can be of different complexities, a recursive decomposition of all the modules will produce a *flat* network consisting of only the basic *processing nodes* provided by the hardware system:

- basic arithmetic and logic operators (add, mul, and, or, ...)
- look-up tables

(2) Scheduling Processing Nodes on Processing Elements

All nodes in the network produced by the previous step can be executed within one clock cycle on a single processing element. If we only consider unconditional flowgraphs and further note that the clock rate of the processing nodes is the same as the signal sampling rate, we see that providing for sharing of processing elements between nodes is not necessary. Thus scheduling on an architecture like Picot is fairly straightforward. We only face the following restrictions:

- Each Picot processor unit provides 1 alu, 1 multiplier, 1 addressing unit and either a long delay-line, a frame memory or a look-up table.
- The processing elements within a single processing unit will only be fully or partly utilized if all the

nodes on this processor have only 2 incoming and 1 outgoing connection with the rest of the graph.

- More than one application module can be downloaded into the program memory of a single processor unit. However, if one of those modules uses the local data memory to store a look-up table, other modules can use the data memory only for the same table.

From the last point in particular, and others specific to the system, the requirement of a central *System Resource Manager* which is responsible for the proper allocation of the resources used by the different applications, becomes apparent.

(3) Mapping to Synchronous Network

Programs on Picot are implemented as systolic networks. Therefore, starting from a delay-free flowgraph (intentional delays have to be represented by special delay nodes), it is necessary to convert the original flowgraph into one considering communication delays between the nodes. Different from the timing problems in most systolic systems[9], we have to consider a variable number of delays on each connection. To ensure a successful mapping it may be necessary to increase some of these delays. This is slightly complicated on Picot, as only certain connections can deliberately increase their delay. The authors developed an algorithm performing the above described task. For further details refer to [10].

(4) Creating code:

After performing the steps described above, we have a network of processing units, where each unit is associated with a set of processing nodes in the "flattened" flowgraph. Some of the connections in this network carry extra delay as calculated in (3). These objects now contain all the necessary information to extract the command sequence for setting up the target system to carry out the desired function.

4 *Conclusion and Further Work*

We described a programming environment for applications in the field of video signal processing. As programmable real-time systems only recently became a reality, many engineers in this field are not very familiar with programming techniques. We therefore tried to develop a visual programming language similar to the very familiar concept of data flowgraphs. Furthermore, the system's independent input allows portability between different target systems.

A weakness of the approach described above is the lack of conditional constructs. Fortunately, the parts of an application requiring conditional signal flow are usually very localized. Encapsuling these parts in assembler modules provides a temporary solution. We are working on extending the "language" to include conditional constructs as well.

A first version of this system was finished recently and includes some of the more important features described above. This will now allow us to study the reactions of typical users and so determine the weak points of our approach.

References

[1] Yagi, Yajima, Enami, et al., "Real-time Video Signal Processing System for Dynamic Images", SPIE Symposium '89, Philadelphia, Pennsylvania, November 1989.

[2] Technical Manual, Interactive Laboratory System, Signal Technology, Goleta, CA.

[3] "I*S*P – The interactive signal processor", Bedford Research, Bedford, MA.

[4] G. Kopec, "The integrated signal processing system ISP", IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-32, pp. 842-851, Aug. 1984.

[5] Lee, Ho, Goei, et al., "Gabriel – A Design Environment for DSP", to appear in IEEE Transaction on ASSP.

[6] Covington, Carter, Summers, "Graphic Oriented Signal Processing Language – GOSPL", Proceedings, ICASSP-87, pp. 1879-1882, 1987.

[7] van Roermund, et al., "A General-Purpose Programmable Video Signal Processor", IEEE Trans. on Consumer Electronics, vol.35, no.3, pp. 249-257, August 1989.

[8] Chin, Passe, Bernard, et al., "The Princeton Engine: a Real-Time Video System Simulator", IEEE Trans. on Consumer Electronics, vol. 34, no. 2, pp. 285-297, May 1988.

[9] H. T. Kung, "Let's design algorithms for VLSI systems", Proceedings of the Caltech Conference on Very Large Scale Integration, Charles L. Seitz, ed., Pasadena, California, January 1979.

[10] M. Ott, K. Enami, M. Hatori, K. Aizawa, "Timing in Systolic Systems with Variable Minimum Connection Delays", IEEE Symposium on Circuits & Systems, New Orleans, May 1990.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

DIGITAL SIGNAL PROCESSOR IMPLEMENTATION OF WAVE DIGITAL LATTICE FILTERS

P. Balsiger, U. Sjöström, F. Pellandini

Institut de Microtechnique Université de Neuchâtel
Rue A.-L. Breguet 2 CH-2000 NEUCHATEL, Switzerland

Abstract: a comparison of different implementation approaches of wave digital lattice filters based on a DSP-architecture reference, derived from a number of commercially available second generation DSPs is presented, and an evaluation of the implementation complexity in function of the filter order.

1 INTRODUCTION

Due to the impressive performance of modern programmable *Digital Signal Processors (DSPs)* many different tasks can be handled on the same DSP. The trend for the near future seems to be fast multiply-accumulate units (10-20ns) and multiple data paths. Considering this evolution into account, the *memory capacity* for programs and data, and the execution time are still important. Because that the complexity of typical applications is growing at least as fast.

In most application fields, filter problems must be treated using reliable filter families. *Wave Digital Filters (WDFs) have* shown to be very useful, especially WDF derived from *lattice reference filters.* This paper gives general estimations on the implementation complexity of *Wave Digital Lattice Filters (WDLFs)* realized as cascaded first and second order all-pass sections and as state-space structures. Furthermore, a performance comparison gives information about the feasibility and the implementation complexity of a given filter. To simplify the estimations, a DSP-architecture reference model has been defined.

2 WAVE DIGITAL LATTICE FILTERS

Wave Digital Filters derived from lattice reference filters, are in some respect the most attractive structures available [2]. WDLFs are known to have many interesting properties. The most important of these concerns their excellent stability behavior, in particular with respect to the nonlinearities due to signal quantization operations by carrying out rounding/ truncation and overflow corrections.

An advantage of using these structures lies in few number of elements in the canonical reference lattice structure in the analog domain. The number of elements directly corresponds to the number of multiplications in the final wave digital lattice filter. WDLFs are also suitable for sampling rate alteration.

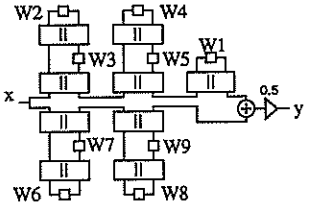


Fig1: Example of signal flow graph of a 9'th order WDLF

Another interesting property of the WDLF is that the structure always is the same, i.e., two parallel branches with first and second order all-pass sections *(fig1)*. Advantage can be taken from this fact when implementing them. Each branch can be realized on separate DSPs resulting in an improved execution speed.

3 DIGITAL SIGNAL PROCESSORS

The high performance of a DSP is achieved through multiple data paths by integration of a hardware multiplier/ accumulator and multiple independent memories.

On most of second generation DSPs, multiply-accumulate units and separate adder/subtracter units process data concurrently and in parallel in the same instruction cycle. Indirect addressing is provided by a dedicated address generation unit able to work in parallel on multiple buses and data processing units.

Simultaneous fetches of memory access can be achieved with fast buses and memories: each instruction cycle would involve several memory accesses. The drawback of fast memories is the high power consumption. Alternative approaches are multi-ported memories, or multiple memory banks.

The programming style, in parallel with the degree of the regularity of the algorithm, affects the execution time. Straight-line programming is often more powerful than looped programming. Looped programming causes some serious problems (additional cycles) due to the pipeline of the architecture. Some DSPs are optimized for a low-

overhead looping capability (DSP56001 and DSP96002).

A high degree of the regularity of the algorithm exploits the address generation unit for the pointer updating. Each irregularity in the algorithm sometimes causes a complicated address computation in parallel with a fetch of the corresponding data.

Usually a regular algorithm enables vector processing. The vectors can be partitioned between each of the processors in a multiprocessor network. This speeds up the overall execution time.

Algorithms having a wordlength smaller or equal to the RAM wordlength are possible to process at full speed. A large wordlength would require double precision arithmetic computation. The execution time immediately increase. This increase is generally not at all linear.

On most fixed-point arithmetic DSPs, only limited scaling modes are supported. Other supported scaling operations that these realized normally by arithmetic shift operations must be performed using some additional instruction cycles.

Some commercially available second generation DSP-chips have been studied and compared in order to be able to define a *DSP-architecture reference.*

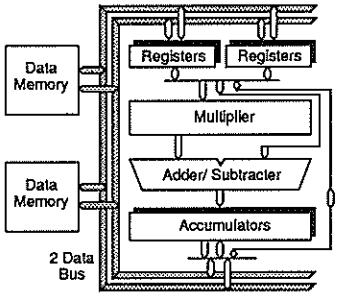


Fig2: Reference architecture of a simplified second generation DSP

The obtained architecture shown in *figure 2* has the following features:

⇒ *simultaneous fetches of an instruction and two operands in the same instruction cycle*

⇒ *four input registers for filter coefficients and filter variables*

⇒ *a dedicated single instruction cycle hardware multiplier in parallel with a single instruction cycle adder/ subtracter*

⇒ *two accumulator registers for the output results.*

Some DSPs are even more powerful and sophisticated offering simultaneous operations on parallel processing units, resulting in fewer instruction cycles for a given algorithm (e.g.,TMS320C30, DSP96002).

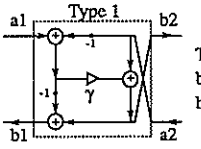
4 IMPLEMENTATION APPROACHES

Four different implementation approaches proposed in [4,9,3,10] for WDLFs are briefly discussed. The first three approaches consider the direct realization of the signal flow graph, shown in *figure 1.* The fourth approach is very different because a so-called state-space transformation of the signal flow graph is utilized. Then the filter algorithm is represented by a matrix description, shown in *figure 6.*

Method 1: Four different two-port adaptors

In [4,5], four types of two-port adaptors are used. The structure of the adaptors is selected so that optimal scaling for a sinusoidal excitation (L_∞-norm) is guaranteed *(example in figure 3).*

The implementation techniques correspond to compute the outputs of the adaptors in function of both inputs. An example of a type 1-adaptor is showed by equations (1). Finally, the filter is realized by connecting the corresponding adaptors. Therefore, the parallel-move instructions provided by DSPs cannot be used. Due to additional variable updating at the end of the sampling period, each adaptor requires one separate data-move.



Type 1:
b1=a1*γ-a2*γ+2*a2-a1 (1)
b2=a1*γ-a2*γ+a2

Fig3: Example of a two-port adaptor type 1 [5]

Method 2: Scaled first-and second-order sections

In [9], the signal flow graph characteristics of WDLFs are considered better than in the first approach. State variables W_i are directly updated. Furthermore, this approach is not limited to a specific scaling norm.

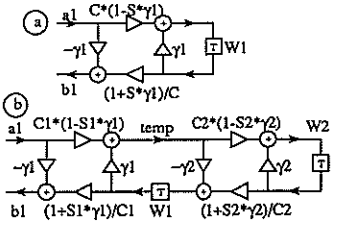


Fig4: First and second-order implementation, (a) first-order section, (b) second-order section [9]

The second-order section in *figure 4b* is described by the equations:

b1(n)= -a1(n)*γ1+W1(n-1)*[1/C1*(1+S1*γ1)] (2)

temp= a1(n)*[C1*(1-S1*γ1)]+W1(n-1)*γ1

W1(n)= -temp*γ2+W2(n-1)*[1/C2* (1+S2*γ2)]

W2(n)= temp*[C2*(1-S2*γ2)]+W2(n-1)*γ2

Each macro is described by the adaptor coefficients γ_i, the scaling coefficients C_i and the sign parameters S_i. To avoid additional cycles for scaling the coefficients C_i and $1/C_i$ are considered as multi-

plier coefficients and not as arithmetic shift operations.

Method 3: Modified adaptors

Fettweis [3] has recently proposed some modified two-port adaptors resulting in a reduced amount of multiply-accumulate operations *(fig5)*. This gives shorter execution time but increases the memory requirements and the coefficient wordlengths.

As can be seen from the modified 4-pole adaptors N1 and N2, described by (3) and (4), each has two new coefficients γ_{ij}. The modifications sometimes result in additional coefficients k_n', k_n''. None of the coefficients γ_{ij} is larger than "1" in module. The presented equations are only simplified overall representations valid for a specific case.

Modified 4-pole adaptor N1 *(fig5a)*:

$$b0=a0*\gamma00+a1' \qquad b1'=a0+a1'*\gamma11 \qquad (3)$$

Modified 4-pole adaptor N2 *(fig5b)*:

$$b2'=a2'*\gamma22+a3' \qquad b3'=a2'+a3'*\gamma33 \qquad (4)$$

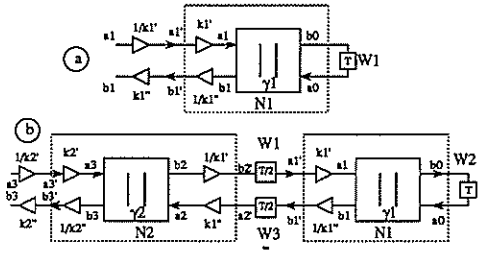


Fig5: Modified adaptors used in [3], (a) first-order section, (b) second-order section

Using this approach the program complexity can be kept low while memory requirements for variables and coefficients are relatively high.

Method 4: State-Space Realizations

The fourth proposed method is based on a state-space representation of the filter algorithm. This representation can be obtained from the original signal flow graph by directly computing the new value of each state variable as a linear combination of the old vector of state variables. An N order filter can generally be represented by an (N+1)*(N+1) matrix. In order to conserve the numerical properties of the original filter it is important to use a numerical equivalent state-space representation [10,11]. However, this implies that each matrix coefficient (α_{ij}) is exactly derived from the filter coefficients γ_i. Usually this results in an increased wordlength. In particular for WDLFs, the matrix elements in the last column will be the ones with the largest wordlength.

To eliminate these difficulties, some auxiliary variables can be introduced. This modifications enables controlling the matrix element wordlengths in certain limits at the expense of an increased matrix dimension. This technique is not developed in this paper.

The numerically equivalent state-space transformed filters have a minimum of quantization points. The noise level will thus be essentially lower than in the other structures [10,6].

$$\begin{bmatrix} W1(kT) \\ W2(kT) \\ W3(kT) \\ W4(kT) \\ W5(kT) \\ W6(kT) \\ W7(kT \\ W8(kT) \\ W9(kT) \\ y(kT) \end{bmatrix} = \begin{bmatrix} \alpha11 & 0 & \alpha13 & 0 & \alpha15 & 0 & 0 & 0 & 0 & \alpha1A \\ 0 & \alpha22 & \alpha23 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha2A \\ 0 & \alpha32 & \alpha33 & 0 & 0 & 0 & 0 & 0 & 0 & \alpha3A \\ 0 & 0 & \alpha43 & \alpha44 & \alpha45 & 0 & 0 & 0 & 0 & \alpha4A \\ 0 & 0 & \alpha53 & \alpha54 & \alpha55 & 0 & 0 & 0 & 0 & \alpha5A \\ 0 & 0 & 0 & 0 & 0 & \alpha66 & \alpha67 & 0 & 0 & \alpha6A \\ 0 & 0 & 0 & 0 & 0 & \alpha76 & \alpha77 & 0 & 0 & \alpha7A \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha87 & \alpha88 & \alpha89 & \alpha8A \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha97 & \alpha98 & \alpha99 & \alpha9A \\ \alpha A1 & 0 & \alpha A3 & 0 & \alpha A5 & 0 & \alpha A7 & 0 & \alpha A9 & \alpha AA \end{bmatrix} * \begin{bmatrix} W1(kT-T) \\ W2(kT-T) \\ W3(kT-T) \\ W4(kT-T) \\ W5(kT-T) \\ W6(kT-T) \\ W7(kT-T) \\ W8(kT-T) \\ W9(kT-T) \\ x(kT) \end{bmatrix}$$

Fig6: State-space transformed 9'th order WDLF shown in figure 1

5 Implementation Complexity

Based on the DSP-architecture reference, general equations for the implementation complexity can be derived for all four methods *(table 1)*. The formulas give estimations of the number of instruction cycles and required memory size in function of the filter order. Phenomena such as double-precision computation and numerical properties have not been taken into account for this estimations.

	#instr. cycles	#memory words
Method 1	p*9+(N-p)*8+N*2+4	N*3
Method 2	14*(N-1)/2+8+4	9*(N-1)/2+5
Method 3	13*(N-1)/2+6+4	9*(N-1)/2+5
Method 4	≤int[4.63*N-2.75+4]	≤int[4.63*N-1.75]

Table 1: Estimation of the implementation complexity. N corresponds to the filter order and p to the number of type 1- or type 4-adaptors used in [4,5]

The bounds on the number of instruction cycles are illustrated in *figure 7*.

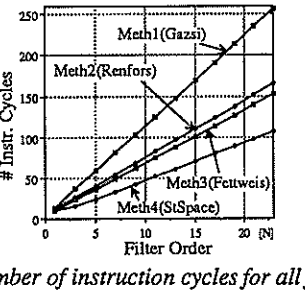


Fig7: Number of instruction cycles for all four methods

The upper bound is given by method 1 and the lower bound by method 4. The approach requiring little data memory is method 1 *(fig8)*. Methods 2, 3 and 4 require a 50% increase of data memory.

The state-space realization (method 4) requires the lowest number of instruction cycles at the expense of an extremely long matrix coefficient wordlengths. Due to the minimum of quantization points the state-space realizations have the lowest round-off noise.

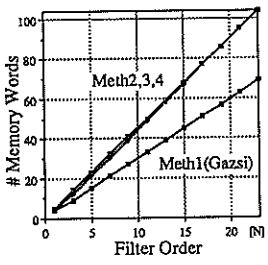


Fig8: Number of memory words for all four methods

Reasonable memory requirements and a few number of instruction cycles are obtained using method 3. The drawback is an enlargement of the coefficient wordlength due to the transformations. Moreover, this enlargement is independent of the filter order.

	Meth1	Meth2	Meth3	Meth4
Fast Execution	-	+	+	++
Small Memory	++	+	+	+
Small Coeffs.	++	+	-	--
Flexible Scaling	+	++	-	++
Signal-to-Noise Ratio	-	+	-	++

Table 2: Attributes for all four approaches

A flexible scaling is provided by methods 2 and 4. The smallest coefficient wordlength in parallel with the lowest memory requirements is provided by method 1 at the expense of a poor execution speed *(table 2)*.

DSP56001 (74ns)	Meth1	Meth2	Meth3	Meth4
fs_{max} (N=9) [KHz]	130	198	217	321
N_{max}	169	113	113	113

Table 3: Implementation example using a DSP56001 (74ns instruction cycle time)

The bounds are illustrated by using a DSP56001 (74ns instruction cycle time). Two kinds of comparisons are shown in *table 3*. The first is related to a specific filter and the second, to the available memory resources on the DSP.

Using the 9'th order WDLF shown in *figure 1*, the maximum achievable sampling frequencies are compared for all four methods. The state-space realization provide the highest sampling frequency of 321KHz.

The second comparison involves the memory resources. These results are rather than symbolic than realistic. The available memory resources show only the theoretical limit of the different methods. This comparison gives an estimation of the degree of leeway for multi-channel processing.

6 CONCLUSIONS

Using the presented estimations, it is now possible to know whether it is feasible to implement a specific wave digital lattice filter structure on DSPs corresponding to the architecture reference.

The results clearly show that the state-space approach seems to be very efficient. At least for fairly low-order filters. Method 3 is also a good candidate for most applications. When memory requirements are the target, method 1 is the best.

There exist a few possibilities to extend the approach described by method 4. To improve the coefficient wordlength of the state-space coefficients, auxiliary variables can be introduced at the expense of an increased matrix dimension. The increased matrix dimension results in a deterioration of the excellent execution speed. Again, this can be improved by reducing redundant operations and appropriate coding of the coefficients. The consequence can be a further reduction of the number of instruction cycles. The excellent numerical properties are conserved, and reduced coefficient wordlengths are obtained.

References

[1] Balsiger P., Pellandini F., "Signal Processor Implementation of an Adjustable Band-Pass Filter", *Proc. ISMM*, June 26-29, Zürich, Switzerland, 1989, pp. 238-241.

[2] Fettweis A., "Wave Digital Filters: Theory and Practice", *Proc.IEEE* , Vol.74, No.2, Feb. 1986, pp. 270-326.

[3] Fettweis A., "Modified Wave Digital Filters for Improved Implementation by Commercial Digital Signal Processors", *Signal Processing*, Vol.16, No.3, March 1989, pp. 193-207.

[4] Gazsi L., "Explicit Formulas for Lattice Wave Digital Filters", *IEEE Trans. on Circuits and Syst.*, Vol.CAS-32, No.1. Jan.1985, pp. 68-88.

[5] Gazsi L., *Falcon: Filter Design Program*, Ruhr University Bochum, FRG, Nov. 1986.

[6] Jackson L. B., *Digital Filters and Signal Processing*, Kluwer Academic Publishers, Boston, USA, 1986.

[7] Lee E., A., "Programmable DSP Architectures, an Overview", *Proc. URSI*, Symp. on Signals, Systems, and Electronics, 18-20 Sept, Erlangen, FRG, 1989.

[8] Van Meerbergen J., "Architectures and Characteristics of Commercially available General Purpose Signal Processors", *Proc. Workshop: CAD for Digital Signal Processing*, IMEC Leuven, Vol.2, Belgium Sept. 9-12, 1986.

[9] Renfors M., Zigouris E., "Signal Processor Implementation of Digital All-Pass Filters", *IEEE Trans. on Acoustics Speech and Signal Processing*, Vol. ASSP-36, No.5, May 1988, pp. 714-729.

[10] Wanhammar L., *An Approach to LSI Implementation of Wave Digital Filters*, Ph.D.Dissertation, No.62, Linköping, Sweden, 1981.

[11] Sjöström U., Defilippis I., Ansorge M., Pellandini F., "CAD Environment for Digital Filters Design and Implementation", *Proc. URSI*, Sept.18-20, Erlangen 1989, FRG, pp. 601-604.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A NEW REAL-TIME SYNCHRONOUS PROGRAMMING APPROACH TO CONTINUOUS SPEECH RECOGNITION

C. LE MAIRE, R. ANDRE-OBRECHT, P. LE GUERNIC

IRISA, Campus de Beaulieu, 35042 Rennes Cedex, France

In this paper we propose the use of intentional synchronous programming concepts as a way to specify and to program algorithms handling complex timing. As an example we use the SIGNAL language to give the description of a new acoustic-phonetic decoder used in an automatic speech recognition system.

1 Introduction

The need for a fast implementation of signal and speech processing algorithms is growing very rapidly. Moreover classical programming methods are not adapted to specify and to program such algorithms. The main cause is that classical languages are based on asynchronous communication mecanisms and they have a chronometric view of the time.
The drawbacks of these methods are at the origin of synchronous languages development. These languages are based on synchronous communication mecanisms and have a chronologic view of the time. We deal with SIGNAL, a data-flow oriented, real-time and synchronous language [7] .
The synchronous hypothesis are presented under two aspects: concerning the internal mechanisms of the system, every action is instantaneous (i.e. has a zero duration) and concerning the communications with the external world, the set of the possible input stimuli is fixed and known in advance, and input data sets are specified through the values they carry and a total ordering of the "instants" at which these values are available at the input ports. These aspects make of SIGNAL a determinist language whereas the data-flow characteristic permit to easily describe parallelism.
Our work consists in developing tools or mechanisms in a synchronous environment in order to facilitate the description and the implementation of signal processing algorithms. This work is illustrated through the implementation of the static part of an acoustic-phonetic decoder used in an automatic speech recognition system. In this paper we give examples to explain the different possibilities of using SIGNAL (see [6] to have a complete study).

2 an acoustic-phonetic decoder used in an automatic speech recognition system [2]

The main idea of this decoder is to use statistical methods to study signal sample by sample. In a first step acoustic segments are produced without a priori phonetic knowledges; a coarse classification of these units is realized to give event labellings and a vector quantization completes the identification of the segments (figure 1).

- the automatic segmentation is based on a statistics test: two autoregressive models (LPC) are identified sequentially and compared by a distance measure; at this "forward" analysis, is added a backward segmentation which processes the signal in the backward sense to detect eventual omission. Two real time programming difficulties arise:
  - a boundary is detected with some delay (threshold)
  - the signal is processed in the forward and the backward sense
- the coarse classification consists of labelling boundaries by some acoustic events (voice and frication onset and termination: VO-VT-FO-FT, closure: C, plosive burst: B). A voiced-unvoiced-silent decision and a plosive bursts detection are performed after a Fourier Transform and after a high pass filtering following by the same forward segmentation. Synchronous tools are necessary to coordinate block analysis and sample by sample analysis.

After the labelling by vector quantization, a global probabilistic model will be used to give the phonetic sentence.

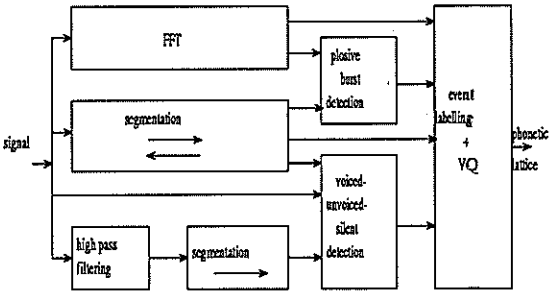


Figure 1: Modular description of a new acoustic-phonetic decoder.

3 SIGNAL: a suitable language to signal processing [5]

The basic objects of the language are named signals. A signal is a pair {flow, clock} where the flow is an ordered file of typed data of unspecified length and its associate clock specifies the instants at which successive values of the flow are available. For example the continuous speech signal can be represented by this object. The flow corresponds to the different values of the speech signal and the clock corresponds to the sampling rate.
The notion of time for a system is defined by relations between the clocks of the different signals of the system, without making reference to an universal clock.

A program is a static network of connected processes (or black boxes); each process can communicate with other processes or with the external world through its input and output signals (or ports). Then, the interconnexion of processes is made by identifying the name of input and output ports with the help of relabelling mecanisms of ports.
For example the decoder is a program which is composed of the segmentation and identification processes. The input port of the decoder process is the continuous speech signal and its output ports are the labelled units.

The SIGNAL language is based on five instructions:

- three "static" processes (the output ports values depend on the input ports values at the instant where they are disponible, without using past values):
  - the *functions* are instantaneous transformations on the data (arithmetic, logic...). For example the statistics test value (named u) depends on its last value, the distance between the two models and a bias. We write $u := lastu + distance + bias$ which is equivalent to $\forall t \;\; u_t := lastu_t + distance_t + bias_t$
  - the *filter* is a conditional overslamping (named "*when*"). For example the distance measure is computed after an initialization period; a logic signal named *active* is introduced, its value is *false* during the initialization and otherwise *true*:
    $distance := distance\_measure \; when \; active$
  - the *merge* with priority (named "*default*"). In the previous example, a new signal is defined to give initial values to the statistics during the first phase (i.e. zero),
    $u\_product := (0 \; when \; not \; active) \; default \; u$
    it takes the values of u when this is present.
- one "dynamic" process: the *delay* express knowledge of the past (named "$"). This instruction is essential because of the non persistent values. For example the previous value of the test u is $lastu := u\$1$ which is equivalent to $\forall t, \; lastu_t := u_{t-1}$.
- the *composition* operator (named "|") to build the network to takes the parallelism between processes into account. This associative and commutative operator allows to write the equations systems. So, to calculate the final value of u, we write

$$
\begin{array}{l}
(| \; lastu := u\$1 \\
\;\; | \; u := lastu + distance + bias \\
\;\; | \; u\_product := (0 \; when \; not \; active) \;\; default \; u \\
\;\; |)
\end{array}
$$

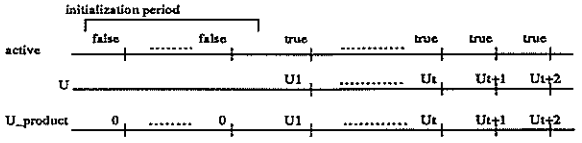


Figure 2: *active*, u and *u_product* signals clocks.

From these five instructions, a static tools library is developed. Let us mention

- *event* operator to access to signal clock. So to know the instants where the speech signal is present, we write $hsignal := event \; signal$
- *memorization* operator (named "*cell*") to memorize a signal to another clock. In the previous example, to compare the test value (u) with the maximum of this test, we must memorize this maximum to the u clock
- *explicit synchronization* operator: "$synchro \; x_1, ..., x_n$" process constrains x_1, ..., x_n to be synchronous
- *regular arrays of processes* to build repetition structure. We give an example in section 5
- *conditional composition* operator (named $if \; condition \; then \; processP \; else \; processQ \; fi$).

Likewise, a dynamic tools library is developed. Let us mention

- different counters: the usual (named "#") counts the past occurences of a signal
- *window* operator is an extension of *delay*. "$x := y \; window \; n$" is a sliding window of length n; at the t instant, $x_t[n] = y_t$, $x_t[n-1] = y_{t-1}$, ..., $x_t[1] = y_{t-(n-1)}$. So the sliding blocks processing is transparent to the programmer which uses the *window* tool.

To build a SIGNAL application we program all modules of the application. In every module we separate control gestion and data processing parts. Then we build the whole module by collecting together the different modules.
In order to facilitate modules and networks constructions, a SIGNAL graphic tool is developed which let us see a textual view or a modular and hierarchic graphic view of an application (figure 4).
More from this whole application we can develop an environment for the application which permit the user to observe the results or to interact during the execution. This opportunity is very useful to study the different parameters of the algorithms, such as bias, thresholds, model orders, etc.
Now we deal with of these different possibilities.

4 Control parts programming.

A part of control gestion is implicit contrary to classical asynchronous languages (FORTRAN, ADA...). The SIGNAL synchronization tools library makes the global control gestion programming easier. We illustrate these two points by concrete examples.
Synchronous programming (chronologic view) makes a part of the control implicit:

- parallelism between two modules is transparent to the programmer using composition operator. If these modules have a same input signal, they could be use signal values at the same logic instant. This is the case for segmentation module and plosive burst detections module. In this last one, Fourier Transforms are computed in parallel to speech signal segmentation
- when a boundary is detected by acoustic segmentation, this boundary must be analyzed by labelling module. This SIGNAL control part is implicit because to connect segmentation output port with labelling input port is sufficient (figure 4). From the temporal point of view the boundary will be analyzed by labelling modules at the logic instant which it will be detected; but segmentation and labelling modules may have different internal clocks between two boundaries detections.

In the same way synchronous programming concepts facilitate control gestion programming:

- initialization period is frequently used in decoder algorithms. As previously, a concrete form of this period may be given by the signal named *cactive*; it is created in using a SIGNAL counter:

$$
\begin{array}{l}
(\mid ny := \#(event\ Y) \\
\mid active := ny \geq length\_I \\
\mid)
\end{array}
$$

- processing again samples of speech signal is not easy to program in a classical language. SIGNAL permit oversampling by creating instants between two samples of speech signal. For example in the segmentation algorithm we must process speech signal in the backward sense. So we put speech signal in a sliding window ("*window*" operator) and we create instants between two samples to reinject the memorized samples
- in the same way coordination of the boundaries labelling is not easy to program in a classical language. A boundary is definitively labelled after three later boundaries locations. SIGNAL lets us delay the boundaries by putting them in a sliding window: *window_boundary* := *boundary window* 4. So the coordination module can label the boundary given by *window_boundary*[1], while it knows the later boundaries. From of the temporal point of view nothing is changed because the analysis is synchronous with the boundary detection

5 Data processing programming.

To help the user, a SIGNAL standard modules library [6] is created: number of zero crossings, autocorrelation vector, distance measure, etc. As example we cite the Burg lattice method programming which identifies one of the autoregressive models:

- Let us define $A_N(z) = 1 + \sum_1^N a_i z^i$ a polynomial filter. Given $e_t(n) = A_n(z)y_t$ and $f_t(n) = A_n^*(z)y_t$. So the recursive formula which define polynomial filter series are: $e_t(n) = e_t(n-1) - k_n * f_{t-1}(n-1)$
and $f_t(n) = -k_n * e_t(n-1) + f_{t-1}(n-1)$
These equations may be represented by a lattice structure (figure 3).

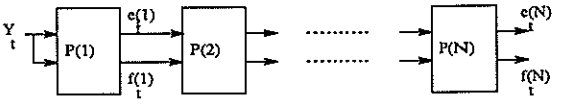


Figure 3: A lattice structure.

The SIGNAL traduction is easy with the help of regular arrays of processes:

$$
\begin{array}{ll}
(\mid zf := f\$1 & \\
\mid array\ n\ to\ N & (\mid e := e[n-1] - k * zf[n-1] \\
\quad of\ P(n)\quad with\ P(n) = & \mid f := -k * e[n-1] + zf[n-1] \\
\quad end & \mid k := ... \\
\mid) & \mid)
\end{array}
$$

- the possibility of using external modules which are not necessary written in SIGNAL (for example FORTRAN...) is offered. They are simply declared as external processes and then used as others processes. For example in our application we use the FFT (Fast Fourier Transform) and the vector quantization labelling as external modules.

6 Building of the application

The graphic tool enables us to build an application with a hierarchic or modular view. Figure 4 shows a graphic view of the decoder presented in section 2 and written in SIGNAL. The decoder has been built from the main modules by an increasing description; i.e. the results of a module can't be called into question by modules which analyze its results. However mixed or decreasing methods are not more difficult to use in SIGNAL; we can give as example the segmentation module building where boundaries can be called into question by the backward analysis.
The same methods may be pursued for a whole continuous speech recognition system. So SIGNAL and its graphic tool permit the programmer to use increasing, mixed or decreasing methods as it is usually defined to describe a system [4] ; all methods may be used to different levels without being concerned by the hardware architecture.

7 An environment for the application.

It's easy to create an environment for the decoder we have built. We still use the synchronous programming advantages and the SIGNAL graphic tool:

- the facility of substituting a module by another (which is not necessary written in SIGNAL) in order to test it in the application. For example to test another segmentation algorithm, we replace the segmentation module by another and we connect its ports
- the facility of transforming any internal signal as output port in order to have a tracking of its signal values
- the facility of using external functions which are connected to the screen or the mouse. So it's possible to synchronize an external function call with an internal or external signal clock. It permits the user to visualize the results of the processings (curve...) and to interact during the execution ("panel" functions to threshold adjustment...). We have developed an environment for the acoustic-phonetic decoder under the SunView window management system (figure 5).

8 Conclusion

The real-time synchronous aspects facilitate the temporal description of the application and the data-flow aspect facilitates the description of parallelism. Moreover the synchronization mecanism development, the graphic tool development, the static verification of the correctness of the timing which is performed by the SIGNAL compiler, the modularity of programs and the internal description of parallelism make SIGNAL a suitable language to signal processing.
To conclude, we can mention the forthcoming translator of SIGNAL programs to OCCAM as a first step going towards the implementation of programs in a transputer's network.

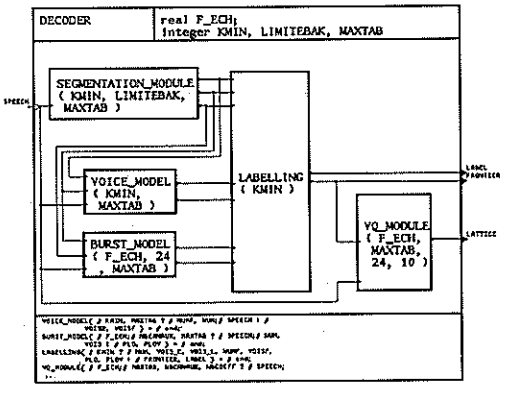


Figure 4: A graphic view of the *DECODER* process. It is composed of an automatic segmentation (*SEGMENTATION_MODULE*), a voiced-unvoiced-silent decision (*VOICE_MODEL*), a detection of plosive bursts (*BURST_MODEL*), a coordination between boundaries labelling (*LABELLING*) and vector quantization (*VQ_MODULE*).

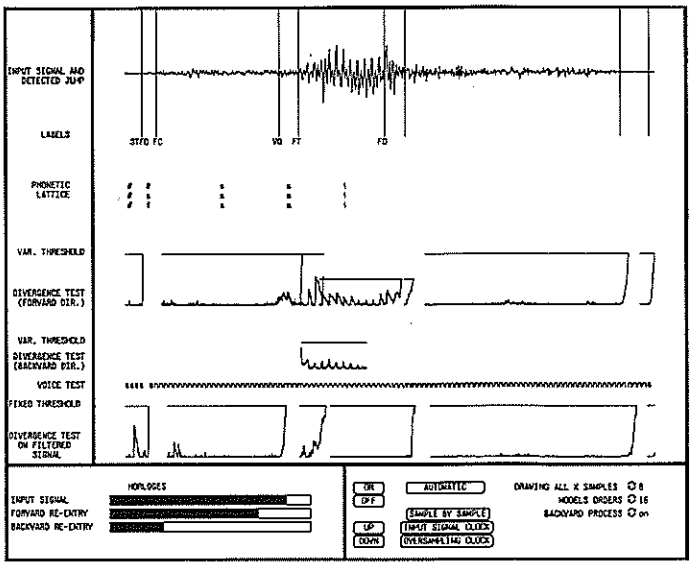


Figure 5: Synchronous environment for an acoustic-phonetic decoder; pronounced digit :"6".

References

[1] R. ANDRE-OBRECHT : *A New Approach for the Automatic Segmentation of Continuous Speech Signals* ; IEEE Trans. on ASSP, ASSP-36 No 1, pp. 29-40, January 1988.

[2] R. ANDRE-OBRECHT, H. Y. SU : *Three acoustic labellings for phoneme based continuous speech recognition* ; SPEECH '88, Edinburgh, Book 3, pp. 943-950, August 1988.

[3] P. BOURNAI, V. KERSCAVEN, P. LE GUERNIC : *Un environnement graphique pour la conception d'applications temps-réel* ; in Didier Plateau, éditeur, Ingéniérie des interfaces homme-machine, IN2-INRIA-LRI, Cargese, Corse, France, Mai 1989.

[4] CALLIOPE : *La parole et son traitement automatique* ; MASSON, Paris, Milan, Barcelone, Mexico, 1989.

[5] T. GAUTIER, P. LE GUERNIC, L. BESNARD : *SIGNAL: a declarative language for synchronous programming of real-time systems* ; Lecture Notes in Computer Science, Portland, Oregon, USA, Volume 274, pp.257-277, September 1987.

[6] C. LE MAIRE : *Le langage SIGNAL: un Exemple en Segmentation Automatique de la Parole Continue* ; Rapport interne IRISA No 527. Rapport de recherche, INRIA FRANCE, mars 1990, à paraître.

[7] P. LE GUERNIC, A. BENVENISTE, P. BOURNAI, T. GAUTIER : *SIGNAL—A Data Flow-Oriented Language for Signal Processing* ; IEEE Trans. on ASSP, ASSP-34 No 2, pp. 362-374, April 1986.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

A POWERFUL ENVIRONMENT FOR SPEECH SIGNAL ANALYSIS AND PROCESSING ON PERSONAL COMPUTERS

G. Bordel, J.M. Alcaide, M.I. Torres, J.M. Tarela

Dpto. Electricidad y Electrónica. Fac. Ciencias. Universidad del País Vasco.
Apdo. 644 - 48080 Bilbao, Spain.

This paper describes a UNIX-based software package that is being developed to support speech signal processing R&D. We discuss its major goals: menu-driven, intensive use of windows, nice 2D and 3D graphical representations, a wide set of signal analysis and processing tools, possibility of simply adding new researcher designed algorithms, statistical routines. We then describe the hardware and programming tools used in the system development. Finally, we present the package software structure consisting of six modules: user-machine interface, graphics manager, object handler, information manager, set of treatment and analysis procedures, and autoprogramming module. The state-of-the-art in computer technology allows to obtain the powerful features characterizing this package on a low cost personal computer.

1. INTRODUCTION

Signal Processing Research in the area of Speech Analysis requires powerful tools capable of manipulating and representing a large number of objects (collections of signals). They should also be able to implement a set of various algorithms which generate new objects as well. In the existing market there are specific machines [1][2] which deal with the problem. There are also software packages [3][4][5], some of them running on large machines [3][6] and other require special hardware [1][5]. The particular characteristics of this type of research require more flexible systems where the developed software structure becomes an important factor [4][7]. Our particular viewpoint of the problem has led us to take into consideration the following aspects at the moment of designing an R & D software package for speech signal processing:

- It is important to have a system which allows to be used as a workbench for new algorithms.
- The accumulation of a large amount of data can make a system unmanageable if the researcher is not released from the task of control the data source and the treatments applied.
- A characteristic of this type of work is the application of a sequence of techniques to a large set of data which results in statistic parameters. This job is extremely tedious and it is necessary to have a tool which does it automatically.

There are some added considerations which deal solely with computer-based activities. These are the following:

- The work system must be agreable, easy to manage and must not cause the user to wait. The user should not need a specific knowledge about the package internal structure at the moment of writing his/her own algorithms.
- The data should be perfectly protected against violations from other users of the same computer or network as well as against possible errors comitted by the researcher.
- The package should not have rigid structures which cause drastic changes when extensions are introduced. In this way the use of a new release will allow the usage of earlier data.
- The package must be as portable as possible. The use of standards, -i.e., software tools, operating system, not specific hardware- is desirable.

These goals have been taken into consideration in the design of the program. As we will discuss below, they are being realized by means of the application of various programming techniques along with an adequate environment.

2. HARDWARE AND PROGRAMMING TOOLS

Our system is designed for running on low-cost workstations. The main requirements are: enough RAM memory (6-8 Mb are adequated), hardware for math processing, and bit-mapped graphics (no color is required, but it is recommended).

An IBM PS/2 model 80 with 8 Mb RAM, math coprocessor and 140 Mb of hard disk storage is being used for our system development. This computer has been equipped with a data acquisition card DT2901 manufactured by Data Translation. This card provides A/D and D/A conversion with a maximum throughput of 50 KHz and 12 bits resolution. This is capable of transferring data to/from the computer memory by using the Direct Memory Access (DMA) mechanism.

The development of a complex software package requires advanced programming techniques and tools. The UNIX operating system [9] provides both. In particular, the AIX operating system (IBM's UNIX implementation) has been chosen.

The UNIX operating system allows multiprogramming techniques for program development [10][11]. Besides, it

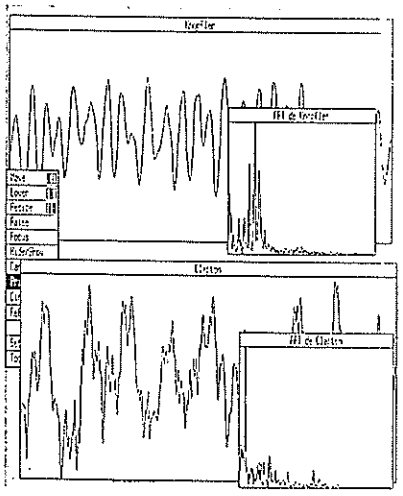


Figure 1 : Screen dump showing four windows.

includes a complete and powerful set of programming tools such as **make** and **SCCS** (for maintaining projects), or **lex** and **yacc** (for creating an input language) [11][12].

C, the UNIX "native" language [13], is being used for writing the whole source code of the system software.
Our system features a user interface based on windows, icons, pop-up menus and a mouse. Its implementation relies on the X Window System [14][15][16]. X Window is at present a rapidly expanding standard. The most remarkable feature of the X Window System is its network transparency. This allows user-program interaction from other nodes at the local network. In this way it is possible to take advantage of a particular node specific performances or simply of the node availability.

UNIX Device Driver for the DT2901 data acquisition card

A UNIX device driver for the DT2901 has been developed [17][18]. It gives indirect, controlled access to the data acquisition card for any program. Our device driver makes use of the DMA capabilities offered by both the computer and the DT2901.

Writing a UNIX device driver is a painful work. Driver routines deal with devices directly, being this the source of numerous problems. Moreover, the efficiency and robustness of the driver code are very important factors.

3. SYSTEM CHARACTERISTICS

The most remarkable features of our system (early version) are:

- Sampling rate up to 50 KHz (12 bits) only limited by the memory size avalaible. Control of the acquisition and reproduction conditions.
- The system is menu-driven and its management is based on the intensive use of windows (see Figure 1). Simplicity in use and other advantages of this sort of systems are well known.
- A wide range of mouse-driven display options to operate the waveform and spectrum graphical representation is provided: markers, cursors, real unit measurements, time scaling, several frequency scales,etc.
- 3-D representations: parallel perspective proyection or use of grey scale intensity as third axe. This allows the graphical display of the time-varying spectral characteristics of speech signals through the use of several transformations: spectrograms (see Figure 2) including a selection of analysis filters, Wigner transform, etc.

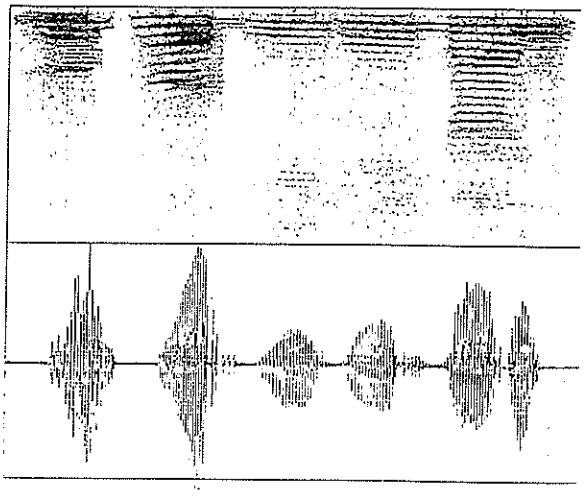


Figure 2 : Acoustic waveform and Spectrogram (FFT) for Spanish /ga/,/ge/,/gi/,/go/,/gu/.

- Information about any object can be obtain from the system at any time (see Figure 3).
- Basic signal analysis and processing tools in both time and frequency domain: windowing, pitch detectors, FFT, Cepstrum, LPC (some algorithms), Wigner and Hilbert transforms. These set of procedures will be as wide as the researcher needs. In fact, an important feature of our system is just the possibility of simply adding new signal processing algorithms at any time.
- Statistical processing of data and parameters. Just like in previous group, new routines can be added.

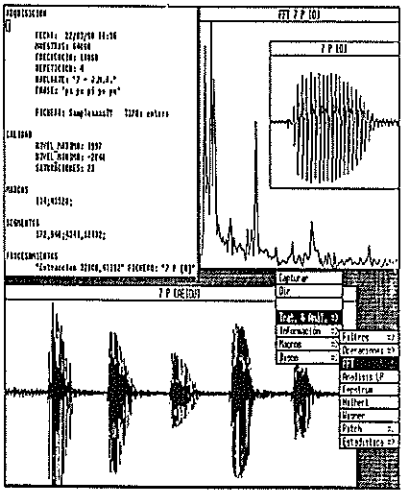


Figure 3 : The window at the left bottom corner displays the acoustic waveform corresponding to a spanish /pa/,/pe/,/pi/,/po/,/pu/ utterance. At the right top corner two windows show a segment of that signal and a sample of its spectrum. The text window contains some information related to the whole signal. The pop-up menus illustrate the selection of the FFT option.

4. SOFTWARE STRUCTURE

Figure 4 shows the package software structure. Apart from the signal acquisition/reproduction driver, the application consists of six modules which have well defined interactions. These are:

- User-Machine Interface (UMI)
- Graphics Manager (GM)
- Object Handler (OH)
- Information Manager (IM)
- Set of Treatment and Analysis Procedures (STAP)
- Autoprogramming Module (AM)

Next, we are going to see the piece of work each module performs.

The User-Machine Interface is the module that allows the user to do all kind of operations in a controlled and friendly manner. The interaction is carried out by means of an undetermined number of windows and pop-up menus. The user can situate and resize these items at his will. All sort of representations can share the display at a time.

The UMI is related with the Information-Manager allowing the user to ask for the information about any object situated on the screen or stored in the disk. This makes possible the addition of new comments to the available information too. The IM is mainly based on syntatic parsers (developed with **lex** and **yacc**) which work with text files that can be modified by the user. Normaly, the information is displayed by the user's editing program (invoked by the package). The changes are checked for syntatic correctness before their acceptance.

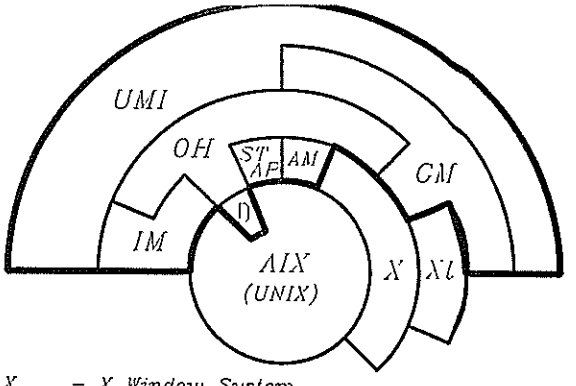
The Object Handler goes between the UMI and the other modules together with the external equipment. This module deals with the data access in a controlled manner using the information supplied by the IM. This makes the work more bearable thanks to the fact that:

- It prevents the user from making errors that could be dangerous for the data.
- It changes the data formats when necessary.
- It avoids repetitions of already applied treatments directly accesing to the previous results.

Also, the OH provides information to the IM about the applied processes, its parameters, and the representations requested. The AM is informed as well to keep track of the actions sequence.

The GM manages the graphic representations and it executes the graphic requests from the UMI (zoom, slice, resize, etc.). This module contains a set of procedures for 2D and 3D representations handling both colors and grey-levels.

The STAP is a set of mathematical procedures that are developed independently of the application structures. New



X – *X Window System*
Xt – *X Toolkit*

UMI – *User-Machine Interface*
GM – *Graphics Manager*
OH – *Object Handler*
IM – *Information Manager*
STAP – *Set of Treatment and Analysis Procedures*
AM – *Autoprogramming Module*
D – *AIX Device Driver (DT2901)*

Figure 4 : System Software Structure.

routines can be linked to the whole system by means of a configuration file and the UNIX linker-loader utility. The addition of a new routine does not require specific knowledge of the system but just that needed to write it. This feature allows to make use of the system as a workbench for testing new algorithms.

The Autoprogramming Module keeps track of the actions sequence reported by the OH in order to apply it to other sets of objects if required. The AM is capable of writing and interpreting text files containing actions sequences expressed in a structured way. So, the user can write his own programs for the package.

5. CONCLUSIONS

A software package for R & D in the speech processing area has been outlined. Its structure and the programming techniques used provides it a set of remarkable features:

- Flexible data structures
- Skillful data management
- Exploitation of UNIX multiprocessing capability
- Use of a powerful window environment
- Can be used as a workbench for algorithms.

Finally, let us remark that the state-of-the-art in computer technology permits us to obtain the whole of these features on a low-cost personal computer.

Final Note: At present (March 1990) the UMI and GM modules are quite fully developed and they are already operative. The other modules are being developed simultaneously in account of the flexibility of the growing technique adopted. They are in an intermediate stage according to the objectives marked for the first version.

ACKNOWLEDGEMENTS

We thank Prof. Dr. José Mariño from the ETSITB (Barcelona), Prof. Dr. Francisco Casacuberta and Prof. Dr. Enrique Vidal from UPV (Valencia) and their research groups. We have taken some ideas from their signal processing systems and their comments.

This work is being supported by the Basque Country University under grant UPV-224310- 0099/ 88.

REFERENCES

[1] Kay Elemetrics Corp., Sonagraph, (Commercial Workstation).

[2] Crump, John M., The Design of a Speech Analysis Workstation, 113th Meeting of the Acoustical Society of America (1987).

[3] Signal Technology Inc., ILS, Interactive Laboratory Systems. (Commercial Software Package).

[4] Shore, John, An Extensible File System for Signal Processing Software, in Proc. IEEE Int.Conf. Acoust., Speech, Signal Processing (1989) pp. 1083-1086.

[5] Morris, L.R., A PC Based Digital Speech Spectrograph, in IEEE MICRO Dec. 1988, pp.68-85.

[6] Kopec, Gary E., The Integrated Signal Processing ISP, in IEEE Transactions on Acoustics, Speech and Signal Processing, vol. ASSP-32, nº4 (1984) pp.842-850.

[7] Joy Mayol, M.A., Estación de trabajo para reconocimiento de voz, (Proyecto fin de carrera E.T.S.I.T. Barcelona, 1988).

[8] Tarela, J.M., Torres, M.I., Alcaide, J.M.,Bordel, G., Diseño y desarrollo de un entorno general para análisis y tratamiento de señal de voz, (Proyecto de investigación UPV/EHU 224.310-0099/88).

[9] McGilton, H. and Morgan R., Introducing the UNIX System (McGraw-Hill, New York NY, 1983)

[10] Rochkind, M., Advanced UNIX Programming (Prentice-Hall, Englewood Cliffs NJ, 1985)

[11] IBM, AIX Programming Tools and Interfaces (AIX Operating System Documentation, 1989).

[12] Kernighan, B.W. and Pike, R., The UNIX Programming Environment (Prentice-Hall, Englewood Cliffs NJ, 1984)

[13] Kernighan, B.W. and Ritchie, D.M., The C Programming Language (Prentice-Hall, Englewood Cliffs Nj, 1984)

[14] Nye, A., Xlib Programming Manual (O'Reilly & Associates, Sebastopol CA, 1989).

[15] Nye, A., Xlib Reference Manual (O'Reilly & Associates, Sebastopol CA, 1989).

[16] O'Really, T., Quercia, V. and Lamb, L., X Window System User's Guide (O'Reilly & Associates, Sebastopol CA, 1989).

[17] Egan, J.I. and Teixeira, T.J. (Writing a UNIX Device Driver, John Wiley & Sons, 1988)

[18] IBM, AIX Technical Reference vol.2, append. C (AIX Operating System Documentation, 1989).

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

EXPERIMENTAL RESULTS IN MINIMIZING ROUNDING ERRORS IN FIXED-POINT WFTA PROGRAMS

Ewa ŁUKASIK

Department of Computer Sciences,
Technical University of Poznań, Poznań, Poland

The paper presents the results of experimenting on the basic structures of the small-N Winograd-Fourier Transform Algorithms (WFTA) in order to minimize their errors in fixed-point arithmetic. Simulations programs have been prepared for the appropriate statistical calculations. The new rules for constructing final parts of the algorithms are formulated.The best of the structures are applied to the WFT modules of greater sizes. The resulting errors are reported.

1. INTRODUCTION

Finite word length effects in various FFT algorithms have been treated in literature repeatedly. In contrast the same problem concerning The WFTA (Winograd Fourier Transform Algorithm) has appeared in a rather small number of publications [1][2]. The error estimation in [1] was pessymistic in comparison with FFT. It was stated that the difference in the accuracy of FFT and WFTA reaches 1-2 bits. Since the WFTA has an interesting from the point of view of rounding errors feature: the unique multiplication stage, the problem has been revised in [3] in order to optimize the algorithm. After a thorough analyse several indications and suggestions for constructing the fixed-point WFTA permitting a minimization of rounding errors were given. This paper presents some practical results of implementing those rules and formulates some additional principles allowing further minimization of rounding errors in WFTA. They concern mainly the last stages of algorithms and became evident after the simulation experiments.

2. STATEMENT OF THE PROBLEM

In the paper real structures of DFT:

$$X(k)=1/N \sum_{n=0}^{N-1} x(n)\exp(-kn2\Pi/N), \quad /1/$$
$$n,k=0,1,..,N-1$$

are considered. For $|x(n)|<1$ that implies $|X(k)|<1$.

The WFTA modules consist of the polynomial reductions, computation of sample X(0) of the DFT, polynomial products (PP), polynomial reconstruction and distribution of data sample x(0) [4]. In real algorithms discussed here all the operations are real-valued. In order to prevent the unnecessary lost of accuracy it is assumed [3], that all the intermediate results are smaller then 1 but as close to 1 as possible. This may be achieved by introducing scalings. For the advisable small DFT modules: 2,3,4,5,8,16,17 ($N=2^r$ and Fermat prime numbers $N=2^r+1$) and for their combinations i.e. $N=2^t*15$ and $N=255*2^t$, t=0,1,2,3... scaling factors are equal to powers of 2, hence they may have form of shifting right after each addition/subtraction without any danger of reducing the result too much (overscaling).

All the multipliers should be fractional numbers, possibly very close to 1. Practically they are divided by N and multiplied by the greatest possible power of 2. Where the polynomial products are concerned it is stated in [3], that the best method allowing the minimal rounding error is to compute PP directly from the definition formula or to use PP mod z^k+1, $k=2^t$, also repeatedly.

The above quoted paper [3] did not deal with the last stages of PP interpolation and reconstruction operations assuming, that their structre is strongly dependent on the way the PP is developed. However the simulation results show, that their influence cannot be omitted, as they, for certain extent, model the final distribution of errors. That is because firstly, they combine the samples carrying different errors from previous stages of the algorithm, secondly, their structure is often asymmetric, i.e. various paths of graphs are divided by different powers of two and may introduce different errors. As the rearrangement of the final subgraphs seemed to be possible, some examinations of those parts of the WFT

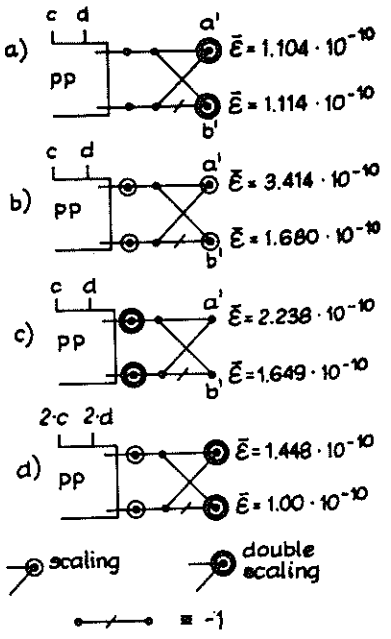


Fig.1. Various scalings in polynomial interpolation and resulting errors. a,b∈(-1,1) c,d∈(-0.5,0.5), PP-polynomial product mod z^2+1

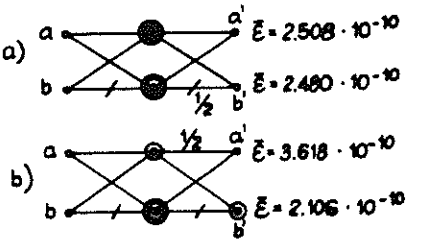


Fig.2. Errors in asymmetric butterflies

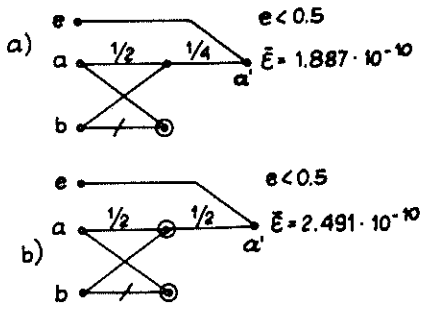


Fig.3. Errors introduced by two subsequent asymetric butterflies

algorithm were carried out in order to find the least erroneous of these structures.

The first structure analyzed is the 15-point DFT module created from 3-point and 5-point ones. Polynomial products mod z^2+1 are left to be computed directly from the definition formula. Other modules of interest are 17-point one as well as recommended in [3] $N=2^3*15=120$ and $N=15*17=255$. The polynomial products to be computed directly have rank 2, so it is necessary to develop PP mod z^4+1 and PP mod z^8+1 algorithms in 17-point module [3].

3. METHOD OF ERRORS MEASUREMENT

Statistical approach is applied to measure the WFTA errors. The Gaussian noise samples form the input data. The error discussed is mean-square error defined as the squared difference between the precise (floating point) and 16-bit simulated WFT samples averaged for 1024 calculations.

Two main criterions for algorithms comparison are introduced:

- maximal mean-square error $\bar{\varepsilon}_{max} = \max_k [\bar{\varepsilon}(k)]$,
- average mean-square error $\bar{\varepsilon}_{av} = 1/N \sum_{k=0}^{N-1} \bar{\varepsilon}(k)$,

$\bar{\varepsilon}(k)$- sum of mean-square errors of real and imaginary part of WFT sample.

For the algorithms of bigger sizes (120-point and 255 point) the additional deterministic criterions are applied:
- maximal value of maximal WFTA square error of sequences of input signals: a) sine- N/2 succesive harmonics, b) cosine - N/2 successive harmonics, c) N successive Dirac impulses,
- maximal value of mean square errors of sequences of N/2 (N) input signals: a)sine, b)cosine, c)Dirac impuls.

They are introduced to find the possible confirmation of another point of view on the minimization results.

All the measures to minimize the WFTA structures are taken mainly for rounding. The effects of optimization are also checked for truncation and random rounding i.e. rounding with the exception of results having a part equal 0.5, which are randomly either truncated or rounded.

4. SIMULATION RESULTS

Consider column 1 in table 1. representing $\bar{\varepsilon}_{re}(k)$ and $\bar{\varepsilon}_{im}(k)$ calculated for 15-point not-optimized module. Some of the samples have much bigger errors than the others (e g. 0,1,2,7,8,9,11). Statistical errors caused by the multiplications are rather small (column 2 in table 1.). On the other hand errors introduced by the last stages of the algorithm are again rather big and, what is more, they have the structure similar to that of final errors (column 3). If we analyse the algorithm we may see, that the most erroneous are these samples, which are the result of the greatest number of additions and of the butterflies of asymmetric structures. The insight into the 17-point module algorithm and its errors confirms the same tendency (see column 1,2 table 2.). The most erroneous are the samples being results of PP mod z^8+1 evaluated by five stage PP mod z^2+1 and these calculated in asymmetric butterflies. Let us then find the optimal structures of interpolation algorithms and asymmetric butterflies. Figs. 1.to 3. present a part of these structures and experimental results received for 4096 averages.

The experiments conclusions are as follows:

1. Later the scaling is done, smaller is the error produced. Preferably scalings should be done at the end of the algorithm. Double scaling is better then simple one (fig. 1. a,b,c,d).
2. The delay in scaling is not recommended on the cost of introducing the butterflies with the "asymmetric" paths entering the node i.e. the structure in which one of the paths (or both) is scaled by the power of 2 (15-point and 17-point DFT modules -b' sample in fig. 2.a,b).
3. Additional earlier scaling produces smaller error then "asymmetric" butterfly (a' sample in fig.2 a,b)
4. If two subsequent "asymmetric" butterflies are unavoidable (15-point DFT module), less erroneous is the structure in fig 3.a.
5. The interchange of addition/subtraction operations leads to higher accuracy.

These conclusions are very useful to rearrange the 15- and 17-point algorithms. It must be stated, that the criterion applied to choose the best butterfly was rather equalization of errors, not the absolute minimization of one error on the cost of the increase of others. But if there is such a need from the point of view of algorithms in which these modules are nested, the structures may be revised and rearranged. The calculation results for optimized 15-point module is presented in columns 4 (final adds) and 5 (WFT), table 1 and for 17-point module in columns 3 (adds) and 4 (WFT) in table 2.

Table 1. Errors introduced by various stages of 15-point WFTA $*10^{-10}$

	WFT	M	FA	FA_{opt}	WFT_{opt}
0.	14.0	1.0	4.4	2.7	9.4
1.	8.3	1.0	8.2	5.1	4.8
2.	4.9	1.1	8.1	4.5	2.4
3.	1.5	1.9	1.1	2.1	1.6
4.	1.4	1.9	1.1	2.3	1.3
5.	1.5	1.3	1.9	2.5	2.2
6.	3.2	1.2	4.3	2.4	3.4
7.	6.6	1.2	4.0	2.6	4.3
8.	6.4	1.9	4.8	3.7	3.9
9.	4.6	2.4	5.0	3.6	2.9
10.	2.0	2.0	9.7	1.0	2.1
11.	7.1	2.0	4.4	3.2	6.8
12.	3.4	1.3	4.2	3.6	2.7
13.	2.1	2.1	2.3	1.9	2.4
14.	2.6	2.9	2.0	2.0	2.0
WFT	ε_{av}= 8.7			ε_{av}=6.6	
	ε_{max}=12.9			ε_{max}=8.9	

Table 2. Errors introduced by final additions in 17-point WFT and final WFT error $*10^{-10}$

	WFT	FA	FA_{opt}	WFT_{opt}	
0.	7.7	1.5	1.5	11.4	
1.	3.8	1.6	1.6	4.3	
2.	15.0	1.6	1.6	12.4	
3.	10.1	1.7	1.7	9.5	
4.	10.2	1.7	1.7	8.2	
5.	10.7	4.8	2.2	4.7	
6.	4.3	6.1	4.5	3.7	
7.	6.8	6.1	5.9	6.6	
8.	3.8	4.7	3.8	3.2	
9.	15.6	15.3	7.1	6.9	
10.	12.4	12.2	10.1	9.2	
11.	14.6	13.5	5.2	5.2	PP mod
12.	17.3	17.4	12.1	13.2	
13.	18.5	19.0	9.5	10.7	z^8+1
14.	18.9	17.9	15.3	17.1	
15.	18.9	19.4	12.2	12.9	
16.	15.4	13.6	9.4	9.8	
WFT	ε_{av}=22.6			ε_{av}=16.6	
	ε_{max}=29.2			ε_{max}=21.6	

For N=15 there is an evident decrease of the errors of sample 0,1,2,7,8, which is related to the appropriate decrease of errors introduced by the final structures of addings. For N=17 the most erroneous are the samples being results of PP mod z^8+1 computation. The simple change of final stages of interpolation give the noticeable improvement reaching $9.3*10^{-10}$ for the sample number 11. Taking into account the complete error for real and imaginary part of the sample the maximal improvement reaches value $13.72*10^{-10}$. The improvements got for both modules for avarage and maximal mean-square errors are also proven (see bottom part of tabl. 1,2).

The above mentioned results refere to the quantisation performed by rounding. The same experiments for random rounding and truncation notify the improvement too, although not so evidently. It may lead to the conclusion, that for any quantisation method applied the analyse should be performed separately to find the optimal structure.

In table 3. the 120- and 255-point old and new structures are compared according to both statistical criterions introduced in Sect.3. The good results achieved by the optimization measurements are rather evident, although the improvements for truncation are not straightforward, as for N=120 average error is slightly bigger then before.

The same tendency is observed for the deterministic criterions. Again for truncation the old structure give sometimes (for N=120) more accurate results.

5. CONCLUSIONS

In the paper the final subgraphs of the WFT algorithms are studied in order to check their influence on the accuracy of the fixed-point calculation results. Several experiments were carried out. They showed the increased (in comparison with multiplications) influence of addition/subtraction errors on the accuracy of final results and proved that the rearrangements of the sequences of the "asymmetric" butterflies (having different scaling factors in their paths) may cause the significant reduction of rounding errors introduced. Implementing the best of these structures to the 15- and 17-point WFTAs and next to the bigger, 120- and 255-point modules, provides satisfactory improvement of final results. It is also stated that the optimization procedure should be performed individually, according to an algorithm and a quantization method. Complete analyse of the effects of all the measures taken to minimize rounding errors in WFTA programs will be given in the forthcoming paper.

Table 3. WFTA optimization results

	WFT	R	RR	T
		$*10^{-10}$		$*10^{-9}$
$\bar{\varepsilon}_{av}$	120	9.2	5.9	4.44
	120_{opt}	7.5	5.5	4.84
	255	33.1	20.7	20.8
	255_{opt}	26.6	19.6	19.3
$\bar{\varepsilon}_{max}$	120	31.7	8.4	22.5
	120_{opt}	23.5	7.5	19.9
	255	96.1	26.1	111.5
	255_{opt}	69.0	26.0	92.1

R-rounding, RR-random rounding, T-truncation

ACKNOWLEDGEMENTS

The author wishes to thank dr R. Stasinski for all the stimulating discussions, theoretical support and encouragement during the experimental works.

REFERENCES

[1] Patterson R.W, McClellan J.H., Fixed-point error analysis of Winograd-Fourier transform algorithms, IEEE Trans. Acous. Speech Signal Proces., Vol ASSP-28, 1978, pp.447-455.

[2] Panda G., Pal R.N, Chatterjee B, On the effect of correlation between truncation errors in fixed-point error analysis of Winograd short-length DFT algorithms, IEEE Trans.Acoust., Speech Signal Proc., Vol. ASSP-30 pp.100-104.

[3] Stasiński R., Łukasik E., Minimization of rounding errors in WFTA programs, Proc. of ICASSP, New York, 1988,vol D, pp 1423-1426

[4] Stasiński R., The WFTA for computing the cosine/sine DFT in one or more dimmensions, in V.Cappellini and A.Constantinides (Eds.): Digital signal Processing - 84" (Elsevier Sci. Publ., North-Holland, 1984), pp.133-139.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

REPRESENTATION AND PROCESSING OF MULTIDIMENSIONAL SIGNALS IN THE OBJECT-ORIENTED SIGNAL PROCESSING SYSTEM QUICKSIG

Matti Karjalainen

Helsinki University of Technology, Acoustics Laboratory
Otakaari 5 A, 02150 Espoo, Finland

Object-oriented programming (OOP) is one of the most important new paradigms to make complex software systems feasible and to improve the productivity of the programmer. The topic of this paper is the formulation of multidimensional signal processing problems in terms of OOP methodology. Some general background concerning OOP in DSP is given with a description of the QuickSig system that is the experimental environment of this study. The representation of multidimensional DSP objects and processing of signal objects are discussed in QuickSig terms. One of the objectives of the study has been to increase the flexibility of experimentation and algorithmic development. This is achieved by software integration, and the interactivity and conceptual clarity of the OOP-based formulation. Another important goal has been the possibility to use the same object-oriented approach when targeting programs to fast signal processors.

1. INTRODUCTION

Object-oriented programming is a paradigm that has started to make a breakthrough in all fields of computer programming. It is both a way of thinking and a tool for knowledge representation in systematic program development. OOP features have emerged in artificial intelligence languages like Lisp and Smalltalk. Flavors [1], Common Loops [2] and the new Common Lisp Object System (CLOS) [3] are examples of flexible object-oriented extensions to Lisp. C++ [4], as an extension to the C language, will probably be the most popular OOP tool in production-level programming.

Common features of OOP languages are *object class definitions* including *inheritance* of class properties like *instance variables* and *method functions*. *Instances* of classes are created to be used for computation by *message passing* or *generic functions*. Some of the most prominent advantages of object-oriented programming are *high modularity* and *good reusability* of software constructs, *easy maintenance* and *reconfiguration* of programs, *compactness* of code, *conceptual clarity*, and highly *improved productivity* by rapid prototyping and incremental redesign.

Object-oriented formulation of signal processing has evolved during the 1980's: originally proposed by Kopec [5], [6] and futher developed e.g. by Myers [7] and Karjalainen et. al [8], [9]. The first experiments and implementations were developed in the highly flexible Lisp language. Smalltalk is also gaining popularity as a rapid prototyping platform and soon these principles will migrate to less flexible but more efficient and standard object languages such as C++. Signals and systems represented and modelled as objects is a natural and systematic way to develop modern DSP software in order to gain the advantages of object-oriented programming.

2. THE QUICKSIG SYSTEM

QuickSig [8], [9] is an experimental object-oriented signal processing environment that includes multiple approaches and features to support DSP research and algorithm development. It is based on the Common Lisp language and the Flavors objects that are to be converted to the standard CLOS objects. QuickSig includes a wide variety of predefined object classes like signals, filters, windowing objects, etc., as well as related generic functions for operating on these objects. There is also support for an interactive graphic user interface, a signal database, code generation for DSP processors, special support for speech processing applications, etc.

The main part of QuickSig is designed for one-dimensional signal processing but multidimensional signals are as well suited to object-oriented formulation. The aim of this paper is to present the different ways of representing and processing multidimensional signals, the main emphasis being on two-dimensional signals.

3. SIGNAL OBJECTS IN QUICKSIG

The class of one-dimensional signals (SIGNAL) in QuickSig is composed of several superclasses. First, a SPAN is simply an encapsulation of two integers to denote an index interval. The SCALE-SPAN is inherited from it by adding two new features: SCALE and SCALER. SCALE keeps a symbol, e.g. time, frequency, etc., to know the dimension (domain) where a signal object is defined. SCALER is a real number to map the index values to the real-valued axis of SCALE. SIGNAL is inherited from SCALE-SPAN and includes a sample array (S-ARRAY) to store the signal samples within SPAN. By default the value of samples outside the span is 0.0 in order to support a virtually infinite span. See Fig. 1 for the class inheritance of these objects.

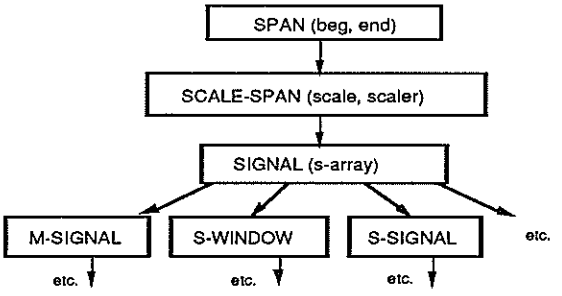


Fig. 1. Part of the class inheritance in QuickSig.

The class SIGNAL can further be used as a root of more complex objects like signal windows (S-WINDOW) and multichannel (M-SIGNAL) as well as signal-valued signals (S-SIGNAL) to be discussed later.

3.1. Span and interval processing

Span is a low level object class which represents index ranges. *Span* objects have two primary integer-valued slots: *beg* (for beginning, the first index included) and *end* (the first index after the span, not included), see Fig. 2. Some related secondary properties are *size* (= end-beg, the number of index points) and *stop* (= *end*-1, last index included).

Interval is another kind of range object that has the primary properties *beg-point*, *end-point* and *scale* with corresponding access functions. Intervals are not related to index numbers in any way. The need for intervals and scale-spans as separate objects arises from the difference between discrete indices and real-valued points as well as from the roundoff error when converting between them.

Automatic span processing is carried out to increase or decrease the size of the span and sample-array as needed by the logic of a given DSP operation. For example, addition of two signals normally results in a union-span of the arguments and multiplication in an intersection-span, respectively (Fig. 2). Other examples are *correlation-span* and *convolution-span*.

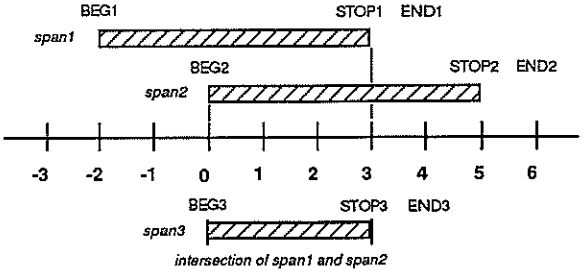


Fig. 2. Example of span processing in QuickSig: *intersection-span* of *span1* and *span2*.

Figure 3 shows how two signal objects are added to yield a new signal:

(add sig1 sig2) => sig3

Automatic span processing by *union-span* is carried out to hide the memory allocation details from the user. If needed, however, the user can control the details of operations by optional and keyword arguments. Data abstraction by classes like SIGNAL and procedural abstraction by generic functions like *add* make QuickSig conceptually clear and flexible for algorithmic development and fast prototyping of DSP software.

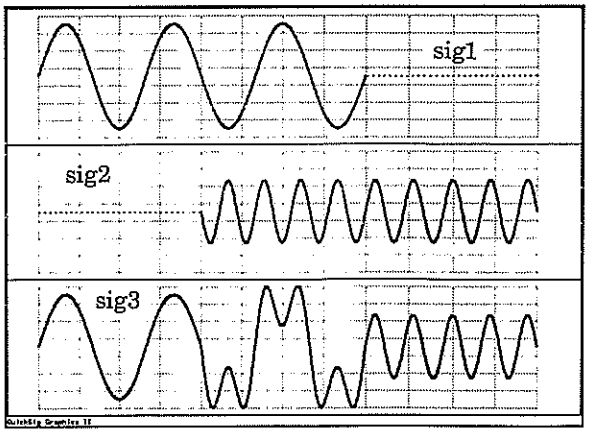


Fig. 3. Addition of two signal objects with the related *union-span* processing.

4. MULTIDIMENSIONAL SIGNAL OBJECTS

There are two simple extensions to the one-dimensional signals: M-SIGNAL (multi-channel

signal) and S-SIGNAL (signal where samples are signals, e.g. for representing spectrograms), see Fig. 1. Both of them are inherited from SIGNAL and are based on the fact that in the Lisp language arrays and also signal objects may have samples of any data type, not only numeric values. These are special cases of multidimensional signals, however. A more general formulation is needed for two- and N-dimensional signals. This may follow the same principles as the implementation of one-dimensional signal.

For higher compatibility of objects of different dimensionalities it is useful to define new classes (data types) of 2D-INDEX, 2D-POINT, 3D-INDEX, 3D-POINT, etc., as generalizations of integers and real numbers. They have a special notation in QuickSig, e.g.:
#i(10 20), #i(3 -4 5) for 2D- and 3D-index values,
#f(1.0 2.0), #f(3.0 4.0 5.0) for 2D- and 3D-points.

Now it is relatively straightforward to define new classes like X-SPAN, Y-SPAN and the combination 2D-SPAN, corresponding SCALE-SPANs, 2D-SIGNAL, three-dimensional object classes, etc. Here we will concentrate on the properties of 2D-SIGNALs.

The domain of a 2D-SIGNAL object is primarily defined by a 2D-SPAN (2D index interval) and secondarily by a 2D-SCALE-SPAN (2D point interval). Samples of a signal are kept in a 2D-array. Samples outside the 2D-SPAN have the value 0.0 so they don't have to be stored. Automatic span processing can be carried out like in the 1D case but some issues of practicality must be considered more carefully.

4.1. Two-dimensional span processing

2D span processing operations like *union-span* and *intersection-span* operate on rectangular index or point intervals. Spans of a more complex form (e.g. round regions or disjoint sets of rectangular spans) are motivated only in special cases. Fig. 4. illustrates the processing of *union-span* in a typical 2D case.

Addition (add) is an operation where this kind of 2D *union-span* is needed. 2D signals corresponding to s1 and s2 in Fig. 4 can be added in nine homogeneous subparts. Each subregion may represent one of three alternatives: use of the default 0.0 if neither s1 nor s2 is covering the subspan, use of a signal sample directly if one signal covers the subspan, or adding the corresponding samples if both s1 and s2 cover the subspan.

The most efficient way of carring out the addition in low-dimensionality cases is normally to precompute the possible subspans and then to process each subspan by a direct loop over the proper index ranges. When the dimensionality (d) grows the number of possible subspans is proportional to the third power of d so that the partitioning of the total span may become a considerable overhead in the addition function. In such cases it might be more efficient and in any case more straightforward to define index or point operations (as generic functions) and to use them for the implementation of addition.

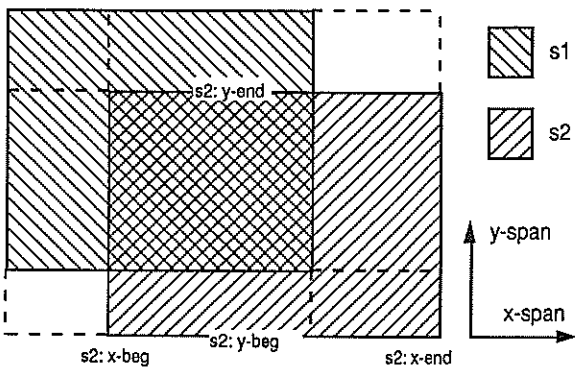


Fig. 4. Processing of union-span of in the 2D case.

An index access function *at* is characterized by the following examples:
(at sig1 1) => value at index 1
(at 2d-sig #i(2 3)) => value at index #i(2 3).

Now the summation of 2D signals can be carried out homogeneously over the full union-span of two source signals because the access function *at* is defined properly for all infinite 2D index points. Notice, however, the extra check needed for the span inclusion and possible array access computation for each index point.

The use of the sample access function *at* is extended to cover the assignment of values to index positions of signals. *Setf* is the generalized assignment form of Lisp and it can be used conveniently for the purpose:
(setf (at 2d-sig #i(5 6)) 3.14)
to write the sample value 3.14 into the desired position. Span processing guarantees that if the index position is outside the the existing span the bounds of the span are first expanded and the sample array updated to include the new index position. This is very convenient to the user or programmer since the span processing is automatic and hidden unless specific control over it is called for. It is also somewhat dangerous because the user might easily ask for unpractical or impossible memory allocation. Due to this, special versions of *setf at* -form are useful where the bounds check is carried out and an error is generated or no assignment at all occurs if writing outside the existing span is attempted.

It is conceptually straightforward to expand the multidimensional signal objects described above to support typical DSP operations, including linear and nonlinear memoryless operations, convolution and correlation, filtering and transforms. The generic nature of the formalism makes it very flexible for fast algorithmic prototyping in a conceptually clear manner.

5. APPLICATION PROGRAMMING FOR DSP PROCESSORS

Automatic memory management, i.e. dynamic allocation and garbage collection, found in the Lisp language is an important flexibility feature in algorithmic prototyping. In practical applications, however, it leads to relatively slow and memory consuming implementations of DSP algorithms and data structures. Especially signal processor applications often need optimized code to achieve real-time performance with a minimum number of processors and memory elements. Automatic span processing of multidimensional signals, for example, is typically too resource consuming. Static allocation of data structures is needed.

We have approached the problem from the point of view of OOP methodology. The present extension to QuickSig contains tools for the TMS 320C30 floating-point signal processor that utilizes Lisp and CLOS to improve the productivity but to retain all efficiency features of assembly programming.

The programming environment is fully integrated to Lisp and CLOS. The starting point is an assembler for the TMS320C30 that follows Lisp syntax. The next step is the formulation of processors as object classes so that multiple instances of the processor can be managed. Finally, a special version of a CLOS object-oriented environment for the C30 is designed with syntactic compatibility with the original CLOS. This is used to define data types and structures (by classes) for the C30. Due to efficiency reasons the procedural part of programming is typically written in the Lisp-syntax assembler. In the future there will also be an optimizing compiler available for the TMS320C30.

This special CLOS environment that runs on the Apple Macintosh II supports static typing and memory allocation for the C30 objects. Built-in classes include e.g. integer, float, complex, boolean, a wide variety of arrays, etc. The DSP class library contains e.g. filters and FFT transform objects. The user can easily define new classes and data types. When instantiated they generate objects that primarily reside on the host Macintosh but those parts of the objects that are needed for time critical or run-time DSP are allocated on the C30 signal processor. The seamless integration of low and high levels improves the productivity of application programming considerably.

To show an example, in the most static form a 2D signal might be allocated so that only the array for the samples resides on the signal processor. All other information like span properties are part of the CLOS object on the host machine and are used only for program development and debugging purposes. In another formulation it might be desirable to allocate span properties on the C30 for run-time purposes. In a third case there might be dynamic memory management for the sample array on the C30 so that resizing of it is allowed, etc.

The object-oriented formulation of the whole programming environment in QuickSig makes it systematic, flexible, efficient and productive on all levels of programming, including multidimensional signal processing.

ACKNOWLEDGEMENTS

This study is a part of project "Symbolic Signal Processing", financed by the Academy of Finland. My thanks are especially due to Toomas Altosaar who has implemented many of the basic features of the QuickSig system.

REFERENCES

[1] New Flavors documentation in: *Symbolics Common Lisp Language (2A)*. Symbolics Inc., Cambridge, MA, 1987.

[2] D.G. Bobrow et al., "CommonLoops: Merging Lisp and Object-Oriented Programming", in *Proc. ACM Conf. on Object-Oriented Systems, Languages and Applications*, 1986.

[3] D.G. Bobrow et al., *Common Lisp Object System Specification*, X 3J13 Doc. 88-002R, 1988.

[4] B. Stroustrup, *The C++ Programming Language*, Addison-Wesley, Reading, MA, 1986.

[5] G.Kopec, *The Representation of Discrete-Time Signals and Systems in Programs*. MIT PhD Thesis, Cambridge, MA, 1980.

[6] G.Kopec, "The Integrated Signal Processing System ISP", *IEEE Transact. ASSP*, vol. ASSP-32, No. 4, Aug. 1984.

[7] C.Myers, *Signal Representations for Symbolic and Numerical Processing*, PhD Thesis, MIT Technical Report No. 521, Aug. 1986.

[8] M.Karjalainen, T.Altosaar and P.Alku, "QuickSig - An Object-Oriented Signal Processing Environment", In *Proc. of IEEE ICASSP-88.*, New York, 1988.

[9] M.Karjalainen, "DSP Software Integration by Object-Oriented Programming: QuickSig Case Study", To be publ. in *IEEE ASSP Magazine*, 1990.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

On Arithmetic Implementation of Orthogonal Linear Algebra Signal Processing Algorithms

R.W. Stewart and R. Chapman

Signal Processing Division
Department of Electronic and Electrical Eng.
University of Strathclyde
Glasgow G1 1XW
Scotland

Abstract

In this paper the arithmetic implementational issues of linear algebraic signal processing algorithms are considered. More specifically this has entailed the design of efficient arithmetic arrays for square root and division which can be exploited in parallel signal processing arrays. The ideas and statements presented in this paper relate primarily to the next generation of fast digital signal processing algorithms and architectures for parallel signal processing.

1 Introduction

Recently many signal processing algorithms and architectures have been presented in forms that avoid the square root computation. The reasons stated for this are simply that this operation is very slow compared to multiply, divide and addition, and would be costly to implement. Furthermore there exists many so called *fast* (square root free) algorithms that seemingly allow (parallel) efficient implementation. As a result a number of authors have chosen to avoid the square root operation.

In this paper it is shown that square roots can be implemented both faster and in a smaller chip area than divisions. Therefore the commonly perceived implementational complexity hierarchy of (1) addition, (2) multiplication, (3) division, and (4) square root, actually has square root in third place and division in fourth. Despite the original intention to reduce the computational complexity the use of square root free linear algebra algorithms actually *increases* the computation load over the original algorithms. Furthermore numerical stability is often compromised. The paper first looks at various methods of calculating the square root and presents arrays for square root/division. Later sections consider the use of the square root in linear algebra algorithms.

2 Square Root Computations

The Direct Method: An iterative, rather than recursive, method of realising the square root is by exploiting the odd-series relationship of perfect squares. This algorithm is also called the *direct* method since each cycle will yield a significant digit of the square root. One method of using this relationship to extract the square root of the binary number A is essentially a restoring division process [16], where successive subtrahends are realised based on the formation of the odd-series. A number of authors independently presented similar arrays for square rooting based around the direct method of square rooting [6] [9] [11].

From an analysis of the non-restoring method a 2-D array for the computation of an N-bit square root from an N-bit operand was realised as shown in Figure 1 [14]. An array to calculate the N-bit quotient from an N-bit dividend and divider based on the non-restoring divide algorithm is easily derived [16]. It can be noted that the square root array requires only half of the silicon area of the division array [15]. These two arrays can be *combined* to give the dual purpose array of Figure 2. A single bit controls whether the array is performing square roots or divisions. From these two figures it can be seen that the critical path for the square root computation is half that of the division, hence in an array with asynchronous cells, square roots will be performed *twice* as fast as divisions. This shared array can be pipelined or efficiently mapped into a one dimensional array [15]. If floating point numbers are to be used the above arrays perform the mantissa calculations, and the exponent handling is done elsewhere (a shift in the case of square roots and a subtraction step for divides).

Look-up Tables: With memory becoming cheaper and device sizes smaller, it is not unlikely that on-chip tables could be used for storing the square root values. However for an N-bit mantissa a direct lookup table would require up to 2^N entries. (Using a 32-bit mantissa, 4 Giga memory locations are required for a look-up table only, note however this would be smaller than the analagous division look-up table.) For parallel linear algebra arrays storing values off-chip is implausible, since there is already considerable I/O bottlenecks in these array processors [10]. A compromise solution is using a reduced look-up table to provide a *good* initial guess for a recursive algorithm which then converges to the solution.

Approximation Methods: Using approximation functions to calculate squares roots has also been suggested. In [17] a simple shift and add approximation function to $\sqrt{x^2+y^2}$ is analysed. The algorithm guarantees an error not worse than $\pm 4\%$. However this level of error is of no use in linear algebra calculations where numerical accuracy is important [5]. (It is interesting to note that using the approximations in a molecule graphics application a systematic error was introduced whereby circular objects when successively rotated became polygon shapes.)

Redundant Number Systems (RNS): RNS systems play an important part in ultra-high speed, dedicated, real time

systems since they can perform addition steps that are *carry free*, as a consequence of the redundancy encoded into the number representations [10]. RNS systems allow computations to proceed MSB first which initially is attractive for divide/square root computations. In [3] an RNS divider unit and a combined multiply, divide and square root unit using RNS was developed.

Logarithmic Number Systems (LNS): Numbers in LNS are represented as a signed radix raised to some sign exponent. If the radix is fixed all the arithmetic in the system can be performed using only the exponent. Using LNS, multiply and divides are now adds and subtracts, and squares and square roots only require a simple shift operations. However adds and subtracts are more complex requiring the use of a look-up table to calculate a logarithmic base two function. Also in an LNS system it would be necessary to convert between say floating point format and LNS. The look-up table could be of considerable size depending on the input range of the system.

DSP Chips: State of the art DSP chips use recursive (or converging) methods, such as Newton Raphson to calculate divisions and square roots [18]. Such techniques fully occupy the on-chip multiplier and ALU. Using a dedicated divider/square root array would be a better and faster solution assuming the chip area is available. To date no DSP chips perform square roots faster than division, hence the prejudice against square rooting is perpetuated.

For all of the above methods it is important to note that the argument that square roots are potentially *simpler* and *faster* to implement still holds true regardless of the number representation, system or base used.

3 Linear Algebra Algorithms

Building on the rapidly advancing parallel array processor developments linear algebra algorithms have become popular in signal processing. For example, Gaussian elimination is an elegant algorithm for LU decomposition, and general linear system solution. The algorithm requires $O(N^3)$ computations which quickly becomes prohibitive for real time solutions as N increases. The high number of computations led to parallel arrays of processors being investigated [10]. These (systolic) processor arrays allow the natural locality and recursiveness of many linear algebraic algorithms to be exploited. However having unearthed the parallelism numerical stability must also be an issue. Using fixed wordlength, algorithms such as Gaussian elimination are very unstable, and for large dimensions will invariably blow-up [5]. Hence the algorithm has limited use. The next step therefore was towards orthogonal methods such as QR decomposition. In its standard *normalised* form this algorithm has excellent numerical characteristics [5].

The QR via Givens transforms maps very elegantly onto a triangular array of processors [10]. The processor nodes for this algorithm require, a capability of square root, division and multiplication. Herein lies one of the areas of prejudice against the square root. A number of authors identified that square roots were *expensive and awkward to calculate,* and therefore reformulated the algorithms and applications to aviod square roots [2], [4], [7] [12]. This is an unfortunate step backwards as the algorithms are now susceptible to overflow/underflow problems and therefore lose their good numerical properties. In an effort to circumvent these problems, stabilising partitioning strategies were introduced. This brings two problems, the first is an increase in computational cost (divisions and multiplication), and secondly the algorithms lose their locality, and regular array implementations become difficult.

Figure 3(a) shows the general QR triarray and Figures 3(b) and (c) shows the standard Givens transformations and one form of the square root free Givens transformations. It can be shown that the square root free Givens transformations will suffer from overflow/underflow problems and some versions can be numerically unstable [15] (although certain anomalies have been noticed [8]). Figure 4 compares the data flow for the ASIC processors for both standard and square root free Givens transformations. Based on the earlier statements that square roots are simpler than divisions, clearly the standard Givens requires less computation contrary to the conclusions of a number of authors.

4 Matrix Square Root Covariance

In many filtering problems such as recursive least squares and Kalman filtering, the propagation of the error in the covariance matrices can result in a matrix that is not positive semidefinite (a theoretical impossibility). This occurs either when a linear combination of the state vectors are known with high precision but others are virtually unobservable, or when the covariance matrix is rapidly reduced by processing very accurate measurements. Both situations can lead to numerical problems of ill conditioned quantities. Reformulating the algorithms in terms of a square root covariance will preserve the non-negative definiteness of the computed covariance.

In linear algebra the Cholesky factor [5] is often referred to as the square root of a matrix. The method of propagating the matrix square root covariance matrix rather than the actual covariance matrix is completely successful in maintaining the positive semi-definiteness of the error covariance matrix. The Cholesky decomposition of a matrix S is denoted as:

$$\mathrm{S} = \mathrm{S}^{1/2}\mathrm{S}^{T/2} \tag{1}$$

where $\mathrm{S}^{1/2}$ is reserved for a lower triangular matrix. Note that the product of $\mathrm{S}^{1/2}\mathrm{S}^{T/2}$ can never be indefinite even in the presence of round-off noise and the numerical conditioning of $\mathrm{S}^{1/2}$ is generally much better than that of S. This can be simply shown by considering the condition number of S denoted by

$$\kappa(\mathrm{S}) = \frac{\lambda_1}{\lambda_n} \tag{2}$$

where λ_1^2 is the maximum, and λ_n^2 is the minumum eigenvalue of $\mathrm{S}.\mathrm{S}^T$. When computing in m-digit arithmetic problems can be expected as $\kappa(\mathrm{S})$ approaches 2^m. The real advantage of the matrix square root approach is clear from considering that:

$$\kappa(\mathrm{S}) = \kappa(\mathrm{S}^{1/2}.\mathrm{S}^{T/2}) = \kappa(\mathrm{S}^{T/2}))^2 \tag{3}$$

i.e. the condition number of $\mathrm{S}^{1/2}$ is the scalar square root of

$\kappa(S)$ and therefore the matrix square root formulation of a problem will not expect problems until $\kappa(S) = 2^{2m}$. Therefore the numerical precision has been effectively doubled.

A scalar square root free form of Cholesky exists, namely the LDL^T factorisation. Two numerical problems can be highlighted with the LDL^T approach. Firstly the matrix square root free orthogonal transformations will suffer from overflow/underflow and in some cases are unstable. The LDL^T decomposition of a matrix S is denoted as:

$$S = LDL^T \tag{4}$$

where L is unit lower triangular and D is diagonal. The condition number of S is denoted by

$$\kappa(S) = \kappa(LDL^T) \tag{5}$$

$$\leq \kappa(L)\kappa(D)\kappa(L^T) \tag{6}$$

Since the eigenvalues of a lower triangular matrix are the same as the diagonals, then λ_{max} and λ_{min} of L is 1. Hence comparing with Eq. 2

$$\kappa(D) \leq \frac{\lambda_1}{\lambda_n} \tag{7}$$

where λ_1^2 is the maximum, and λ_n^2 is the minimum eigenvalue of $S.S^T$. As stated above when computing in m-digit arithmetic, problems can be expected as $\kappa(S)$ approaches 2^m. Unlike the real matrix square root factorisation (Cholesky) this LDL^T computation requires to compute D which can expect numerical problems when $\kappa(S) = 2^m$, whereas the standard matrix square root formulation will not expect problems until $\kappa(S) = 2^{2m}$. Therefore one of the advantages of the square root covariance method has been lost, namely the effective doubling of numerical. Hence there is considerable computational and implementational advantages to be made by using Cholesky (with square roots) rather than LDL^T (without square roots) [13].

5 Conclusions

This paper has presented arrays for square root and division. It has also presented the arguement that in actual fact square roots are *half* as simple to implement as divisions in VLSI terms. (A possible conjecture here is that as squaring is *simpler* than the multiplication of two numbers, primarily because only one operand is used, it may be argued that the same holds true for square rooting when compared to division.) Part of the signal processing community's bias against the square root has been caused by the slower benchmarks of square root computations compared to division on current state of the art processors. Furthermore most people are aware how to calculate multiplications and divisions mentally, however are not aware of the standard methods for square roots. Hence they have a natural built-in resistance to using square roots, and will avoid them if possible. For standard Cholesky and QR algorithms and the new class of fast QR and lattice algorithms square roots are imperative for fast, efficient and stable calculation of these algorithms.

References

[1] M.A. Andrews. *Mathematical Microprocessor Software.* IEEE Micro, Vol.2, No.2, pp. 63-75, May 1982.

[2] J. Barlow and I.C.F. Ipsen. *Scaled Givens rotations for the solution of linear least squares problems on systolic arrays.* SIAM J. Sci. Stat. Comput, Vol. 8, No. 5, pp. 716-733, September 1987.

[3] M.D. Ercegovac and T. Lang. *Implementation of module combining multiplication, division and square root.* In Proc. International Symposium on Circuits and Systems, May 1989.

[4] W.M. Gentleman. *Least squares computation by Givens transformations without square roots.* J. Inst. Mathematical Applications, Vol. 12, pp. 329-336, 1973.

[5] G.H. Golub and C.F van Loan. *Matrix Computations.* John Hopkins Press, 1989.

[6] H.H. Guild. *Cellular logical array for nonrestoring square root extraction.* Electronics Letters, Vol. 6 No. 3, pp. 66-67, 1970.

[7] J.M. Jover and T. Kailath. *A parallel architecture for Kalman filter measurement update and parameter estimation.* Automatica, Vol. 22, No. 1, pp. 43-57, 1986.

[8] J.G. McWhirter. *Private Communication.* March 1990.

[9] J.C. Majithia and R. Kitai. *A cellular array for the non-restoring extraction of square roots.* IEEE Trans. on Computers, pp. 1617-1618, December 1971.

[10] S.Y. Kung. *VLSI Array Processors.* Prentice-Hall 1987

[11] R.C. Devries and M.H. Chao. *Fully iterative array for extracting square roots.* Electronics Letters, Vol. 6 No. 8, pp. 255-256, 1970.

[12] C.C. Paige and M.A. Saunders. *Least Squares Estimation of Discrete Linear Systems using Orthogonal Transforms.* SIAM Journal of Numerical Algebra, Vol. 14, No. 2 pp. 180-193.

[13] R.W. Stewart, R. Chapman. *Fast stable Kalman filter algorithms utilising the square root.* Proceedings of ICASSP 90, Albuquerque, U.S.A., April 1990.

[14] R.W. Stewart, R. Chapman, T.S. Durrani. *The Square Root in Signal Processing.* SPIE Real Time Signal Processing XII, San Diego, U.S.A., August 1989.

[15] R.W. Stewart. *On Parallel and Orthogonal Linear Algebraic Signal Processing.* Ph.D. Thesis, University of Strathclyde, Scotland, March 1990.

[16] K. Hwang. *Computer Arithmetic, principles, architecture, and design.* John Wiley and Sons, Inc., 1979.

[17] W.T. Adams and J. Brady *Magnitude approximations for microprocessor implementation.* IEEE Micro, pp. 27-31, October 1983.

[18] Special issue on DSP Processors. IEEE Micro, Vol. 8, No. 6, December 1988.

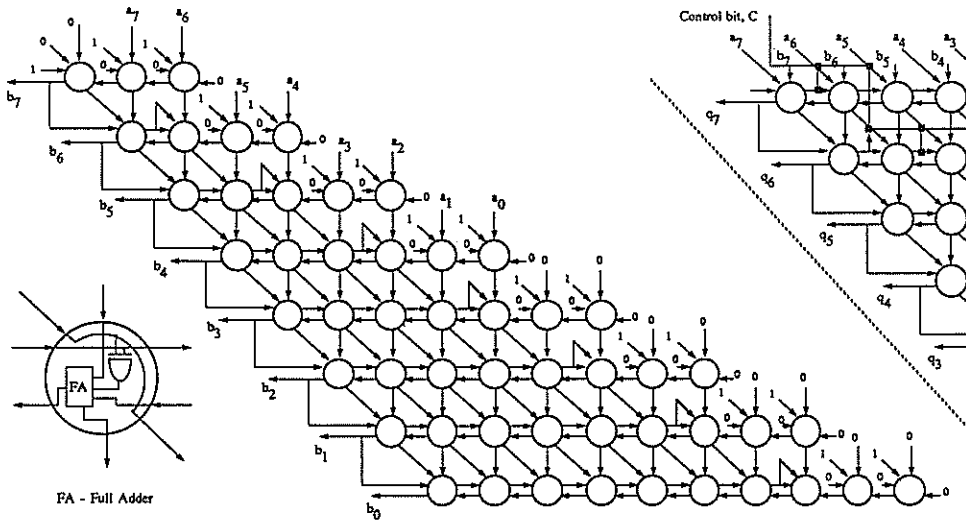


Figure 1: *Square root array for 8 bit mantissa.*

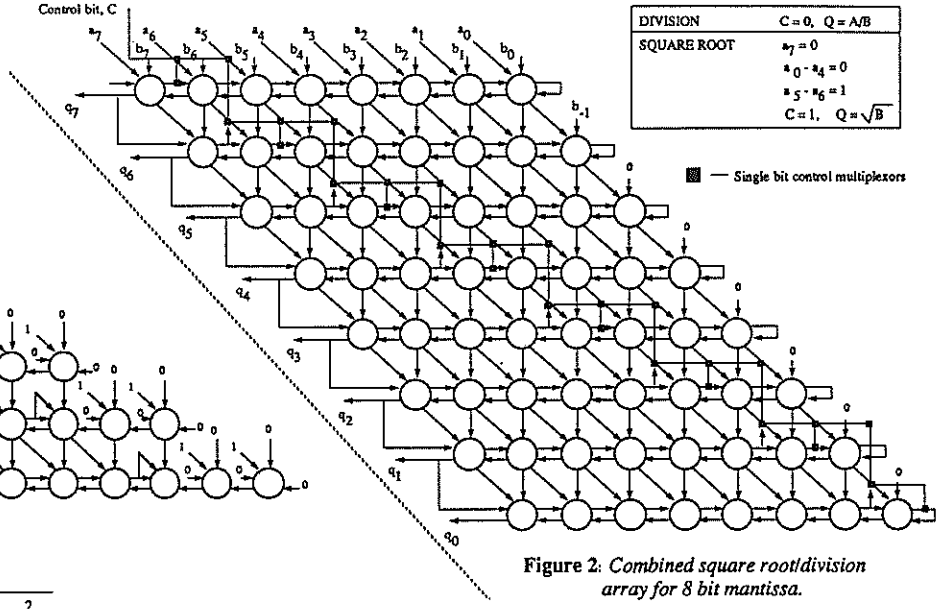


Figure 2: *Combined square root/division array for 8 bit mantissa.*

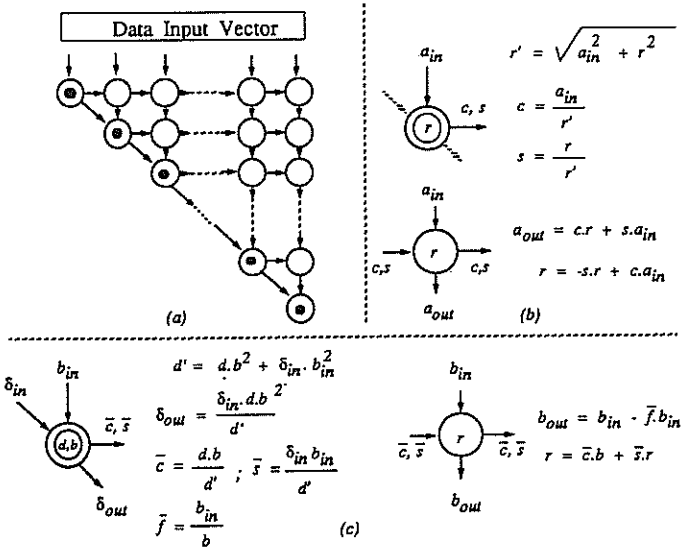


Figure 3: *(a) Givens QR Triarray; (b) Standard Givens Processors; (c) Square Root Free Givens Processors*

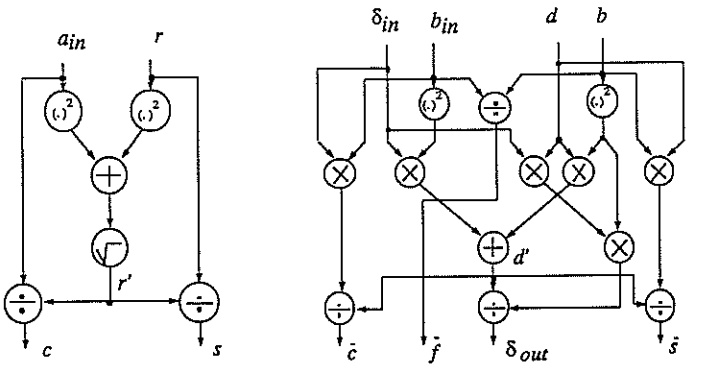


Figure 4: *Data flow for (a) standard Givens generation, and (b) Square root free Givens generation processors.*

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A DYNAMIC RANGE COMPRESSOR ARCHITECTURE FOR AUDIO, USED AS A TEST-VEHICLE FOR TYPE-HANDLING IN THE CATHEDRAL-2ND SYNTHESIS ENVIRONMENT †.

M. Pauwels, F. Catthoor, K. Schoofs, M. Masschelein, H.De Man *.

IMEC, Kapeldreef 75, B-3030 Leuven, Belgium.

In this paper an efficient chip architecture is presented for the real-time implementation of a digital Dynamic Range Compressor algorithm for audio. The micro-code controlled architecture is made in an application specific way. The efficiency is due to the adopted CATHEDRAL-2nd architectural style and especially the use of multiple precision arithmetic. Its use has allowed to minimise the data path dimensions and to get a maximal utilisation of the data path operators at the expense of larger micro-code programs. It is also shown in this paper that numerical types of the signals should be handled carefully in order to guarantee a 'bit-true' mapping of an algorithm onto an architecture, especially when signal word lengths and operator dimensions don't match, as it is the case in this application. Solutions for the alignment problem, which is strongly related with type-handling, are also presented.

1 Introduction.

The dynamic range of digital audio sources is typically very large and -in certain environments- even too large to be used as such. The purpose of a compressor is to adapt the dynamic range of the recording to the listening situation (e.g. car radio, apartment living room), where the acceptable dynamic range is determined by the background noise (for the low-level passages) and the maximal acceptable music level (for the high-level passages).

The algorithm which is used in the design discussed here, was published in [Sti86]. It allows both expansion and compression of the dynamic range. The algorithm was evaluated by means of subjective auditive tests on real music samples [Wag86]. It turned out that it was of very good quality and of practical interest: the compression effect is clearly audible while the artificial sounds and distortion are minimal.

In the second section of this paper, the main characteristics of the algorithm are enumerated, in order to motivate the choice of a micro-coded, application-specific (ASIC) processor architecture for this application. Also examples are given of the different numerical types that occur in this design. This type information is an essential part of the specification, and it must be handled carefully to map the algorithm in a 'bit-true' and efficient way on an architecture [Pau89].

In the third section, three candidate architectures are introduced from which the one with a reduced multiplier and ALUs is the most efficient.

In order to map the algorithm with large data word lengths efficiently on the data path, multi-precision arithmetic is used; the detailed implementation of some rules which are required in this application are given as an example in the fourth section.

Finally, the last section presents the use of the prototype CATHEDRAL- 2nd synthesis environment [Lan90], which partly supports the mapping to the architectural style used here. Also its further extensions with new tools is discussed. This extended synthesis environment will allow to evaluate architectures of designs such as the one presented in this paper, more easily and partially automatically.

2 The characteristics of the algorithm and its 'bit-true' mapping.

The Dynamic Range Compressor (DRC) fits into high volume applications and as such it is justified to go to the most efficient architecture and overall design. Therefore we investigated an ASIC realisation of the DRC.

The main signal flow of the algorithm is shown in Fig.1: the stereo signals are sent through an offset filter and a delay line, while the gain factor by which they must be multiplied, is calculated in parallel in a feed-forward control path.

A micro-coded processor architecture is very well suited for this application, mainly because of the relatively large ratio between sampling rate (audio standards: 32, 44.1 or 48kHz) and the achievable clock rate (10MHz up to 20MHz in $3\mu m$ CMOS). Moreover, a large variety of operations have to be performed, and the required degree of programmability is rather high for an ASIC. Indeed, in the current design the user can choose from 7 predefined expansion and

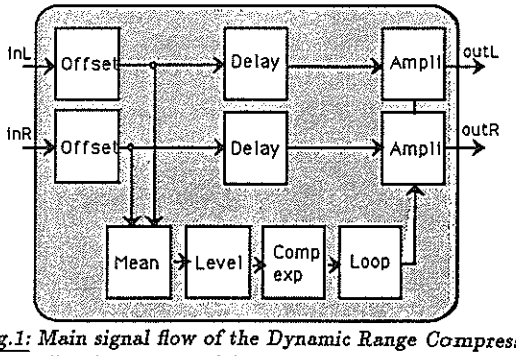


Fig.1: Main signal flow of the Dynamic Range Compressor system. Level, compexp, delay and ampli are programmable.

† This research has been partially sponsored by the SPRITE project of the EC.
* Professor at the Katholieke Universiteit Leuven, Belgium.

compression curves: therefore 'compexp' (Fig.1) contains parametrizable power functions (x^m with $m = 1, 2$ or 3) and 'ampli' a parametrizable factor. Also the release time of the peak-hold function implemented in 'level', must be set according to the kind of music which is processed (classic,pop); one out of 4 predefined factors can be chosen.

The length of the delay line depends on whether the expansion (25ms.) or compression mode (35ms.) is chosen, and on the used sample frequency. It can require up to 1680 samples of 24 bit words (10080 Byte); in the current state of technology, this motivates the use of an off-chip RAM. Fortunately no high data communication rate is needed (4 read-write operations per frame).

Not only the operations characterize the algorithm, but also the numerical types of the signals. They determine the accuracy and bit-level (quantisation and overflow) behaviour of an algorithm. Correct type-handling is very important in order to obtain a 'bit-true' compilation and architectural synthesis [Pau89]. An example is found in section 4. Bit-true compilation means that at least the I/O bit-level behaviour of the synthesized IC must be identical to the one that is specified, independently from the chosen architecture.

Moreover, we want to exploit this type information as much as possible in order to come up with an efficient ASIC architecture (e.g. by using multi-precision arithmetic and variables, the data path dimensions can be minimised). For a realisation on an off-the-shelf signal processor, typically all the signal widths are made compatible with the processor width. However, for an efficient ASIC realisation types are best first optimised in order to exploit this information during synthesis. Indeed, types are related with the required dimensions of the data paths, which are parametrizable in the case of an ASIC. Therefore, the behaviour of the DRC was described in the applicative SILAGE language [Hil90] and simulated both without and with finite word length effects [Sch88]. First the signal types, as well as the type and values of coefficients were fully optimised by using simulation, analysis [Cla86], and optimisation programs [Cat88]. In a second step, similar types were made identical to reduce the alignment hardware (section 4).

Because of the hard time-domain specifications of the 'loop' filter, signals need to be locally 48 bits long in the multiply-accumulate instructions, while all other functions require maximally 24 bits. Fortunately, these large signals do not occur many times and so it is adequate to use multi-precision arithmetic.

3 Dynamic Range Compressor architecture.

The novel CATHEDRAL-2nd synthesis system described in section, addresses highly complex DSP applications with sample rates from a few KHz up to 1 MHz and scalar, vector, matrix and decision making operations. Therefore, highly multiplexed micro-coded multi-processor configurations are very well suited [Cat90].

For the DRC application, a single processor (Fig.2) is sufficient. According to the adopted architectural style, it consists of a customised data path, a micro-code controller and I/O communication circuits. The data path is composed of parametrizable execution units (EXU: e.g. customised ALU), which are constructed from functional building blocks (FBB: e.g. adder, shifter). Various memory structures allow to store variables efficiently on-chip: register files for temporary storage in foreground, pointer addressed memories in EXU feedback paths, and RAMs for a centralized background storage. The EXUs and memories are connected with each other via a dedicated bus network. All programmable data paths are controlled by a micro-code controller that allows for fast decision making and multiple branching.

In an audio system, typically a micro-processor is used as supervising controller of all devices. This micro-processor will also control the DRC processor (start, reset, operation mode control,..). Communication goes via a micro-processor interface. A standard I^2S audio interface [I2S] allows a bit-serial communication of the audio channels between different devices of the audio system. Serial-parallel and parallel-serial conversion is done in the interface. Internally data communicate in a bit-parallel way. In the rest of this section, we will mainly concentrate on different alternatives for the data path of the DRC processor.

First of all, a multiplier-ALU structure is considered. A 24-bit ALU is sufficient to perform all operations on, except the few 24x48-bit multiplications that can be done in double precision on a 24x26 multiplier (MPY). The second input of the MPY must be 26 bits wide because our module library contains only a twos-complement MPY and because only even numbers of bitslices are allowed. Indeed, in this way the 24-bit unsigned, least significant part of a 48-bit double precision signal can be represented as a 25-bit signed signal at the MPY input. The area of a 24x26 MPY is however large (Table 1) and the utilisation very low (Table 2).

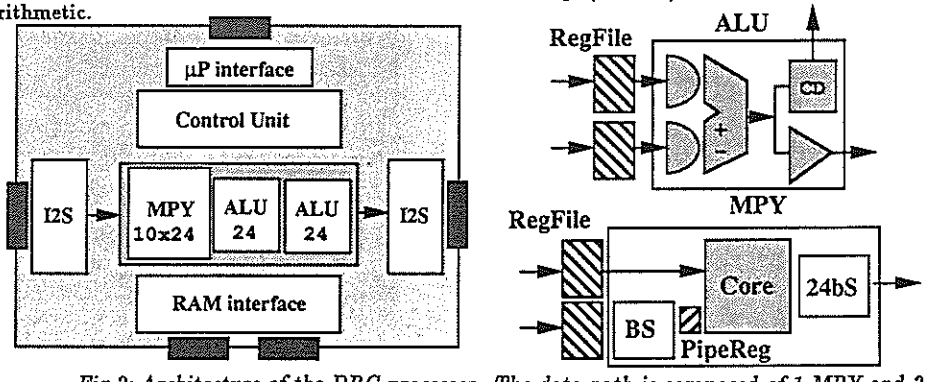


Fig.2: Architecture of the DRC processor. The data path is composed of 1 MPY and 2 dedicated ALUs (BS=Byte-Select, 24bS=24bit-Select).

MPY (24x26)	23.4
ALU (24)	10.2
1 MPY, 1 ALU	33.6
General ALU (26)	11.1
3 ALUs	33.3
MPY (10x24)	12.3
Dedicated ALU (24)	7.5
1 MPY, 2 ALUs	27.3

Table 1: Area estimates in mm^2 of the data paths of the 3 architectures in a 3μm CMOS process.

Function	Multiplications kind	♮	MPY ♮ cycl.	ALU ♮cycl.
offset2	$p24 = c24 \times v24$	2	2	6
level	$p24 = c24 \times v24$	3	3	18
compexp	$p24 = v24 \times v24$	3	3	75
loop	$p72 = c24 \times v48$	6	24	160
ampli2	$p24 = v24 \times v24$	2	2	50
	$p48 = c24 \times v24$	2	2	36
Total utilisation			36	345

Table 2: Utilisation of a 24x26 MPY vs. 26-bit ALU for all kind of mult. occuring in the algorithm (ci, vi, pi: constant, variable and product of i bits). On the ALU a Booth mult. is used for var × var and shift-add mult. for cte × var.

function	Progr.Length ♮cycles	Utilisation MPY	ALU1	ALU2
offset2	21	12	7	7
mean	8	-	-	6
level	37	18	22	-
compexp	30	18	16	-
loop	120	69	55	61
ampli2	41	24	18	18
delay	14	-	-	14
addr. calc.	33	-	-	33
DRC	225	141	118	139

Table 3: Program lengths of manually scheduled functions on the (10x24)-MPY and two (24bit)-ALU architecture. The utilisation (in ♮cycles) on MPY and ALUs is indicated. Note that due to overlapping schedules, the length of DRC is smaller than the sum of individual functions.

As a second solution all multiplications could be mapped on a multi-ALU data-path. A Booth algorithm would be used to multiply 2 variables, and multiplications with a constant expanded in add-shift sequences. Both methods are very time-consuming (Table 2); estimates of the required number of cycles show that at least 3 ALUs must be allocated. They must be 26 bits wide to perform all multi-precision operations on. If they need to be very programmable, what still must be investigated, this solution would be area inefficient (Table 1). Also the micro-code controller will become larger because of the larger programs.

On the other hand, using a 24x10 MPY and two 24-bit ALUs (Fig.2) is more efficient in terms of area (Table 1) and utilisation of the data path operators (Table 3), but requires multi-precision arithmetic. A byte-wise multiplication of 24x48 bits signals is explained as an example in the next section. A first consequence of multi-precision multiplication is that the partial products must be post-processed on an ALU. Therefore 2 ALU instances must be allocated. Fortunately these can be made very dedicated and area efficient. Secondly, efficient placement of registers and dedicated data selection blocks are crucial in order to guarantee a parallel processing on the MPY and ALUs (e.g. Byte-Select and 24Bit-Select, Fig.2).

The large delay lines motivate the use of an external standard RAM. Because of the low I/O rate, and in order to save on the pin-count, the communication can be organised byte-wise, even with relatively slow RAMs (e.g. 32KByte RAM with an access time of 150ns.). Interface circuit like this are the target application for our (asynchronous) interface synthesis environment [VBe89]. In the case of a 32KB RAM, 15-bit unsigned integer addresses must be generated. So, yet another type occurs in the description of the algorithm. The address calculation can easily be done on one of the 24-bit ALUs, because enough free cycles are available. In order to easily increment addresses, the 15-bit variables should be aligned at the least significant side of the ALU.

4 Multiple Precision arithmetic rules.

A major challenge in this design is the use of multi-precision arithmetic with its specific alignment problems. It allows to gain on the area of the operators and get a better utilisation of the operators in time. The penalty is that the programs become larger. So, determining the degree of multi-precision is an optimisation problem with time-area trade-offs.

As an example we describe the byte-wise multiplication of a 24 and 48-bit variable. The scheme in Fig.3, shows the six 32-bit partial products and how each of them must be split in two 24-bit words to be post-processed on a 24-bit ALU, in order to obtain the 72-bit product as three 24-bit words. To calculate the full product takes 15 cycles with a utilisation of 12/11 cycles on the MPY/ALU. In section 3, the need for a 10x24 bit MPY was explained. It has a 34-bit output. Because the I/O dimensions of the MPY are slightly larger than the signal word lengths, the signals must be well aligned on the MPY ports, with the necessary sign extensions and/or zero-fillings to obtain correct 32-bit partial products. Dedicated routing blocks (Fig.2, Byte-Select and 24Bit-Select) are used for this alignment.

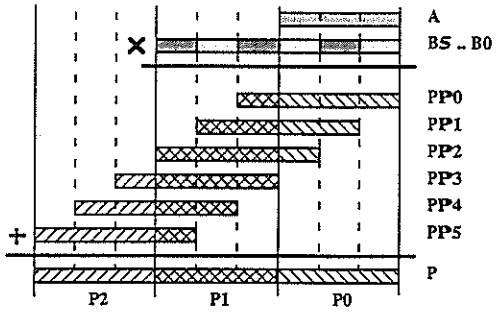


Fig.3: Byte-wise multiplication scheme of A (24bits) with B (6x8bits) = P (72bits). PPi = 32-bit partial products.

Nevertheless the full numerical types must be considered carefully to guarantee a bit-true mapping. E.g. the multiplication of 2 signals of types $num < 24, 22 >$ and $num < 48, 45 >$; this would give a $num < 72, 67 >$ product. A product of type $num < 48, 43 >$ corresponds with the 2 most significant 24-bit words obtained on the ALU. However, for a product of type $num < 48, 45 >$, the three 24-bit words should be upshifted first over 2 bits.

5 The CATHEDRAL-2nd architectural synthesis environment.

The novel CATHEDRAL-2nd synthesis environment under development at IMEC, has the same target domain of applications but a more flexible architecture as the one supported in CATHEDRAL-2 [Rab88]. Mainly because of the increased architectural flexibility, but also in order to enhance the level of user interaction, it uses a new synthesis approach [Lan90]. It consists of a set of tools that are called from a script and gradually transform the signal flow specification of the algorithm into a control and signal flow description, mapped on an architecture.

In the current prototype, the data path must be allocated manually and given as an input to the compiler. An expansion, chaining and foreground memory management tool is available and has been successfully used for this application. First, the high level operations (e.g. a*b) are expanded into FBB operations (e.g. byte-wise multiplication and additions) The next tool chains FBB operations together in 'super- operations' (e.g. shift-add-compare operation if a shifter is followed by an adder and a comparator in the data path). The foreground memory tool performs the hardware binding of the super-operations on the FBBs in the data path and determines in which register to store the variables. From this, a register-transfer description can be deduced and scheduled. Unfortunately, at the moment of this writing, no schedule results are available to compare with the manually obtained results of Table 3.

In future versions of the compiler, new tools will be added to the script. In particular, the automated synthesis of the routing network in order to solve the alignment problem will be addressed. Also tools for the automated selection of the level of multiple precision and the definition of the dedicated execution unit, are needed.

6 Conclusions.

An efficient micro-coded architecture was proposed for a Dynamic Range Compressor application. The efficiency is due to the CATHEDRAL-2nd architectural style and the use of multi-precision arithmetic. It was shown that type-handling and alignment are crucial when an exact and optimised architecture is purchased, certainly for the development of ASICs for real-time signal processing applications. This design was successfully evaluated with the current CATHEDRAL-2nd prototype. Because of its interesting properties, this application will be used as a test-vehicle for further improved and extended versions of CATHEDRAL-2nd.

Acknowledgement.

The authors wish to express their gratitude to the colleagues in the project, and especially to J.Van Ginderdeuren (PHILIPS), and Dirk Lanneer and Peter Van Bekbergen (IMEC) for the stimulating discussions.

References

[Sti86] E.F.Stikvoort, "Digital Dynamic Range Compressor for Audio", J.Audio Eng.Soc., Vol 34, No. 1/2, 1986 Jan/Feb.

[Wag86] W.M.Wagenaars et al., "Subjective Evaluation of Dynamic Compression in Music", J.Audio Eng.Soc., Vol 34, No. 1/2, 1986 Jan/Feb.

[Pau89] M.Pauwels, F.Catthoor, D.Lanneer, H.De Man, "Type-handling in Bit-true Silicon Compilation for DSP", Proc. Eur. Conf. on Circuit Theory and Design, ECCTD, Brighton, U.K., pp.166-170, Sep. 1989.

[Lan90] D.Lanneer, F.Catthoor, G.Goossens, M.Pauwels, J.Van Meerbergen, H.De Man, "Open-ended System for High-Level Synthesis of Flexible Signal Processors", Proceedings EDAC90, pp. 272-276, Glasgow, Scotland, March 1990,

[Hil90] P.N.Hilfinger, J.Rabaey, D.Genin, C.Scheers, H.De Man, "DSP specification using the Silage language", Proc. Int. Conf.on Acoustics, Speech and Signal Processing, Albuquerque, NM, April 1990.

[Sch88] C.Scheers, "User manual for the S2C SILAGE to C compiler", IMEC report, IMEC, Heverlee, Sep.1988 (available upon request).

[Cla86] L.Claesen, F.Catthoor, H.De Man, J.Vandewalle, S.Note, K.Mertens, "A CAD Environment for the thorough Analysis, Simulation and Characterisation of VLSI implementable DSP Systems", Proc.IEEE Int.Conf. on Computer Design, Port Chester NY, pp.72-75, Oct.1986.

[Cat88] F.Catthoor, J.Vandewalle, H.De Man, "Simulated-annealing based Optimisation of Coefficient and Data Word-lengths in Dig. Filters", Int.Journ. on Circ. Theory and Appl., Vol.16, pp.371-390, Sep.1988.

[Rab88] J.Rabaey, H.De Man, J.Vanhoof, G.Goossens, F.Catthoor, 1988, "CATHEDRAL II: A Synthesis System for Multi-processor DSP Systems", in "Silicon Compilation", D.Gajski (ed.), pp.311-360.

[Cat90] F.Catthoor, H.De Man, "Application-specific architectural methodologies for high-throughput digital signal and image processing", IEEE Trans. on Acoustics, Speech and Signal Processing, Vol.37, No.2, pp.176-192, Feb.1990.

[I2S] "The I2S bus specification", PHILIPS report, PHILIPS Electronic Components and Materials.

[VBe89] P. Vanbekbergen, "Race-free Time-optimised synthesis of asynchronous interface circuits." Int. Workshop on Logic Synthesis at MCNC. May 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Novel Architecture for Fast, Numerically Stable DCT on Single-Chip DSP

Christoph D. Cavigioli

Analog Devices, Inc. -- DSP Division
P. O. Box 9106, Norwood, MA 02062 USA

When the DCT function is required in a system, three VLSI considerations must be weighed. First, which of several computational approaches is employed dictates the numerical stability and accuracy of the transform. Next, pixel throughput rate is especially important for real-time applications. Finally, total system cost is gauged not only by the price of the VLSI chip, but also in development effort required and reliability of the system. These tradeoffs are addressed by showing the novel architecture single-chip DSP.

1. Introduction

The discrete cosine transform (DCT) is used to compress image data. The transform consolidates the picture's most important information in a smaller area than the original pixels occupy. The less important information can then be disregarded by one of several different ways. This is how data reduction occurs. Applications of DCT image compression include:

- CCITT H.261 video telephony
- ISDN applications
- digital FAX
- scene recognition
- photo / film archival

Just as the discrete Fourier transform (DFT) can be computed significantly faster using the fast Fourier transform (FFT) method, similar algorithms exist for the DCT called fast cosine transform (FCT or FDCT) algorithms. A two dimensional transform of an NxN block is calculated by doing N one dimensional N-pt transforms on the rows, followed by N transforms on the columns. Benchmark execution times and memory storage requirements are shown in figure 1.

2. Algorithmic Trade-Offs

Many fast algorithms have been devised for accelerating the computation of the DCT. Most algorithms can be classified into one of three categories:

- 2N-pt FFT with phase shift
- matrix factorization
- recursive computation

Indirect computation involves doubling the length of an N-pt sequence to a 2N-pt sequence with its mirror image, then performing a 2N-pt FFT on that sequence and multiplying the result with a complex exponential phase shift vector [1]. Considerable data moves must be done as well as FFT calculations involving complex numbers and storage. Some FFT ASIC chips have been used to implement this method using transposition memory although the resulting board-level complexity can be prohibitive for practical use.

	N - pt	N x N block
N = 8	153 cycles **12.2 µs** 138 PM code 7 PM data 8 DM data	2428 cycles **0.19 ms** 183 PM code 7 PM data 8 DM data
N = 16	309 cycles **24.7 µs** 278 PM code 15 PM data 16 DM data	9789 cycles **0.78 ms** 325 PM code 15 PM data 16 DM data

Figure 1 - DCT Benchmarks - ADSP-2105 (12.5 MHz)

The formal equation for the one-dimensional DCT [2] can be rewritten as a matrix multiply if the DC scale coefficient is ignored and both the input and output

sequences are reordered in certain ways. Using this as a starting point, matrix factorization techniques have yielded several fast algorithms which require significantly less arithmetic operations than the full matrix multiply. Most important is that there are no complex operations and relatively few data moves. One drawback is that different values of N require factoring the matrix differently. Flexibility is thereby compromised. Examples of this method would be: Chen, Wang, and Lee [3], [4], [5]. Wang requires the use of two different types of DCTs, and Lee requires inversion or division of the cosine coefficients. A commercial ASIC has been announced which uses the Lee algorithm. Unfortunately, that algorithm inherently causes numerical instabilities due to roundoff errors in finite length registers.

Probably the best (numerically and arithmetic efficiency) and most elegant is a recursive method proposed by H. S. Hou [6]. Hou's FDCT algorithm is numerically stable, fast and recursive. Similar to the Cooley-Tukey FFT algorithm, this algorithm generates the next larger DCT matrix from two identical smaller DCT matricies. This deviates from direct factorization algorithms, which change as N (points) changes.

As shown in figure 2, a DCT of size N=16 is recursively separated into smaller DCTs of size N={8,4,2}. Notice the two 8-pt DCTs embedded in

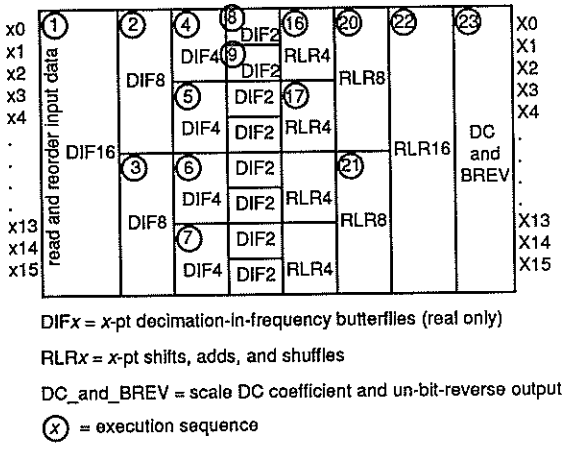


Figure 2 - Recursive Modularity of 16-pt FDCT using Hou's Method

the 16-pt DCT signal flow. The code presented here performs the cosine transform in-place. That is, the input data values are read from the image buffer (arranged in normal, sequential order) and the results are written back to the same buffer in the same sequential order. Hou's method was chosen because it makes two dimensional transforms simple, and it uses very little on-chip memory space. If a not-in-place DCT is desired, simply change the pointer to point to a different output buffer before the final bit-reversing routine is called.

3. Chip Architecture

Figure 4 shows the architecture of the ADSP-2105. This architecture is ideally suited for Hou's DCT algorithm. Notice three memory spaces: dedicated data memory (DM = data storage #1) and shared program memory (PM = instruction storage and data storage #2). A fetch from all three memory spaces can occur during every instruction (or clock) cycle. The data address generator #1 can fetch a data value from DM using the DMD bus in parallel with data address generator #2 fetching a coefficient from PM using the PMD bus in parallel with the program sequencer independently fetching the next

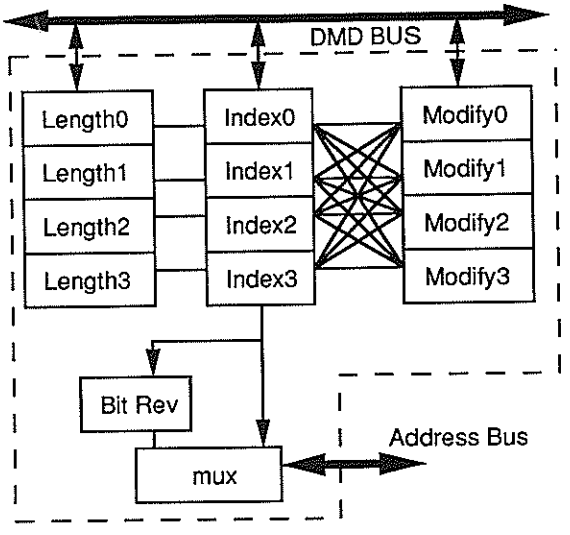


Figure 3 - Looking Inside One of the Two On-Chip Data Address Generators Reveals Flexibility

instruction on a dedicated instruction bus also connected to PM. This can all be happening in parallel with the data computation units being used for data calculations as well as all the necessary address computations taking place inside the dedicated data address generators. No speed penalty is incurred when accessing external image memory.

Three computation units, the ALU, a dedicated MAC, and a powerful barrel shifter, are organized in parallel. Also visible is a fourth data bus called the result bus or R bus. The result bus allows data computation results to be used as input values to any

of the three computation units. This eliminates output register to input register data copies as well as eliminating the corruption of the previous input register contents.

The program sequencer includes hardware loop management circuitry in order to execute recursive, looped code without loop management overhead.

physical method allows bit-reversing of addresses generated in any sequence, or singular addresses, whereas the reverse-carry method requires addresses to be generated sequentially from a base. It is interesting to note that the same subroutines are called whether the DCT is being calculated on row data or on column data even though the individual data elements are spaced differently in memory

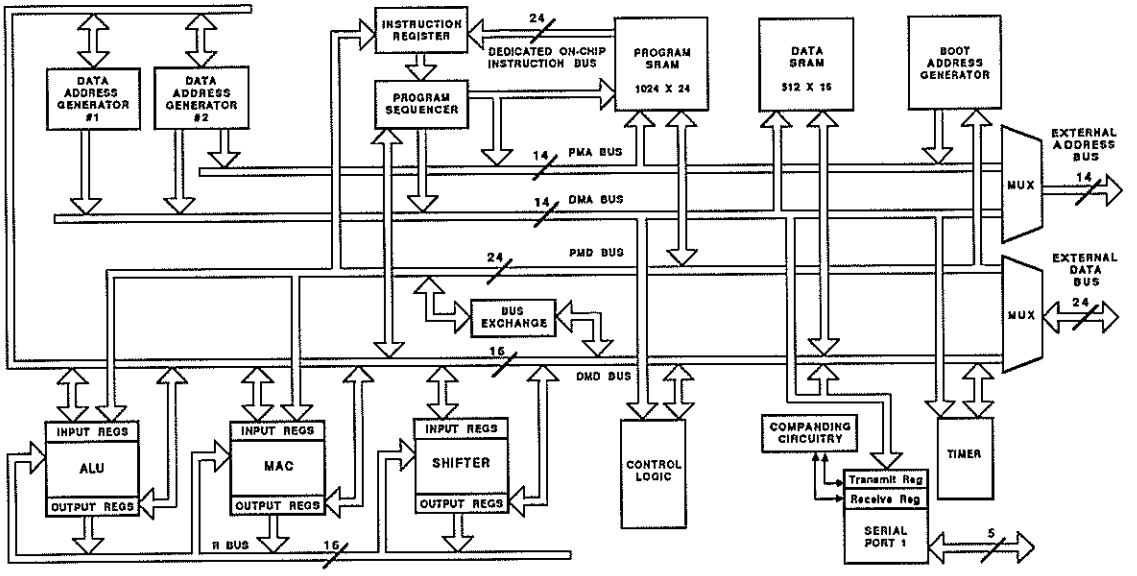


Figure 4 - ADSP-2105 Internal Architecture

This zero-overhead looping in conjuction with the recursive nature of Hou's algorithm gives two dimensional DCTs an extraordinarily fast execution time.

Hou's DCT involves complicated data sequencing. Figure 3 indicates the programmer's ability to choose the association of index and modify registers in both data address generators, eliminating the problem of how to store data and coefficents for efficient execution. Simply pick the correct modify register during a data access such that the index register points to the correct next data value in the following cycle. This feature is also handy when zig-zag scanning the DCT output coefficients.

By setting a bit in a mode register, one address generator will output its addresses in bit-reverse fashion. This greatly simplifies data shuffling involved within sections of Hou's algorithm. An important clarification is that the bit-reversing operation is done by physically bit-reversing the address bus. This differs from the reverse-carry arithmetic used elsewhere in the industry. The

when accessed by rows or by columns.

Even with all this flexibility and parallelism, the resulting algebraic instructions are easy to read, self documenting, and separable into small, manageable modules. See figure 5.

Data memory storage required to compute the DCT is very small. Sixteen data memory locations are used on-chip for a 16-pt DCT (8 locations for an 8-pt DCT). See figure 1. These locations make up the TMP buffer. Two additional data memory locations store pointers to the correct row and column inside the array. See figure 5. Notice that external transposition memory is not required since transpositions are not done. The only external memory used is the picture memory itself. There is no speed penalty for accessing off-chip memory in the DCT algorithm.

Program memory storage is also efficient. Only 15 real, cosine coefficients are stored in program memory for the 16-pt transform (7 for the 8-pt transform). The instructions occupy few locations

because they are stored in short, recursively called subroutines. The code and cosine values can be ROM-coded on-chip or booted from external EPROM.

```
.module             fast_16x16_dct;
.external DIF16,DIF8,DIF4,DIF2,RLR4,RLR8,RLR16,DC_AND_BREV;
.var/dm/circ/abs=0  tmp[16];       { temp scratch memory }
.global             tmp;
.var/dm             xadr, xadr2;
.var/dm             x[256];        { block to transform }
.var/pm/ram         cosvals[15];   { cosine coefficients }
.init  cosvals[00]: h#7F6200, h#70E200, h#513300, h#252800,
                    h#F37500, h#C3AA00, h#9D0E00, h#858300;
.init  cosvals[08]: h#7D8A00, h#471C00, h#E70800, h#959300;
.init  cosvals[12]: h#764100, h#CF0500;
.init  cosvals[14]: h#5A8200;

setup:  l0=0; l1=0; l2=0; l3=0; l5=0; l6=0;
        m6=1; m7=-3; se=1;

rows:   si=^x;    dm(xadr)=si;  i2=si;
        si=^x+15; dm(xadr2)=si; i3=si;
        m5=1;
        cntr=16;                { do 16 rowdcts }
        do rowdcts until ce;
          i6=^cosvals;
          m2=2;
          m3=-2;
          call DIF16; call DIF8; call DIF4; call DIF2;
          call RLR4;  call RLR8; call RLR16;
          si=dm(xadr);
          i5=si;
          call DC_AND_BREV;
nextrow:  ay0=16;               { incr row,col ptr by 16 }
          ax0=dm(xadr);  ar=ax0+ay0; dm(xadr)=ar;
          i2=ar;
          ax0=dm(xadr2); ar=ax0+ay0; dm(xadr2)=ar;
rowdcts:  i3=ar;

cols:   si=^x;     dm(xadr)=si;  i2=si;
        si=^x+240; dm(xadr2)=si; i3=si;
        m5=16;
        cntr=16;                { do 16 coldcts }
        do coldcts until ce;
          i6=^cosvals;
          m2=32;
          m3=-32;
          call DIF16; call DIF8; call DIF4; call DIF2;
          call RLR4;  call RLR8; call RLR16;
          si=dm(xadr);
          i5=si;
          call DC_AND_BREV;
nextcol:  ay0=1;                { incr row,col ptr by 1 }
          ax0=dm(xadr);  ar=ax0+ay0; dm(xadr)=ar;
          i2=ar;
          ax0=dm(xadr2); ar=ax0+ay0; dm(xadr2)=ar;
coldcts:  i3=ar;
        rts;
.endmod;
```

Figure 5 - DSP Source Code for Hou's 16x16 FDCT

4. Processor or ASIC

Total system cost in terms of board complexity and reliability is reduced when using general purpose DSP processors because each DSP has on-chip memory and on-chip address generation. Cost in terms of development time is reduced because the design is easier and can be debugged with software simulation tools and in-circuit emulators. Moreover, the chips are reusable for other functions or existing functions can be upgraded with a software change. Even though ASICs usually have faster throughput rates, DCT compression lends itself to organizing multiple DSPs in parallel since there is no shared information between image blocks, thereby achieving similar throughput rates for competitive prices. For example, the ADSP-2105 has been announced in 1990 for less than $10.

Additional information about the DSP chip family, development tools, and image compression code [7] can be obtained from the author.

5. References

[1] Makhoul, J., A Fast Cosine Transform in One or Two Dimensions, IEEE Trans, vol. ASSP-28, no. 1, Feb. 1980.

[2] Rosenfeld, A., and Kak, A., Digital Picture Processing, Second Edition, (Academic Press, New York, 1982).

[3] Chen, W.H., Smith, C.H., and Fralick, S.C., A Fast Computational Algorithm for the Discrete Cosine Transform, IEEE Trans, vol. COM-25, Sept. 1977.

[4] Wang, Z., Fast Algorithms for the Discrete W Transform and for the Discrete Fourier Transform, IEEE Trans, vol. ASSP-32, Aug. 1984.

[5] Lee, B.G., A New Algorithm to Compute the DCT, IEEE Trans, vol. ASSP-32, no. 6, Dec. 1984.

[6] Hou, H.S., A Fast Recursive Algorithm for Computing the Discrete Cosine Transform, IEEE Trans, vol. ASSP-35, no. 10, Oct. 1987.

[7] Cavigioli, C.D., The Discrete Cosine Transform, Analog Devices, DSP Division, To appear in: ADSP-2100 Family Applications Handbook Vol. 4.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

TRANSPUTER BASED QUADTREE DATA STRUCTURE FOR ADAPTIVE TRANSFORM CODING

M.N. Chong, J.J. Soraghan

Signal Processing Division, Department of Electronic and Electrical Engineering,
University of Strathclyde, Glasgow G1, SCOTLAND.

Abstract- The concurrent implementation of adaptive transform coding algorithm using quadtree data structure, variable blocksize Discrete Cosine Transform (DCT) is presented. The computational intensity and highly parallel nature of such algorithms motivate the use of muitiple processor network to execute the algorithms in close to real-time. A speed up of 6.5 is achieved using a system network of eight T800 transputers. This paper presents the method used in segmenting and distributing the images for parallel processing. The system configuration and flow-diagram of these parallel algorithms are presented.

1 INTRODUCTION

In conventional transform coding algorithms, the image is divided into image sub-blocks of *equal size.* Transformation, thresholding, and quantization are subsequently performed on these image sub-blocks. In order to enhance the efficiency of transform coding, adaptive blocksize coding is applied. A block diagram of the variable blocksize transform coding system is illustrated in figure 1. The overhead information in representing the variable blocksizes *should not* degrade the overall bit rate of the coder. The quadtree structure is used to segment and represent the variable blocksize of sub-images efficiently. The resulting overhead in coding the quadtree is insignificant as compared with the overall bit rate. Both improvement in mean square error and subjective image quality were reported in [1].

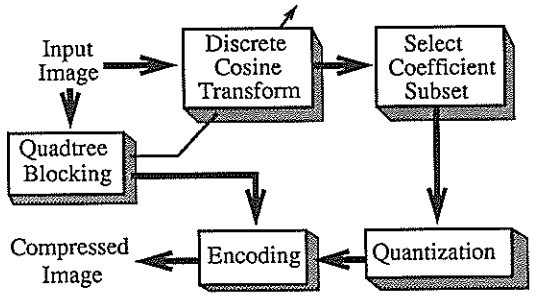


Figure 1 Adaptive Blocksize Transform Coding Layout

Adaptive transform coding algorithms require enormous amount of computing resource. In conventional sequential computing machines, the image sub-blocks are processed in a sequential order. Sequential program execution is inherently slow and real-time processing seems inapproachable. To exploit the parallelism in the program, the data flow-diagrams of these algorithms are examined. Parallel implementation of these algorithms is achieved such that no data communication is required between each image sub-block by setting a limit to the size of quadtree. This results in great speed improvement. The Inmos T800 transputer is chosen as the target computing device. Various topologies were investigated and a *tree structure* was chosen for the final implementation, due to the data flow structure of these algorithms. The advantages in using T800 transputer are :

(i) a 2-D block of data can be moved in one execution cycle.

(ii) the floating point arithmetic unit in T800 further enhance the speed performance in these algorithms as floating point arithmetic is essential for the accurate reconstruction of the compressed image.

The possibilities of implementing the adaptive transform coding algorithms in real-time motivates the work in this paper.

In section 2, the parallel flow-diagram of the adaptive blocksize transform coding is presented. The method of segmenting the original image into sub-images for parallel processing is discussed. The quadtree building algorithm [2] and the fast 2-dimensional Discrete Cosine Transform (DCT) algorithm based on the work of Haque [3] are outlined. The in-depth discussions of using the above mentioned algorithms in transform coding can be found in [4]. The system configuration is discussed in section 3. Section 4 contains the timing results of running the algorithms on a network of processors. The final section summarises the work to date, and focuses on the remaining algorithms associated with the transputer based parallel transform coder, namely coefficients selection and block quantization.

2 FLOW-DIAGRAM

The original image is first segmented into sixteen image sub-blocks of size 256 by 16 which is to be mapped onto a total of sixteen *root* and *leaf* processors. There is no overlapping in this segmentation process and therefore each image sub-block can be processed independently without the need of communicating with each other. Algorithm implementation is co-ordinated by the *host* and *sub-host* processors. These two processors co-ordinate the jobs of initial segmentation, distribution, collection and restoration of the sub-images. The flow-diagram of the main program is illustrated in figure 2a. The sub-images are distributed to the *root/leaf* processors.

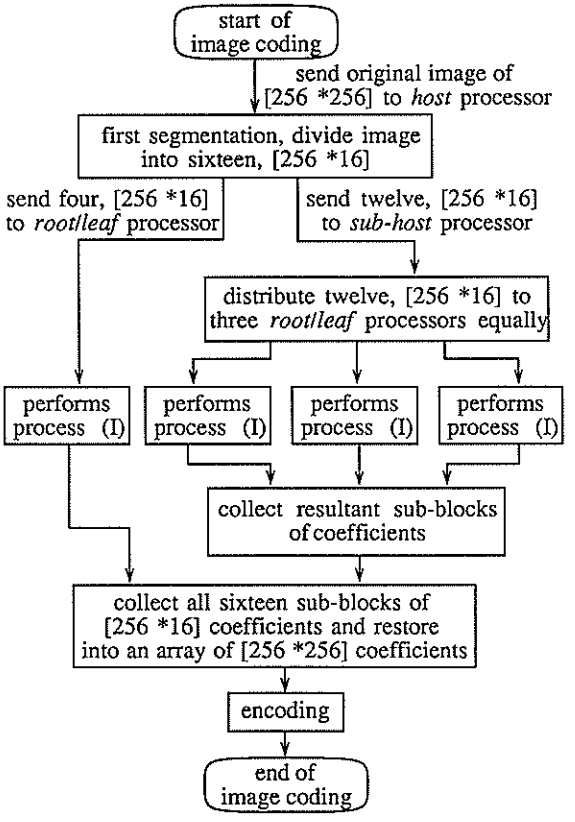


Figure 2a Main program flow-diagram

At the *root*/leaf processors, the image sub-blocks of [256*16] are further segmented into smaller sub-blocks of [16*16] in order that a three level quadtree data structure may be constructed. The transform coding algorithms are performed concurrently at the *root*/leaf processors. The additional work load on the *root* processor is the further distribution and collection of the sub-blocks of [256*16] to the three leaf processors. The detail flow-diagram at the *root*/leaf processors is shown in figure 2b and 2c. The additional work load on the *root* processor is insignificant compared with the execution of the adaptive transform coding algorithms.

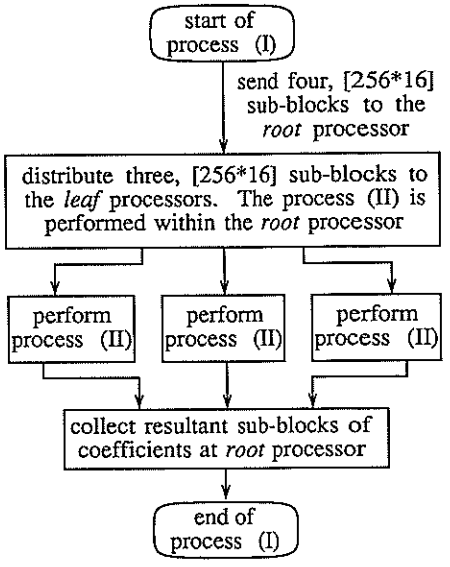


Figure 2b Process (I) flow-diagram

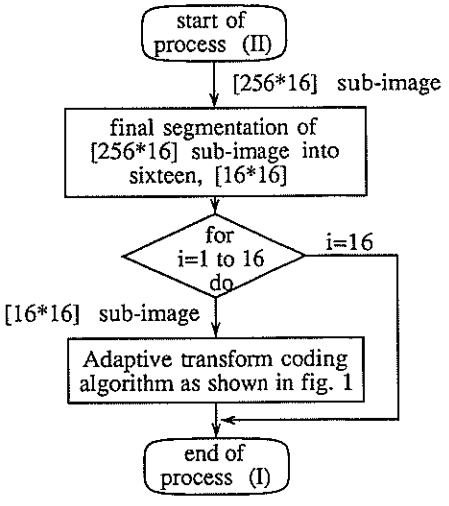


Figure 2c Process (II) flow-diagram

A three level quadtree data structure is built within each 16 by 16 sub-block as shown in figure 3a. Hence, three variable block sizes result, i.e. a whole block of size 16, four blocks of size 8, sixteen blocks of size 4 or a mixture combination of the above mentioned block sizes. Quadtree is a class of hierarchical data structures based on the principle of recursive decomposition of space. The principle guiding the decision (Top-down) or merging (Bottom-up) is that

of a decision rule. A pre-ordered bottom-up quadtree building algorithm is used. The mean of the smallest (4 by 4) blocks are first computed, these means are compared with a threshold value to decide for merge or split. In our algorithm, a '1' represents merge whilst '0' represents split at each node of the quadtree. With the use of the pre-determined quadtree traversal building algorithm (see figure 3b for the order of direction used in this algorithm), these binary representation of merge or split can be used to address the 3-level quadtree nodes, thus minimising the overhead code required to address the variable block sizes. The equivalent quadtree structure can be seen in fig. 3c. The average overhead coding of the quadtree is 0.01 bit/pixel. The detail of the quadtree coding can be found in [4]. The overhead bit/pixel is insignificant as compared to the encoded bits of the compressed image.

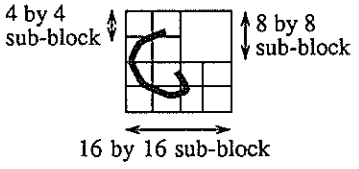


Fig. 3a Quadtree of a 16 by 16 image

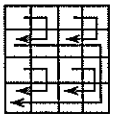


Fig. 3b Direction in constructing the quadtree

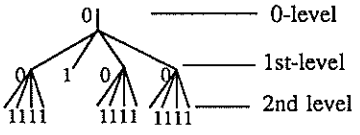


Fig. 3c Labelling of the quadtree

The 2-D Fast Discrete Cosine Transform (DCT) algorithm has a similar structure to the quadtree. The original spatial data matrix of size N is decomposed into four sub-blocks, each of size N/2. These sub-blocks are further decomposed into smaller sub-blocks of N/4. This process of decomposition is continued until the minimum sub-block of size 1 is obtained. DCT is performed on these smallest sub-blocks and subsequently reconstructed back to the original block size of N. The advantage of performing decomposition and reconstruction is that the number of real multiplications involved is $(3/4)\times N^2 \log_2 N$. Further details can be found in [3].

3 SYSTEM CONFIGURATION

The Concurrent algorithms discussed above have been implemented in occam 2 on a network of up to 16 T800 transputers. The tree topology is selected in the final implementation due to the data-flow of these algorithms as discussed in the preceding section. The parallel image coder system architecture is shown in figure 4.

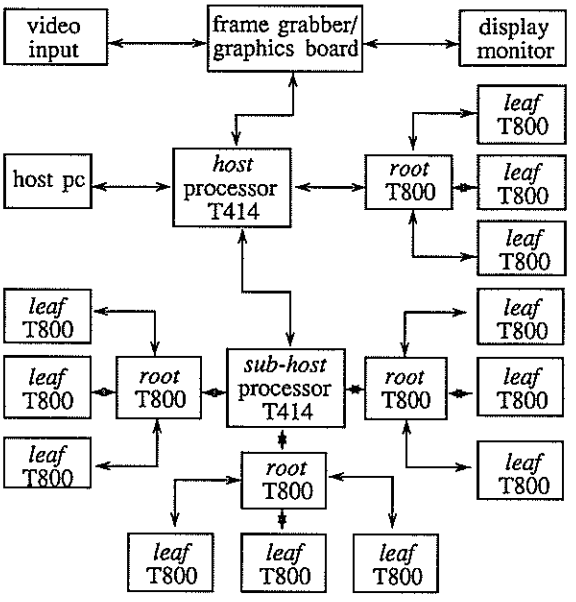


Figure 4 Parallel Image Coder System Architecture

Part of the occam 2 source code [5] is presented in figure 5a, 5b and 5c for better understanding of the system. The main program is running in the *host* processor and three type of procedures, namely *sub-host, root, and leaf* procedures are running concurrently and synchronous with each other in the transputer network.

```
PROC subhost(CHAN OF prot frm.host, CHAN OF prot to.host,
             CHAN OF prot to.root1,CHAN OF prot frm.root1,
             CHAN OF prot to.root2,CHAN OF prot frm.root2,
             CHAN OF prot to.root3,CHAN OF prot frm.root3)
PAR
  SEQ
    SEQ i = 0 FOR 4
      frm.host ? tag ; sub-image
      to.root1   ! tag ; sub-image
    SEQ i = 0 FOR 4
      frm.host ? tag ; sub-image
      to.root2   ! tag ; sub-image
    SEQ i = 0 FOR 4
      frm.host ? tag ; sub-image
      to.root3   ! tag ; sub-image
  SEQ i = 0 FOR 12
    ALT
      frm.root1 ? tag ; sub-coefficients ; quadtree_code
        to.host   ! tag ; sub-coefficients ; quadtree_code
      frm.root2 ? tag ; sub-coefficients ; quadtree_code
        to.host   ! tag ; sub-coefficients ; quadtree_code
      frm.root3 ? tag ; sub-coefficients ; quadtree_code
        to.host   ! tag ; sub-coefficients ; quadtree_code
```

receive sub-images from host and send sub-images to root

receive sub-coefficients from root and send sub-coefficients to host

Figure 5a Procedure on the *sub-host* transputer

```
PROC root(CHAN OF prot  frm.link, CHAN OF prot  to.link,
          CHAN OF prot  to.leaf0, CHAN OF prot  frm.leaf0,
          CHAN OF prot  to.leaf1, CHAN OF prot  frm.leaf1,
          CHAN OF prot  to.leaf2, CHAN OF prot  frm.leaf2)
 PAR
   SEQ
     SEQ i = 0 FOR 4
       frm.link ? tag ; sub-image
       to.leaf0   ! tag ; sub-image
     SEQ i = 0 FOR 4
       frm.host ? tag ; sub-image
       to.leaf1   ! tag ; sub-image
     SEQ i = 0 FOR 4
       frm.host ? tag ; sub-image
       to.leaf2   ! tag ; sub-image
   SEQ i = 0 FOR 12
     ALT
       frm.leaf0 ? tag ; sub-coefficients ; quadtree_code
         to.host   ! tag ; sub-coefficients ; quadtree_code
       frm.leaf1 ? tag ; sub-coefficients ; quadtree_code
         to.host   ! tag ; sub-coefficients ; quadtree_code
       frm.leaf2 ? tag ; sub-coefficients ; quadtree_code
         to.host   ! tag ; sub-coefficients ; quadtree_code
   SEQ
     to.leaf3 ? tag ; sub-image
     ..... sequential process of Image Coding
     frm.leaf3   ! tag ; sub-coefficients ; quadtree_code
```

Figure 5b Procedure on the *root* transputer

```
PROC leaf(CHAN OF prot  frm.root, CHAN OF prot  to.root)
  SEQ
    frm.root ? tag ; sub-image
    ..... sequential process of Image Coding
    to.root   ! tag ; sub-coefficients ; quadtree_code
```

Figure 5c Procedure on the *leaf* transputer

4 RESULTS

The timing results for various sized networks are given in Table 1. The speed improvement by using multiple transputers is plotted and shown in figure 6. The speed-up of the parallel algorithms is compared with the theoretical model. It can be seen that the speed improves significantly from the use of one transputer to eight transputers and modest improvement in speed is obtained using sixteen transputers.

Number of processors	1	4	8	16
Timing in seconds	13.350	3.600	2.060	1.538

Table 1 The table shows the timing (in sec)of running the algorithms in various number of transputers

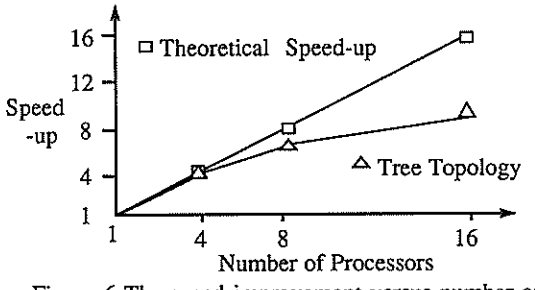


Figure 6 The speed improvement versus number of transputers used

5 CONCLUSIONS

It can be seen from the results that the addition of processors will bring improvements in performance nearly proportional to the increase in processors. However, there is a limitation to the degree of parallelism in our implementation. As more processors are added , the computational load at each node is reduced whilst the communication overhead is increased, leading to a saturated network. Such an effect of parallel implementation saturation is inherent.

The present work focuses on the adaptive transform size and DCT algorithms. Future research will examine the design on parallel adaptive coefficient selection and adaptive quantization algorithms such that a cost efficient, real time image compressor can be achieved using parallel processing technique.

ACKNOWLEDGMENTS

Much of the work was carried out at the Parallel Signal Processing Center, University of Strathclyde. The financial support has come from the Overseas Research Studentship award and Strathclyde University research studentship. I would like to acknowledge Gary Kelieff for his programming assistance.

REFERENCES

[1] C. T. Chen , *Adaptive Transform Coding via Quadtree-Based Variable Blocksize DCT* , Proc. of IEEE Inter. Conference on Acoustics , Speech and Signal Processing, May 1989, pp. 1854-1857.

[2] Y. Cohen, M.S. Landy and M. Pavel, *Hierarchical Coding of Binary Images*, IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. PAMI-7 pp. 284-298, May 1985.

[3] M. A. Haque, *A Two-Dimensional Fast Cosine Transform*, IEEE Trans. on Acoustics, Speech, And Signal Processing , Vol. ASSP-33 , No.6, December 1985, pp. 1532-1538

[4] M.N. Chong, J.J. Soraghan, T.S. Durrani , *Parallel Implementation and Analysis of Adaptive Transform Coding Algorithm* , Proc. of IEEE Inter. Conference on Acoustics, Speech and Signal Processing, April 1990.

[5] INMOS Limited, *Occam 2 Reference Manual* , Prentice Hall International , 1988.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SYSTOLIC VOTES COLLECTION FOR THE GENERALIZED HOUGH TRANSFORM

M. Grazia Albanesi, M. Ferretti and R. Megazzini

University of Pavia, Dipartimento di Informatica e Sistemistica
Via Abbiategrasso 209, 27100 Pavia, ITALY

In this paper a new system for the collection of votes in the Generalized Hough Transform (GHT) is presented. The exploitation of the capabilities of VLSI CAD design tools leads to a design methodology based on a *multi-level hierarchical* description and simulation of the circuit. Two different architectures are proposed, based on a semi-sistolic and a sistolic structure, respectively. The system, which can operate in real time, represents the accumulator space in a GHT implemtation with limited memory. The final result of a possible integration of this architecture with a standard cell approach, in terms of area complexity and timing analysis, is shown.

1. INTRODUCTION

The collection and the analysis of data distributions is a very common technique in classical pattern recognition. Particularly, mode analysis and peak detection are very important phases of object classification. There are situations in which both the generation of data and their collection and analysis must be performed in real time; it becomes necessary to devise an efficient system that supports these statistical operations with dedicated hardware.
In this contribution one such situation is analyzed: the detection of 2-D shapes in an image by means of a structural technique, the Generalized Hough Transform (GHT in the following).
In the sequel, after a brief description of the transform, it will be shown that the collecting phase of the process can be organized to benefit from the regular data flow of systolic architectures. The various stages of decomposition of the overall detection process into multiple, hierarchical sub-processes and their structural and behavioural simulation are the central part of the paper; the detailed design of a vote collecting integrated circuit, capable of real time operation, that embodies one of the just mentioned sub-processes, completes the contribution.

2. THE GENERALIZED HOUGH TRANSFORM

The GHT [1] is a powerful tool to describe the structure of a generic shape. It extends a previous formulation, the Hough Transform [2], that characterizes analytic shapes with a limited number of parameters. The generalized version, instead, describes a shape by means of a reference point r and a set of boundary points b. By using the gradient information of the contour of the shape, it is possible to encode the structure of the shape in a table, the so called *reference table*, R, in a way that is insensitive to rotations and translations. Let d(b) be the direction of the gradient at b and d(b-r) the direction of vector connecting the reference point r to a boundary point b; then the reference table stores, as a function of d(b-r)-d(b), the module of b-r. The number C of rows in this table (the *cardinality*) gives an indication of the detail with which a generic shape is described. It can be easily verified that such a representation is rotation insensitive. The recognition phase, indeed, can be organized as the pipeline of three independent processes.
The first process accepts as input the image and produces as output an edge image, which carries both the module and the direction information for each boundary points.
The second process uses the reference table and generates, for each boundary point, a set of C "votes", indicating the positions, relative to that instance of a boundary element, of the possible reference points for the shape.
The third process accumulates and organizes such votes and eventually detects peaks associated to positions in the image where the probabilities that the shape exist are very high.
This abstract description of the generalized Hough process can be thought of as a transformation from the image space to a parameter space. If one is interested

in an actual realization of this process into an architecture, many alternatives are possible.

2.1. Alternative implementation paradigms

A straightforward implementation on a classical Von-Neumann computer of the GHT has time complexity $O(bC+C_p)$, being b the number of boundary points in the image, C the cardinality of the reference table and C_p the cardinality of the accumulator space; the latter term accounts for the analysis phase necessary to identify peaks. The space complexity is $O(C_p)$ for a parameter space storing the location of the reference points.
Parallel implementations for the GHT are much more difficult than analogous solutions for the HT. Indeed, no regular data flow can be identified on mesh structures running the generalized version of the algorithm. Special purpose devices have been proposed and build for the HT; a review of such architectures is available in [3].
As far as the GHT is concerned, previous work [4] has addressed the problem of reducing the space complexity of the method by using a limited memory, cache based system. Since the utilization of the parameter space is generally poor, (sparse votes distribution), as a rule the histogram of the accumulated votes has a few strong peaks and a large majority of weak contributions. A limited memory can therefore accommodate generated and accumulated votes as long as free "slots" are available. When an overflow occurs, a flush must be performed to eliminate weak contributions and to preserve stronger ones, meanwhile making room for more incoming votes. A prototype "vote tallying chip" has been actually fabricated to this purpose [5]. The flushing strategy is crucial to the success of the technique. The work reported here extends this activity and analyzes the vote collecting process so as to highlight smarter strategies to perform the flushing operation.

3. THE NEW ALGORITHM

In a large number of applications, it can be safely stated that the shape(s) of the object(s) is much smaller than the image of scene containing the objects. Let N be the linear dimension of the image. In terms of the structural description just introduced, this means that the following relation holds:

$$\max(b-r) \ll N \quad \forall \; b-r \in R$$

Actually, one can rely on a ratio $N/\max(b-r)$ in the range 20÷40. In other words, the shape intercepts between 13 and 26 rows of an 512x512 image.

This situation has no special bearing in massively parallel implementations of the algorithm. In special purpose, serial implementations, instead, it offers a viable strategy to perform flush operations in a limited memory accumulator. Let us assume that the image enters serially a pipeline of the three processes outlined above. Be y_C the current row number; than, the process that generates the votes produces votes in a range of row addresses determined by y_C and the maximum displacement stored in R: $[y_C-\max(b-r), y_C+\max(b-r)]$. The accumulating process, managing the limited memory, can exploit this situation: upon an overflow, it can flush out weak contributions (having a count below a chosen threshold T) that can no longer receive additional votes, because their row number y is such that $y<y_C-\max(b-r)$.
This strategy has been analyzed in some detail in actual cases; the outcome of this analysis [6] encourages the realization of a dedicated integrated circuit that performs the accumulation in a limited memory and carries out both the flush operation necessary upon memory overflow and a smart collection of the consolidated votes for peak detection.

4. DESIGN METHODOLOGY

The design of the chip has been carried out in several phases, each corresponding to a different level of detail of description of the system. This approach is based on the exploitation of the *multi-level, hierarchical* design methodology [7] which is offered by the CAD tool used in the project. According to these concepts, the circuit has been described and simulated at different levels, starting from a behavioural level to gate level. In the first case, the circuit is described with a Hierarchical Hardware Description Language (HHDL [8]), a high level, Pascal-based language, which can manage concurrent processes. The description and simulation is performed at different, nested levels of hierarchy, in a top-down way, where the behavioural information of the preceeding hierarchical step is the base for a further refinement of the circuit design. In the second case, the circuit is described in terms of elementary gates, which have been characterized according to the electrical specifications of two different standard cell libraries, [9-10].

5. THE IMPLEMENTATION

At the highest level, the scheme of the circuit is shown in fig. 1. Some basic blocks may be identified:

• a RDFIL (ReaDFILe) block which provides the GHT votes. In the simulation of the actual process, it accounts for the extraction of the gradient information from the image and for the table look-up operations associated to the Reference table in the GHT, as explained in section 2. It represents information coming form the outside world and stores the GHT votes, in the form of address-vote, in a file and provides the data to the other modules, when requested. It has three input signals: for opening (S1) and closing (S3) the file and the enable of read operation (S2); and two output signals: the concatenation of the x-y coordinates with the associated vote v (S5) and a mark for end-of-file (S4).
• a MEM (MEMory) block, which accumulates the votes in a limited memory. The block has two input signals: the flag for insertion/extraction modality (S8) and the coordinates-vote data (S9); and two outputs: the concatenation of x-y coordinates with the vote v (S10), and a flag for the overflow condition (S7). This memory can be conceptually described as a modified priority queue, and it is represented in different ways, according to the level of simulation adopted.
• a CT (ConTrol) block, for the control and synchronization of the system operations. It receives as input the coordinates-vote data (S5) and the control signals provided by the other two blocks (S4, S7) and generates the input for the queue (S9), the control signals for the MEMory block (S8), for the RDFIL block (S2,S3), and for the external world (S6, overflow management).

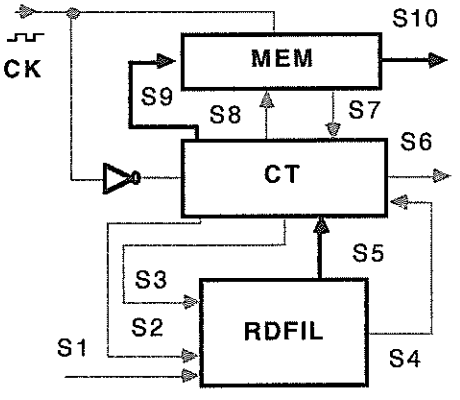


Figure 1. High-level scheme of the system

Blocks CT and RDFIL have been described only at the highest level, while the MEMory block has been represented and simulated also at lower levels, down to gate level. In order to point out the different approaches used in the design, let focus the attention on the different implementations of the limited memory block.

5.1. Systolic collection of votes

The limited memory (namely the accumulator space, block MEM) has the architecture of a modified *priority queue* which may operate in the normal status of insertion/extraction or in the status of overflow management. Each datum stored in the queue consists of three fields, i.e., the x and y coordinates of the generated vote and its value v. In the high-level description, the queue is represented by a vector, in which the data obtained by the concatenation of x, y and v bits are stored. In the Insertion modality, the datum coming from the file (provided by the controller) is compared with all the elements of the vector: if one of them contains the same x-y pair, its vote is increased otherwise it is stored in the first free element of the vector. After each insertion the vector is ordered by decreasing values of votes, since the extraction performed at the first element provides the maximum of the accumulator space. When the vector becomes full, the overflow modality is enabled, with the appropriate signal (S7) flagging this condition. As a consequence, the controller disables any insertion operation and enables only extractions from the queue. In the overflow condition, the input to the MEM block is a particular datum which contains the value of the flush threshold T and the value y_{min} such that $y_{min} < y_c - \max(b-r)$; this is the lower coordinate delimiting the region of interest for the flush test. The overflow management consists of a scanning of the vector, with the deletion of the votes according to the flush algorithm above described and the subsequent vector compaction.
After this fist high-level characterization, the block MEMory has been further refined at lower levels. In a first solution, the accumulator has been implemented by a semi-systolic architecture (see fig. 2). The queue is now represented by a sequence of identical cells, where the flow of information has the same global characteristics defined at the preceding design level: the first cell gives the maximum value of the queue, where the data accumulation and ordering are performed inside each cell. Each cell has three inputs and three outputs; it performs the basic function of accumulation of votes and ordering of the three resulting values. In fact, if two of the inputs have the same x-y address, their votes are summed and attributed to the higher in the ordering; the other is reset to the special configuration ($-\infty$) which stands for the smallest possible vote count. As the votes enter the queue, they are accumulated and ordered, and the higher ones move towards the top of the queue. An overflow occurs when the lowest output of the last cell assumes a value different from $-\infty$. The overflow signal becomes active and this inhibits any insertion operation, while

allowing extraction operations in order to flush the queue.

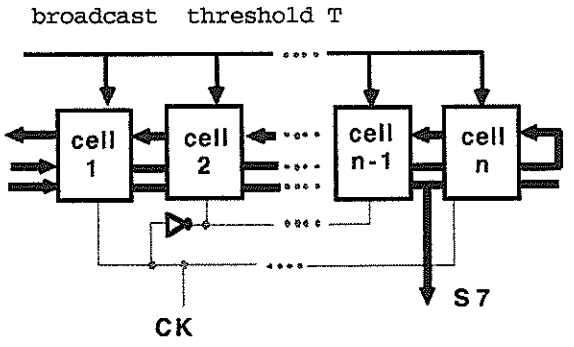


Figure 2. The implementation of the limited memory with a priority queue.

This approach leads to a semi-systolic system, owing to the presence of broadcast signals from the controller to all the cells of the queue. A second solution, aiming at the design of a systolic system, implements the management of the overflow condition in a different way: the broadcast signal has been substituted with a *local* signal, provided by the controller, entering the queue from its low-end (least significant) cell. This signal propagates in the queue with a fixed delay for each cell, carrying the value of the threshold T used in the flush operation and the value of the y coordinate (y_{min}) delimiting the region for the flush test.

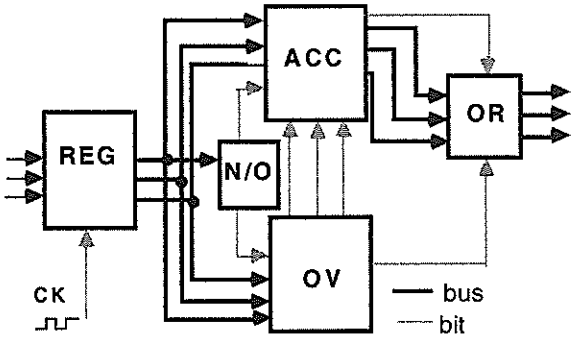


Figure 3. The functional blocks of a cell of the queue.

The hierarchical design has been refined in the systolic solution; a single cell has been further defined by identifying a set of distinct functions, each assigned to separated blocks (see fig. 3). The functions are the following: 1) the storing of data (block REG); 2) the identification of the input as a normal datum (x-y-v) or as an overflow datum (y_{min},T_v), which determines the operative mode of the queue (Normal or Overflow, block N/O); 3) accumulation of data (ACC); 4) overflow management (OV); 5) ordering of votes (OR). At the lowest level of representation and simulation, each block of fig. 3 has been implemented in terms of elementary gates (logical ports, flip-flops...) provided by the two considered libraries of standard cells of 2μm, CMOS technology with two levels of metal. In the case of the ES2 library a possible realization of the circuit shows a complexity of nearly 1800 equivalent gates for each cell of the priority queue, while the timing analysis shows a possible operating frequency of 10 Mhz.

6. CONCLUSIONS

This work, which is partially supported by CNR grant N. 86.01868, is the feasibility study of an actual device under construction as the partial fullfillment of a nation wide project aiming at the realization of a real-time Hough Engine.

REFERENCES

[1] Ballard, D.H., Generalizing the Hough Transform to Detect Arbitrary Shapes, PR 13, (1981) 111-122.

[2] Hough, P.V.C., A Method and Means for Recognizing Complex Patterns, U.S. Patent 3,069,654.

[3] Albanesi M.G., Architectures for the Hough Transform: A Survey, University of Pavia, RI-DIS-08, 1990.

[4] Brown, C.M. and Sher, D.B., Hough Transformation into Cache Accumulators: Considerations and Simulations, CSD, University of Rochester, TR 114, 1982.

[5] Sher, D. and Tevanian A., The Vote Tallying Chip: a Custom Integrated Circuit, CSD, University of Rochester, TR 144, 1984.

[6] Albanesi M. G. and Ferretti M., Geometry Constraint and Limited Memory for the Generalized Hough Transform, 5th International Conference on Image Analysis and Processing, 20-22 September, Positano, Italy.

[7] Hill D.D. and Coelho D.R, Multi-level Simulation for VLSI Design (Kluwer Academic Publishers, 1987).

[8] Sajjan G. Shiva, Computer Hardware Description Languages - A Tutorial, Proceedings of the IEEE, Vol. 67, No. 12, December 1979, pp. 1605-1615.

[9] 2-micron Standard Cell Databook, AMS Austria Mirko Systeme International GmbH, 1988.

[10] Training Guide for SDA Environment, 2.0 Release, 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SYSTOLIC VLSI IMPLEMENTATION OF 2-D DIGITAL FILTERS BASED ON MATRIX DECOMPOSITION

B.G. MERTZIOS
Department of Electrical Engineering
Democritus University of Thrace
67 100 Xanthi, Greece

and A.N. VENETSANOPOULOS
Department of Electrical Engineering
University of Toronto
Toronto M5S 1A4, Canada

Implementation structures of 2-D FIR and IIR linear digital filters via VLSI array processors are presented. The underlying realizations are based on the matrix decomposition approach. The resulting structures are pipelined, paraller, modular, regular, use only local communications and internal local feedback loops and achieve very high throughput and sampling rates.

1. INTRODUCTION AND PRELIMINARIES

The tremendous growth in both number of images and bit rates that have been recently experienced, coupled with the need for fast processing of these images, has led to an intense interest for real-time image processing. This need became evident with the expanding utilization of television imaging to the industrial, medical and military environments [1].

The present trend to meet efficiently the requirement of fast processing, is the use of special purpose hardware and in particular VLSI array processors (APs). Systolic and wavefront arrays [2]-[4],which consist the main classes of VLSI APs, are special purpose VLSI planar arrays of simple processing elements (PEs) that feature regular, modular, nearest-neighbor interconnection computing networks. They are characterized by high computational concurrency, which is achieved by exploiting pipelining and parallelism, and ensures high troughput rates. A number of implementations of one-dimensional (1-D) and 2-D digital filters via systolic and wavefront APs have been presented [5-10].

The recursive bottleneck that appears in the realizations with feedback loops may be overcome by recasting the algorithm using the principle of *look-ahead computation*, in order to increase the number of delays in the feedback loops,and *retiming* to effectively pipeline the computation within the loops [7-9]. Additional techniques for fast processing of recursive algorithms are the bit-level pipelining [10], the block processing [11,12] and the use of internal local feedback loops, whenever this is possible, since they increase the throughput rate.

This paper refers to the implementation of linear 2-D FIR and IIR digital filters, based on the matrix decomposition approach, via VLSI APs. This approach uses the decomposition of the 2-D (or m-D) polynomials and results to general, regular, parallel and modular realization structures of 2-D and m-D digital filters. Special forms of matrix decomposition structures, depending on the particular matrix decomposition,are the Jordan decomposition, the Singular Value decomposition, the Lower-Upper triangular decomposition and the Walsh-Hadamard Transform decomposition [13].

In order to implement both the row and column delays z_1 and z_2 within the execution time of each PE, we use the concurrent 2-D processing. The resulting pipelined implementations are modular, pipelined, regular, with only local communications, use internal local feedback loops and achieve high throughput processing rates.

Using the general matrix decomposition theorem, an arbitrary 2-D rational function of the form.

$$H(z) = \frac{n(z_1,z_2)}{d(z_1,z_2)} = \frac{n(z_1,z_2)}{1+\bar{d}(z_1,z_2)} \qquad (1)$$

where

$$n(z_1,z_2)=\sum_{i=0}^{n_1}\sum_{j=0}^{m_1} n_{ij}z_1^i z_2^j = \sum_{j=0}^{p} \alpha_j(z_1)\beta_j(z_2) \qquad (2)$$

$$d(z_1,z_2)=1+\sum_{\substack{i=0\\(i,j\neq(0,0)}}^{n_2}\sum_{j=0}^{m_2} d_{ij}z_1^i z_2^j = \sum_{j=0}^{r} \gamma_j(z_1)\delta_j(z_2) \qquad (3)$$

where

$$p \geq \text{rank } D, \quad r \geq \text{rank } N \qquad (4)$$

and $D\in R^{(n_1+1)x(m_1+1)}$, $N\in R^{(n_2+1)x(m_2+1)}$ are coefficient matrices of the polynomials $\bar{d}(z_1,z_2)$,and $n(z_1,z_2)$ respectively. It is noted that there is an infinity of the polynomials $\alpha_j(z_1)$,$\beta_j(z_2)$, $\gamma_j(z_1)$,$\delta_j(z_2)$, depending on the decomposition of the matrices N and D.

These latter polynomials have the form

$$a_j(z_1)=\sum_{i=0}^{n_1} a_{ij}z_1^i,\ \beta_j(z_2)=\sum_{i=0}^{m_1}\beta_{ij}z_2^i,\ j=1,2,\ldots,p \tag{5}$$

$$\gamma_j(z_1)=\sum_{i=0}^{n_2} \gamma_{ij}z_1^i,\ \delta_j(z_2)=\sum_{i=0}^{m_2}\delta_{ij}z_2^i,\ j=1,2,\ldots,r \tag{6}$$

2. DESCRIPTION OF THE IMPLEMENTATION

The underlying realization structure of the 2-D IIR digital filters, which is used as the basis for the systolic VLSI implementation, is the general decomposition structure of the direct form II and is shown in Figure 1.

The scanning of each image considered for processing may be organized so that all the pixels belonging to a line accross the one dimension of the image are processed concurrently [15]. The set of the pixels that can be processed simultaneously is called "Concurrent Computational Region" (CCR).

It is seen from Figure 1 that there are in general r branches in the feedback block and p branches in the forward block.

Each one of the 1-D factors $\alpha_i(z_1)$, $\beta_i(z_2)$, $\gamma_i(z_1)$, $\delta_i(z_2)$ is realized by tapped delay structures. The whole implementation is based on the *parallel assembling* of the dynamic elements in both the forward and feedback blocks and in the pipelining of the resulted structure. The layout diagram of the systolic implementation, which is shown in Figure 2, consists of $n_2 + m_2 + 2$ bi-directional processing units (PUs) in cascade configuration. Two types of PUs are used, whose structure is shown in Figures 3 and 4 respectively. Each PU has a number of multipliers and adders, which operate in parallel. Moreover note that the last (m_2+1) PUs that contain the delays z_2, consist of r+p elementary PEs operating independently and in parallel (Figure 4). The execution time of each PU is given by

$$\tau = Mu + Ad \tag{7}$$

However, due to the existence of a local feedback loop in the first PU, one "blank" data is needed among the successive input samples in order to ensure proper synchronization. Thus, the time scaling factor of each block, and therefore of the whole implementation, equals to $\alpha = 2$ [4]. Actually α equals to the latency of the local feedback loop.

The throughput rate of the output vector y(k) is then given by

$$R = \frac{1}{T} \leq \frac{1}{\alpha T} = \frac{1}{2\tau} \tag{8}$$

The p-1 additions of the partial results of the p forward branches may be executed within the cycle time $T \geq 2\tau$.

It is seen from Figure 3 that the implementation uses block pipelining. The total throughput delay of all the successive PUs, are found to be

$$\Lambda = (n_2+1)(m_2+1)T \geq 2(n_2+1)(m_2+1)\tau \tag{9}$$

The derived implementations are comprised of similar PEs or PUs in a linear configuration with local communications and are characterized by modularity, regularity, high concurrency and high throughput rates. Both parallelism and pipelining contribute to the system's concurrency.

Two types of delays are involved in the two-dimensional implementation structure: a row delay operator z_1 and a column delay operator z_2, defined by

$$z_1[x(k,l)] = x(k-1,l)$$
$$z_2[x(k,l)] = x(k,l-1) \tag{10}$$

Due to the nature of concurrent 2-D scanning, the delays z_1 and z_2 correspond to time delays

$$z_1 = (L-1)T$$
$$z_2 = LT \tag{11}$$

where L is the number of pixels on the antidiagonal, where the pixel (k,l) belongs.

Note that the principle of rescaling of time unit is applied by setting $z_1' = z_1^{1/2}, z_2' = z_2^{1/2}$ in the corresponding PUs [2], [3], since $\alpha = 2$.

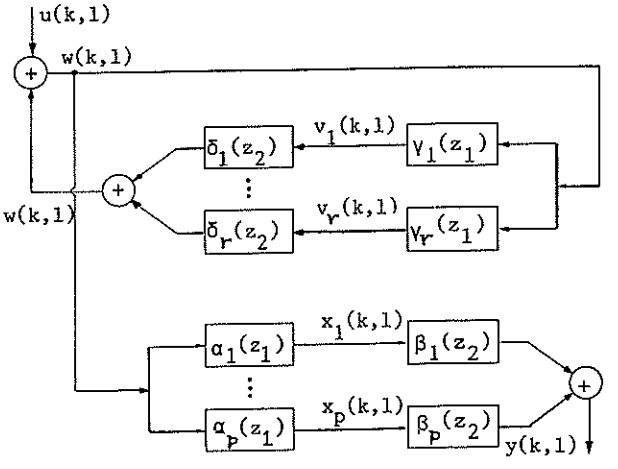


Figure 1. Block diagram of the direct form II block decomposition based implementation.

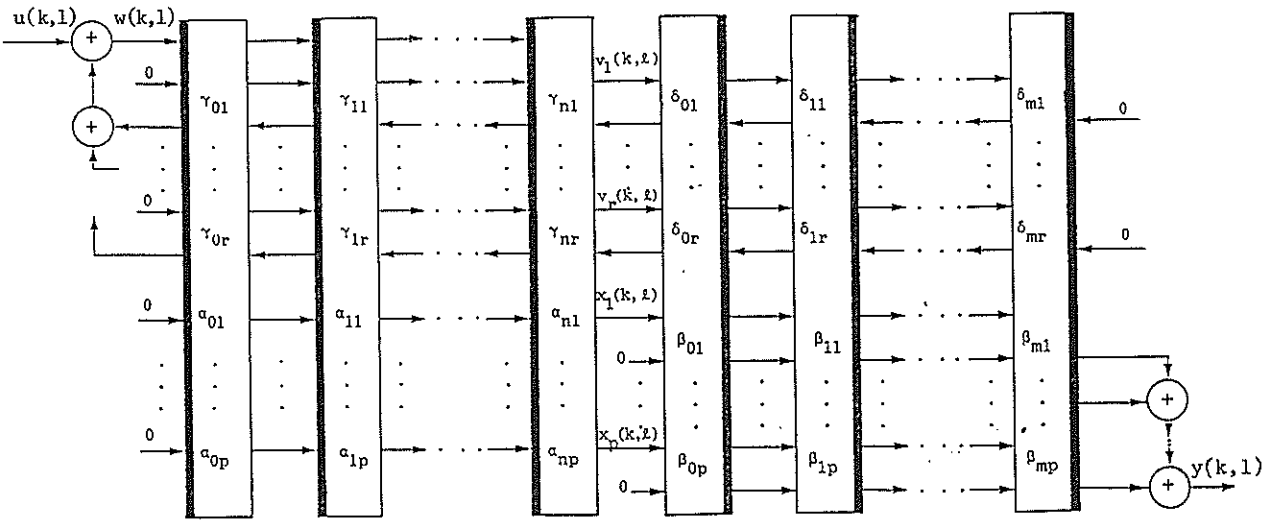


Figure 2. The layout diagram of the systolic implementation of the 2-D IIR digital filter.

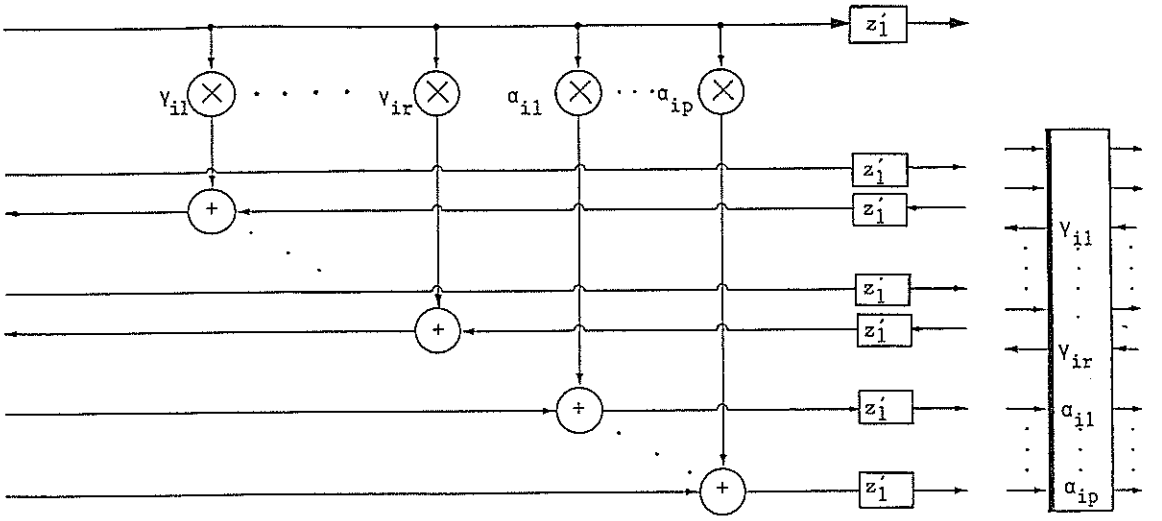


Figure 3. The structure and the symbol of the processing unit that implements the blocks containing the delay z_1.

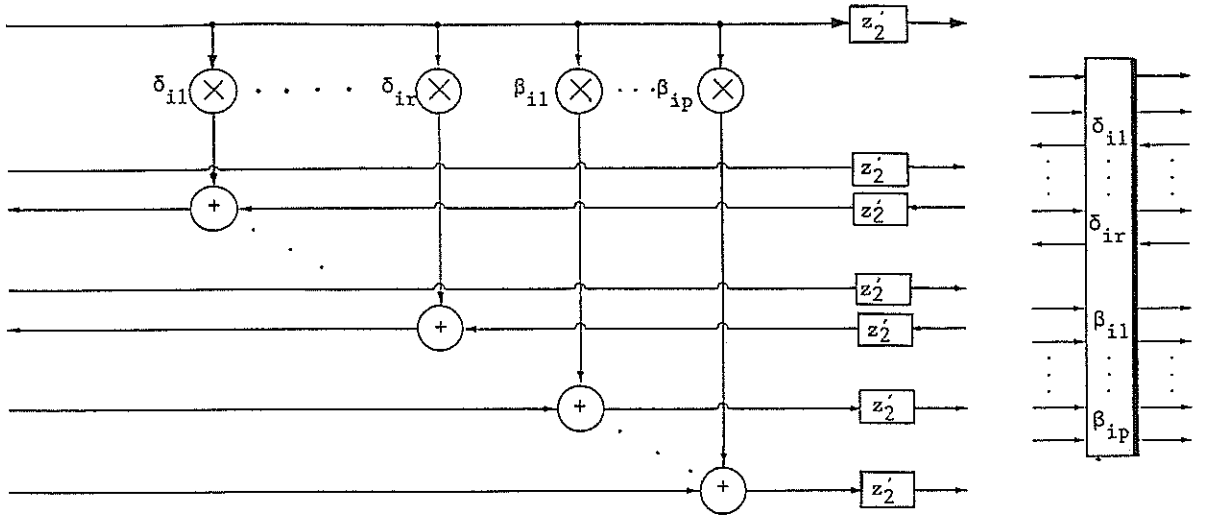


Figure 4. The structure and the symbol of the processing unit that implements the blocks containing the delay z_2.

REFERENCES

[1] A.N. Venetsanopoulos and V. Cappellini, "Real-Time Image Processing", *in Multidimensional Systems: Techniques and Applications*, Marcel Dekker Inc., ch. 8, pp. 345-399,1986.

[2] S.Y. Kung ,*VLSI Array Processors*, Prentice Hall, Englewood Cliffs, N.J. 1987.

[3] H.T. Kung, "Why systolic architectures", *Computer*, vol. C-15, pp. 37-45,January 1982.

[4] S.-Y. Kung, "On supercomputing with systolic/wavefront array processors", *Proc. IEEE*, vol. 72, pp. 867-884, July 1984.

[5] H.H. Lu, E.A. Lee, and D.G. Messerschmitt, "Fast recursive filtering with multiple slow processing elements", *IEEE Trans. Circuit Syst.*, vol. CAS-32, pp. 1119-1129, November 1985.

[6] S.K. Rao and Th. Kailath, "VLSI arrays for digital signal processing", *IEEE Trans. Circuits Syst.*, vol. CAS-32, pp. 1105-1118, November 1985.

[7] K.K. Parthi and D.G. Messerschmitt, "Concurrent cellular VLSI adaptive filter architectures", *IEEE Trans. Circuits Syst.*, vol. CAS-34, pp. 1141-1151, October 1987.

[8] K.K. Parthi and D.G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters-Part I: Pipelining using scattered look-ahead and decomposition", *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-37, pp. 1099-1117, July 1989.

[9] K.K. Parthi and D.G. Messerschmitt, "Pipeline interleaving and parallelism in recursive digital filters, Part II: Pipelined incremental block filtering", *IEEE Trans. Acoust., Speech Signal Process.*, vol. ASSP-37, pp. 1118-1134, July 1989.

[10] K.K. Parthi and D.G. Messerschmitt, "A bit-parallel bit level recursive filter architecture", *Proc. IEEE Int. Conf. Comput. Design*, New York, 1986.

[11] B.G. Mertzios, "Block realization of 2-D IIR, digital filters", *Signal Processing*, vol. 7, pp. 135-143, Oct. 1984.

[12] B.G. Mertzios, "Systolic and wavefront arrays block implementation of two dimensional digital filters", *AEU*, vol. 44, No. 1, pp. 50-58, 1990.

[13] A.N. Venetsanopoulos and B.G. Mertzios, "A decomposition theorem and its implications to the design and realization of two-dimensional filters", *IEEE Trans. Acoust., Speech Signal Processing*, vol. ASSP-33, pp. 1562-1574, Dec. 1985.

[14] B.G. Mertzios and A.N. Venetsanopoulos, "Implementation of quadratic digital filters via VLSI array processors", *AEU*, vol. 43, No. 3, pp. 153-157, 1989.

[15] A. Fettweis, "Multidimensional Circuits and Systems Theory", *Proc. IEEE Int. Symposium Circuits Syst.*, Montreal, Canada, May 1984.

ACKNOWLEDGEMENT. The authors acknowledge the partial support of the NATO Grant for International Cooperation in Research No. 86/0063.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

VLSI DATA-PATH STRUCTURE FOR A PIPELINE 2D-FHT IMPLEMENTATION

Juan A. MICHELL, Angel M. BURON, José M. SOLANA, Gustavo A. RUIZ

Departamento de Electrónica. Universidad de Cantabria.
Avda. los Castros s/n. 39005. Santander. SPAIN

A computing architecture for the 2D-FHT tailored to its ASIC implementation is presented. Having two sections, one comprising only 1D-FHTs and another one dealing only with 2D-FHTs, the architecture, internal operation and single ASIC implementation of the latter one is described.

1. INTRODUCTION.

Transform based image coding techniques generally have a transform stage and a coding stage. From the point of view of its real-time implementation, the transform stage results the most demanding one and is the main purpose of the work reported here.

The use of algorithms termed as *fast* leads to a reduced number of arithmetic operations. Organizing those computations in order to optimize properties for its circuit implementation like regularity, recursivity, adequate concatenation of operations, simplicity of control logic, size of intermediate storage, etc. may result in improvements of the same order and even greater than those attainable through the use of fast computation algorithms which generally aim to the reduction of computation time assuming general purpose processor architectures.

The Haar Transform has been used mainly for image processing due to its low computing requirements[2,5]. Using a *decimation* methodology, similar to the one employed in the derivation of *fast* algorithms for other transforms, the 2D-FHT can be expressed in terms of only 2x2 2D-FHTs and 1D-FHTs of diverse lengths. On previous works we have explored various approaches[1,3,4,6] to the implementation of the Fast Haar Transform (FHT), in one and in two dimensions.

In the present work we report a computing architecture for the 2D-FHT having advantageous properties when considered for its ASIC implementation. It results adequate for implementing fixed length transforms, and it leads to a simplified implementation due to the highly regular structure of the intermediate storage, affordable complexity for the data flow control and reduced number of arithmetic blocks.

2. 2D-FHT ARCHITECTURE FOR REAL-TIME IMPLEMENTATION.

The architecture we propose for the real-time implementation of the 2D-FHT is as fig. 1 shows. Input data values are supposed to arrive in raster order. The computation of a NxN 2D-FHT can be done[4] by computing first N/2xN/2 2x2 DHTs, they produce N/2xN/2 coefficients and 3(N/2xN/2) intermediate results, which we label A,B and C. With results A, a new set of N/4xN/4 coefficients are obtained, along with 3(N/4xN/4) intermediate results (A', B' and C') by performing 2x2 DHTs on them. With results B and C, N/2xN/2 new coefficients can be obtained as well as N/2xN/2 new intermediate results by doing add&subtract operations with them. Iteration of this process $\log_2(N/2)$ times allows to obtain 2(N/2xN/2) coefficients from sets B and C. The successive add& subtract operations can be arranged as two different sets of 1D-FHTs of length N/2; the first one by considering N/2 sequences formed by N/2 consecutive B data as they are obtained from raster ordered data, and the second one, by considering N/2 sequences formed by taking one data every N/2 of them from C data, keeping also its ordering. The same process is to be applied to results B' and C' but changing N/2 by N/4, and so on.. This leads to the conclusion that the computation of a NxN 2D-FHT can be done in terms of 2x2 DHTs and variable length 1D-FHTs as the proposed architecture shows. The storage for intermediate results is incorporated to each processor as its structure and operation are quite different. The three main blocks operate in pipeline and also each one of them have internal pipeline architecture.

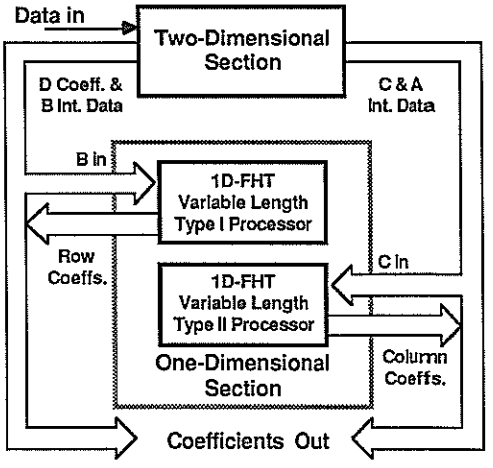


Figure 1: 2D FHT Architecture.

Within the one-dimensional section, the type I processor block (TI) in fig.1 performs a half of the 1D-FHTs needed in such an scheme and it has been already prototyped in single chip form[1]. The type II processor block (TII) performs the other half of the 1D-FHTs needed and has a different architecture than that of TI. Its associated register file needs a more elaborate control, and, due to the ordering of its input data, it shares some characteristics with the control for the data path presented here.

The two-dimensional section performs all the 2x2 2D-FHTs needed and to its architecture and implementation in a single ASIC is mainly dedicated the present paper.

3. DATA-PATH STRUCTURE OF THE TWO-DIMENSION SECTION.

Fig. 2 shows the block diagram of the 256x256 FHT two-dimensional section. It includes a 2x2 DHT computational element (CE), a shift register array (SRA) implementing a delay commutator and a control unit. The CE operates on input data pairs, or delayed input data pairs, or delayed intermediate data pairs produced at a previous computational level. Its two outputs give alternatively [A&B] and [C&D]. **A**, **B** and **C** are intermediate data; **A** to be stored in the SRA, **B** and **C** the respective inputs of TI and TII processors. **D** are final coefficients.

The memory section comprises a shift register bank to store an input data line and seven shift register banks to store intermediate **A** data, one of them for each computational level. The width and length of each bank are adapted to the requirements of the corresponding level.

The control signals for the CE, the clock signals of the SRA and the signals enabling access from each shift register bank to the CE inputs are all derived from a single external clock input signal having a frequency twice of the input rate.

3.1. The 2x2 DHT Computing Element.

Fig. 3 shows the block diagram of the CE. It has three blocks in pipeline; two of them are adder and subtracter circuits and the third one a delay commutator. The arithmetic circuits are 22&23 bits parallel ones designed to minimize ripple carry& borrow propagation delays as well as area. They have been arranged in slices of two bits each, in order to maximize the use of common logic for addition and subtraction and to reduce the number of logic levels in the ripple carry&borrow path.

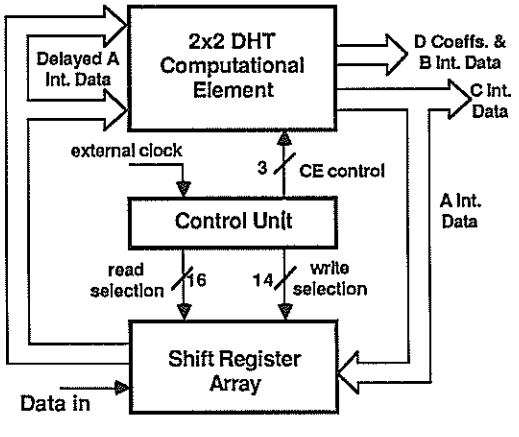


Figure 2: Two-Dimensional Section for 256x256 FHT.

2x2 transforms are continuously being computed by the two adder&subtracter connected in pipeline by the delay commutator at a rate equal to the input data. This arrangement halves the needs of computing elements as well as the number of data lines required for achieving the transform, having both reductions a strong impact in total complexity and silicon needs. The pipeline has then two fully combinational circuits (+&-) and a sequential one (delay commutator), its control requiring just the signals shown in fig. 4. In order to get some insight about its operation let us suppose that at time **i** begins the computation of a 2x2 DHT and:

$$X = A(0,0)\ ;\ Y = A(0,1) \quad \text{so that:}$$
$$Q_+ = A(0,0) + A(0,1);\ \ Q_- = A(0,0) - A(0,1).$$

At time **i+2**:

$$X = A(1,0)\ ;\ Y = A(1,1) \quad \text{so that:}$$
$$Q_+ = A(1,0) + A(1,1);\ \ Q_- = A(1,0) - A(1,1).$$

While **Id**=1 these values are latched at the input of the delay commutator and later undergo two internal transfers between dynamic storage cells while **ht**=1 and **vt**=1 respectively, so that at time **i+3**:

$$U = A(0,0) - A(0,1)\ ;\ V = A(1,0) - A(1,1) \text{ and:}$$
$$O_+ = A(0,0) - A(0,1) + A(1,0) - A(1,1) = C_{XX}$$
$$O_- = A(0,0) - A(0,1) - A(1,0) - A(1,1) = D_{XX}$$

Similarly, at time **i+5**:

$$U = A(0,0)+A(0,1);\ V = A(1,0)+A(1,1) \text{ so that:}$$
$$O_+ = A(0,0) + A(0,1) + A(1,0) + A(1,1) = A_{XX}$$
$$O_- = A(0,0) + A(0,1) - A(1,0) + A(1,1) = B_{XX}$$

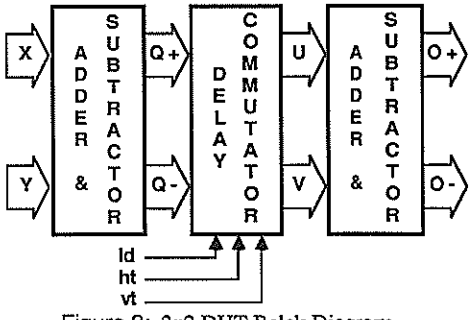


Figure 3: 2x2 DHT Bolck Diagram.

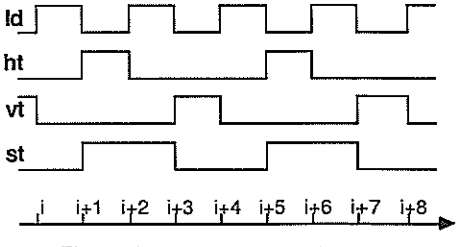


Figure 4: 2x2DHT Control Signals.

The signal pairs [A,B] and [C,D] will be valid during time intervals twice the period of signal **Id**. With reference to fig. 5, the signal **st** it is so from time **i+1** to time **i+3**, from **i+5** to **i+7**, and so on. This signal is used to obtain the ones enabling the storage of intermediate data A in the SRA.

3.2. The Shift Register Array.

The storage of all intermediate data in the two-dimensional section is done in shift register banks (SRB) made up of two or four shift register chains (SRC) grouped as fig. 5 shows. The first computational level operates on delayed input data. Alternate rows of scan data are entered in consecutive pairs in the uppermost SRB (2x128x8). When the following row of data arrives they enter the CE in pairs alternating with data pairs stored in the SRB. Using matrix notation and supposed at the beginning of a frame, the pairs entering the CE would be:

{A(0,0), A(0,1)} {A(1,0), A(1,1)} -> 1st. DHT
{A(0,2), A(0,3)} {A(1,2), A(1,3)} -> 2nd. DHT
{A(0,4), A(0,5)} {A(1,4), A(1,5)} -> 3rd. DHT

and so on.

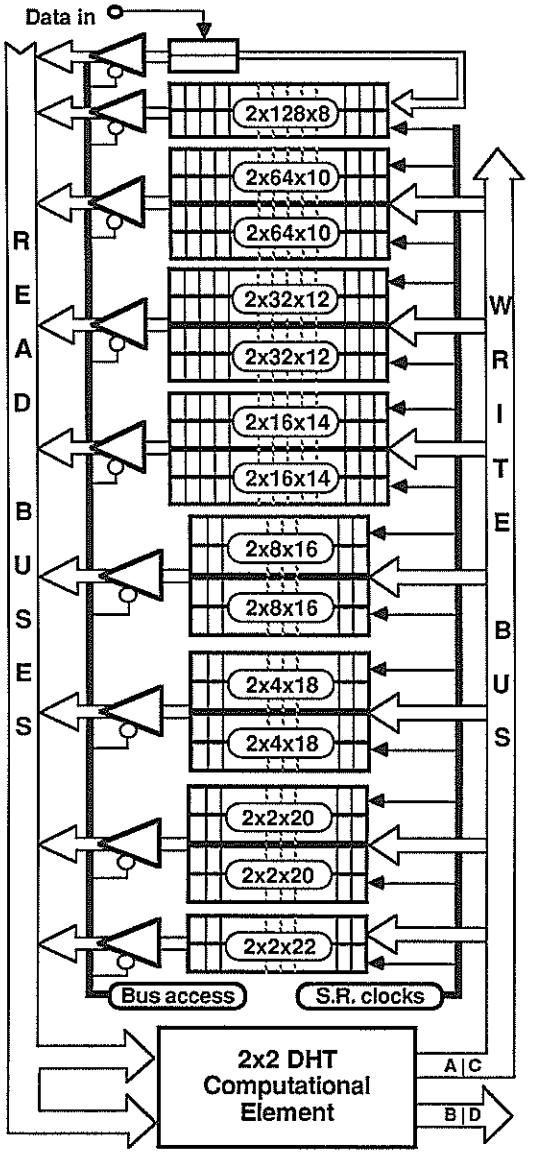


Figure 5: Shift Register Array Structure of Two-Dimensional Section for 256x256 FHT.

The remaining SRBs store intermediate data produced one for each 2x2 DHT of the previous computational level (previously labeled **A**). SRBs arranged having **a** width of four data (all except the ones at the top and the bottom) are written (filled-up) first the two upper SRCs and then the two lower ones and read (emptied) alternating two data from the upper SRCs with two from the lower ones and so on. The last SRB stores the four data needed for the last 2x2 DHT.

Although the read and write buses have a width of 2x22 and 24 lines respectively, in order to accommodate the greatest possible result that can be attained (all 256x256 points having a hex. value of FF) the SRBs have been kept in its size to the maximum value attainable in its computational level, extending its MSB to enter the read buses as the arithmetic is done on 2's complement representation and all input data are considered positive integers. The total number of data that is needed to be stored for the 256x256 FHT equals 766 and this represents 8104 shift register stages with the arrangement shown in fig. 5. The upper four SRBs have been implemented using dynamic storage so that area and power drain are minimized. They correspond to nearly 90% of the total shift registers used. The remaining four SRBs (1048 shift register stages) have been implemented using static storage as they result less frequently accessed.

3.3. The Control Section.

All the control signals are generated from a single external clock input having twice the frequency of the incoming data (pixels). This is accomplished by a 17 bit binary counter and the combinational logic conceptually partitioned in blocks as shown in fig. 6.
Control signals for the CE **-ld,ht,vt-** and the signal to enable the storage of A data **-st-** (fig.4) are obtained from **Q0,Q1,Q2**.

The SRA needs for its operation 16 read signals: 8 for even SRCs (**Re**), 8 for odd SRCs (**Ro**), and 14 write signals: 7 for even SRCs (**We**) and 7 for odd SRCs (**Wo**). As it can be suspected from fig. 6, each of these signals is generated from one index level qualifier and the adequate (read or write) length qualifier. The read and write length qualifiers are the same signals for each level except for a delay of eight clock periods, which is the delay from input to output of the CE due to its pipeline operation. Both sets are obtained from **Q1-Q8** The computational levels are eight, but 15 signals are needed for this purpose, 8 for reading, 7 of them also used for writing in odd SRCs, and another 7 for writing in even SRCs, they are all obtained from **Q9-Q16**.

Boolean expressions for the combinational part of the control unit have been derived following a Rademacher-Walsh[7] spectral synthesis methodology

The circuit implementation has been carried out having as main criteria layout area minimization and control signal skews avoidance. The binary counter has been designed as a chain of D flip-flops and its outputs latched so that all

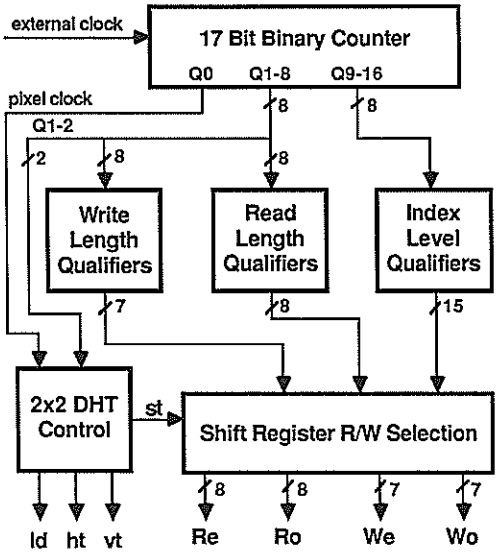


Figure 6: Block Diagram of the Generation of Control Signals for 256x256 FHT.

appear to change simultaneously to the combinational circuit. This reduces somewhat the maximum attainable counting frequency (more than twice the needed here anyhow) but increases greatly layout regularity and reduces area when compared with a synchronous counter implementation. Part of the index level qualifiers signals are obtained in a ripple carry-like circuitry placed as an extension of the D flip-flop chain and previous to the latches, so avoiding also for them the delays associated with signals depending on many outputs of the counter through several levels of logic. Counter outputs, index level qualifiers and/or its complements (as needed) form the input signals of a NMOS-like PLA circuit whose buffered outputs constitute the set of control signals for the rest of the chip.

4. IMPLEMENTATION FEATURES

A 2μ double-metal CMOS full-custom single ASIC has been designed to implement the two-dimensional part of the 256x256 2D-FHT (8-bit input points 2x24-bit output coefficients and intermediate data). Most of its area is taken by the SAR; the biggest SRB is the set of four SRCs having size 64x10, with its 10320 transistors and 1630x1676 μm^2. The CE takes 1912x422 μm^2 and 2568 transistors. Total area is slightly under 30 mm^2, and total transistor count nearly 49000 Its fabrication is scheduled for July'90 as part of C.N.M. national M.P.C. From simulation results, the maximum input data rate can be conservatively expected to be in excess of 15 Mpxls/sec.

5. CONCLUSIONS.

In the present work we have presented a computing architecture for the 2D-FHT tailored to its ASIC implementation. It can be split in two sections, one comprising only 1D-FHTs and another one dealing only with 2D-FHTs. The architecture and internal operation of the two-dimensional section have been described.

ACKNOWLEDGEMENT

The authors wish to acknowledge the Spanish Government Research Commission (CICYT) for the financial help received in support of our group of work.

REFERENCES

[1] A.M.Burón, J.A.Michell, J.M.Solana. **Single-Chip Fast Haar Transform at Megahertz Rates.** 3rd. International Workshop on Spectral Techniques. October 1988, pp.8-17. Dortmund.

[2] R.T.Lynch, J.J.Reis. **Haar transform image coding.** Proc. 1976 National Telecom. Conf. Dallas, TX.

[3] J.A.Michell, A.M.Burón, J.M.Solana. **1D&2D FHT for Real Time Video Data Processing.**MIMI'88. 38th. ISMM International Symposium. June 1988, pp. 600-604, San Feliu, Spain.

[4] J.A.Michell, A.M.Burón, J.M.Solana. **Single-chip-Based Recursive Structure for Real Time Image Transform** on ***Signal Processing IV: Theories and Applications.*** EUSIPCO'88. Elsevier Sc. Publ. B.V. (North-Holland). September 1988, pp. 315-318. Grenoble.

[5] A.Netravali,J.Limb. **Picture Coding: A Review.** Proceedings of the IEEE, vol. 68, n.3, March 1980. pp. 366-406.

[6] J.A.Michell, A.M.Burón, J.M.Solana. **Architectures for VLSI real-time FHT.** MIMI'89. 39th. ISMM International Symposium. June 1989, pp. 213-216. Zurich, Switzerland.

[7] S.L.Hurst, D.M.Miller, J.C. Muzio. **Spectral Techniques in Digital Logic.** Academic Press, 1985.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

OPTIMAL ARCHITECTURE AND TIME SCHEDULING OF A DISTRIBUTED ARITHMETIC BASED DISCRETE COSINE TRANSFORM CHIP

I. Defilippis, U. Sjöström, M. Ansorge, F. Pellandini

Institut de Microtechnique Université de Neuchâtel Rue A.-L. Breguet 2
CH-2000 NEUCHATEL, Switzerland

This paper presents a chip for the computation of 16x16 point DCTs. The chip architecture and the time scheduling have been optimized considering several criteria. The obtained circuit is powerful, flexible, compact and easy to test.

1. INTRODUCTION

The use of the Discrete Cosine Transform (DCT) [1] for image processing has dramatically increased during the past years. The DCT is particularly well suited for image transform coding applications. Taking several important performance criteria (such as rate distortion, data decorrelation, energy compaction, etc.) into consideration, the DCT is superior to other orthogonal transforms [2].

A powerful chip for the computation at video rates of the DCT on 512x512 point images subdivided into 16x16 point frames was presented in [3]. Successively, its architecture and time scheduling have been optimized and improved. This, considering three criteria: elimination of the hardware redundancies, elimination of the time inactivities of the various sub-systems and improvement of the chip test facilities. The final circuit is also flexible: both forward and inverse, 1- or 2-Dimensional DCT, with 4-, 8-, or 16-point vectors can be performed with the same hardware.

The emphasis of the paper is put on the description of the chip architecture and time scheduling. Original architectural choices related to this implementation will be explained in details.

2. DCT ALGORITHM

Many variants of the DCT algorithm have been proposed over the years. For the present implementation, the Modified Symmetric DCT (MSDCT), first described in [4], has been chosen. The MSDCT is a variant of the Symmetric DCT [5] offering some additional advantages. Given a N point data sequence $\{x(n)\} = x(0), x(1),..., x(N-1)$, the 1-D forward and inverse MSDCT have the following form:

$$X(k) = \sqrt{\frac{2}{N-1}} \sum_{n=0}^{N-1} c_n \, x(n) \cos\left(\frac{nk\pi}{N-1}\right) \qquad k = 0, 1,..,N-1 \quad (1)$$

$$x(n) = \sqrt{\frac{2}{N-1}} \sum_{k=0}^{N-1} c_k \, X(k) \cos\left(\frac{nk\pi}{N-1}\right) \qquad n = 0, 1,..,N-1 \quad (2)$$

where
$$c_i = \begin{cases} 1/2 & , \text{ i=0 and i=N-1} \\ 1 & , \text{ otherwise} \end{cases} \quad (3)$$

This can be expressed in a matrix form ($\mathbf{x}$ and $\mathbf{X}$ are N-point vectors while $\mathbf{A}$ is the NxN point transform matrix):

$$\mathbf{X} = \mathbf{A}\,\mathbf{x} \;,\; \mathbf{x} = \mathbf{A}\,\mathbf{X} \quad (4)$$

Extension to 2-Dimensional transform is easy due to the separability of the algorithm:

$$\mathbf{X} = \mathbf{A} \times \mathbf{A}^T \quad (5)$$

Note that $\mathbf{X}$ and $\mathbf{x}$ are now NxN point matrices. A 2-D transform can be computed using a 1-D transform hardware and making a matrix transposition of the intermediate result in between.

3. ARITHMETIC

Distributed Arithmetic (DA) [6] is very suited for calculating short length discrete transforms. It also leads to highly regular and easy-to-test structures. Essentially, Distributed Arithmetic supports direct calculation of innerproducts. Thus, direct realization of the transform algorithm is possible. This allows a maximum of parallelism on the algorithmic level: one DA-Processor (DAP) can be used for each row of the transform matrix. In this way, a full 1-D transform can be computed at the same time.

The basic innerproduct (vector multiplication), is simply calculated by adding, shifting and accumulating precomputed code-words stored in a *look-up-table*. All arithmetic and logic operations are performed by a special ALU called the Shift-Accumulator (SA), while the look-up-table is normally implemented as a ROM. The size of the ROM is dependent on the number of terms in the innerproduct (the number of points in the

transform). In the straightforward case, this size is equal to 2^N (65536 words ROM for a 16 point transform). Fortunately, due to row or column symmetries in the transform matrix A, the input or output data can be pre- or post-added respectively. The number of terms in each innerproduct is then reduced to the half. Consequently , the ROM tables have to contain $2^{N/2}$ words (256 for a 16 point transform). Using a technique called Offset Binary [7,8], the look-up-tables size can be further reduced by a factor of two. The final ROMs only contain $2^{N/2-1}$ words (128 words in the 16 point case).

4. BASIC BUILDING BLOCKS

This section briefly discusses the basic building blocks of the developed circuit. A more detailed description of the same blocks can be found in [9].

The 1-D transform is performed by the *1DDP* (1-D DCT Processor) shown in figure 1. The 1DDP contains 16 DA-processors, which allow the concurrent computation of 16 innerproducts. The row symmetries of the transform matrix are used here and lead to the pre-adder pre-subtractor structure. The XOR row before the decoders is due to the use of the Offset Binary technique. Shimming delays (not shown on figure 1) must be inserted at the output of the SAs in order to bring the latency time to a multiple of the data wordlength. This allows the synchronization of the 1DDP with the other chip building blocks.

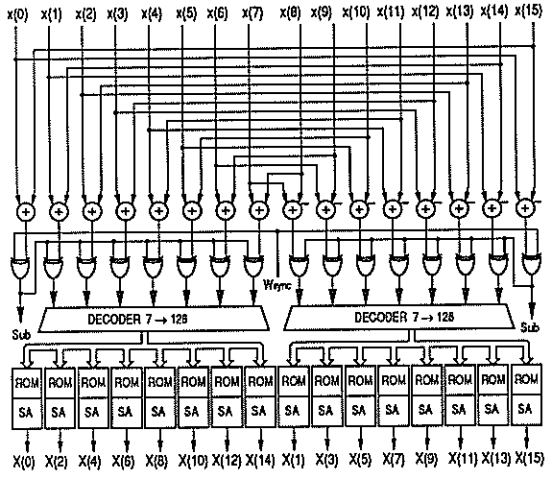


Figure 1 1-Dimensional DCT Processor

Storage and transposition of the intermediate frames are insured by the IRM (Intermediate Result Memory). The IRM is a special purpose high speed single access dynamic RAM. On-chip storage of intermediate data is not strictly necessary but it represents a non-negligible saving of external hardware and a reduction of the I/O bandwidth by a factor of two. Furthermore, a RAM exactly tailored to the application can be employed. In fact, since the DCT is a fixed algorithm, the IRM addressing is cyclically repeated. For this reason, pipelining can be introduced between the RAM components (decoder, RAM matrix, input and output buffers). A pipelined memory is well suited for future high speed versions (50-100 MHz) of the chip.

Serial input- and output- data is required for Distributed Arithmetic. On the other hand, parallel data is to prefer for chip I/O and intermediate data storage. The necessary serial to parallel and parallel to serial conversions are performed by bidimensional Shift Register Banks (SRB) (figure 2).

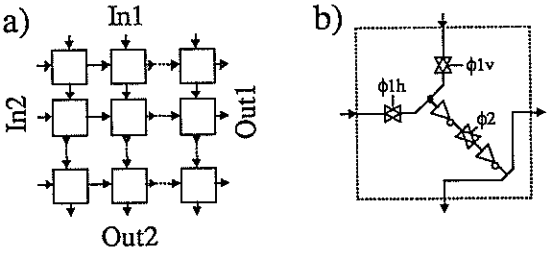


Figure 2 Shift Register Bank:
a) Organization.
b) Basic Cell.

5. CHIP SPECIFICATIONS

A wordlength W_d of 16 bit was chosen for both chip I/O and internal data representation. This is convenient from an implementation point of view and it also results in a sufficient dynamic range (an input wordlength of 8 to 9 bit and an output wordlength of 12 bit is usually enough for image processing applications). Simulations have indicated that a coefficient wordlength W_c of 10 bit for the DAP ROMs is adequate for normal applications. Following these specifications, all other basic building blocks parameters can be set. For the 1DDP, the ALU will be 10 bit wide when the ROMs will contain 10 bit coefficients. The IRM must store a complete frame of 16x16 pixels at 16 bit/pixel. In other words, a 4096-bit storage is necessary. Finally, each SRB contains 16 registers of 16 bit each.

6. CHIP ARCHITECTURE AND TIME SCHEDULING

The architecture of the chip and the time scheduling are strictly interdependent. For this reason both subjects will be treated in this section. Two different architectures will be discussed. The first one, architecture-A, is rather intuitive but not optimal. Thanks to a modified time scheduling, the

second one, architecture-B, has shown to be optimal considering several criteria. For this reason, it has been chosen for integration.

A first solution for the DCT execution is the straightforward calculation of the algorithm, frame-after-frame. First, a whole frame is transformed and stored line-by-line in the IRM. Then, the intermediate frame is read column-by-column from the IRM, (performing an implicit matrix transposition), and it is transformed in the second dimension. This leads to the architecture-A (figure 3). Here, 16 data words are loaded in parallel form into SRB1 from the external input or from the IRM. Subsequently, the same data words are serially shifted into the 1DDP. The innerproducts are computed in bitserial form by the DA-processors of the 1DDP. Once the computation of the innerproducts is finished, the results are read (again in parallel form) from SRB2 and are written into the IRM or transferred outside the chip.

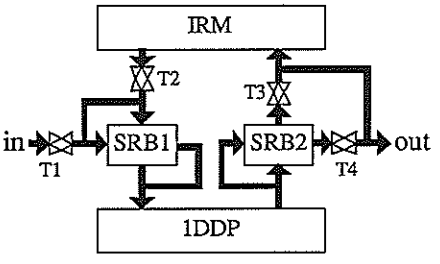


Figure 3 Architecture-A. T1 to T4 are three-state bus switches (CMOS transmission gates).

It is convenient to include the pre-adders and pre-subtractors of the 1DDP in the SRB1 block. Three-state buffers can be added at the output of these pre-adders/subtractors. Consequently, no multiplexers have to be inserted between the SRB1 output buses.

It has shown to be advantageous to represent the data with 16 bit. All blocks can then work synchronously. If a shorter wordlength was chosen, 4 SRBs would be indispensable, leading to a more complex circuit.

For a specific block, the delay time T_D is given by the time separating the input and the output of a given data element. It can easily be verified that, $T_{DI} = 2*W_d = 32T$ for the 1DDP and that $T_{DS} = W_d = 16T$ for both SRB1 and SRB2, where T stands for the basic clock cycle time.

Due to these delays, architecture-A has a serious drawback. The transformation in the second dimension can only start if the intermediate frame is available and transposed. Consequently, the pipeline chain formed by SRB1, 1DDP and SRB2 must first be totally emptied at the end of the transformation in the first dimension. Thus, 64 clock cycles are lost for each frame in between the row and the column transformation. Furthermore, the control signals become irregular and different for each block. This gives a total (frame) cycle time T_{CA} of 256+64+256=576T, while the total delay time T_{DA} equals 384T. The total latency time (time between the first data is loaded on chip and the last result is unloaded) T_{LA} equals 640T.

A much simpler and more efficient solution is obtained by allowing a higher degree of pipelining. The resulting architecture is shown in figure 4.

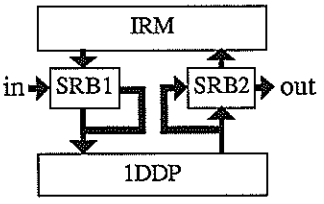


Figure 4 Architecture-B.

Here, SRB1 is loaded alternatively with data coming from the external input bus or from the IRM. SRB2 works in an analog way. This implies that the chip *always works concurrently on two consecutive frames*. The 1DDP transforms alternatively one line from the n:th frame and one column of the n-1:th frame. When the n:th frame starts to be loaded and transformed row-by-row on the chip, the (n–1):th frame is fully stored in the IRM and is ready to be transformed in the vertical direction. Line-by-line, the transformed rows of the n:th frames will replace the columns of the (n-1):frame in the IRM. For 64 clock cycles, even three consecutive frames are processed concurrently.

Architecture-B has many advantages over architecture-A. All blocks work all the time and they are simple to synchronize. Control signals become regular, and can easily be generated from a single main clock signal. In addition, this main clock becomes exactly two times the pixel rate; this dramatically simplifies the interfacing of the chip with external circuitry.

Architecture-B takes full advantage of the pipelining. The highest possible throughput is obtained (T_{CB} = 512T). This maximal performance is paied in terms of an increased delay- and latency-time; T_{DB} = 624T and T_{LB} = 1120T. However, these two last parameters are rarely critical in image processing applications.

7. TEST FACILITIES AND ALGORITHMIC VARIANTS

Due to the chip complexity, it is important to be able to test each block separately. For this reason, some extra bidirectional bus switches have been

inserted in the final implementation (figure 5). These bus switches represent a very small increase of the total chip area (simple CMOS transmission gates are used). As it can be easily verified, now, each single block can be tested separately.

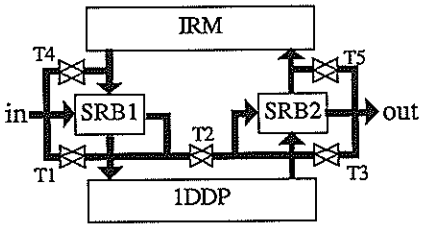


Figure 5 Test architecture.

The test hardware also adds new features to the chip. 1-D transforms can now easily be performed since the 1DDP is directly loadable and unloadable from the external world. Moreover, parallel- or serial-form can be independently selected for the input- and the output data bus. (If serial form is chosen, the corresponding SRB is bypassed).

4- and 8-point transforms can also be calculated. It is sufficient to compute a 16 point transform with the unused input data set to 0 (e.g. each second input word must be set to 0 for an 8-point transform). However, for 2-D transforms, the unused data of the intermediate frame as well must be set to 0. This is best done at the output of the SRB1 block.

8. ACHIEVED RESULTS

A 2μm poly-gate CMOS technology (VTI cmn20a) was used for the implementation of the chip. The core occupies approximately 4.5 x 5.0 = 22.5 mm^2 of silicon, resulting in a total chip area of 30.2 mm^2. The total number of transistors is about 72'000, subdivided into the following blocks:

• 1DDP	34'000 MOS
• SRB's	6'000 MOS
• IRM	22'000 MOS
• Control & Test Logic	10'000 MOS

This gives the relatively high density of 3200 MOS/mm^2. The full design has been made using a methodology and a cell library described in [10]. The expected maximal speed is far higher than the required 16.7 MHz clock frequency for a 512x512 point image transform at 32 images/second. All basic blocks should in fact work at approximately 50 MHz, with the only exception of the chip I/O circuitry.

9. CONCLUSIONS

An optimal architecture for implementing a DCT based on Distributed Arithmetic has been presented. It has been shown how a 2-D DCT can be realized using a 1-D DCT Processor, an Intermediate Result Memory, and two Shift Register Banks. Allowing maximal pipelining, the hardware has been reduced to a minimum and the sequencement has been simplified. Using a special extension of the architecture, all blocks can be tested separately, and new features are implicitly added to the chip.

Using the described techniques and a down-scaled CMOS technology, it is believed that a larger DCT, (e.g. 32x32 point), can be implemented. Another future extension could be the use of the 2-D DCT block as a subsystem on a larger and complete image coding chip.

ACKNOWLEDGEMENTS

This project was supported by the Swiss Foundation for Research in Microtechnology, under Grants FSRM 88/14 and FSRM 87/1.

REFERENCES

[1] N. Ahmed, T. Natarjan, K. R. Rao: "Discrete Cosine Transform", *IEEE Trans. Computers*, Vol. C-23, pp. 90-93, January 1974.

[2] S. Matsumura: "Discrete Cosine Transforms - Theory and LSI Implementation", *Linköping Studies in Science and Technology*, Thesis No. 43, Linköping University, Sweden, August 1985.

[3] I. Defilippis, U. Sjöström, M. Ansorge, F. Pellandini: "A 2-Dimensional 16 Point Discrete Cosine Transform Chip for Real Time Video Applications", Proc. Symposium GRETSI-89, Vol. 2, pp 813-816, Juan-Les-Pins, France, June 1989.

[4] S. Matsumura, B. Sikström, U. Sjöström, L. Wanhammar: "LSI Implementation of an 8 Point Discrete Cosine Transform", *International Conference on Computers, Systems & Signal Processing*, Bangalore, India, December 1984.

[5] H. Kitajima: "A Symmetric Discrete Cosine Transform", *IEEE Trans. Computers*, Vol. C-29, pp. 317-323, 1980.

[6] A. Peled, B. Liu: "A New Hardware Realization of Digital Filters", *IEEE Trans. Acoustic Speech and Signal Processing*, Vol. ASSP-22, No. 6, pp. 456-462, December 1974.

[7] M. Büttner and H-W. Schüssler: "On structures for the Implementation of the Distributed Arithmetic", Nachtrichtentechn. Z., Vol. 29, No. 6, pp. 472-477, June 1976.

[8] S. G. Smith and P. B. Denyer: "*Serial-Data Computation*", Kluwer Academic Publisher, Boston, USA, 1988 .

[9] U.Sjöström, I. Defilippis, M. Ansorge, F. Pellandini : "A Discrete Cosine Transform Chip for Real Time Video Applications", Proc. Int. Symposium on Circuits and Systems, New Orleans, Louisiana, May 1990.

[10] U.Sjöström, I. Defilippis, M. Ansorge, F. Pellandini : "A Methodology for ASIC Implementation of Digital Filters", Proc. Symposium GRETSI-89, Vol. 2, pp. 797-800, Juan-Les-Pins, France, June 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A SYSTOLIC ARRAY FOR MVDR BEAMFORMING BASED ON THE MODIFIED GRAM-SCHMIDT METHOD AND ITS APPLICATION TO RLS

Hideaki SAKAI

Division of Applied Systems Science, Faculty of Engineering, Kyoto University, Kyoto 606, Japan

A new systolic array for the minimum variance distortionless response (MVDR) beamforming in array processing based on the modified Gram-Schmidt method for the recursive least squares (RLS) problem is presented. The array is a counterpart of the one derived by McWhirter and Shepherd based on the Givens rotation method. Our array does not contain an undesirable operation appearing in the array of McWhirter and Shepherd. It is also shown that our MVDR array can be used with a slight modification for computation of the regression coefficients of the usual RLS problem on a sample by sample basis.

1. INTRODUCTION

In this paper we present a new systolic array for the minimum variance distortionless response (MVDR) beamforming problem in adaptive array processing, based on the modified Gram-Schmidt (MGS) orthogonalization method for the recursive least squares (RLS) problem. Recently, for this MVDR problem McWhirter and Shepherd [1] have derived a very efficient single systolic array based on the QR decomposition method by Givens rotations. The array is an extension of the one for the usual RLS problem first derived by Gentleman and Kung [2] and later adapted by McWhirter [3] and has two modes of operation.

On the other hand, the modified Gram-Schmidt (MGS) method for RLS has been used independently by several authors (Kawase, Sakai, and Tokumaru [4], Ling and Proakis [5], Kalson and Yao [6]). And Ling *et al.* [7] have shown that using the a priori error formulation McWhirter's array can be derived by the MGS method.

Using the MGS method with the a priori errors, we derive a counterpart of the MVDR array by McWhirter and Shepherd. The resulting array consists of a triangular part which is same with the array for RLS and a rectangular part. The cells in the rectangular part of the array of McWhirter and Shepherd contain an undesirable operation. That is, in updating a quantity there is a divison by a number less than 1 when the forgetting factor is less than 1. In our array, such an operation is removed so that we expect that our array has better numerical property.

It is well known that the triangular array for RLS only produces the residuals as an output and if we want to have the values of the regression coefficients, an extra linear array to perform the back substitution is needed [2]. But the back substitution is known to be numerically unstable and the whole structure is not appropriate for continuous adaptive operation, since new data cannot be processed by the array during the operation of back substitution. It is shown in this paper that our MVDR array can be used with a slight modification for computation of the regression coefficients on a sample by sample basis by setting constraint vectors appropriately.

2. THE MVDR BEAMFORMING PROBLEM

Here we give a brief review of the MVDR beamforming problem [1]. Let

$$x_n = (x_1(n), \cdots, x_p(n))^T \tag{1}$$

be the p-element (complex) vector where $x_i(n)$ denotes the value of the ith antenna element at time n. Also let

$$\boldsymbol{X}_i(n) = (x_i(1) \cdots x_i(n))^T \tag{2}$$

be the n-element (complex) vector and the $n \times p$ (complex) data matrix X_n is defined by

$$X_n = B_n(\boldsymbol{X}_1(n) \cdots \boldsymbol{X}_p(n)) \tag{3}$$

where B_n is the $n \times n$ diagonal matrix such that

$$B_n = \mathrm{diag}(\lambda^{\frac{n-1}{2}} \cdots \lambda^{\frac{1}{2}}\ 1) \tag{4}$$

with λ the forgetting factor in the usual RLS problem. Then the problem is to find the weight vector $w_n^{(k)}$ which minimizes

$$\| e_n^{(k)} \| = \| X_n w_n^{(k)} \| \tag{5}$$

subject to a linear equality constraint

$$c^{(k)T} w_n^{(k)} = \mu^{(k)} \tag{6}$$

and to compute the a posteriori residuals at time n

$$e_n^{(k)} = x_n^T w_n^{(k)} \tag{7}$$

efficiently for each $k = 1, \cdots, K$.

Using the MGS method ([4]–[7]) to form a new matrix $Q_n = (q_1(n) \cdots q_p(n))$ whose columns are orthogonal, from the column vectors of X_n, we have

$$Q_n = X_n R_n^{-1} \tag{8}$$

where R_n^{-1} is a $p \times p$ upper triangular matrix with the unit diagonal elements and $Q_n^H Q_n = \mathrm{diag}(s_{1,n} \cdots s_{p,n}) \equiv \Lambda_n$ where "H" denotes the Hermitian transpose. The details of the decomposition will be described in the next section. Then, as in [1] the desired solution is given by

$$w_n^{(k)} = \mu^{(k)} \frac{R_n^{-1} \Lambda_n^{-1} b_n^{(k)}}{\| \Lambda_n^{-1} b_n^{(k)} \|^2}, \quad b_n^{(k)} = R_n^{-H} c^{(k)} \tag{9}$$

$$e_n^{(k)} = \mu^{(k)} \frac{g_n^T \Lambda_n^{-1} b_n^{(k)}}{\| \Lambda_n^{-1} b_n^{(k)} \|^2}, \quad g_n = R_n^{-T} x_n. \tag{10}$$

It has been shown in [5]–[7] that the last element of $q_i(n)$ in (8) denoted by $u_{i,n}$ and $s_{i,n}$ can be efficiently computed by a triangular systolic array similar to that of [2] and [3] based on the Givens rotation method. Moreover, it will be seen that the ith element of g_n is equal to $u_{i,n}$. Thus to derive a systolic array for computing (10), it is only necessary to obtain a time-recursive relation for $b_n^{(k)}$ in (9). For notational simplicity, from now on, the superscript (k) in (9)–(10) is dropped.

3. THE MGS METHOD FOR RLS

It is well known that by the standard Gram-Schmidt method, $q_i(n)$ $(i = 1, \cdots, p)$ are generated by

$$\begin{aligned} q_1(n) &= B_n X_1(n) \\ q_i(n) &= B_n X_i(n) - \sum_{m=1}^{i-1} \frac{d_{i,n}^m}{s_{m,n}} q_m(n) \\ &\quad (i = 2, \cdots, p) \end{aligned} \tag{11}$$

with $s_{m,n} = q_m^H(n) q_m(n)$, $d_{i,n}^m = q_m^H(n) B_n X_i(n)$. Defining the intermediate vectors by

$$\begin{aligned} q_i^j(n) &= B_n X_i(n) - \sum_{m=1}^{j} \frac{d_{i,n}^m}{s_{m,n}} q_m(n) \\ &\quad (j = 1, \cdots, i-1) \end{aligned} \tag{12}$$

and writing the nth component of $q_i^j(n)$ as $u_{i,n}^j$, from (12) we have the fundamental recursion

$$u_{i,n}^j = u_{i,n}^{j-1} + \alpha_{i,n}^j \, u_{j,n} \quad (j = 1, \cdots, i-1) \tag{13}$$

with $u_{i,n}^0 = x_i(n)$ and $\alpha_{i,n}^j = -d_{i,n}^j / s_{j,n}$. Note that $q_j^{j-1}(n) = q_j(n)$ and the nth element of $q_j(n)$ is $u_{j,n}$. The time-update recursions for $s_{j,n}$ and $d_{i,n}^j$ are given by

$$s_{j,n} = \lambda s_{j,n-1} + |u_{j,n}|^2 / \alpha_{j,n} \tag{14}$$

$$d_{i,n}^j = \lambda d_{i,n-1}^j + u_{j,n}^* \, u_{i,n}^{j-1} / \alpha_{j,n} \tag{15}$$

where "$*$" denotes the complex conjugate and $\alpha_{j,n}$ is the so-called likelihood ratio variable and is order-updated by

$$\alpha_{j+1,n} = \alpha_{j,n} - \frac{|u_{j,n}|^2}{s_{j,n}} \tag{16}$$

with $\alpha_{1,n} = 1$ ([4]–[6]). The notations here are slightly different from those in [4]. The equations (13)–(16) consist of the Escalator algorithm in [4] and are put together as an efficient systolic array in [5] [6] which is similar to the original array in [2] and [3]. The equivalence between these two arrays has been shown by Ling *et al* [7]. They have introduced the a priori error formulation. Let the a priori (estimation) error of $x_i(n)$ using the coefficients based on the data $\{X_1(n-1), \cdots, X_j(n-1)\}$ be denoted by $\bar{u}_{i,n}^j$. It is well known [7] that the relation between the a posteriori error $u_{i,n}^j$ and the a priori error $\bar{u}_{i,n}^j$ is given by

$$u_{i,n}^{j-1} = \alpha_{j,n} \, \bar{u}_{i,n}^{j-1} \quad (j = 1, \cdots, i) \tag{17}$$

Combining (13)–(17), we have the basic equations

$$\bar{u}_{i,n}^j = \bar{u}_{i,n}^{j-1} + a_{i,n-1}^j \, \bar{u}_{j,n} \tag{18}$$

$$a_{i,n}^j = a_{i,n-1}^j - \frac{\alpha_{j,n}}{s_{j,n}} \bar{u}_{j,n}^* \, \bar{u}_{i,n}^j. \tag{19}$$

4. A NEW ALGORITHM

From (8), (11), and (13) it can be seen that

$$R_n = \begin{pmatrix} 1 & -a_{2,n}^1 & \cdots & -a_{p-1,n}^1 & -a_{p,n}^1 \\ & 1 & \cdots & -a_{p-1,n}^2 & -a_{p,n}^2 \\ & & \ddots & & \\ 0 & & & 1 & -a_{p,n}^{p-1} \\ & & & 0 & 1 \end{pmatrix} \tag{20}$$

From (9) we have

$$R_n^T b_n^* = R_{n-1}^T b_{n-1}^*. \tag{21}$$

Writing the ith element of b_n^* as $b_{i,n}$ and using (20), the following relation is obtained

$$b_{i,n} - \sum_{j=1}^{i-1} a_{i,n}^j b_{j,n} = b_{i,n-1} - \sum_{j=1}^{i-1} a_{i,n-1}^j b_{j,n-1}. \tag{22}$$

Let us define the following quantity

$$e_{i,n} = \sum_{j=1}^{i} \frac{u_{j,n}}{s_{j,n}} b_{j,n}^*. \tag{23}$$

Note that $g_n^T \Lambda_n^{-1} b_n$ in (10) is equal to $e_{p,n}$. Then we can show the following simple relations.

$$e_{i,n} = e_{i-1,n} + \frac{\alpha_{i,n} \bar{u}_{i,n}}{s_{i,n}} b_{i,n}^* \tag{24}$$

$$b_{i,n} = b_{i,n-1} - \bar{u}_{i,n} e_{i-1,n}^*. \tag{25}$$

The order-update recursion (24) is obvious from the definition (23). The time-update recursion (25) can be proved by induction about i but the proof is omitted here. Note that (24) and (25) are dual to (18) and (19).

When we estimate $x_p(n)$ by a linear combination of $x_1(n), \cdots, x_{p-1}(n)$ in the least-squares sense, the regression coefficient vector $\beta_n = (\beta_{1,n} \cdots \beta_{p-1,n}\, 1)^T$ lies in the last column of R_n^{-1}, that is,

$$\beta_n = R_n^{-1}(0 \cdots 0\, 1)^T, \tag{26}$$

so that $\beta_{i,n}^* = (R_n^{-H}(\overbrace{0 \cdots 1}^{i}\, 0 \cdots 0)^T)^T (0 \cdots 0\, 1)^T$. From (9) this means that $\beta_{i,n}^*$ is just the last element of $b_n^{(i)}$ (, *i.e.*, $(b_{p,n}^{(i)})^*$) with the constraint vector $c^{(i)} = (\overbrace{0 \cdots 1}^{i}\, 0 \cdots 0)^T$ in the MVDR problem. Also, from (10) and (26) we note that

$$u_{p,n} = x_n^T \beta_n = g_n^T (0 \cdots 0\, 1)^T \tag{27}$$

and similarly for other $u_{i,n}$.

5. SYSTOLIC ARRAY IMPLEMENTATION

The equations (14), (16), (17), (18), (19), (24) and (25) can be put into a single systolic array similar to that in [1]. The array consists of triangular and rectangular parts. From the right end cells of the triangular part, $\bar{u}_{j,n}$, $s_{j,n}$ and $\alpha_{j,n}$ are transfered to the rectangular part. To use (25), we need to properly initialize $b_{i,n}$. As in [1], first R_p is formed in the trangular array ($M = 1$). Then, from the top of the triangular part the constraint vectors $c^{(1)}, \cdots, c^{(K)}$ are input to produce the initial vectors $b_p^{(1)}, \cdots, b_p^{(K)}$ and during this phase $a_{i,p}^j$, $s_{j,p}$, $\alpha_{j,p}$ are frozen ($M = 0$). This is because from (27) $u_{i,p}$ appears from the ith cell in the right-end, so that by replacing x_n with $c^{(k)}$ ($k = 1, \cdots, K$), $b_{i,p}^{(k)}$ sequentially appear from that cell. After these vectors are transfered to the rectangular part, the operations of the triangular part return to its normal mode ($M = 1$) and the cells of the rectangular part compute (24) and (25) together with the computation of the denominator of (9) in a recursive manner by

$$\nu_{i,n}^{(k)} = \nu_{i-1,n}^{(k)} + \frac{1}{s_{i,n}} |b_{i,n}^{(k)}|^2. \tag{28}$$

Thus, $e_n^{(k)}$ is given by $\mu^{(k)} e_{p,n}^{(k)} / \nu_{p,n}^{(k)}$. The block diagram of the array is shown in Fig. 1.

The array in McWhirter and Shepherd [1] contains a numerically undesirable operation. The equation corresponding to (25) updates a quantity by dividing it by a number less than 1. Though they state that "this division is offeset by the fact that, the operation of all cells within the main triangular array involves multiplying the stored value by the same factor λ", in small word length computations, the numerical errors of both operations may not be exactly cancelled and may destory the normal operation of the array. Fig. 2 shows the simulation results done by T. Kagawa where $y_t = w_1 x_{1t} + w_2 x_{2t} + w_3 x_{3t} + \varepsilon_t$ and the signals are all white and the regression coefficients $w_1 \sim w_3$ are estimated by the two algorithms in 15 bits arithemetic. From these it may be concluded that in a word length less than the single precision, our array is much more numerically robust than the array in [1].

6. CONCLUSION

We have presented a new systolic array for MVDR beamforming based on the modified Gram-Schmidt method. The array has better numerical properties than the existing one. For the RLS problem due to the special forms of the constraint vectors, the annexed rectangular array in Fig. 1 becomes lower triangular so that the overall array look like a diamond. This two-dimensional array can be efficiently projected into a one-dimensional linear array. Its implementation on a transputer system is now being conducted.

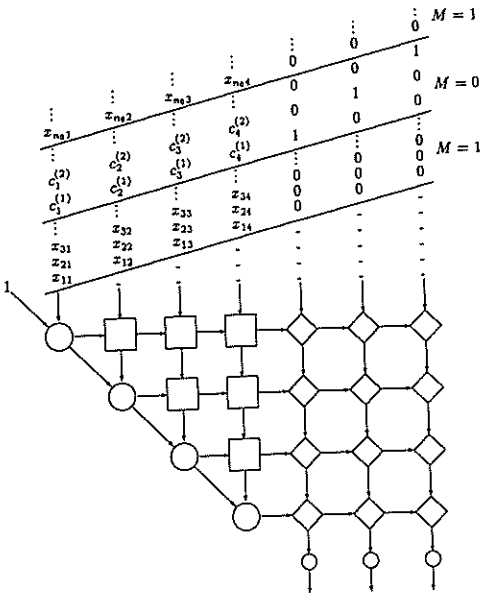


Fig.1 (a) Systolic array for MVDR beamforming.

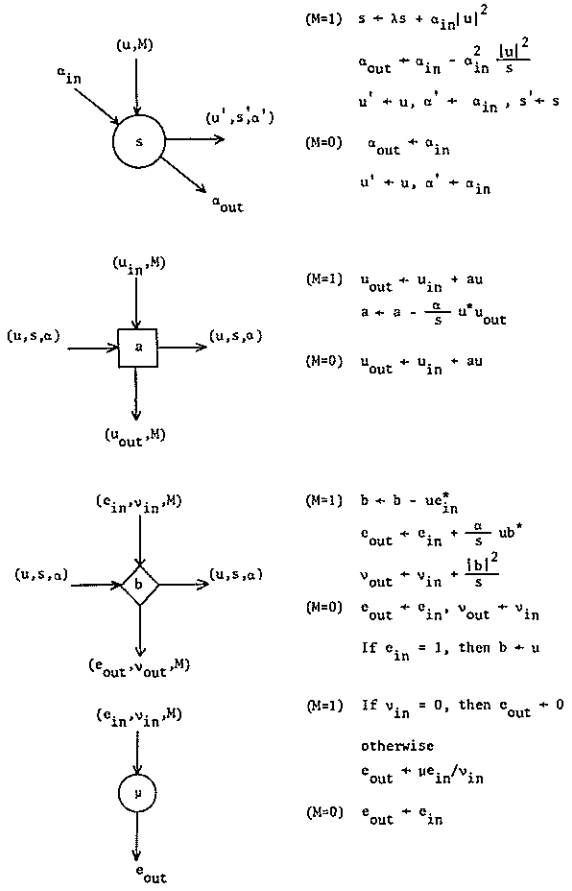
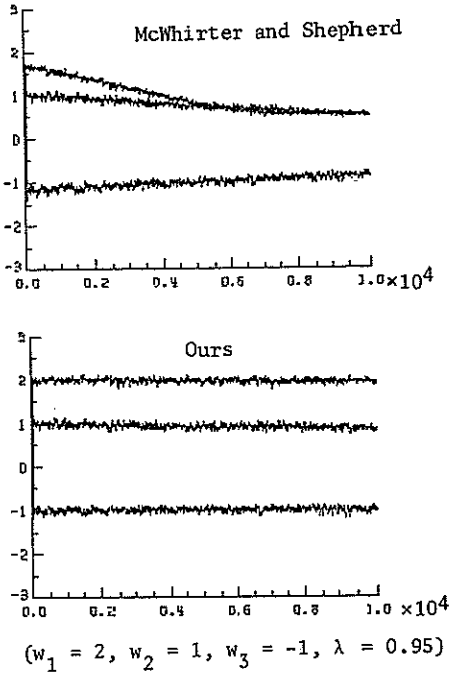


Fig.1 (b) Cell descriptions.



($w_1 = 2,\ w_2 = 1,\ w_3 = -1,\ \lambda = 0.95$)

Fig.2 Simulation results of the algorithm of McWhirter and Shepherd without square root and our algorithm in 15 bits arithemetic.

References

[1] J. G. McWhirter and T. J. Shepherd, "Systolic array processor for MVDR beamforming," *IEE Proc.*, vol. 136, pt. F, pp. 75–80, Apr. 1989.

[2] W. M. Gentleman and H. T. Kung, "Matrix triangularization by systolic arrays", *Proc. SPIE*, vol. 298, 1981.

[3] J. G. McWhirter, "Recursive least-squares minimization using a systolic array", *Proc. SPIE*, vol. 431, 1983.

[4] T. Kawase, H. Sakai, and H. Tokumaru, "Recursive least squares circular lattice and escalator estimation algorithms", *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP–31, pp. 228–231, Feb. 1983.

[5] F. Ling and J. G. Proakis, "A recursive modified Gram-Schmidt algorithm with applications to least squares estimation and adaptive filtering", *Proc. ISCAS*, Montreal, May 1984.

[6] S. Kalson and K. Yao, "Systolic array processing for order and time recursive generalized least-squares estimation", *Proc. SPIE*, vol. 564, 1985.

[7] F. Ling, D. Manolakis, and J. G. Proakis, "A flexible, numerically robust array processing algorithm and its relationship to the Givens transformation", *Proc. ICASSP*, Tokyo, pp. 2127–2130, 1986.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

WAVEFRONT ARRAY IMPLEMENTATION OF SCATTERING AND INVERSE SCATTERING SOLUTION METHODS

Yoshimi MONDEN[1], Masayasu NAGAMATSU[1] and Satoshi OKAMOTO[2]

[1]Naruto University of Education
Takashima, Naruto-shi, 772 Japan
[2]Niihama National College of Technology
Yakumocho, Niihama-shi, 792 Japan

In this paper, taking an example of scattering phenomenon as a typical of the physical real world, we will present a systematic approach for solving its direct and inverse scattering problems based on digital signal processing theory. We also advocate the importance of unifying these trans-disciplinary fields to form a 「new science」 which will play a crucial roll in the era of high technology.

1. INTRODUCTION

The main purpose of the computer is to model the (physical) real world. And the world is inherently concurrent. Its events happen in both time and space, sometimes in the same place one after the other in time (sequentially), but sometimes in different places at the same time (concurrently). However almost every commercially available computer is traditionally sequential. Consequently there has been a serious mismatch between the concurrent nature of the real world and the sequential nature of the digital computer.

However, with the advent of the VLSI technology and the increasing demand for high-speed and massive computing capabilities for signal processing and visual simulation, a number of VLSI-oriented array algorithms has been proposed. The most promising among these is WAP(Wavefront Array Processor) whose principle is to successively pipeline the computational wavefronts according to the data-flow principle [1]. This concept of the computational wavefronts should be compare with the electromagnetic wavefronts which obey Huygens's principle, and could be considered to be a concept systematically incorporating both sequential and concurrent natures of physical phenomena like wave propagation.

Despite these important applications, not many languages have provided explicitly for describing concurrent execution. One of the few exceptions is the programming language OCCAM. In fact, OCCAM together with its hardware support transputers is now the only commercially available programming language that provides facilities for both concurrent programming and execution and is also very suitable for programming wavefront-type array processing. Nevertheless it has not been fully and systematically exploited to describe a variety of physical phenomena from the view point of modeling the real physical world.

On the other hand, every physical phenomenon has two sides to it. One is associated with the direct problem and the other with the inverse problem. The direct problem is required to determine a system response for some given input and known physical functions, while the inverse problem is to determine unknown physical functions from its response to some probing input. Among a variety of the inverse problems the most well-known and well-established is the inverse scattering problem.

The extensive studies on the inverse scattering problem had been done associated with the quantum scattering theory, and culminated in the work of Gel'fand and Levitan in 1951, who showed that the potential could be determined from the spectral measure by solving a linear Fredholm equation, and the subsequent work of Marchenko, Krein and others established the integral equation method for solving the inverse spectral problem [2]. In 1970, with the advent of speech analysis, Gopinath and Sondhi [3] succeeded in formulating the vocal tract inverse problem as the inverse impulse response problem. Though the inverse scattering problem could be considered as a kind of system identification problem and might be interesting from the engineering point of view, it had seldom attracted attention of the engineers until recently because of its physical and mathematical difficulty.

The vocal tract inverse problem found its breakthrough in 1973 when Wakita discovered the important relationship between PARCOR and reflection coefficients under the special input and boundary conditions. This relationship essentially relied on the Levinson-Durbin algorithm relevant to computer implementation. Since then the Levinson-Durbin algorithm as well as its alternative, so-called Schur algorithm has been widely applied to speech analysis and has been considered as efficient and effective solution methods for the vocal tract inverse problem.

Recently Bruckstein & Kailath [4] categorized these solution methods into the layer-peeling and the layer-adjoining ones and present the general layer-adjoining method for the discrete inverse

scattering problem, covering discrete equivalents of Gel'fand-Levitan, Marchenko, and Krein' as its special cases. They also advocated the importance of using the discrete layer-peeling method as a direct and simple solution method also amenable to WA implementation. Thus the inverse scattering problem has enlarged its area including the inverse spectral and inverse impulse response problems and has come to be closely related with digital signal processing.

Therefore, in this paper, taking an example of scattering phenomenon as a typical of the physical real world, we will present a systematic approach for solving its direct and inverse problems based on digital signal processing. To be more specific, let us first deduce SFGs (Signal Flow Graphs) for scattering and transmission processes by applying the wave-transfer rule to the continuity conditions of both volume velocity and sound pressure and then transform the SFGs to their delay-transfered ones with the aid of delay-transfer rule. Secondly the SFGs are transformed to topologically equivalent DFGs (Data Flow Graphs) by an SFG/DFG equivalence transformation. Finally we will describe WAs (Wavefront Arrays) of scattering & transmission processes for the direct problem and the layer-peeling method for the inverse problem, and present their OCCAM implementation in terms of multi-transputer network.

2. DISCRETE NONUNIFORM ACOUSTIC-TUBE MODEL

This section is concerned with some preliminary results required for SFG & DFG description of scattering and transmission processes within a nonuniform acoustic tube. Let us first begin with deriving its discrete acoustic tube model.

Let a lossless nonuniform acoustic tube be approximated by a series of (M+1) uniform cylindrical sections of unit length L whose mth section has a constant local area S_m. The wave propagation within section m can be described by a set of equations

$$\rho\,\delta u_m(x,t)/\delta t = -S_m\,\delta p_m(x,t)/\delta x$$
$$S_m\,\delta p_m(x,t)/\delta t = -\rho c^2\,\delta u_m(x,t)/\delta x$$

where $u_m(x,t)$ and $p_m(x,t)$ denote volume velocity and sound pressure respectively in the mth section as a function of time t and distance x measured from its left end. The solution to these equations is expressed as a linear combination of right- and left-going waves in the following form

$$u_m(x,t) = u^+_m(x,t) - u^-_m(x,t)$$
$$p_m(x,t) = (\rho c/S_m)[u^+_m(x,t) + u^-_m(x,t)]$$

where $u^+_m(x,t) = u^+_m(x+ct)$. Then the whole wave propagation profile within the discrete nonuniform acoustic tube can be obtained by using the continuity conditions of volume velocity and sound pressure at the boundary of two contiguous sections. These conditions for section m-1 and section m are described as

$$u_{m-1}(L,t) = u_m(0,t)$$
$$p_{m-1}(L,t) = p_m(0,t).$$

Applying wave-transfer rule

$$u^+_m(0,t) = u^+_m(L,t+T/2)$$
$$u^-_m(0,t) = u^-_m(L,t-T/2), \quad T = 2L/c$$

to the continuity conditions and defining right- and left-going waves of section m as

$$u^\pm_m(n) = u^\pm_m(L,nT), \quad n=0,1/2,1,3/2, \ldots$$

we are led to the following equations describing the wave propagation profile between section m-1 and section m

$$u^+_m(n+1/2)+u^-_{m-1}(n) = u^+_{m-1}(n)+u^-_m(n-1/2)$$
$$u^+_m(n+1/2)-(S_m/S_{m-1})u^-_{m-1}(n) = (S_m/S_{m-1})u^+_{m-1}(n)-u^-_m(n-1/2),$$

from which directly follow a couple of sets of equations corresponding to different representations of the same wave evolution process. One is a causal scattering equation in the scattering representation relating incoming waves to outgoing ones and the other is a signal transfer equation in the transmission representation which transfers right- and left-going waves of section m-1 to those of section m. In the sequel, for the sake of brevity and later convenience, we are only concerned with the transmission representation and process.

The transmission equation and its associated diagram are shown as follows:

$$\begin{bmatrix} u^+_m(n) \\ u^-_m(n) \end{bmatrix} = \begin{bmatrix} d & 0 \\ 0 & d^{-1} \end{bmatrix} Q_m \begin{bmatrix} u^+_{m-1}(n) \\ u^-_{m-1}(n) \end{bmatrix}$$

$$Q_m = (1+\kappa_m)^{-1} \begin{bmatrix} 1 & -\kappa_m \\ -\kappa_m & 1 \end{bmatrix}$$

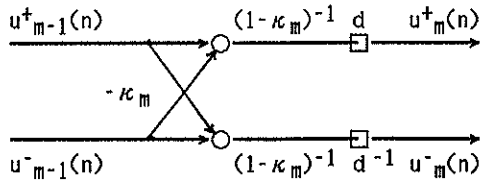


Fig.1

where κ_m is the so-called reflection coefficient between sections m-1 and m defined as

$$\kappa_m = (S_{m-1}-S_m)/(S_{m-1}+S_m).$$

These elementary diagrams will be combined in series to form SFGs and DFGs of the whole wave

evolution processes, i.e., scattering and transmission processes which play an important roll in tackling the discrete inverse scattering problem mentioned in Section 3 and 4.

3. SFG & DFG FOR SCATTERING AND TRANSMISSION PROCESSES

In this section, based on digital signal processing theory, we will successively convert a SFG of the transmission process to other types of SFGs and DFG that are crucially important in WAP implementation of the layer-peeling method.

Let us begin with a SFG of the transmission process, resulting from combining the elementary diagrams of Section 2. The first in SFG conversion process comes delay-transfer rule [1] that states: 「the timing relationships of input-input, input-output and output-output are not affected by advancing k time units on all the outbound edges and delaying k time units on the inbound edges for any given cut-set of a time-invariant SFG」. Applying the rule to the SFG of the transmission process, we obtain a delay-transfered SFG. Then redefining right- and left-going waves of volume velocity as

$$w^+_m(n) = u^+_m(n-m/2)/\nu_m$$
$$w^-_m(n) = u^-_m(n-m/2)/\nu_m$$

we have a normalized SFG of Figure 2, from which directly follows the associated DFG of Figure 3 by using SFG/DFG equivalence transformation [1]. This transformation is defined as follows: 「The computation of any SFG can be equivalently executed by a self-timed, data-driven machine with a topologically identical DFG. The number of initial tokens assigned on each DFG edge is equal to the number of delays on the corresponding SFG edge」. Note that for each delay in the SFG of Figure 2, there is an initial data token (denoted as a dot on a bar) assigned to the corresponding DFG edge of Figure 3. The initial data token distribution plays a vital role in assuring the correct sequencing in a data-driven computing network.

4. LAYER-PEELING METHOD FOR SOLVING THE DISCRETE INVERSE SCATTERING PROBLEM

The inverse scattering problem for the acoustic tube model is to determine its local areas(local reflection coefficients) given the input and response sequences, $w^+_0(n)$ and $w^-_0(n)$, under the assumption that the tube was initially quiescent. Bruckstein & Kailath [4] have shown that, among a variety of the inverse scattering solution methods, the layer-peeling one is most suitable and promising for WA implementation. Therefore, before embarking on its WAP implementation, we will give a brief summary of the layer peeling method along the same line as Bruckstein & Kailath's [4].

It should be first noted that, by the causality property and the delay structure of the acoustic tube, there are no right- and left-going waves at section m for at least m/2 time units i.e., $u^+_m(n)=u^-_m(n)=0$ for $n<m/2$. Consequently we have $w^+_m(n)=w^-_m(n)=0$ for $n<m$.

On the other hand, it hold $w^-_{m-1}(n)=\kappa_m w^+_{m-1}(n)$ for $m-1\le t<m$, since it will take at least m time units for any left-going incident wave $w^-_m(n)$ to appear as an input to section m-1. Therefore we can determine the mth reflection coefficient κ_m as $\kappa_m = w^-_{m-1}(m-1)/w^+_{m-1}(m-1)$. By the same argument we observe $\kappa_{m+1} = w^-_m(m)/w^+_m(m)$. However we do not have at hand $w^-_m(m)$ and $w^+_m(m)$ which are needed to determine κ_{m+1}. But, by using the transmission matrix $\underline{Q}_m$, which is completely specified once we now know κ_m, we can compute

$$\begin{bmatrix} w^+_m(n) \\ w^-_m(n) \end{bmatrix} = \underline{Q}_m \begin{bmatrix} w^+_{m-1}(n-1) \\ w^-_{m-1}(n) \end{bmatrix}$$

and now κ_{m+1} can be obtained. In effect, once we have determined the reflection coefficient of the mth section, we use its associated $\underline{Q}_m$ matrix to peel off the effect of section m-1 of the tube and put ourselves in the same position as before with a tube parametrized by $\{\kappa_{m+1}, \kappa_{m+2}, \ \dots \ \}$

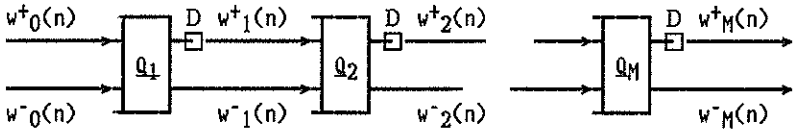


Fig. 2

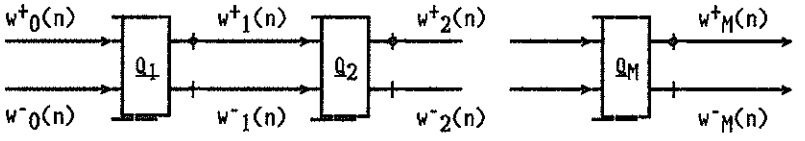


Fig. 3

In other words, using κ_m and the original scattering data we are able to produce the "artificial scattering data" for the tube extending over $[m+1,\infty)$.

We can continue in this way, successively determining a reflection coefficient and peeling off the associated section, to determine as much as we wish of the extent of the acoustic tube. In the next section, pipeline processing of this layer-peeling method will be presented in the programming language OCCAM.

5. WAP AND OCCAM IMPLEMENTATION OF THE LAYER-PEELING METHOD

As afore-mentioned, OCCAM together with its hardware support transputer provides both software and hardware facilities for concurrent programming and execution and is now going to be a standard programming language for describing pararell array processing. So, in this section, we will present WAP implementation of wave evolution process and its direct inversion procedure, so called layer-peeling method in terms of OCCAM and transputers. However, due to shortage of space, we will cite only an OCCAM program of the layer-peeling method with the aid of a DFG of the transmission process.

Figure 4 shows a DFG for the layer-peeling method, where L's denote elementary processes of the DFG and c's and w's denote its global channels and variables which should be declared in advance on the top of OCCAM program.

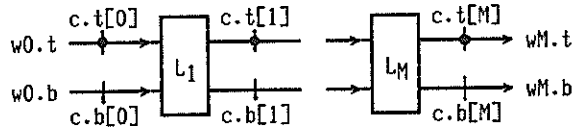


Fig. 4

A DFG for the elementary process L_m is structurally very similar to that of the transmission process which as well as its local channels and variables are depicted in Figure 5.

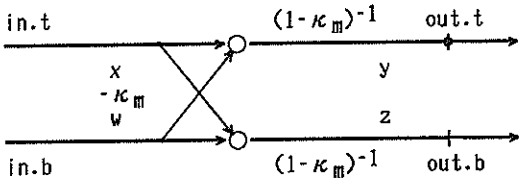


Fig. 5

With the aid of this DFG, OCCAM implementation of the layer-peeling method is described as follows:

```
...Layer Peeling Method
VAL M IS 20:
[M+1]CHAN OF REAL32 c.t, c.b:
[M+1]REAL32 k:
REAL32 w0.t, w0.b, wM.t, wM.b:

PROC peeler(VAL REAL32 kk, CHAN OF REAL32 in.t,
                in.b,out.t,out.b)
  REAL32 w,x,y,z:
  SEQ
    in.t ? x
    in.b ? w
    kk:=w/x
    y:=(x-(kk*w))/(1.0(REAL32)-kk)
    WHILE TRUE
      out.t ! y
      in.t ? x
      in.t ? w
      y:=(x-(kk*w))/(1.0(REAL32)-kk)
      z:=(w-(kk*x))/(1.0(REAL32)-kk)
      out.b ! z
:
PAR
  WHILE TRUE
    PAR
      c.t[0] ! w0.t
      c.b[0] ! w0.b
      c.t[M] ? wM.t
      c.b[M] ? wM.b
  PAR i=1 FOR M
    peeler(k[i],c.t[i-1],c.b[i-1],c.t[i],c.b[i])
```

where each elementary process begins with the computation of its reflection coefficient by the use of the first incoming inputs, followed by the usual elementary transmission procedure. Note that before a new value of y is computed the old value of y is output via channel out.t. In this way we are able to simulate a initial data-token.

6. CONCLUSIONS

In this paper, based on DSP theory we have presented a unified approach to solving the scattering and the inverse scattering problems from the view point of modelling the physical real world. Our results will strengthen the belief that the inverse problem is now going not only to enlarge its area covering the inverse spectral problem, the inverse impulse response problem, the profile inversion, and the image reconstruction, but also to combine with the trans-disciplinary research fields such as DSP, FEM etc. to form a 「new science」. It is only by means of unification of these trans-disciplinary fields that a goal of dimensional extension of the inverse scattering problem could be attained in the near future.

REFERENCES

[1] Kung, S.Y., VLSI Array Processors (Prentice-Hall, 1988).
[2] Chadan, K. and Sabatier, P.C., Inverse Problems in Quantum Scattering Theory, 2nd ed. (Springer-Verlag, 1989).
[3] Sondhi, M.M., A Survey of the Vocal Tract Inverse Problem, in: Santosa, F. et al. (eds.), Inverse Problems of Acoustic and Elastic Waves (SIAM, Philadelphia, 1984) pp. 1-19.
[4] Bruckstein, A.M. and Kailath, T., An Inverse Scattering Framework for Several Problems in Signal Processing, IEEE ASSP Magazine (1987).

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A MODIFIED GRIFFITHS-JIM ADAPTIVE BEAMFORMER BASED ON GIVENS ROTATION USING THE SYSTOLIC TRIARRAY

Kuang-Chih Huang and Shun-Hsyung Chang

Institute of Electrical Engineering,
National Sun Yat-Sen University,
Kaohsiung, 80499, Taiwan, R.O.C.

ABSTRACT

In this paper, we have applied the Givens-rotation-based systolic triarray structure to the Griffiths-Jim adaptive beamformer. In the context of advanced signal processing system, a proposed structure has been developed to improve the convergence of the closed-loop gradient descent algorithm. Computer simulations demonstrate the capability of the systolic triarray in rapid adaptation. The results show that a cancellation performance is obtained under stationary signal statistics when only a small number of data samples is processed.

1. INTRODUCTION

An adaptive array system can automatically adjust its directional response to null the interferences or jammers and thus enhances the reception of the desired signals. A number of adaptive algorithms have been proposed for array processing in recent years [1, 2, 7, 11, 12]. The 'close-loop algorithm,' in one extreme, such as LMS algorithm, essentially using stochastic gradient descent techniques to obtain their updating equations. The convergence or tracking behavior is dependent on the external environment and can be very slow if jamming is severe. The 'open loop' algorithms, in the other extreme, possess very fast convergence behavior and is independent of the noise environment. They, however, are much more complicated to implement.

In this paper, we investigate the systolic arrays for high performance of adaptive beamforming [1]. We aim at developing an array processor which is more efficient in deriving the residues of output. And, the systolic array architecture proposed by Kung [9] is particularly suitable for signal processing. We use the systolic triarray to improve the closed-loop adaptive control algorithm of the Griffiths-Jim adaptive beamformer [7]. Computer simulations demonstrate the capability of the systolic triarray in rapid adaptation. The results show that a cancellation performance is obtained under stationary signal statistics when only a small number of data samples is processed.

2. THE MODIFIED STRUCTURE

Illustrated in Fig. 1 is a simplified block diagram of an adaptive antenna employing a 'direct solution' control processor. The weight vectors can be obtained through the method known as QR decomposition [10]. Minimization of the norm of the residual vector is the derivation of straightforward least-squares.

The orthogonal triangularization process may be carried out by using various techniques such as Householder transformations [5, 6] or Givens rotations [4]. The triangularization is performed by a process under the Givens rotation method whereby coefficients of the lower half of the data matrix are successively eliminated by manipulating pairs of row vectors.

An important feature of the QR method using Givens rotations is the ability to apply the algorithm in a recursive form whereby the triangular set of equations can be updated on a sample-by-sample basis as each new row of data enters the computation [3]. In the recursive problem, a sequence of elementary transformations are applied to the new data vector and the rows of the existing triangular system in order to eliminate successively the leading coefficients of the former matrix.

The main objective, in many least-squares methods, is to compute the least-squares residual since the corresponding weight vector is not of direct interest. In his previous research, McWhirter [10] has described a modified version of the QR recursive least-squares algorithm in which the least-squares residual is quite easily produced at each stage of the recursive process. The modified algorithm is much more robust because it does not involve the solution of a linear system equation which could be ill-conditioned. In addition, since the back-substitution and a separate beamforming network are eliminated, its results in significant simplification of the complexity of

the subsequent circuit implementation.

Shown in Fig. 2 is a block diagram of the Griffiths-Jim beamformer [7]. It is another realization of the Frost adaptive beamformer [2] and achieves the same constraint as Frost's [12].

In Fig. 3, we perform the proposed structure which combines Griffiths-Jim algorithm and the systolic triarray. We use the systolic triarray to improve the closed-loop adaptive control algorithm of the Griffiths-Jim adaptive beamformer. Shown in Fig. 3, the systolic triarray comprises three distinct sections: the basic triangular array labelled ABC, the top row of cells labelled DE, and a final processing cell labelled F. The array may be controlled by a clock. In every cycle, each cell receives its data from the directions shown, performs its computation, and delivers appropriate values to neighboring cells on the subsequent cycle. Each cell within the basic triangular array stores the corresponding element of the recursively evolved triangular matrix which is initialized to zero at the outset of the least-squares calculation and then updated every clock cycle. Cells in the top row store one element of the evolved vector which is also initialized to zero and updated every clock cycle.

In McWhirter's research [11], it was shown that a systolic triarray could be used in an efficient recursive manner to evaluate the sequence of a posteriori least squares residuals, which is shown as

$$e(t_n) = \gamma(n)\alpha(n), \qquad (1)$$

where $\alpha(n)$ is the value produced by the internal cell E at time t_n, and $\gamma(n)$ is the corresponding value produced by the boundary cell C. The product in Eq (1) is computed by the final cell F in the Figure 3. And, $\gamma(n)$ can be expressed as

$$\gamma(n) = \prod_{i=1}^{n} c_i(n) , \qquad (2)$$

i.e., $\gamma(n)$ is the product of all cosine parameters associated to the sequence of Givens rotation used to eliminate new data vector.

3. COMPUTER SIMULATIONS

In this paper, all the programs are written in FORTRAN LANGUAGE and simulated on the VAX-11/8300 minicomputer.

The environment is considered such that it contains a target signal and a jammer that are noncoherent with each other. The frequencies are normalized so that $f = 1$ corresponds to Nyquist rate. In this case, the power of the target signal is normalized to 1 watt and the background noise is white noise with zero mean. The desired signal is incident from the broadside. We choose the interelement spacing of the array $d = \lambda/2$, where λ is the wavelength. The environmental parameters are as follows:
The sensor number of the array = 11,
The incident angle of the target signal = 0°
The incident angle of the jammer signal = 27°
SNR = -3 dB; JNR= 10 dB,
The frequency of the target signal = 1/8,
The frequency of the jammer signal = 1/4.

In Fig. 4, which shows the interference cancellation on the arrival angle of 27 degrees, illustrated is a comparison between the Adaptive Beam Pattern (ABP) of nonadaption and the ABP after 5000 adaption cycles of the Griffiths-Jim adaptive beamformer applying the LMS algorithm [12]. The beamformer exhibits a reasonably effective nulling capability.

In Fig. 5, which shows the interference cancellation on the arrival angle of 27 degrees, we compare the ABP of nonadaptive with the ABP of the proposed adaptive beamformer applying systolic triarray under the recursive QR Decomposition Least-Squares algorithm. It is clearly displayed that no more than 250 adaption cycles is needed to null the interference.

The time series of the desired signal and the residual output are illustrated in Fig. 6(a) and (b), both of which are based on the conditions of Fig. 5. The conversion factor $\gamma^2(n)$ of the systolic array, defined by Haykin [8] is derived under these conditions, as shown in Fig. 7.

Therefore, it is obvious that the open loop techniques applied in our structure achieves a much faster convergence in computation than the closed-loop gradient descent algorithm.

4. CONCLUSIONS

In this paper, we have applied the Givens-rotation-based systolic triarray structure to the Griffiths-Jim adaptive beamformer. The improved structure is advantageous in that it requires only limited input data to describe accurately its external environment. In the context of advanced signal processing applications, the use of regularly structured processing is believed to be the one of the most practical approach to obtain real-time performance.

REFERENCES

[1] Chang, Shun-Hsyung and Huang, Kuang-Chih, "A comparison between the multiple beamforming network and the modified Griffiths-Jim structure in adaptive beamforming using a systolic array," Proceedings of the

National Science Council (Part A), Taiwan, R.O.C., Vol. 14, to be published in March 1990.
[2] Frost, O. L., "An algorithm for linearly constrained adaptive array processing," Proc. IEEE, Vol. 60, pp. 926-935, 1972.
[3] Gentleman, W. M., Kung, H. T, "Matrix triangularization by systolic arrays," Proc. SPIE, Vol. 298, 1981.
[4] Givens, W.,"Computation of plane unitary rotations transforming a general matrix to triangular form," J. Soc. Ind. Appl. Math., Vol. 6, pp. 26-50, 1958.
[5] Golub, G. H., "Numerical methods for solving Linear least-squares problems," Num. Math., Vol. 7, pp. 206-216, 1965.
[6] Golub, G. H., and Van Loan, C. F., 1983, *Matrix Computations* (Maryland: John Hopkin University Press).
[7] Griffiths, L. J. and Jim, C. W., "An alternative approach to linearly constrained adaptive beamforming," IEEE Trans. AP, Vol. 30, pp. 27-34, Jan. 1982.
[8] Haykin, S., 1986, *Adaptive Filter Theory*, (New Jersey: Prentice-Hall).
[9] Kung, H. T., "Why systolic architectures?", Computer, Vol. 15, pp. 37-46, 1982.
[10] McWhirter, J. G., "Recursive least-squares minimization using a systolic array," Proc. SPIE, Vol. 431, p. 2983, 1983.
[11] Ward, C.R., Hargrave, P.J., and McWhirter, J.G., "A novel algorithm and architecture for adaptive digital beamforming," IEEE Trans. on AP, Vol. 34, pp. 338-346, March 1986.
[12] Widrow, B, and Stearns, S., 1985, *Adaptive Signal Processing*, (New Jersey: Prentice-Hall).

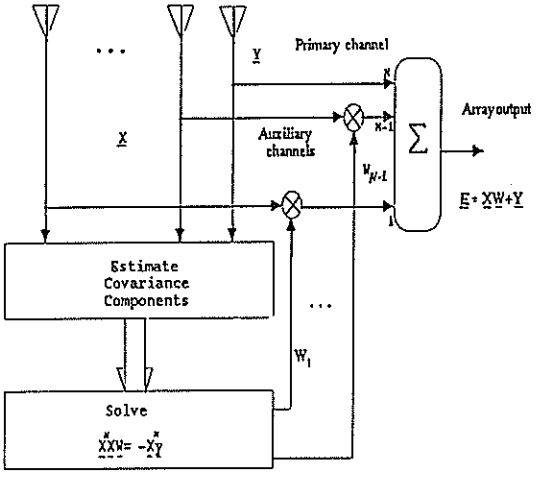


Fig. 1. A simplified block diagram of an adaptive antenna system.

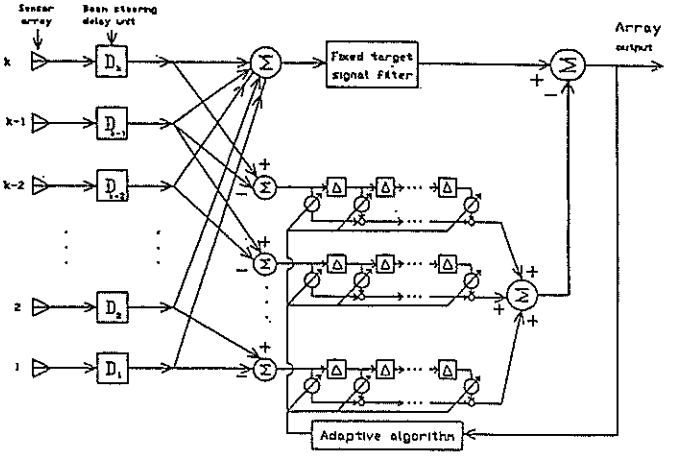


Fig. 2. Griffiths-Jim version of the Frost adaptive beamformer.

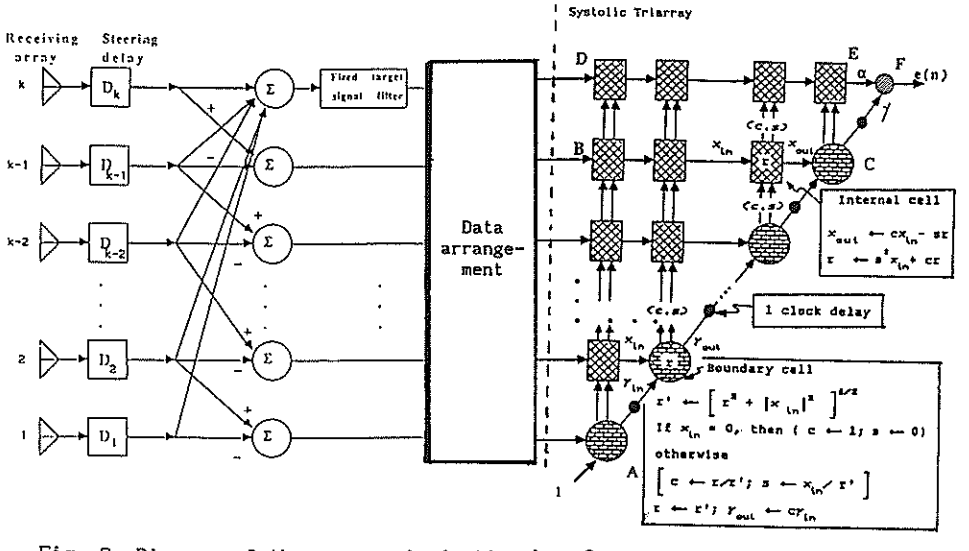


Fig. 3. Diagram of the proposed adaptive beamformer.

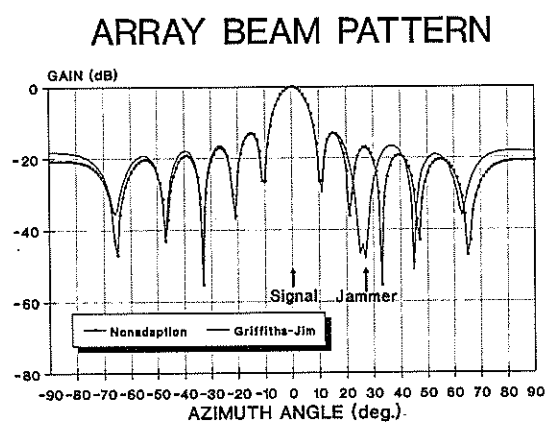


Fig. 4. A comparison between the ABP of nonadaption and the ABP after 5000 adaption cycles of the Griffiths-Jim adaptive beamformer applying the LMS algorithm.

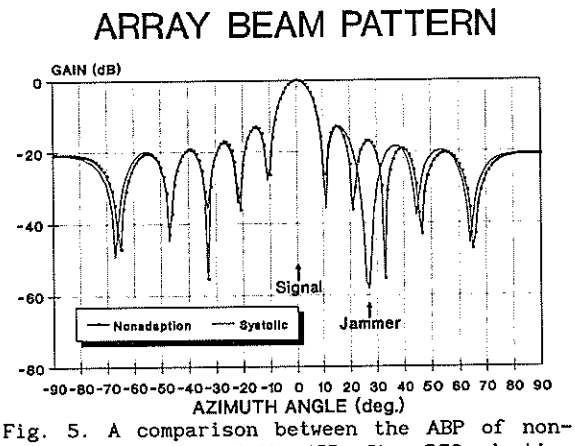


Fig. 5. A comparison between the ABP of non-adaption and the ABP after 250 adaption cycles of the proposed adaptive beam-former applying systolic triarray under the recursive QR Decomposition Least-Squares algorithm.

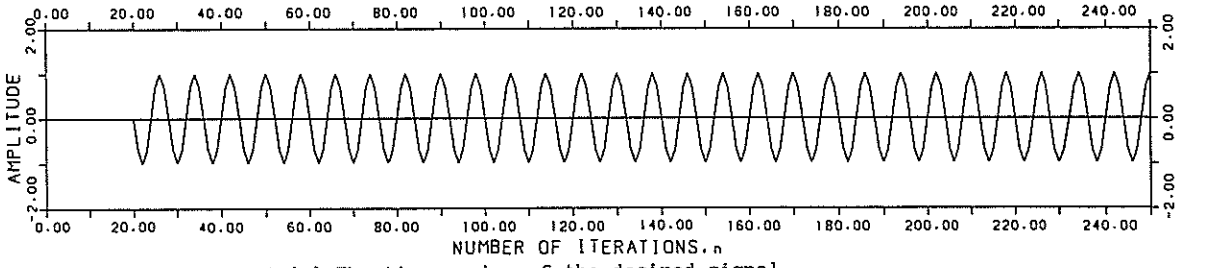


Fig. 6.(a) The time series of the desired signal.

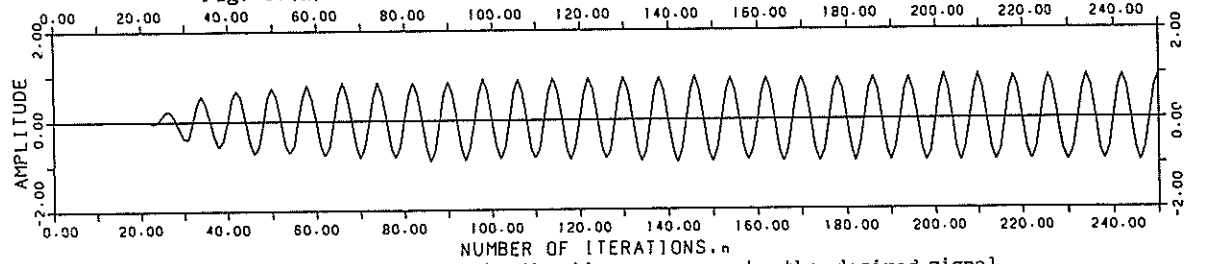


Fig. 6.(b) The residual output: the time response to the desired signal.

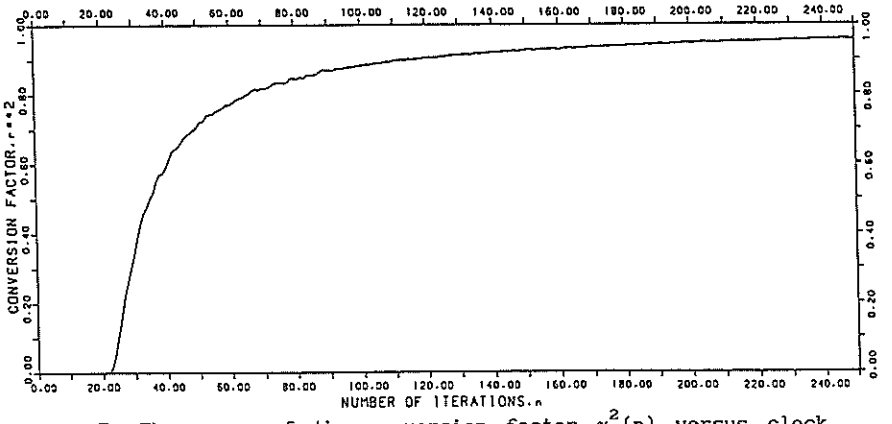


Fig. 7. The curve of the conversion factor $\gamma^2(n)$ versus clock cycle (n) of the proposed adaptive beamformer using the recursive QR Decomposition Least-Squares algorithm.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

A SYSTOLIC ARRAY FOR QR DECOMPOSITION USING PIPELINED FUNCTIONAL UNITS

Miguel Valero-García, Núria Torralba, Juan J. Navarro and José. M. Llabería

Dept. Arquitectura de Computadors, Univ. Politècnica Catalunya, Pau Gargallo 5, 08028 Barcelona Spain

This paper describes a linear systolic architecture for QR decomposition. The processing elements of this architecture are based on pipelined functional units. A second goal of this paper is to ilustrate our methodology for adapting systolic algorithms to the hardware that will execute them.

1. INTRODUCTION

The QR decomposition is the first step in many engineering applications, as for example, SONAR or RADAR beamforming. However, this operation is computationally expensive, and usually represents the bottleneck of the application. Systolic Array Processors (SAPs) [Kung78] are a good architectural solution to implement computationally expensive operations using a high degree of parallelism at a low cost.

In this paper we describe a SAP for QR decomposition. Our intention is not only to present an efficient architecture but also to propose a systematic method to efficiently implement a given Systolic Algorithm (SA) in hardware.

The main features of the proposed SAP are: (a) it is a one dimensional (1D) architecture, (b) it can perform the QR decomposition of a matrix with any size, and (c) operations of the SA are executed using Pipelined Functional Units (PFUs).

The SA executed by our SAP was previously proposed in [Torr88]. It is a problem size independent 1D SA that was obtained from a 2D SA proposed by Heller and Ipsen [Hell82]. In section 2 we briefly review the main features of these SAs.

PFUs allow to improve the throughput of an architecture because a PFU can initiate an operation before the completion of the previous ones. However, in a SA, every cell initiates one operation only when the previous one has completed. In section 3 we describe the proposed method to transform a SA in order to efficiently exploit the PFUs used to execute it.

Finally, section 4 discusses the implementation of the SAP.

2. SA DESIGN

In this section we briefly describe the SA executed by the proposed SAP. Figure 1 depicts the procedure to derive this SA. The original SA (figure 1.a) is due to Heller and Ipsen. This is a 2D SA that uses Givens rotations. In figure 1.a, circles represent cells which compute rotations (OP_a in figure 1.d) and squares represent cells which updates the matrix with the actual rotation factors (OP_b in figure 1.d).

This 2D problem size dependent SA is mapped into a 1D problem size dependent SA via cut&pile [Nava87]. Figure 1.b shows the resulting SA. This SA is then transformed into a 1D SA with an arbitrary number of cells, say w. This transformation is also done via cut&pile. The resulting problem size independent SA is shown in figure 1.c. The ordering of operations is done by a temporal mapping based on DBT techniques [Nava87].

Figure 1.d specifies the operations performed by the cells. All the cells must perform OP_b. Cell 1 must also perform OP_a in some cycles. Signal c_i determines the type of operation to be initiated by cell 1 in every cycle. In this SA it is assumed that every operation takes one cycle. This fact is modelled by associating a delay of one cycle to every link between cells. These delays are represented in figure 1.c by black rectangles. Finally, every cell of the SA performs a valid operation every 2 cycles. For this reason it is said that the SA is 2_slow [Leis81].

The main intention of this paper is to demonstrate how the SA shown in figure 1.c can be efficiently executed using PFUs. For this reason, we have not described the SA in depth. A more detailed description can be found in [Torr88], including a specification of the I/O patterns of the SA.

3. SA TRANSFORMATION

In this section we describe the hardware to be used to implement the SA presented in the previous section. Moreover, we propose a procedure to adapt the SA to the selected hardware.

3.1. Hardware to be used

We supose that, in order to implement the operations of the SA shown in figure 1.c, we use pipelined multipliers and adders like those offered by Weitek [Weit83]. Specifically, the square root required to execute OP_a can be performed by multiplications and additions, as the following expression shows:

$$R1 = 0.5*R0*(3.0 - A*R0*R0)$$

$$\frac{1}{\sqrt{A}} \approx 0.5*R1*(3.0 - A*R1*R1)$$

Work supported by the Ministry of Education of SPAIN (CAICYT PA85-0314 and CICYT TIC 299/89)

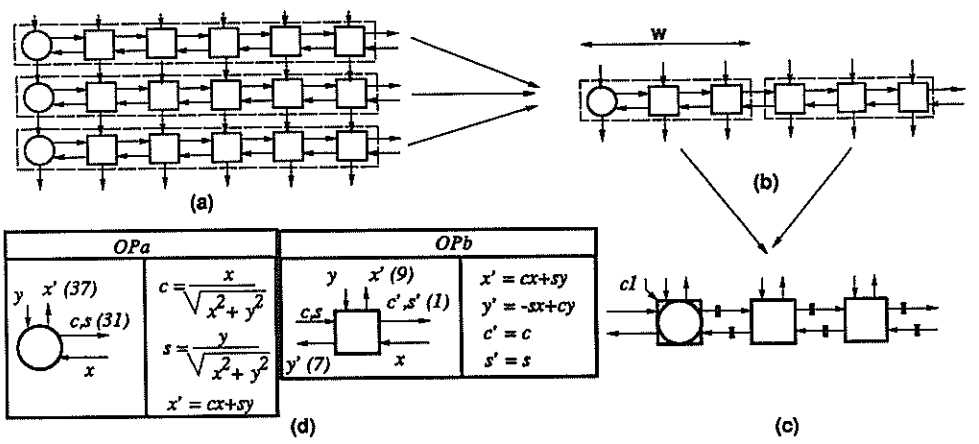


Figure 1: Design of a problem size independent 1D SA for QR decomposition. (a) Original SA due to Heller and Ipsen. (b) 1D SA obtained via cut&pile. (c) Problem size independent SA obtained also via cut&pile. (d) Operations performed by the cells.

R0 is an approximation of $1/\sqrt{A}$ obtained by indexing a table with some bits of *A*. The accuracy of the initial reciprocal *R0* is extended from only about 8 bits to about 24 bits.

Figure 2 shows two reservation tables which describe how to implement OP_a and OP_b using only one adder and one multiplier, both pipelined into three stages. In these reservation tables, rows represent the activity of each stage of the adder and multiplier in every cycle. A mark in position *(row, column)* indicates that, in order to perform the required operation, it is necesary to use the stage associated with *row* in cycle associated with *column*. For the sake of simplicity, we assume that the time required to access the table of approximations can be neglected.

It is easy to design a PFU able to perform OP_a and OP_b as indicated in the reservation tables. The inputs to that PFU are signals *Y*, *C*, *S*, and *X*. The outputs are signals *X'*, *Y'*, C'_a, S'_a, C'_b and S'_b. Values *c* and *s*, produced by OP_a, are obtained through C'_a and S'_a. Values *c* and *s* produced by OP_b are obtained through C'_b and S'_b. In figure 1.d it has been indicated, in parentheses, next to each value produced in every operation, the number of cycles required to compute this value in the PFU. Note that, when the PFU performs OP_b, the values obtained through C'_b and S'_b are equal to the values received through *C* and *S*. However, in order to avoid data broadcast, we associate a delay of one cycle with the computation of these values.

3.2. Adapting the SA to the hardware

The SA shown in figure 1.c cannot be directly executed using PFUs like that described previously. It is necessary to transform this SA in order to adapt it to the hardware. To do that, we propose a procedure based on a temporal transformation and a spatial transformation of the original SA.

Temporal transformation

A temporal transformation permits to modify the cycle in which every cell of the original SA, say *A*, performs its operations. This transformation will be used to modify *A* in such a way that the dependences imposed by the PFUs are preserved. The proposed temporal transformation is a combination of two transformation: slowdown and retiming [Leis81].

Slowdown consists in multiplying every delay in *A* by a constant *c*. This constant is the parameter of the transformation. The result is a new SA, equivalent to *A*. The slow of the new SA is *c* times greater. So, the number of cycles required to perform the whole computation has been multiplied by *c*. Moreover, it is necessary to modify the input data sequences from the outside. Specifically, if every cell of *A* receives a data item from the outside every *k* cycles, each cell of the new SA receives a data item every *ck* cycles.

Retiming is based on the fact that it is possible to obtain an equivalent SA by substracting a delay of d_i cycles to every input link to cell *i* and adding this delay to every output link from this cell. Retiming is parametrized by the following tuple:

$$rt = (d_1, \ldots, d_i, \ldots, d_n)$$

where *n* is the number of cell of the original SA.

Note that retiming does not modify the slow of the SA and that both, slowdown and retiming, do not modify either the number of cells or the interconnection topology.

As a conclusion, slowdown permits to introduce new delays in the SA and retiming permits to redistribute these delays among the links. The new delays will be used to model the time required by every cell to perform its operations using the available hardware.

In our example, it is necessary to introduce a delay of, at least, 31 cycles in the link from cell 1 to cell 2. Moreover, it is necessary a delay of, at least, 7 cycles in the link from cell *i* to cell *i-1*. The required temporal transformation can be done by applying slowdown with parameter *c = 19*, and retiming with parameter:

$$rt = (0, -12, 6, 24, 42, 60, \ldots)$$

The obtained SA, say *A'*, is that shown in figure 3.a. Note that only 7 of the 37 delays in the link from cell *i* to cell *i-1* *(i>3)* are used to model the time required by the PFU to

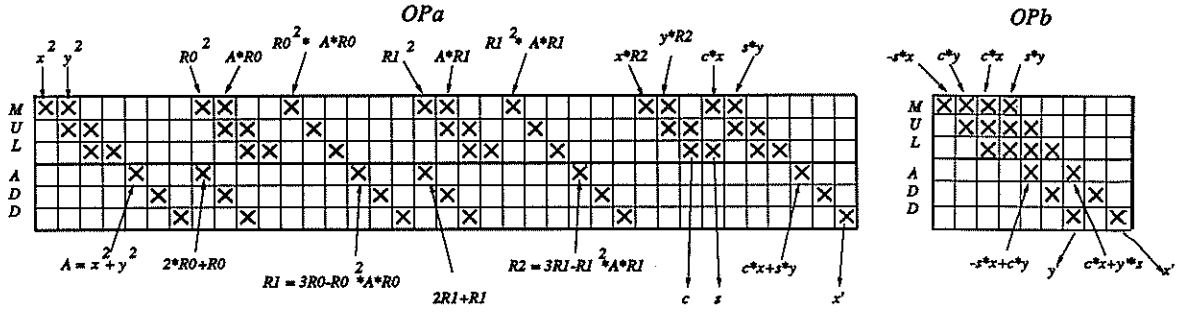


Figure 2: Reservation tables to implement OP_a and OP_b using only a multiplier and an adder, both pipelined into three stages.

produce the value to be sent through that link. The remaining 30 extra delays are required to preserve the correctness of the SA. In figure 3.a it has been indicate in parentheses, the number of extra delays associated with each link.

The case for cell 1 is slightly different. When this cell performs OP_a, the whole delay (31 cycles) model the time required to perform the computation. However, when the cell performs OP_b, only 1 delay models the computation time, and the rest are extra delays.

Note that, after the temporal transformation, cell 1 performs the first valid operation in cycle 0, cell 2 in cycle 31, cell 3 in cycle 32, etc. Moreover, due to the applied slowdown, the slow of A' is 38. Therefore, each cell performs only one valid operation every 38 cycles. On the other hand, the SA does not exploit the throughput offered by the PFU because every operation is initiated when the previous one has completed.

Spatial transformation

The transformation procedure is completed by applying a spatial transformation. The goal of this transformation is to increase the utilization of the PFUs.

A spatial transformation permits to redistribute the operations of the SA among the cells. However, each operation is still performed in the same cycle. The result of the transformation is an equivalent SA, say A^*, that spends the same number of cycles to perform the whole computation but using a lesser number of cells.

In this paper we propose the use of coalescing as spatial transformation [Leis81]. This transformation maps a set of adjacent cells of A' to every cell of A^*. The transformation can be parametrized by the tuple:

$$cl = (p_1, \ldots, p_i, \ldots, p_q)$$

In this tuple, p_i indicates the number of adjacent cells of A' assigned to cell i of A^*.

In order to apply coalescing in our example, we have to determine how many adjacent cells of the SA shown in figure 3.a can be implemented using a single PFU. To do that, we must take into account two important points: (a) every cell repeats its operations every 38 cycles, and (b) when an operation is initiated the PFU is used as described by the reservation tables in figure 2.

The SA in figure 3.a shows that cell 2 initiates one of its operations (OP_b) 31 cycles after cell 1 does. The reservation tables for these operations show that OP_b cannot be initiated 31 cycles after OP_a because the conflict in the use of the multiplier. This conflict can be avoided by delaying 4 cycles the last multiplication and addition in OP_a. In this way, the multiplier and adder become idle for a period long enough to perform OP_b. Note that this modification only affects the time required to compute the value x' produced by OP_a.

The next step is to determine if it is possible to execute cell 3 together with cells 1 and 2, in the same PFU. Cell 3 initiates its operations (OP_b) 1 cycle after cell 2 and 32 cycles after cell 1. It is easy to see that the PFU cannot initiate two OP_b in two consecutive cycles because the conflict in the use of the multiplier. In this case, the conflict can be avoided by delaying the initiation of operations in cell 3. Specifically, if cell 3 initiates its operations 19 cycles after cell 2 then cells 1, 2 and 3 can be executed in the same PFU. This modification can be done by applying retiming to the SA shown in figure 3.a. The parameter for this transformation is:

$$rt = (0, 0, -18, -18, -18, \ldots.)$$

In this way, a delay of 19 cycles is introduced in the link from cell 2 to cell 3.

Finally, cell 4 can also be executed in the same PFU if this cell initiates its operations 10 cycles after cell 3. This modification is also done by retiming with parameter:

$$rt = (0, 0, 0, -9, -9, -9, \ldots.)$$

The new SA A', obtained through the last two transformations is that shown in figure 3.b.

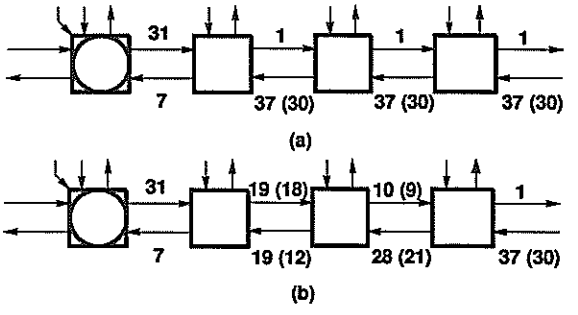


Figure 3: (a) SA A' obtained via slowdown and retiming. (b) New SA A' obtained via slowdown and retiming. Now, cells 1, 2, 3 and 4 can be implemented using a single PFU.

The described procedure should be repeated for the rest of cells of A'. It can be seen that it is possible to group cells from 5 to 13, cells form 14 to 22 and so on. These groupings require further transformation of the SA, always done by retiming. At the end of this process, it is possible to determine the parameter for coalescing. In our example, the parameter is:

$$cl = (4, 9, 9, 9, \ldots)$$

4. SAP IMPLEMENTATION

Once the SA has been adapted to the PFUs, we have to define the final structure and control for each of the PEs of the SAP. Figure 4 shows the structure and control of PE1, which executes cells 1, 2, 3 and 4 of the original SA. This PE has been obtained from the PFU and including some feedback links. Specifically, each feedback link corresponds to one of the links that in A' communicates two cells that now are executed in the same PFU. The delay associated with the feedback link is equal to the extra delay associated with the corresponding link in A'. As an example, the feedback link from output Y' to input 1 of multiplexor M_r corresponds to the link from cell 4 to cell 3 of A'. As we saw before, the link from cell 1 to cell 2 is an special one because the associated extra delay depends on the operation performed in cell 1. The feedback link corresponding to this link of A' is implemented by using a multiplexor $M_{r,0}$ which permits to apply the appropriate extra delay to the link, depending on the operation performed by the PFU.

Figure 4 also shows the control scheme for the PE 1. The control is based on a module 38 counter. This counter is used to determine both the type of operation to be initiated in every cycle and the source of its operands. When the operation to be initiated is one of those executed in cell 1 of A', signal c'_1 determines the type of this operation (OP_a or OP_b). Signal c'_1 has been obtained from signal c_1 (figure 1.c), taking into account the temporal transformations applied to the original SA.

Note that it is possible to simplify the design of PE 1 by sharing delays among feedback links when possible.

The design of PE 2, which executes cells form 5 to 13, is very simple and is not described here. The rest of PEs are equal to PE2. Therefore, using a SAP with P EPs it is possible to execute a SA with $w = 4 + (P-1)*9$ cells. Value w is used to determine the partitioning of a problem with size greater than w.

5. CONCLUSIONS

In this paper we describe an efficient implementation of a 1D SA for the QR decomposition of a matrix. We have paid special attention to the procedure for adapting the original SA to the hardware used to execute it. This hardware is based on PFUs.

The proposed procedure is automatic. In [Vale89] we propose a formal model for the used transformations and we give some algorithms which obtain the parameter for the transformations to be applied in every step. This method can be used as a basis for a software tool oriented to the design of specific purpose processors.

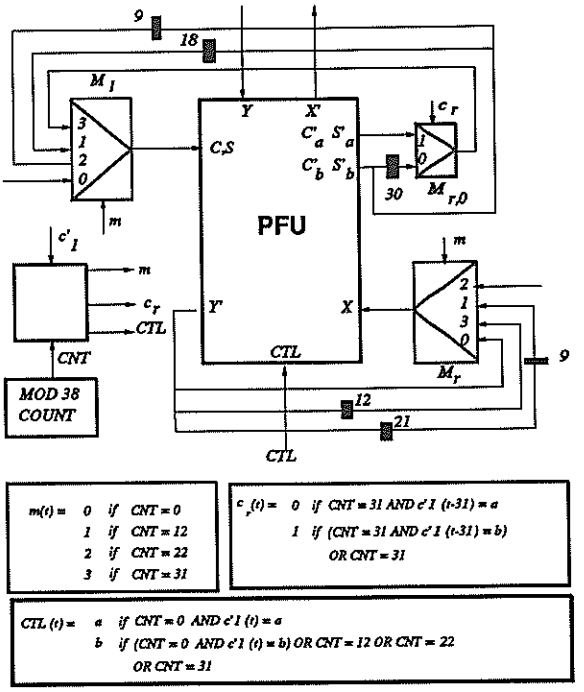


Figure 4: Internal structure and control for PE 1 which executes cells 1, 2, 3 and 4 of the original SA.

However, there are some problems that have not been mentioned in this paper but that, in any case, should be undertaken when designing a SAP. The most important problem is the high communication bandwidth between the host computer and the SAP, required to feed the PEs at maximum rate. A first solution to this problem is to include a local memory in each PE. The host computer has to load this memory before the SAP starts working. In that case, the PE must also include an addressing mechanism responsible for sending data to the PFU in the order stablished by the temporal transformations.

6. REFERENCES

[Hell82] D.H. Heller and I.C.F. Ipsen, "Systolic Networks for Orthogonal Equivalence Transformations and their Application", Conference on Advanced Research in VLSI, M.I.T. 1982, pp. 113-122.

[Kung78] H.T. Kung and C.E. Leiserson, "Systolic Arrays (for VLSI)", Sparse Matrix Symp, SIAM, 1978, pp. 256-282.

[Leis81] C.E. Leiserson and J.B. Saxe, "Optimizing Synchronous Systems", Proc. 22nd Annual Symp. on Foundations of Computer Science, October 1981, pp. 23-36.

[Nava87] J.J. Navarro, J.M. Llabería and M. Valero, "Partitioning: An Essential Step in Mapping Algorithms into Systolic Array Processors", IEEE Computer, July 1987.

[Torr88] N. Torralba and J.J.Navarro, "A One Dimensional Systolic Array for Solving Arbitrarily Large Least Mean Square Problems", Proc. Int'l Conf. on Systolic Arrays 1988, pp. 103-112.

[Vale89] M. Valero-García, J.J. Navarro, J.M. Llabería and M. Valero, "Systematic Hardware Adaptation of Systolic Algorithms", 16th Annual Int'l Symp. on Computer Architecture, 1989, pp. 96-104.

[Weit83] Weitek, Floating Point Division/Square Root/IEEE Arithmetic WTL 1032/1033 Application Note 1983.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A UNIFIED APPROACH FOR THE REALISATION OF MULTIDIMENSIONAL DIGITAL SIGNAL PROCESSING

Mohamed B.E. Abdelrazik

Brunel University, Dept. of Electrical Engineering & Electronics
Kingston Lane, Uxbridge, England UB8 3PH.

A unified approach for the realisation of multidimensional digital signal processing in time- and frequency-domain is described in this paper. Two concepts, hypermatrix and multilinear form, have been evolved in order to achieve such approach. This approach produces general algorithms for digital signal processing which possess high degree of regularity, modularity and parallelism. The resulting realisations are flexible and suitable for VLSI/ULSI/WSI implementation.

1. Introduction

The realisation of multidimensional digital signal processing (m-D DSP) requires efficient algorithms in order to implement flexible and high performance processors. This paper describes a unified approach to realise and implement the time- and frequency-domain of m-D DSP. That is because the decomposition and realisation of m-D DSP systems is quite different to 1-D realisation. The basic theorem of algebra in one dimension cannot be extended to higher dimensions. However, there are many approaches to realise m-D DSP. Such approaches were based on the Singular Value Decomposition (SVD) [1], Lower/Upper (LU) triangular decomposition [2], QR factorisation [3], canonical factorisation [4] and Jordan decompositions [5]. In addition, Chinese Remainder Theorem (CRT) has found applications in the design of fast algorithm for 1-D and 2-D digital filters [6]. Most of these approaches are either special cases or approximation to the general decomposition problems and the realisation problems.

The proposed approach produces general algorithms for DSP which possess high degree of regularity, modularity and parallelism. In addition, the algorithms are highly flexible in order to achieve high performance systems. The application of this approach to time-domain can be seen as a hierarchical decomposition of a general m-D real rational transfer function [7]. The most important point is that the proposed approach produces algorithms for frequency-domain and adaptive filters which are highly flexibly to improve the cost/performance criterion. The resultant realisations are flexible and suitable for VLSI/ULSI/WSI implementation [8,9].

There are two approaches for obtaining the hardware realisations, namely the state space approach and Signal Flow Graph (SFG). The proposed approach is a direct realisation to the transfer function and can be mapped onto hardware. A fundamental issue of the unified approach is to express parallel algorithms in a notation that can be easily understood and directly mapped onto parallel processors. There are two concepts have evolved to facilitate and generalise the realisation, namely the hypermatrix and multilinear form, which have been described in [7,9].

2. Terminology and Notations

In order to simplify the realisation problem, the following terminology has been adopted. The DSP system can be completely specified by its dimensionality and the orders of the transfer function. The DSP system can be denoted by a 4-tuple as follows:

$$DSP(m,N) = [X,Y;I,S] \tag{1}$$

where m is finite dimensions, N is a finite nonempty ordered set of the orders of the system, X is a finite nonempty ordered set of the input sequence, Y is a finite nonempty ordered set of the response sequence, I is a finite ordered set of the initial conditions and S is the structure function, which may be denoted by a 3-tuple as follows:

$$S(m,N) = [R_1,R_2;*] \tag{2}$$

where R_1 and R_2 are finite sets of registers, '$*$' is the Linear OPerator (LOP). The contents of the R_1 and R_2 may be constants or variables depend on the various realisations. The operator '$*$' can be defined as the mapping function of R_1 and R_2 onto the structure function S() which can be denoted as the cartesian product

$$R_1 \times R_2 \dashrightarrow S$$
$$S = R_1 * R2 \tag{3}$$

The ';' in (1) is a separator between the variables which are applied in serial and those which are applied in parallel respectively. For example, digital filters can be denoted as follows:

$$DF(m,N) = [X,Y;I,S]$$
$$S(m,N) = [R(c),R(z);*]$$

The input set X is applied sequentially, and the output set Y is the sequential response of the filter. The initial conditions set I, which is preloaded prior to the computation phase, applied in parallel. While S() is a parallel structure of the digital filter. The set of registers denoted by R(z) represents the delay elements of the transfer function. The set of

registers denoted by R(c) contains the coefficients (constants or variables). While the operator '$\star$' is performed in parallel. The building cell of the structure can be represented by an element of the cartesian product $(r(z_i),r(c_i))$, as described by (3), under the binary operator LOP '$\star$'.

The discrete Fourier transform (DFT) can be denoted as follows:

$$\begin{aligned} DFT(m,N) &= [X(k);X(n),I,S] \\ S(m,N) &= [R(x),P(w);\star] \end{aligned}$$

The output set X(k) is calculated sequentially, while the input set X(n) is applied in parallel. The set of registers denoted by R(x) contains the input sequence. While the set of registers denoted by P(w) contains the variable coefficients which can be computed recursively as follows:

$$\begin{aligned} w_0 &= I, \text{ and} \\ w_i &= w_{(i-1)} \circ w_1 \end{aligned}$$

where $I = [1\ 1\ \dots\ 1]$, $w_1 = [1\ W\ W^2 \dots W^{(N-1)}]$, $i \in [0,N-1]$ and '$\circ$' is elementwise multiplication operator. The building cell of the structure can be represented by the cartesian product $(r(x_i),p(w_i))$, as described by (3), under the binary operator '$\star$'.

The important of the initial conditions set is that it contains all the information about the past behaviour of the system which is necessary to calculate the future state of the system and the present output. Moreover, the set I for the DFT realisation is an important factor to achieve high performance.

The structure function S() can be extended recursively to higher dimensions as follows:

$$S(m+1,N^*) = S(m,N) \star R_{m+1}$$

where N^* is the union of the set N and the element $\{N_{m+1}\}$, i.e., $N^* = N \cup \{N_{m+1}\}$.

It is convenient to denote the sum and product of a number of variables by $\Sigma_i x_i$ and $\pi_i x_i$, where $i \in [1,N]$ which can be read as the sum or product of x_i for $i = 1$ to $i = N$.

3. The Realisation of Digital Filters

The m-D causal recursive digital filter can be described by the transfer function

$$H(z_1, \dots ,z_m) = \frac{\sum_{i_1} \dots \sum_{i_m} a_{i_1,\dots,i_m} z_1^{-i_1} \dots z_m^{-i_m}}{\sum_{i_1} \dots \sum_{i_m} b_{i_1,\dots,i_m} z_1^{-i_1} \dots z_m^{-i_m}} \qquad (4)$$

where $z_j^{-i_j}$, $i_j \in [0,N_j\text{-}1]$ and $j \in [1,m]$ are the unit delays along different orthogonal axes of the m-D space.

The transfer function, equation (4), can be expressed by the multilinear form [9] as follows:

$$\begin{aligned} H'(z_1, \dots ,z_m) &= (((\mathbf{C} \star \mathbf{z}_1)_{i_1} \star \mathbf{z}_2)_{i_2} \star \dots \star \mathbf{z}_m)_{i_m} \\ &= \langle Cz_1,z_2, \dots ,z_m \rangle \end{aligned} \qquad (5)$$

where $\mathbf{C}_{i_1,\dots,i_m}$ is a hypermatrix, whose elements are the vector $[a_{i_1,\dots,i_m} \text{-}b_{i_1,\dots,i_m}]$, in the m-D space, $\mathbf{z}_j$ where $j \in [1,m]$ are vectors in the directions of the orthogonal axes $i_1,i_2, \dots ,i_m$ of the space, and

$$\mathbf{z}_j = \begin{bmatrix} 1 \\ z_j^{-1} \\ \dots \\ z_j^{-(N_j-1)} \end{bmatrix}$$

Then, the transfer function may be expressed [9] by

$$H'(z_1, \dots ,z_m) = \frac{Y(z_1, \dots ,z_m)}{X(z_1, \dots ,z_m)} \qquad (6)$$

where $X(z_1, \dots ,z_m) = [X(z_1, \dots ,z_m)\ \ Y(z_1, \dots ,z_m)]$ is a vector which represents the Z-transform of the system inputs.

Equation (6) shows that this technique is applicable to a system with multiple inputs; and can be extended to a system with multiple outputs. The structure function of such a system, which is a direct mapping of the transfer function (5), can be written as

$$\begin{aligned} S(m,N) &= (((R(c) \star r(z_1))_{i_1} \star r(z_2))_{i_2} \star \dots \star r(z_m))_{i_m} \\ &= \langle R(c)r(z_1),r(z_2), \dots ,r(z_m) \rangle \end{aligned} \qquad (7)$$

where R(c) is the set of registers which contains the coefficients a's and -b's. As described above, the structure function S() is recursive function, then the system can be realised hierarchically as follows:

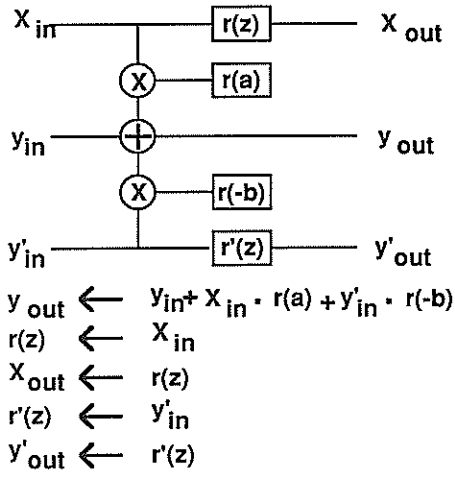
$$\begin{aligned} S(1,N_1) &= \langle r(c),r(z_1) \rangle \\ &\dots\dots\dots\dots \\ S(i,N'_i) &= \langle S(i\text{-}1,N'_{i-1}),r(z_i) \rangle \\ &\dots\dots\dots\dots \\ S(m,N'_m) &= \langle S(m\text{-}1,N'_{m-1}),r(z_m) \rangle \end{aligned}$$

where r(c) and $r(z_i)$ are two sets of registers, and N'_i is a set whose elements are the orders of the transfer function, i.e., $N'_i = \{N_1, \dots , N_i\}$. Each structure function can be realised in two forms, namely form I and form II. The two forms are the result of the commutative axiom of the LOP (7-9) as follows:

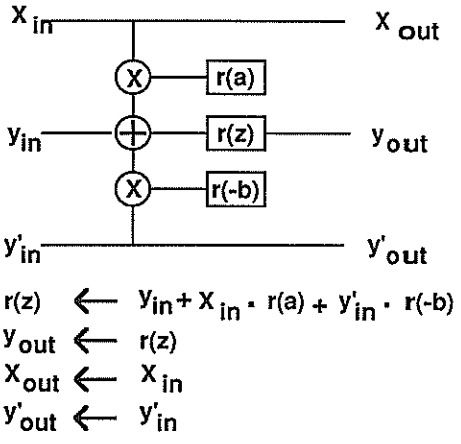
$$\begin{aligned} S(i,N'_i) &= \langle r(z_i),S(i\text{-}1,N'_{i-1}) \rangle, \quad \text{Form I} \\ S(i,N'_i) &= \langle S(i\text{-}1,N'_{i-1}),r(z_i) \rangle, \quad \text{Form II} \end{aligned}$$

In general, the total number of realisations (structure functions) is $2m(m!)^2$. This is the result of the commutative axiom of the multilinear form with respect to LOP and the permutation of the set N'_i [7-9]. Figure 1 shows the building cells and their definitions for form I and II, a linear array and 2-D array structures. FIR can be also realised by this technique, however, the area complexity will be reduced by factor 2, where r(c) = r(a) and r(z) = r(x).

The cost of recursive structure can be defined in terms of the number of binary adders, multipliers, registers and shift



Building cell form I



Building cell form II

registers as follows:

$$\#(R(c)) = \#(Mul) = 2\,\pi_i\,N_i,$$
$$\#(Add) = 2\,\pi_i\,N_i + \Sigma_j\,\pi_k\,N_k,$$
$$\#(r(z)) = 2\,(N_i\text{-}1)\,\pi_j\,N_j + \Sigma_i\,(N_i\text{-}1)\,\pi_n\,N_n + (N_m\text{-}1).$$

where $i \in [1,m]$, $j \in [2,m]$, $k \in [j,m]$, $l \in [2,m\text{-}1]$ and $n \in [l,m]$.

The number of adders differs from one form to another, especially when using some of the design styles (for example tree-like style). Each of the realisation has different throughput. The throughput of a structure is just the reciprocal of the cycle time. The cycle time can be calculated as follows:

$$T_c = T(Add)\,.\,(\lceil \log_2(m\text{-}d+1) \rceil + \lceil \log_2(\pi_i\,N_i) \rceil) + T(Mul) + T(Reg)$$

where T(Mul) is the multiplication time, T(Add) is the addition time, T(Reg) is the access time of a memory or the shifting time, $\lceil \ldots \rceil$ is the ceiling function and d is the number of registers r(z)'s to the right of R(c) in equation (7).

4. The realisation of m-D DFT

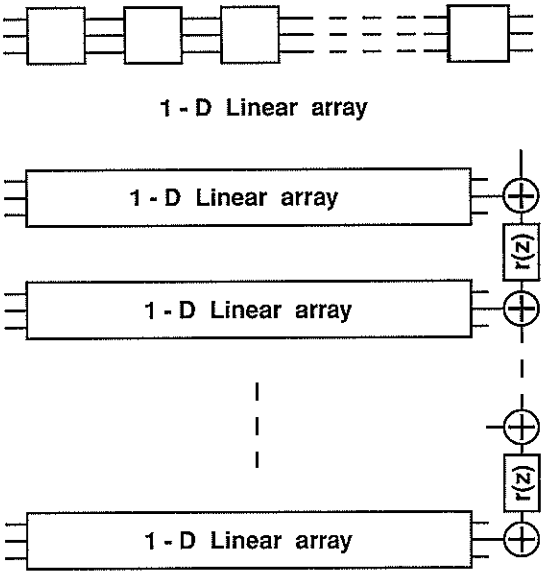
The N-point DFT may be defined as follows:

$$X(k) = \Sigma_n\,x(n)\,W^{nk} \qquad (8)$$

where $n,k \in [0,N\text{-}1]$. Equation (8) can be expressed in matrix notation [10] as

$$\mathbf{X}(k) = \mathbf{W}\,\mathbf{x}(n)$$

Each row in the matrix **W** may be written as



2 - D Array structure

Figure 1. Digital filter realisation

$$\mathbf{w}_k = [1\ W^k\ W^{2k}\ \ldots\ W^{(N-1)k}]$$

Hence, equation (8) may be expressed in multilinear form as

$$X(k) = \mathbf{x}(n) \ast \mathbf{w}_k = \langle x, w_k \rangle \qquad (9)$$

The structure function of DFT is a direct mapping of equation (9) as follows:

$$S(1,N) = [R(x),P(w);\ast] = R(x) \ast P(w), \text{ where}$$
$$P(w) = [P(w),w_1;\circ] = P(w) \circ w_1.$$

The DFT realisation and the cell definition are illustrated in figure 2. The cycle time for each point (X(k)) can be calculated as follows:

$$T_c = 2.T(Mul) + \log_2(N).T(Add)$$

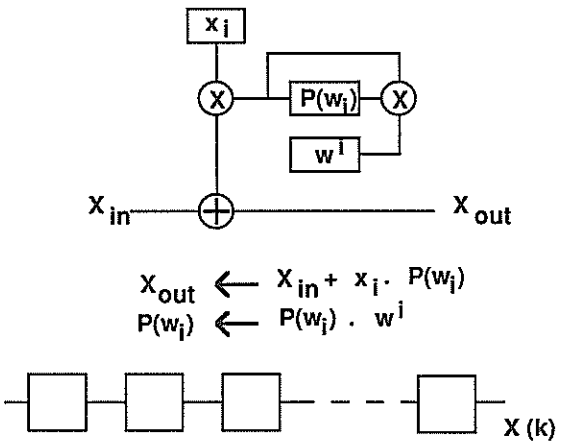


Figure 2. Building cell and Linear array for DFT

The time complexity of this structure is of the order O(N). In order to achieve high performance, we can use multiple structures with different initial conditions. For instance, using two structures whose initial conditions are

$$I_1 = [1\ 1\ \ldots\ 1],$$
$$I_2 = [1\ W^{N/2-1}\ W^{2(N/2-1)}\ \ldots\ W^{(N-1)(N/2-1)}].$$

Hence, the cost, in terms of the number of structures, and the performance ($T_t(1)/T_t(N)$) are increased by factor 2. The cost/performance criterion is still constant. The m-D DFT has the feature of separability. For instance, the 2-D DFT can be achieved by calculating 1-D of each column (row) followed by 1-D DFT of each row (column). Figure 3, illustrate the realisation of 2-D DFT. The time complexity of such structure is of order $O(N^2)$. However, high performance can be achieved by multiple structures as described above.

5. Conclusions

A unified approach for mapping m-D DSP onto VLSI/ULSI/WSI array structures is described. The resulting structures possess high degree of regularity, modularity and parallelism. The total structures of a transfer function for digital filters is $2m(m!)^2$. A cost function and cycle time are quantified to evaluate the cost/performance of each structure. The recursivity of structure function provides a hierarchical structure which supports a linear expandable structure. Moreover, these features of the mapping technique facilitate the automatic layout generation (silicon compiler) from the system description. This approach has been employed in order to design 2-D digital filter and parallel DFT structures

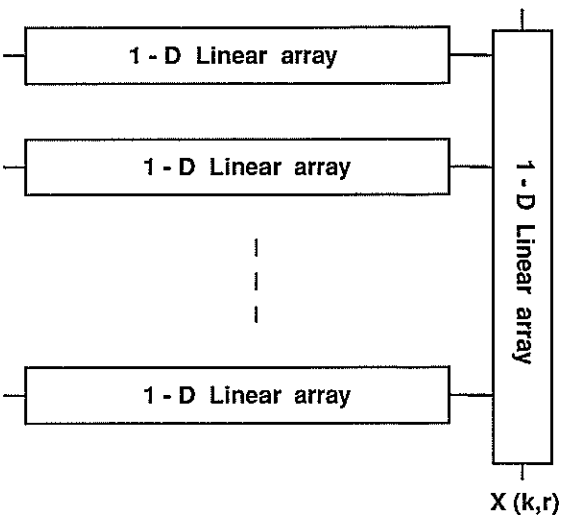


Figure 3. Parallel structure for 2 - D DFT

for image processing applications. These structures will be implemented in CMOS (2um feature size and two-level metal). The designed and verification issues will be achieved by using the Siemens design system "VENUS" at Brunel University.

Acknowledgement

The author gratefully acknowledges the stimulation and the support of Prof. G. Musgrave the head of the Department of Electrical Engineering and Electronics at Brunel University.

References

[1] S. Treital and J.L. Shanks, IEEE Trans. Geosci. Electron., Vol. GE-9 (1971) 10-27.
[2] C.L. Nikias, A.P. Chrysafis and A.N. Venetsanopoulos, IEEE Trans. on ASSP (1985) 694-711.
[3] S.Y. Kung, H.J. Whitehouse and T. Kailath, VLSI and Modern Signal Processing (Prentice-Hall, Inc., 1985).
[4] Mitra, S.K., A.D. Sagar and N.A. Pendergrass, IEEE Trans. on CAS, (1975) 177-184.
[5] K.M. Ty and A.N. Venetsanopoulos, IEEE Trans. on ASSP (1987) 904-907.
[6] J.K. Pitas and A.N. Venetsanopoulos, IEEE Trans. on CAS (1985) 1029-1040.
[7] M.B.E. Abdelrazik, 32nd Midwest Symposium on Circuits and Systems (1989).
[8] M.B.E. Abdelrazik, European Conference on Circuits Theory and Design (1989) 425-429.
[9] M.B.E. Abdelrazik, PhD Thesis, Brunel the University of West London (1988).
[10] Auslander, L., E. Fieg and S. Winograd, IEEE Trans. on Comp., (1983) 388-403.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

AN ARTIFICIAL NEURON BASED ADAPTIVE CLASSIFIER WITH A NOVEL UPDATE ALGORITHM

Yalçın TANIK, Mehmet Ali TUĞAY

Dept. of Electrical & Electronics Engineering, Middle East Technical University, Ankara — Turkey

In this work, a linear adaptive binary classifier which implements the Bayes test is presented. This classifier requires no a priori knowledge on the distribution of the observations to be classified. The algorithm used in the classifier directly attempts to minimize the probability of misclassifications. It resembles the Highleyman algorithm, in that, both methods try to minimize the relative frequency of misclassifications. However, the new algorithm operates on a real-time basis. It is suitable for implementation on an artificial neuron. The basic properties of the algorithm are given and tested by computer simulations.

1. INTRODUCTION

In the field of electrical engineering, *classification* may be defined as, given an observation, the process of determining the particular element of a finite set to which the observation fits the best. To achieve this, a *classifier* needs to know the basic properties that relate the observations and the set elements (the classes). When these properties are not exactly known or are subject to changes, the best way of implementing a classifier is to "teach" it: an *adaptive* classifier thus results.

Most of the adaptive classifiers use the *mean square error* (MSE) criterion, which is based on the minimization of the difference between the classifier response to an observation and a signal indicating the class of that observation; the "desired" response [1,2]. Such classifiers are generally well defined and their behavior is analytically tractable. However, their main drawback is that their output must be very well controlled so as to be compatible with the desired response.

The algorithm presented in this work attempts to directly minimize the probability of misclassification (PM), rather than dealing with a MSE criterion. It resembles the Highleyman algorithm [3] for minimization of the relative frequency of misclassifications (RFM) over a finite number of observations and can be implemented on an artifical neural network; the most versatile type of adaptive system.

2. DESCRIPTION OF THE ALGORITHM

In [3], Higleyman proposed an iterative algorithm for the determination of a linear boundary between two distinct classes in an observation space. This algorithm has been shown to generate, in an N–dimensional observation space, the hyperplane which minimizes the RFM. The Highleyman algorithm is based on a finite number of observations from each one of the two classes, and assumes no a priori knowledge on the distribution of these observations. This algorithm, when applied to a binary classification problem with two classes H_0 and H_1 , can be summarized as follows:

1. Given, an N–dimensional observation space and a vector $X = [1\ x_1\ x_2\ \ldots\ x_N]^T$, one of the M augmented observation vectors, define a hyperplane by the $N+1$ coefficients w as
   $S:\ W^TX = w_0 + w_1x_1 + w_2x_2 + \ldots + w_Nx_N = 0$.
   The classifier is known to divide the observation space into two parts, each being one side of this hyperplane.
2. Let $C(j,W)$ be the "cost" of a classifier decision, given the vector W of coefficients. $C(j,W)$ will be equal to C_F, if a vector X_j in H_0 is misclassified; to C_M, if X_j in H_1 is misclassified; or to zero if the decision is correct. The total cost may then be formulated as
   $$J(W) = \sum_{j=1}^{M} C(j,W)\ .$$
   $J(W)$ must be minimized with respect to W. Note that, if $C_F = C_M = 1$, $J(W)$ is equivalent to RFM.
3. The minimization of $J(W)$ might be made by gradient search techniques if $C(j,W)$ were not discontinuous on the hyperplane defined by W. Highleyman overcame this difficulty by letting $C(j,W) = C(s_j,W)$ be a continuous function of the distance s_j between X_j and the hyperplane: as s_j is increased, $C(s_j,W)$ gradually decreases if X_j is in the right side of the hyperplane, and gradually increases if X_j is in the wrong side.

The Highleyman algorithm takes, at a time, all of the finite number of observations into account. This algorithm may be modified by not restricting the number of observations, and by using, only a single observation at a time, to update the total cost. To formulate the algorithm, the additional definitions below are necessary:

t: the two–valued random variable indicating the true class of a given observation: the desired response,
Ω_t: the desired response space,
Ω_x: the observation space,
$u(.)$: the unit step function,
$C(X,W,t)$: the cost function with continuous argument X,

$\mathcal{R}(W)$: the expected value of the cost function; the "risk",
$p(X,t)$: the joint pdf of X and t.
The mathematical expression for the risk is

$$\mathcal{R}(W) = \int\int C(X,W,t)p(X,t)dXdt\,, \tag{1}$$

where the integration is over Ω_x and Ω_t. Assuming that t can arbitrarily be given the values $+1$ for H_1 and -1 for H_0, using the definitions of C_F and C_M, and assigning zero cost to correct decisions, the cost function can be written in a compact form:

$$\begin{aligned} C(X,W,t) &= \frac{1}{2}(1+t)C_M[1-u(W^TX)] \\ &\quad +\frac{1}{2}(1-t)C_F u(W^TX)\,. \end{aligned} \tag{2}$$

The risk thus becomes:

$$\begin{aligned} \mathcal{R}(W) = \tfrac{1}{2}\iint \big\{&[(1-t)C_F-(1+t)C_M]u(W^TX)p(X,t) \\ &+(1+t)C_M p(X,t)\big\}\,dXdt \end{aligned} \tag{3}$$

The aim is to find the vector W which minimizes this risk. A necessary condition is:

$$\nabla\mathcal{R}(W) = 0\,. \tag{4}$$

While evaluating this gradient, a problem will arise from the discontinuity of the unit step function on the hyperplane defined by $W^TX=0$. Similar to the Highleyman algorithm it must be replaced by a continuous and monotonically increasing function of W^TX, whose gradient is defined for all W. Let this function be denoted by $g(W^TX,\alpha)$, such that:

$$\lim_{\alpha\to\infty} g(W^TX,\alpha) = u(W^TX) \tag{5}$$

α is a parameter which controls the steepness of the function around $W^TX=0$. With this substitution, the gradient of the risk becomes:

$$\begin{aligned} \nabla\mathcal{R}(W) &= \frac{1}{2}\int\int[(1-t)C_F-(1+t)C_M] \\ &\qquad Xg'(W^TX,\alpha)p(X,t)dXdt \qquad (6) \\ &= \frac{1}{2}(C_F-C_M)\mathrm{E}\left\{Xg'(W^TX,\alpha)\right\} \\ &\quad -\frac{1}{2}(C_F+C_M)\mathrm{E}\left\{tXg'(W^TX,\alpha)\right\}\,. \qquad (7) \end{aligned}$$

If $C_F = C_M = 1$, the risk becomes equal to the PM, and the expression (7) reduces to:

$$\nabla\mathcal{R}(W) = -\mathrm{E}\left\{tXg'(W^TX,\alpha)\right\}\,. \tag{8}$$

In the sequel, unless otherwise stated, this specific form of $\mathcal{R}(W)$ will be considered. Eq.(8) is very suitably applicable to a stochastic approximation type algorithm [1]: the idea behind using the iterative procedure

$$W(k+1) = W(k) + \mu t(k)X(k)g'(W^T(k)X(k),\alpha) \tag{9}$$

is the same as that which led to the well-known LMS algorithm [4]. Therefore, the direction of the vector sequence $W(k)$ is expected to converge to the direction of a vector W which satisfies eq.(4), provided that the constant μ is properly chosen.

From the point of view of implementation, the only difficulty that may be encountered is the term $g'(W^T(k)X(k),\alpha)$ in eq.(9). This term depends on the special function $g(.,\alpha)$. The most suitable function seems to be the "sigmoidal" nonlinearity [2]:

$$g(z,\alpha) = \frac{1}{1+e^{-\alpha z}}\,, \tag{10}$$

since its derivative is easily expressed as:

$$g'(z,\alpha) = \alpha g(z,\alpha)\,[1-g(z,\alpha)]\,. \tag{11}$$

Thus, the classifier which implements the modified Highleyman algorithm may be a single artificial neuron with sigmoidal nonlinearity. Defining $y(k) = g(W^T(k)X(k),\alpha)$, as the neuron output, the algorithm (9) can be rewritten as:

$$W(k+1) = W(k) + \beta t(k)X(k)y(k)\,[1-y(k)]\,, \tag{12}$$

where β is the product of α and μ.

3. PROPERTIES OF THE ALGORITHM

First of all, it is important to notice that, since the coefficient vector W defines a hyperplane, given a vector W^* which satisfies eq.(4), any vector which is linearly dependent with W^* will also be a solution.

A. Optimality of the Solution:

For the binary classification problem, with the desired response t taking values ±1, the joint pdf of the observation X and t can be written as:

$$p(X,t) = q\delta(t+1)p(X|H_0) + (1-q)\delta(t-1)p(X|H_1)\,, \tag{13}$$

where q is the a priori probability that $t=-1$, and $\delta(.)$ is the dirac delta function. Using eq.(13) in eq.(8), and writing the expectation in integral form:

$$\begin{aligned} \nabla\mathcal{R}(W) &= -\int_{\Omega_x} g'(W^TX,\alpha)X \\ &\qquad [-qp(X|H_0)+(1-q)p(X|H_1)]\,dX\,. \end{aligned} \tag{14}$$

From eq.(5), it can easily be seen that the limiting form of $g'(W^TX,\alpha)$ is the dirac delta function $\delta(W^TX)$. Using this result, and eq.(14), eq.(4) can be written as:

$$\begin{aligned} &\lim_{\alpha\to\infty}\nabla\mathcal{R}(W) = 0 \\ \Leftrightarrow\quad &\int_{S_x(W)} X[qp(X|H_0)-(1-q)p(X|H_1)]\,dX = 0 \end{aligned} \tag{15}$$

where $S_x(W)$is the $(N-1)$–dimensional subspace of Ω_x whose elements (vectors) X are orthogonal to W.

Assuming that, for a given problem, the optimum classifier is linear, eq.(15) can be satisfied by two specific types of

vector W:

1. W defines the boundary hyperplane for the Likelihood Ratio Test (LRT) (or Bayes test): i.e. $S_x(W) = S_{xL}$, such that, for any X in S_{xL};
$(1-q)p(X|H_1) = qp(X|H_0)$. (16)
This is the global solution for the optimum vector W .
2. W is such that, the equality (15) holds, but eq.(16) is not satisfied for at least some X in S_{xL}. In other words, $S_x(W) \neq S_{xL}$. It can be shown that, in such a case, $S_x(W)$can be "rotated" to get closer to S_{xL}, decreasing $\mathcal{R}(W)$. Consequently, W is a saddle point for $\mathcal{R}(W)$. The small perturbations due to the stochastic approximation nature of the algorithm will avoid convergence to such vectors.

B. Convergence of the Algorithm:

We assume a binary classification problem that can be solved by a linear classifier which performs the LRT. Then, the unique global minimum of $\mathcal{R}(W)$ is attained when W defines the subspace $S_x(W)$satisfying eq.(16).

At a first glance, the selectivity of $g'(W^TX, \alpha)$ around $S_x(W)$seems prone to cause start up problems: if $p(X)$ happens to be close to zero around $\S_x(W(0))$, the convergence may never start at all. This difficulty may easily be overcome by choosing α small enough to assure that $g'(W^T(0)X, \alpha)$ is not close to zero over a wide region around $S_x(W(0))$. $W(k)$ is thus frequently updated at start up. On the other hand, α must not be excessively small to loose its selectivity in Ω_x.

The parameter μ must guarantee that $\|\mu g'(.,\alpha)X\|$ is not too large (say at most 10%) as compared to $\|W(0)\|$. This will be useful in avoiding large updates in wrong directions due to the stochastic approximation nature of the algorithm.

A very important feature of the algorithm is that the selective term $g'(W^T(k)X(k), \alpha)$ enhances the update vectors which are orthogonal to the current coefficient vector $W(k)$. Therefore, the magnitude of $W(k)$ will globally increase. The effect of this increase will be equivalent to increase α, while keeping the norm of $W(k)$ constant (adjusting only its direction). Consequently, as the algorithm proceeds, the selectivity of $g'(W^TX, \alpha)$ increases, causing $W(k)$ to be updated only by vectors almost orthogonal to it. The algorithm will thus have no steady-state, since $W(k)$ may increase without a bound. A solution $W(\infty)$ with finite norm may be attained only if $g'(W^T(\infty)X, \alpha)p(X)$ is identically zero. Since $g'(W^TX, \alpha)$ is strictly positive for all finite α, this condition is never exactly satisfied, unless an implementation with finite precision arithmetic is in question.

4. COMPUTER SIMULATIONS

In this section, three example problems which reveal the outstanding features of the algorithm are presented.

Problem 1: Binary Data Corrupted by Gaussian Noise.

This very simple problem is used in investigating the basic properties of the algorithm. The classifier device makes its decisions based on single observations. The conditional pdf of the observations is

$$p(x|H_i) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left\{-\frac{(x-m_i)^2}{2\sigma_i^2}\right\}, \qquad i = 0,1, \quad (17)$$

where $\sigma_0 = \sigma_1 = \sigma$ is varied, $m_0 = -1$ and $m_1 = 3$ are kept constant. The optimum treshold thus must be $x = 1$. To evaluate the properties of the algorithm, the ratio $-w_0/w_1$, which defines the boundary hyperplane is considered. The optimum ratio must be unity. The convergence time τ, to 90% of this optimum ratio (an average of 500 simulations, all starting with the same arbitrarily selected initial coefficient vector $W(0) = [-3 \;\; 1]^T$) and the perturbations σ^2_W (the mean square deviation of $-w_0/w_1$ from unity, starting with $W(0) = [-1 \;\; 1]^T$ and averaged over 100000 iterations) are shown in Table 1 for various values of μ, α and σ.

σ^2	μ	α	τ	σ^2_W	$\|W(10^5)\|$
0.8	0.1	3	250	0.005	7.5
0.8	0.2	3	120	0.009	9.5
0.8	0.4	3	50	0.014	12.5
0.8	4	0.3	40	0.004	55
1.5	4	0.3	40	0.007	55

Table 1: Convergence time and perturbations.

It can readily be seen that, as expected, ...
a) ...if α is decreased, the norm of $W(k)$ rapidly increases, leading to a relatively stable solution.
b) ...the convergence time decreases and the perturbations after convergence increases with increasing μ.
c) ...the perturbations after convergence increase with the increasing density of $p(X)$ (here, related to σ) around S_{xL}.

Problem 2: Arbitrary Classification of 3-D Observations.

This is an artificial problem intended to evaluate the ability of the proposed classifier to solve peculiar classifications on multidimensional observations. For this specific data,

$$p(X) = p(x_1, x_2, x_1) = p(x_1)p(x_2)p(x_3)\,, \qquad (18)$$

where x_1 is normal with unit variance and mean 0.2, x_2 is uniformly distributed over [-2,2], and, x_3 is normal with variance 4 and mean -1. The classification boundary is defined as:

$$S_{xL}: \quad -1 + 0.5x_1 - 0.5x_2 + x_3 = 0\,. \qquad (19)$$

For various values of the initial coefficient vector and of the algorithm parameters, the classifier generates the coefficients which describe S_{xL}. If $\mu = \alpha = 0.5$, convergence to

90% of the optimum coefficient values occurs within 3000 steps.

Problem 3: Nonlinear Classification Boundary.

This problem is designed to emphasize the PM–minimizing feature of the algorithm. Observation data is uniformly distributed over the rectangular area shown in fig.1, and the classification imposed is indicated. The PM is minimized when

$$S_x(W): \quad -1 + 1.5x_1 + 1.5x_2 = 0\,. \tag{20}$$

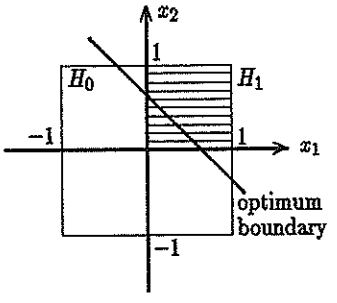


Figure 1: Observation space and classes.

For $\alpha = 1$ and $\mu = 0.5$, and with various choices of $W(0)$, the algorithm effectively generates this boundary within 2000 iterations. Fig.2 shows sample learning curves for the ratios $-w_0/w_2$ and $-w_1/w_2$.

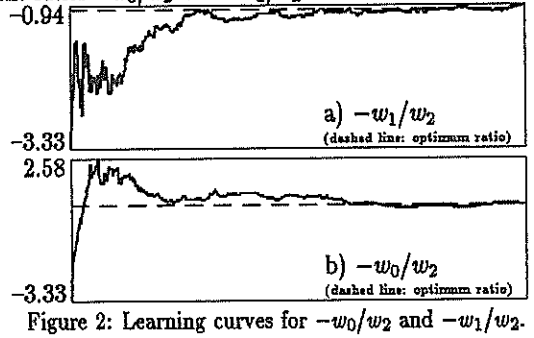


Figure 2: Learning curves for $-w_0/w_2$ and $-w_1/w_2$.

5. DISCUSSIONS

The PM–minimizing algorithm presented in this work uses a stimulus and response controlled reinforcement procedure [5], rather than an error–correction method (that of the MSE–based algorithms), since it does not compare the classifier output to a desired response. A MSE–based, error–corrective algorithm devised for the same type of neural element (a single–neuron version of the well–known "back propagation training algorithm" [2] for perceptron type neural networks) uses the iteration procedure:

$$W(k+1) = W(k) + \beta[t(k) - y(k)]X(k)y(k)[1 - y(k)] \tag{21}$$

Comparing equations (12) and (21), it can be seen that the substitution of the term $t(k)$ by $t(k) - y(k)$ would gradually slow down the convergence, but reduce the perturbations around the solution. However, as $\|W(k)\|$ gets large, these perturbations would be nearly the same for both algorithms.

The algorithm is shown to be capable of implementing a linear binary classification device with no restriction on the distribution of observations. The main practical problem about the algorithm is the lack of well defined general rules for the selection of the parameters α and μ. Although the global convergence is not affected for a very wide range of these parameters, the convergence time and the deviations after convergence are highly affected by them: optimal choices for α and μ are problem dependent and, therefore, can empirically be obtained.

It is also interesting to note that, since there is no theoretical upper bound on the norm of the coefficient vector W, its update rate is considerably decreased as its norm increases. Therefore, if the decision boundary is subject to changes (a kind of nonstationary problem), the norm of W must be occasionally reduced in order to keep track of them.

The theoretical analysis is carried out for the direct minimization of the PM. The algorithm can be generalized to specific problems where $C_F \neq C_M$. Given C_F and C_M, eq.(12) must be modified, according to eq.(7), as:

$$W(k+1) - W(k) = \\ +\beta\,[(C_F - C_M) + (C_F + C_M)t(k)]\,X(k)y(k)\,[1 - y(k)]$$

The resulting coefficient vector will minimize the risk defined in eq.(3).

For nonlinear classification problems, $g(f(W^TX), \alpha)$ may replace $g(W^TX, \alpha)$, where $f(.)$ is a nonlinear function. As in many applications, $f(.)$ may be a Taylor series expansion [1]. Obviously, such a modification necessitates some a priori knowledge on the decision boundary, and a more careful choice of the parameters α and μ.

6. REFERENCES

[1] Young T.Y. and Calvert T.W., *Classification, Estimation and Pattern Recognition*; American Elsevier Publishing Inc., New York, 1974.

[2] Rumelhart D.E., Hinton G.E. and Williams R.J., *Parallel Distributed Processing, Volume 1: Foundations*; Rumelhart & McClelland (eds.), MIT Press, Massachussetts, 1986.

[3] Highleyman W.H., "Linear Decision Functions with Application to Pattern Recognition"; *Proc. IRE*, Vol. 50, No. 6, June 1962.

[4] Widrow B. et al., "Stationary and Nonstationary Learning Characteristics of the LMS Adaptive Filter"; *Proc. IEEE*, Vol. 64, No. 8, Aug. 1976.

[5] Rosenblatt F.D., *Principles of Neurodynamics*, Spartan Books, Washington D.C., 1959.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

CONTINUOUS LEARNING: THE USE OF A DESIGN METHODOLOGY FOR FAULT-TOLERANT NEURAL NETWORKS WITH UNSUPERVISED LEARNING

Vincenzo Piuri

Department of Electronics, Politecnico di Milano, piazza L. da Vinci 32, 20133 Milano, Italy

A general design methodology for fault-tolerant neural networks is presented by considering the nominal behavior of the network and the effects of faults. Its use is also evaluated with particular regard to artificial neural networks with unsupervised learning mechanisms.

1. INTRODUCTION

Massive computation is required by many applications, e.g. in signal and image processing and in pattern and speech recognition. A number of advanced architectures were proposed in literature to achieve a considerable throughput for computational-intensive algorithms: artificial neural networks are one of these paradigms. These architectures, their functioning, the related learning rules were studied time ago [1] [2]. They have received new attention which has been continuously increasing since the recent advances in integration technologies. These results allow efficient, effective realisations by using VLSI, ULSI and WSI integrated circuits. Besides, the costs and the performances of neural networks become competitive with respect to traditional computing structures.

The lack of reliability and computation availability, due to end-of-production defects and to life-time faults [3], cannot often be tolerated. In mission-critical or strategic applications error detection and system survival are appreciated to guarantee a high quality of the computing architecture and a high availability of its services.

The present paper is a contribution in the direction of designing fault-tolerant neural networks. The approach here proposed is a general methodology, which can be effectively used in a number of network structures and learning rules. Particular attention is given to neural networks with unsupervised learning mechanisms [1] [2]. In section 2 we briefly review neural models and learning mechanisms, while in section 3 we discuss the effects of faults on the computation and the error-correcting capabilities. In section 4, we present the design methodology: *continuous learning*. In section 5, we present the application to unsupervised networks, the architectural solutions and the performance evaluations.

2. NEURAL ARCHITECTURES AND LEARNING ALGORITHMS

Artificial neural networks are a class of architectures which are particularly suited for massive computation [1] [2]. This characteristic is due to the distribution both of the computational resources and the information. Different aspects of neurocomputing are studied to point out the main characteristics of this computational model and to exploit new computational power for highly-parallel applications in engineering and research.

As a natural neural network, the artificial networks are complex structures containing a number of computing units, the *neurons*, which are densely interconnected [1] [2]. I_i is the input of the i-th neuron from the external world, x_j is the input from the j-th neuron, w_{ij} is the weight of the interconnection from the neuron j to the neuron i. The output y_i is generated by applying the non-linear transfer function $f_i(.)$ to the weighted sum of the inputs. Often neurons are organised into groups, called layers. Usually, only a subset of neurons (the input layer) is connected to the external world: neurons connected to the final outputs constitute the output layer. The other layers are called hidden layers. A artificial neural network is completely defined by the characteristic parameters of the computation, the overall structure of the network, the interconnection weights and the non-linear transfer function.

Recent researches [1] [2] identified specialised architectures for classes of applications and learning techniques to configure the characteristic parameters. These procedures simulate learning in natural neural networks. Sequences of examples are supplied during training to the artificial network: knowledge is stored in the network by updating the interconnection weights. Then, the network configuration is frozen and the learnt algorithm is executed: no further updating is performed. This is the *operational* phase (or *recall*).

The learning mechanisms may be classified in two main groups: the supervised and unsupervised techniques. In *supervised* learning techniques, the learning phase consists in adjusting the synaptic weights to minimise the difference between the expected and the actual outputs, under the guide of a teacher. In *unsupervised* learning techniques, the synaptic weights are updated by extracting information only from the nominal input stimuli during training, without any knowledge about the expected outputs. To show effectiveness of *continuous learning* for fault tolerance, we consider as examples the recirculating networks [4], the Boltzmann machines [5] and Kohonen's self organisation [6].

3. TECHNIQUES FOR FAULT TOLERANCE

The effect of deviations from the nominal behavior of the digital architecture of a neural network appears as an *error* in the computation. It is the difference between the actual output of a neuron and the output expected from the behavioral definition. From the point of view of the network outputs, *externally-observable errors* appear at the external outputs of the neural network and, thus, they may be directly observed. Conversely, *externally-masked errors* remain hidden inside the network, due to topological characteristics, to the distributed structure of the computation and to the actual inputs.

By introducing additional test points at the output of each neuron or inside each neuron, it is possible to increase the granularity of testing. Any difference between the actual and the expected outputs of a neuron is an *observable error*, when proper circuits and techniques have been provided to identify such difference. Some errors at the output of a circuit are *masked errors* if they remain hidden in neurons. For example, whenever the evaluation of the transfer function is not affected by an error in the computation of the weighted sum, the error is masked in the final output of the neuron. Error masking induces latency of the error and, possibly, superposition of subsequent errors.

One of the most important sources of errors is related to the presence of end-of-production defects [3]. They are due to improper production, occasional errors or random events during some production phase. they are present at the termination of the production or appear immediately at the beginning of the life of the circuit. After the burn-in period, all these faults are present and do not change in

time. The second main source of errors is given by life-time faults [3]. They may be generated by the functioning stress or by external sources. The main difference with the previous kind of faults is that they are not present at end of production: they appear, often one at a time, during the normal functioning. Faults and defects can be described by using various models at different abstraction levels. In this paper we assumed the traditional model at gate level, as proposed in [3]. Any fault is represented by a functioning error at the output of some logic gates: the nominal behavior of these gates is modified by a stuck-at-value or a bridging with other signal lines.

Testing applied to artificial neural networks cannot often identify and localise all faults for lack of information or observation and control points. For example, in multiple-layered networks may be not easy to verify completely the hidden layers and localise faults. Self-testing architectures and design for testability may be used to improve detectability and diagnosability in the network, while fault-tolerance techniques based upon redundancy and reconfiguration enhance reliability and availability.

Fault tolerance in artificial neural networks should guarantee a correct behavior of the neural network according to the nominal description, even in presence of error sources. Usually, fault tolerance is composed by some of the following activities:

- *error detection*, i.e. the identification of the presence of erroneous behaviors in the network with respect to the nominal specifications,
- *error correction*, i.e. the modification of the erroneous outputs to mask externally the presence of error sources,
- *localisation of the error source*, i.e. the identification of the source of the error and, possibly, of the position of the source itself in the network architecture,
- *network repair*, i.e. the use of techniques for excluding the sources of errors from the actual computation or for including and masking them definitely.

Implementation and adoption of some of these phases is related to the characteristics of the neural network and to the requirements of the specific application. Besides, each technique must accurately consider the class of errors that must overcome in the network.

Error detection is the basic requirement of any fault-tolerance approach. Defects may be detected by applying traditional testing scheme to the digital/analog neural network [3] [8], possibly by introducing additional hardware to support diagnoeis. The same techniques can also be used during life time by suspending the nominal activities of the network and by starting a periodic testing. Partially, it is also possible to extend these techniques for on-line verification of a generic digital circuit. All these approaches, since they do not consider the specific characteristics of the circuits, are usually expensive in terms of silicon area for the additional circuits, in computational delay and in availability of the system.

Many techniques were presented in literature for on-line detection of errors in digital circuits [8]. They are based for example upon hardware modular redundancy and data coding. The former method introduces identical copies of the basic computational unit: computation is performed in parallel in all of them on the same data and outputs are compared to verify the presence of errors. In the second technique, input data are transformed through a coding rule: computation is then performed in the code space. Results are checked and converted back into the nominal data space.

Error correction should be used whenever the application is particularly critical and continuous generation of correct results is mandatory. Many extension of the detection techniques were proposed in literature for traditional digital circuits [8]. Correction may be achieved by increasing the hardware redundancy through voting or by improving the capabilities of coding.

To assure system survival even in presence of error sources, we must adopt techniques for network repair. A traditional approach is based upon dynamic redundancy. In particular, reconfiguration techniques [9] may be used to exclude the faulty units from the active computation. To support this approach, spare units, interconnections and programmable switching elements must be introduced into the basic structure of the network. The nominal computation is performed upon a subset of neurons and interconnections, while the others remain unused until an error is detected. Localisation techniques are required to identify the position of the faulty units which produced the error. Testing techniques could be used both for off-line diagnosis and for on-line localisation during the nominal computation. After localisation, host-driven or on-chip reconfiguration rebuilds a functioning neural network by excluding the faulty neurons or interconnections. This approach is effective until enough redundancy is available.

4. A STRUCTURED DESIGN METHODOLOGY FOR FAULT TOLERANCE

The common aspect of fault-tolerance techniques discussed in the previous section is that they do not consider the characteristics of the behavior and the internal structure of the neural networks. In fact they were proposed in literature for generic digital circuits and, often, for a quite traditional view of the computation. In this paper, we propose *continuous learning* as a general design methodology based upon an extensive consideration of the structural and behavioral characteristics of neural networks. The final goal of this methodology is to improve the fault-tolerance capabilities of the neural network by allowing continuous adaptation of the network itself to the internal functioning status.

After production, the initial learning is used to configure the neural network and to define the computational parameters which generate the nominal behavior. During this learning phase some defects may be masked. Faults and defects may be divided in two classes from the point of view of learning: the *critical faults* and the *non-critical faults*. In the latter class, there are all defects and faults that can be overcome simply by learning the nominal behavior. In this case the network contains enough redundancy to mask completely the errors at the final outputs by redistributing computation and information inside the network itself. In the first class we include all other defects and faults, which cannot be hidden by a learning procedure. An example is given by the stuck-at defects in the output gates of the output layer.

Initial learning is effective to hide non-critical defects by concentrating and adjusting information and computation onto the fault-free neurons. When these neurons are insufficient to perform the nominal algorithm, the whole neural network is not able to survive to the considered defect distribution with the nominal behavior. In many cases, the neural network containing defects may be still used, even if with *degraded* performances. For a small amount of critical defects, the network may be able to execute an algorithm which is similar to the nominal one, but with reduced capabilities. For example, a network for pattern classification needs a number of neurons related to the granularity of recognition capability. When few critical defects affect the nominal structure, the system is often still able to distinguish patterns, but with a smaller granularity. In this case, classes should be grouped and restructured with respect the nominal system. When no degradation is acceptable, modular hardware redundancy must be considered [9].

After initial learning, faults during life time may induce new errors in the computation and in the knowledge stored in the network. Continuous learning may be used to overcome these faults. The overall strategy comes from observation of natural neural networks. They have a continuous learning mechanism and control on the evolution of the environment and on the internal status of the organism. In fact, learning and adaptation never finish in any biological system. Fault tolerance may be introduced in artificial neural networks by imitating this natural behavior. A child learns a great amount of behaviors, operations and techniques during its first years of life. In a similar way, we can teach problem solving to an artificial neural network through initial training. Then, a man is still able to learn new behaviors to adapt himself to the surround-

ing environment or to overcome his own internal failures (e.g. the death of some neurons or the lack of an arm). In a similar way, the artificial network may be allowed to continue its learning to tolerate some hardware faults. In other word, continuous learning in artificial neural networks modifies the configuration of interconnections and computation even after initial learning of the nominal behavior. The behavior is not stored statically once, but it is dynamically updated during the life of the network to overcome internal failures.

Implementation of continuous learning, for any learning mechanism, may be achieved by using the following steps:

1. The initial configuration of the network is given by initial learning. Such configuration defines the nominal behavior of the network and is able to mask non-critical defects. At the end of the initial learning, the configuration is frozen.
2. When a fault occurs and produces an observable error, the final results of the computation delivered by the output layer may contain errors. Suited techniques based upon algorithmic redundancy may be used to identify the presence of errors concurrently with the nominal computation.
3. After error detection, the learning procedure is restarted to reallocate the computation and the knowledge in the network. At the end of the new learning session, the new configuration is frozen and the nominal computation is reactivated. The system reenters in the step 2 to continue on-line error detection.

Error occurrence during life time is thus considered as a new defect distribution for initial learning. Errors are masked by relearning of the nominal algorithm. This cyclic learning mechanism may be repeated until until saturation of available computational and storage redundancy. Subsequent faults become critical since no relearning procedure allows system survival. Definition of the saturation boundaries is a complex task since it requires a detailed knowledge of the hardware architecture, the nominal algorithm and the defect/fault distribution.

The external effect of critical faults onto the computation is different according to the position and the kind of faults. If a bounded percentage of errors and a bounded variance of the actual outputs around the expected values may be accepted, the neural network may still be used without activating continuous learning, even if it has been affected by a sequence of faults. This approach is useful to reduce the total repair time and to increase the overall availability of the neural network. The network is considered definitely faulty only when a given number of errors occurred and the actual behavior is too far from the nominal one.

Fault-tolerance capabilities and behavior degradation may be improved by enhancing the features offered by continuous learning and adaptation to a faulty system. By introducing additional neurons in the basic structure, it is possible to enhance the computational capabilities of the network with respect to the minimal network which solves the specific problem. Such neurons are redundant since they are not required to identify the final solution, but are used for fault tolerance. Redundant neurons are included in the active computation during initial learning. The configuration of the network is thus different from the configuration of the minimal network. Redundant neurons are used in continuous learning in a way which is completely different from traditional reconfiguration strategies [9]. In reconfiguration, on spare neurons no computation of the nominal algorithm is mapped. In continuous learning, they participate to the nominal computation and, possibly, they enhance the nominal behavior until enough redundancy is available. In some applications (in particular with unsupervised learning), continuous learning may often be implemented by using the actual input data without suspending the activities of the system, but accepting temporary errors. Reconfiguration needs always suspension and remapping of the computation. The main drawback of continuous learning is that, even if the basic schema is general, it must be tailored for each specific application to achieve the best performances. Reconfiguration does not need any customisation since it is independent from the algorithm. An other disadvantage of continuous learning is the learning time, which could be too high and which could be reduce the availability of the system. The choice between continuous learning and reconfiguration based upon hardware modular redundancy is strictly related to the specific application, to the hardware architecture, to fault distribution, to the error detection techniques and to the strategies for fault localisation.

5. APPLICATION TO UNSUPERVISED NETWORKS

Design of fault-tolerant neural network by using continuous learning requires the definition of an enhanced hardware structure which is able to identify the presence of errors, to reactivate the learning mechanism and to overcome errors due to faults. In other words, the augmented structure must be able to execute the redundant algorithm, deduced from the nominal one, which can detect errors and allow system survival. Detailed definition of the redundant computation is strictly related to the specific application of the neural network.

Initial configuration defines the interconnection weights by applying the specific learning mechanism. During life-time, the network performs two computation in parallel: the nominal computation produces the nominal outputs as in traditional networks, while the redundant computation provides information to detect errors due to faults. Error detecting circuits operate during life time on the nominal and redundant outputs of the network to verify the presence of errors. As soon as they detect errors, they enable reactivation of the nominal learning procedure, i.e. the updating of interconnection weights. Learning terminates when the nominal algorithm has been taught again, or when a system failure occur due to complete consumption of hardware redundancy.

In continuous learning the *algorithmic redundancy* is adopted to detect error. Effectiveness and efficiency of such approach are due to the complete exploitation of the characteristics of neural networks and the detailed knowledge of the specific application. The basic idea consists of the identification of an upgraded algorithm: the basic part of such computation generates the nominal outputs as required by the application, while the additional computation produces redundant information which can be used to detect the error occurrence in the nominal outputs.

Algorithmic redundancy does not use data coding at the level of single data to execute the redundant computation. According to the specific nominal algorithm, additional input data are generated from subsets of the nominal input data by using simple mathematical operators. Such additional data are treated as in the nominal computation by the same or by redundant circuits. Then, checking information are extracted from the results of the nominal computation and compared with the results of the redundant computation. Extraction is also performed by means of simple mathematical relationships. This approach exploits specific relationships which hold between subset of nominal input data and which are preserved (or modified in a known way) by the computation. For example, in classification problems with unsupervised learning, the output classes can be grouped in super-sets according to a suited criterion. Nominal input data are treated by the network which performs the nominal computation. The outputs identify the class to which each input belongs. In parallel, the redundant computation generates checking information from the same input data. The outputs of such network identify the super-set to which the input belongs, i.e. the group of classes in which there is the class of the input. Check on final outputs means to verify that the class, identified by the nominal classification, belongs to the super-class recognised by the redundant computation. The application of such design principle is effective if the redundant classification in super-set may be performed in simple way by using a reduced number of neurons. The detailed rules which must be used to aggregate the classes are related to the specific application and the learning algorithm.

Reactivation of nominal learning mechanisms for continuous learning may be executed by one of the following approaches: *off-line* and *on-line* learnings. In off-line learning, the subsequent acti-

vations of the learning mechanism are completely equivalent to the initial configuration, even if they operate on an architecture which contains a higher number of faults. The nominal computation is suspended and learning is executed. At the end of this configuration phase, the interconnection weights are frozen again and the newly learnt computation is restarted. In on-line learning, the nominal learning mechanism is restarted on the actual inputs. In other words, the nominal computation is not suspended, but it is adapted to the newly faulty state. In this case we assume that the behavior that must be taught is not so different from the behavior of the network before error occurrence. Configuration takes advantage of the previous experiences embedded in the neural network during the previous learning phase. At most, during relearning, the network produces results which may be temporarily erroneous, even if the correct behavior is finally learnt. The presence of possible noise in the input patterns may lead to a less accurate final configuration, i.e. to an imprecise execution of the nominal algorithm.

Until enough redundancy is available, the neural network has performances which are similar to the nominal one or, at least, which satisfy a predefined lower bound. In these cases the system survives at faults. Otherwise, computational performances gracefully degrade progressively towards a completely-faulty state, since internal redundancy has been exhausted.

The use of continuous learning has been extensively studied and experimented by using computer simulations of artificial neural networks with unsupervised learning mechanisms. In particular, we considered the architectures for recirculation networks, Boltzmann machines and self-organising networks. Detailed evaluations of the costs and the performances of such approach are strictly related to the specific application and may differs between architectures and learning mechanisms.

The most interesting results of our research and our evaluations are presented in the following figures for experimental design of neural networks with unsupervised learning mechanisms. In fig. 1, the capability of detecting errors is given, with respect to the different learning mechanisms. Detection capability is the probability that an error is detected in the architecture: it decreases as the number of faults in the structure increases since the available redundancy is progressively exhausted. Experiments have shown also a dependence from the possible reconfiguration of the interconnections: if reconfiguration is interleaved to continuous learning, the detection probabilities of the off-line approach are improved. In fig. 2 we give the survival probability of the artificial neural network. It is the probability, versus the number of faulty neurons, that the network is able to execute the nominal computation without performance degradation. Pure continuous learning, pure reconfiguration and mixed solutions are considered. A small dependence on the relearning approach has been identified: off-line relearning redistributes the computation and information better than on-line relearning and, thus, achieve a better survival.

Degradation of the behavior due to the presence of faults in the neural network is given in fig. 3 with respect to the nominal behavior required to the network. It is a measure of the reduced capabilities of correct execution in presence of observable errors. Performance degradation is given as the ratio between the number of actual results which are correct from the point of view of the application and the expected number of correct results which are computed by the nominal fault-free architecture. Since continuous learning may use the redundant neurons during the nominal computation, the quality of the computation (i.e. the computational performance) is higher than the nominal one when a small number of faults occurred.

In figg. 4 and 5, we show the costs of the experimental implementation considered in computer simulations. Technology independent evaluations of digital implementation has been adopted. In particular the hardware cost is given by the percentage of hardware redundancy (i.e. the percentage of additional gates), that must be introduced for fault tolerance. Similarly, the computational delay has been computed as the percentage of additional gates that are involved in neural computation and in output checking.

REFERENCES

[1] J.A. Anderson, E. Rosenfeld, editors, *Neurocomputing: foundations of research*, The MIT Press, 1988

[2] D.Z. Anderson, editor, *Neural information processing systems*, The American Institute of Physics, 1988

[3] J.A. Abraham, W.K. Fuchs, "Fault and error models for VLSI", *Proc. IEEE*, May 1986

[4] G.E. Hinton, J.L. McClelland, "Learning representations by recirculation", *IEEE Proc. on Neural Information Processing Systems*, 1988

[5] D.H. Ackley, G.E. Hinton, T.J. Sejnowski, "A learning algorithm for Boltzmann machines", *Cognitive Science*, 1985

[6] T. Kohonen, *Self organisation and associative memory*, Springer-Verlag, 1984

[7] S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, "Optimisation by simulated annealing", *Science*, 1983

[8] D.P. Siewiorek, R.S. Swarz, *The theory and practice of reliable system design*, Digital Press, 1982

[9] F. Distante, M. Sami, R. Stefanelli, G. Storti Gajani, "A configurable array architecture for WSI implementation of neural nets", *Proc. IPCCC'90*, Phoenix, 1990

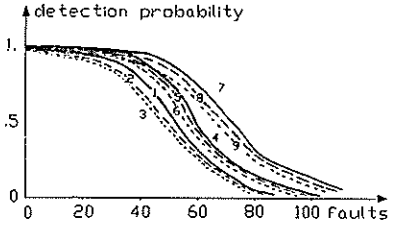


Fig. 1 - Detection capability

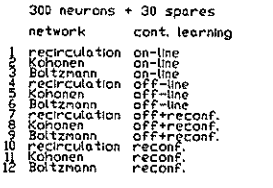
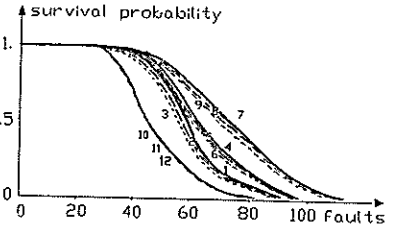


Fig. 2 - Survival probability



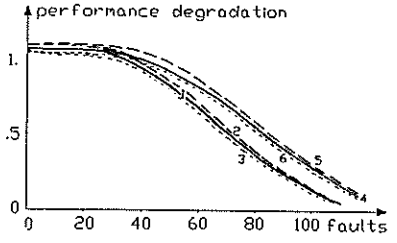


Fig. 3 - Performance degradation

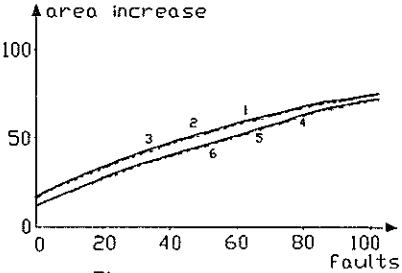


Fig. 4 - Hardware cost

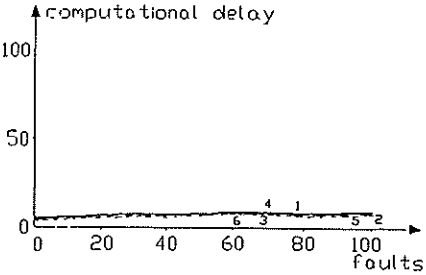


Fig. 5 - Computational delay

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A CLASS OF CONTINUOUS LEVEL BIDIRECTIONAL ASSOCIATIVE NEURAL NETWORKS

Zhong-Kai Yang Sheng-Wei Zhang Li-He Zou

Information and Control Engineering Dept., Xian Jiaotong University
Xian 710049, P. R. China*

A novel class of Bidirectional Associative Neural Networks (BANN's) is proposed in this paper. The formulation we present here is an extension of that in Alternative Projection Neural Networks (APNN's), and so the BANN's retain the attributes of APNN's such as ease of analysis in signal space, high accuracy of the steady-state solution, high storage capacity, and fast training process. The network can be configured as either a heteroassociative memories or classifier. Simulation results on storing and retrieving bipolar patterns in BANN's are presented.

I. INTRODUCTION

Departing from homogeneous or feed forward multilayer neural networks such as Hopfield model or backpropagation networks, bidirectional associative neural networks consist of two or more layers of neurons which transmit signal in both forward and backward directions. Such a closed-loop information transferring mechanism of BANN's is more like that in human nervous systems and is expected to perform better than homogeneous or unidirectional network[1].

Recently, a Bidirectional Associative Memory (BAM), proposed by Kosko as a heteroassociative memory and a pattern associator, extends the symmetric unidirectional autoassociator [2] of Cohen and Grossberg and Hopfield [3,4]. Like the more conventional technique of forming an energy metric for the neural network, the BAM, establishing a lower energy bound and showing the energy reduce in each iteration, may also arrive at its stable state which corresponds to the system local minimum energy rather than global minimum energy. Such procedure generally does not address the accuracy of the final solution. Furthermore, the BAM does not scale well, i.e. its performance is greatly degraded with the increase of the numbers of stored paired-data associations $(\underline{X}_r, \underline{Y}_r)$.

In contrast to the conventional technique, Marks II et al. presented a class of continuous level neural networks, called Alternating Projection Neural Networks (APNN's) [5], which perform by alternatively projecting between two or more constraint sets. It has been shown that APNN's has advantages over other neural network architectures, such as ease of analysis and geometrical interpretation in signal space, high accuracy of steady-state solution for both synchronous and asynchronous operation, high storage capacity, and fast training process. In this paper, we extend the work of Marks II et al by introducing the Bidirectionality into APNN's to form two-way associative search for stored associations $(\underline{X}_r, \underline{Y}_r)$. Unlike the APNN, the new Bidirectional Neural Networks perform by alternatingly mapping between two different Hilbert spaces X and Y. Passing information through T_X gives one direction; passing it through T_Y gives the other. Sufficient conditions for proper convergence are established. Theoretical analysis and simulation results on noised character recollection in BANN's are given to illustrate its good performance.

II. Formulation of BANN's

Consider a BANN of two layer neurons, with N neurons in layer A and M neurons in layer B, as shown in Fig.1. The network can transmit signal in either A-->B or B-->A direction. Let $\underline{X}$ and $\underline{Y}$ denote the state vectors in layer A and B, respectively. We denote x_i as the ith neuron state in layer A, y_i in layer B.

It is well known that there are two operation models, synchronous or asyn-

*This research was supported by the Natural Science Foundation of China.

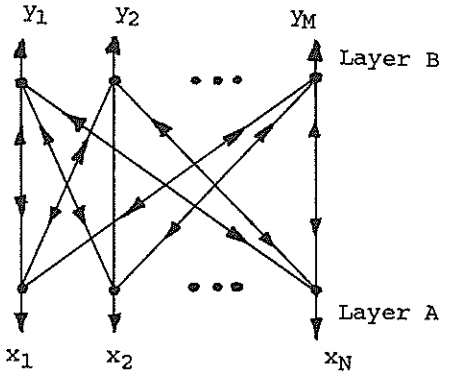


Fig.1 diagram of the BANN Structure

chronous models, in neural dynamic process. A special case of the asynchronous operation is the sequential one. As for APNN's, effects of synchronous and sequential operation are the same, and we only discuss synchronous operation in this paper. The BANN's dynamics under general asynchronous operation is very complicated and could not be contained in such a limited paper.

Suppose that there are L key / recollection vector pairs $(\underline{X}_r,\underline{Y}_r)$, $\underline{X}_r\epsilon R^N$, $\underline{Y}_r\epsilon R^M$, L<min(N, M), named library patterns, stored and recalled in the BANN. Now we construct two library matrices by:

$$F_X=[\underline{X}_1 \mid \underline{X}_2 \mid \ldots \mid \underline{X}_L] \quad (1)$$

$$F_Y=[\underline{Y}_1 \mid \underline{Y}_2 \mid \ldots \mid \underline{Y}_L] \quad (2)$$

and choose the T_X as forward (A--->B) connection Matrix, T_Y backward (B-->A) connection matrix, that is:

$$A\text{--->}B:\ T_X=F_YF^+_X \quad (3)$$

$$B\text{--->}A:\ T_Y=F_XF^+_Y \quad (4)$$

where + denotes the Moore-Penrose inverse of matrix. When $\{\underline{X}_r\}$ and $\{\underline{Y}_r\}$ are both linearly independent sets, we have

$$F^+_X=(F^T_XF_X)^{-1}F^T_X \quad (5)$$

$$F^+_Y=(F^T_YF_Y)^{-1}F^T_Y \quad (6)$$

The association may start from layer A or layer B, resulting in two associating processes.

(1) Starting A:
$\underline{X}(0)\text{-->}\underline{Y}(0)\text{-->}\underline{X}(1)\text{-->}\underline{Y}(1)\text{-->}\ldots$

(2) starting B:
$\underline{Y}(0)\text{-->}\underline{X}(1)\text{-->}\underline{Y}(1)\text{-->}\underline{X}(2)\text{-->}\ldots$

That is, the information is alternatively transferred between layer A and B. The steady-state vector pair $(\underline{X}^*,\underline{Y}^*)$, if exists, is the association output.

As in APNN, the neurons in BANN can be clamped to a preassigned values and provide the network stimuli or can float in accordance to the stimuli of other neurons. The status of a neuron as clamped or floating is related to associating process as well as to applications. When the BANN starts association from layer A, the stimuli are laid on A and some of neurons in A are clamped, some are floating, while all the neurons in B are floating. When association starts from B, some of B neurons are clamped, while the remaining neurons of B and all the A neurons are floating.

III. STEADY-STATE SOLUTIONS

Here, we only need to discuss the dynamic properties of BANN when association starts from A, similar results can be obtained when association starts from B.

Without loss of generality, suppose that neuron 1 through P (P<N) are clamped and the remaining Q=N-P are floating in layer A. In this case neurons in layer B are all floating. By partition notation of matrix, we can express F_X as

$$F_X=\begin{bmatrix} F_{XP} \\ F_{XQ} \end{bmatrix} \quad (7)$$

where F_{XP} denote the first P rows of the key library matrix F_X, we define a corresponding clamping operator, η, on an arbitrary vector $\underline{a}$ as:

$$\eta\underline{a}=\eta\begin{bmatrix} \underline{a}_P \\ \underline{a}_Q \end{bmatrix}=\begin{bmatrix} \underline{x}_P \\ \underline{a}_Q \end{bmatrix} \quad (8)$$

Then, in synchronous form, the network performs the operation:

$$\underline{Y}(n)=T_X\underline{X}(n) \quad (9)$$

$$\underline{X}(n+1)=\eta[T_Y\underline{Y}(n)] \quad (10)$$

The two sequences of vectors, $\{\underline{X}(n), n=1,2,\ldots\}$ and $\{\underline{Y}(n), n=1,2,\ldots\}$ will be generated by $\underline{X}(0)$, the stimulating vector on layer A. For the steady-state solution of these two vector sequences, we have following result.

The sequences $\{\underline{X}_{(n)}\}$ and $\{\underline{Y}_{(n)}\}$ will converge respectively to some $\underline{X}_{(\infty)}$ and $\underline{Y}_{(\infty)}$ if the matrices F_{Xp} and F_Y are all full column rank. Furthermore, if the segmental vector $\underline{x}_p$ in (8) is a segmentation of a library pattern vectors, then $\{\underline{X}_{(\infty)}, \underline{Y}_{(\infty)}\}$ is a library pattern pair.

We only need to prove the convergence of sequence $\{\underline{X}(n)\}$ for synchronous operation, because convergence of sequence $\{\underline{Y}(n)\}$ can be assured if convergence of $\{\underline{X}(n)\}$ occurs.

Proof: For synchronous operation, the network iteration (9),(10) can be written as:

$$\underline{X}(n+1)=\eta[T_Y\underline{Y}(n)]=\eta[T_YT_X\underline{X}(n)]$$
$$=\eta[T\underline{X}(n)] \qquad (11)$$

If matrix F_Y has full column rank, then

$$T=T_YT_X=F_X(F_X{}^TF_X)^{-1}F_X{}^T \qquad (12)$$

Obviously, the T matrix orthogonally projects any vector onto a N-dimensional subspace $[F_X]$, formed by the closure of the column vectors of F_X. The clamping operator, η, orthogonally projects a vector onto a Q-dimensional linear variety which formed by the set of all N tuplets with their first P elements equal to $\underline{x}_p$. It is proved in [2] that the sequence $\{\underline{X}(n)\}$ will converge strongly to a library vector $\underline{X}_r$ corresponding to $\underline{x}_p$ if F_{Px} is a full column rank matrix.

V. ADAPTIVE TRAINING

The equations (3) through (6) are available only if we know the all of N library patterns. If the patterns are obtained sequentially, or a new pattern is waiting to join the library, thus an adaptive training is going to be in need.

Let $F_{X,r-1}$, $F_{Y,r-1}$ be the library matrices of r-1 patterns, and $T_{X,r-1}$, $T_{Y,r-1}$ be the corresponding association matrices, assume that we are going to add a new pattern pair $(\underline{X}_r, \underline{Y}_r)$ into the library. new library matrices $F_{X,r}=[F_{X,r-1}\ \underline{X}_r]$ and $F_{Y,r}=[F_{Y,r-1}\ \underline{Y}_r]$ can be constructed as [7].

$$F^+_{X,r}=\begin{bmatrix} F^+_{X,r-1}(I-\underline{X}_r\underline{D}_r) \\ \underline{D}_r \end{bmatrix} \qquad (13)$$

and

$$\underline{D}_r=\begin{cases} \dfrac{(I-F_{X,r-1}F^+_{X,r-1})\underline{X}_r}{\|(I-F_{X,r-1}F^+_{X,r-1})\underline{X}_r\|^2} & \text{if denominator}\neq 0 \\ \dfrac{(F^+_{X,r-1})^TF^+_{X,r-1}\underline{X}_r}{1+\|F^+_{X,r-1}\underline{X}_r\|^2} & \text{otherwise} \end{cases} \qquad (14)$$

where I is the identity matrix, and $\|.\|$ the norm of inner product.

Let $[F_X]$ be the subspace spanned by the columns of F_X, and $\underline{X}_r$ be the difference between $\underline{X}_r$ and its projection on $[F_{X,r-1}]$. We can get

$$T_{X,r}=\begin{cases} T^+_{X,r-1}(\underline{Y}_r-T_{X,r-1}\underline{X}_r)\underline{X}_r{}^T/\|\underline{X}_r\| & \text{if } \|\underline{X}_r\|\neq 0 \\ T_{X,r-1} & \text{if } \|\underline{X}_r\|=0 \end{cases} \qquad (15)$$

The $\underline{X}_r$ can be calculated by the Gram-Schmidt orthogonalization procedure. Similar equations for $T_{Y,r}$ can also be obtained.

VI. Computer Simulations

Computer simulations on storing and retrieving bipolar patterns in BANN are carried out. Layer A has 12x10 neurons, N=120; layer B has 14x12 neurons , M=168. Twenty (L=20) patterns (A, B, C, D, E, F, G, H, I, J, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9) form a ten library pattern pairs (0,A), (1,B), ..., (9,J) which are stored with association matrices T_X, T_Y constructed by SVD method [7]. Bernoulli clamping approach is used to identify the clamped neurons and floating neurons [8]. Various noisy patterns are stimulated on the layer A. The BANN can converge properly to related ideal pattern pairs in spite of up to 35% points being noisy. In addition, incorrect stable result due to the improper choice of clamped neurons in BANN's is also illustrated. A number of results are shown in Fig. 2-4.

VII. Conclusion

In this paper, Bidirectionality, forward and backward information flow, is introduced to APNN's to develop a novel Bidirectional Associative Neural Networks. Theoretical analysis and computer simulations show that BANN's have the same favorite attributes as APNN's do. However, BANN's are of more potential applications than APNN's, i.e.

BANN's can be used as heteroassociative memories while APNN's can only be used as autoassociative memories.

References

1. B. Kosko, IEEE Trans. Syst. Man. Cybern., vol. SMC-18, no.1, pp49-60, Jan./Feb. 1988.
2. T. Kohonen, SELF-ORGNIZATION AND A ASSOCIATIVE MEMORY. Berlin: springer-verlag, 1984.
3. M. A. Cohen and S.Grossberg, IEEE Trans. Syst. Man. Cybern., vol. SMC-13,pp815-826, Sept/Oct. 1983.
4. J. J. Hopfield, Proc. Nat. Acad. Sci. USA, vol. 79, pp.2554-2558, 1982.
5. Robert J. Mark II, et al., IEEE Trans. Circuits and Systems, vol. CAS-36, no.6, pp.846-857, 1989.
6. -----, ISDL Report, Report A-88.
7. T. N. E. Graeville, SIAM Rev. 2, 15 ,1960.
8. K. F. Cheung, International Symposium on Computer Architecture and Digital Signal Processing, pp. 41-45, 1989, Hong Kong.

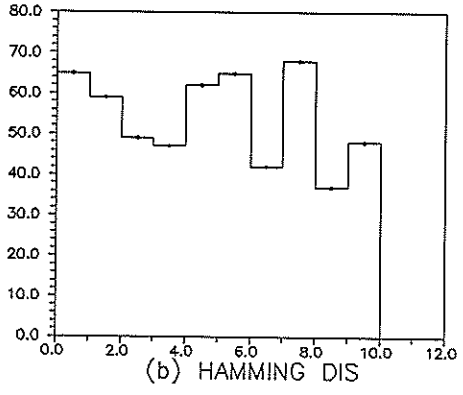


Fig. 2 Ten binary pattern pairs (X, Y) are coded on a (12x10, 14x12) grid. The letter pair (8, I) corrupted by flip noise, is shown in (a). The Hamming distance between this perturbation and each letter is shown in (b), After six iterations in BANN's using Bernoli clamping, the letter pair is restored.

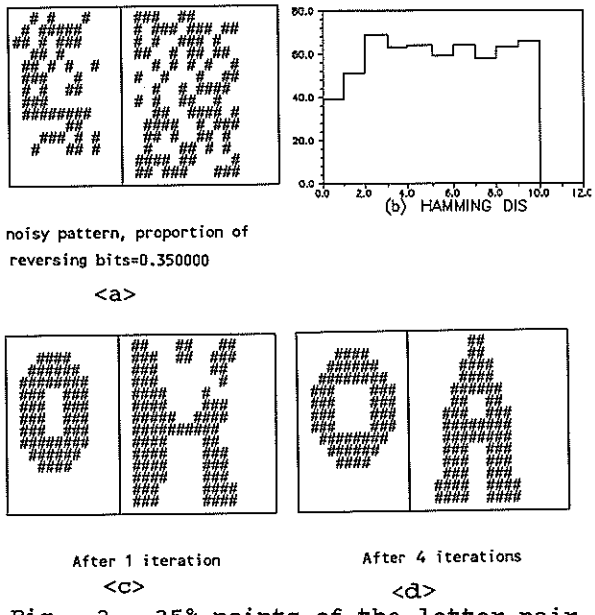


Fig. 3 35% points of the letter pair (0, A) are corrupted by flip noise. After four iterations, correct result is retrieved

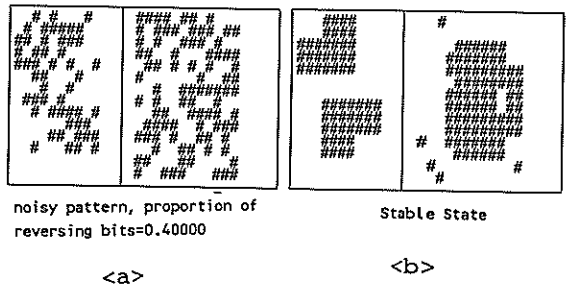


Fig. 4 When 40% point of the pattern (0, A) are corrupted by flip noise, the result is clearly incorrect, because it become impossible to choose clamed points properly.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

THE BACK PROPAGATION USING THE CONJUGATE GRADIENT METHOD.

E.Monte,J.B.Mariño and E. Lleida.
Department of Signal Theory and Communications.
E.T.S.E. Telecomunicació. Apartat 30.002. 08080 Barcelona
SPAIN.

ABSTRACT.

In this paper we present a back propagation algorithm, that does the search of the minima of the cost function by means of a conjugate gradient method. The reason that justifies the use of the new algorithm which has a higher aritmetical complexity, is the fact that the back propagation algorithm has a very slow convergene rate. The underlying idea of the new algorithm is to use more efficiently the data that is presented to the network so that the time that it needs to reach an acceptable perfomance is greatly reduced. In order to update the weights of each unit a gradient calculated by means of the tipical back propagation equations [2] and once this information is known the update of the weights is done by means of the conjugate gradient method [1].

INTRODUCTION.

The algorithm that we present, uses the information about the gradient of the cost function. This information is back propagated to the hidden layers of the network, and then is used to update the weights of each unit by means of the conjugate gradients method. The conjugate gradient method does a gradient serach of the minimum of a function with only the knowledge of the gradient of the function, which is the information that the equations of the back propagation yield.

In the first part of the paper, we present the deduction of the back propagation equations, afterwards in the second part we shall introduce the conjugate gradient method that uses the information about the gradient, that is propagated backwards.

PART I.

First of all we shall present the deduction of the back propagation equations so that we shall have the gradient of the cost function with respect to the weights of each unit. This gradient will be used afterwards in the conjugate gradient method.

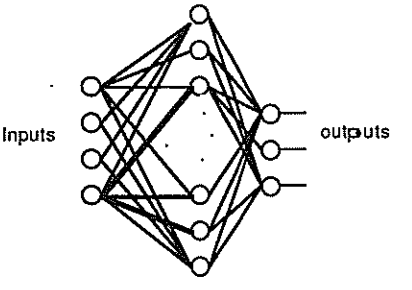
The cost function is defined as follows:

$$E=\frac{1}{2}\sum_{p}\sum_{i}\left(y_{i,p}-ref_{i,p}\right)^2 \qquad (1)$$

Where the meaning of each parameter is:

E : The cost function.
i : Index of the output unit.
p : Index that corresponds to the input pattern.
$y_{i,p}$: Output of the network that corresponds to the "i" output unit, when the "p" pattern is presented.
$ref_{i,p}$: Reference of the unit "i" that corresponds to the "p" patterns.

The arquitecture of the network is the following:



A feedforward neural network.

The update of the equations in the back propagation as described in [2] is the following:

$$w_i(n+1) = w_i(n) + \varepsilon \cdot \frac{\delta E}{\delta w_i} + \alpha \cdot \Delta w_i \quad (2)$$

Where:

w_i : Is the weight "i" of a unit.

$\frac{\delta E}{\delta w_i}$: Is the gradient of the error function E with respect to the weight w_i which is obtained by means of the back propagation equations.

α: Is the momentum term.

ε: Is the adaptation step.

It is important to notice that the back propagation yields at each moment the gradient of the cost funtion in relation to a given weight. This information is quite usefull in order to use the conjugate gradient search, because we have a local information of the gradient which we can use, in order to do the search. One way of doing the search of the minimum of the cost function is to search along conjugate directions. In part II we will present the way of using the conjugate gradient method with the information that we have from the gradient which is obtained by means of the back propagation.

PART II.

The back propagation equations yield the derivative of the cost function with respect to each of the weights of the network, once one has these derivatives the gradient of the cost function can be calculated for each unit. This is the only information that is needed in order to use the conjugate gradient search. In this paper the conjugate gradient method will not be deduced, this is already done in [1]. Instead, we will present directly the algorithm and explain how to connect it with the back propagation algorithm.

ALGORITHM.

Note that in this section the simbols **w,g,d** with the subscript **k** mean a vector, at the iteration number k. So when we use w_0 we mean the vector of weights of a given unit at the moment zero.

0- Firts of all, calculate the gradient of the cost function at each unit by means of the back propagation algorithm. The gradient of the error function is:

(3)

$$\nabla E(w_k) = \left(\frac{\delta E}{\delta w_1}(k), \frac{\delta E}{\delta w_2}(k), \ldots\ldots, \frac{\delta E}{\delta w_N}(k)\right)^T$$

Note that when we use the vector notation, the subscript means time instead of the index of the weight of the unit.

Where:

N : is the number of weights of the unit.

$\nabla E(w_k)$: is the gradient of the cost function at the unit, when we have the weights at the time k.

$\frac{\delta E}{\delta w_2}(k)$: Is the derivative of the cost function with respect to the unit number 2, at the moment k.

1- Once we have the value of the gradient for the first set of weights of a unit (w_0), repeat the following iteration.

$$g_0 = \nabla E(w_0)^T \quad (4)$$

2- then for k=0 to the number of weights of each unit.

$$w_{k+1} = w_k + \alpha_k d_k \quad (5)$$

Use an α that minimizes $E(w_k + \alpha_k d_k)$

$$g_{k+1} = \nabla E(w_k)^T \quad (6)$$

$$d_{k+1} = -g_{k+1} + \beta_k d_k \quad (7)$$

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{g_k^T g_k} \quad (8)$$

If the cost function E> ε then w0=wn and return to step 1.

When we use this algorithm, the search is done by means of orthogonal directions, and thus the algorithm has a better estimation of the true gradient, yielding better results.

CONCLUSION.

In this paper we present a version of the back propagation algorithm that uses a conjugate gradient method to accelerate the convergence of the networks. The new algorithm makes use of the fact that the back propagation algorithm yields the gradient of the cost function with respect to the weights of each unit, which is the information that the conjugate gradient method needs to update each unit.

REFERENCES.

(1) D.Luenberger, "Linear and Nonlinear Programming". Addison-Wesley Publishing Company 1984.

(2) D.Rumelhart, G. Hinton, and the PDP Research Group, "Parallel Distributed Processing: Explorations in the structure of cognition."
Cambridge,MA. Bradford Books. 1986.

This work was supported by the PRONTIC grant nº 105/88

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A PERCEPTRON CONVERGENCE MODEL FOR GAUSSIAN INPUT SIGNALS

John J. Shynk † and Sumit Roy ‡

† Center for Information Processing Research, Department of Electrical and Computer Engineering, University of California, Santa Barbara, CA 93106, USA

‡ Department of Electrical Engineering, University of Pennsylvania, Philadelphia, PA 19104, USA

A convergence analysis of an adaptive algorithm is presented for a single-layer perceptron that has a Gaussian input pattern. It is demonstrated that the stationary points of the algorithm are not unique; they depend on the output variance of the perceptron, which in turn depends on the values chosen for the algorithm parameters. This paper focuses on the convergence properties of the output variance, and it presents computer simulations that support the analysis. We also describe a convenient model for the desired response, which provides an improved understanding of the algorithm, and it leads to some useful analytical results.

1. INTRODUCTION

A single-layer perceptron [1] with N input signals and an output hard limiter is shown in Fig. 1. This structure is also known as an adaptive linear neuron (ADALINE) [2], and it is the simplest feed-forward neural network structure, which corresponds to a single "neuron" element. The perceptron learning algorithm examined in this paper has the following recursive form [3]:

$$W(n+1) = W(n) + 2\mu e(n)X(n) + \alpha[W(n) - W(n-1)] \quad (1)$$

where $W(n)$ is the adaptive weight vector

$$W(n) = [\, w_1(n), \ldots, w_N(n) \,]^T , \quad (2)$$

and $X(n)$ is the corresponding input signal vector

$$X(n) = [\, x_1(n), \ldots, x_N(n) \,]^T \quad (3)$$

The output error is given by $e(n) = d_q(n) - y_q(n)$, where $y_q(n) = \mathrm{sgn}(y(n))$ is the quantized filter output, sgn is the sign function, $y(n) = W^T(n)X(n)$ is the intermediate filter output, and $d_q(n)$ is the (binary-valued) desired response. The step size μ and the momentum constant α, where $\mu > 0$ and $|\alpha| \leq 1$, determine the convergence properties of the algorithm. The component given by $\alpha[W(n) - W(n-1)]$ is the so-called momentum term [4].

By examining the expected value of (1), the stationary points of the algorithm can easily be determined; these will be denoted by W_*. Assuming that μ is sufficiently small so that we can ignore second-order convergence effects [5], the stationary points are specified by the following *orthogonality condition* [3]:

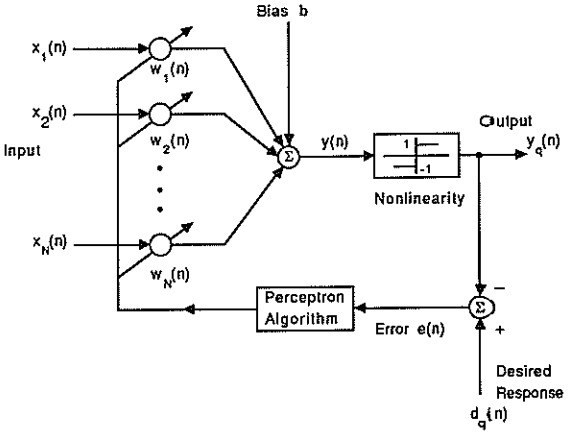


Fig. 1. Single-layer perceptron with hard limiter.

$$E[e_*(n)X(n)] = 0 , \quad (4)$$

where $e_*(n)$ is the output error generated when the weights are at W_*. Substituting the previous expression for the output error, and assuming that $X(n)$ is a Gaussian random vector with zero mean and correlation matrix $R = E[X(n)X^T(n)]$, we can derive the following equation that corresponds to a convergence point of the algorithm [3], [5]:

$$W_* = c\sigma_{y_*}R^{-1}P_q , \quad (5)$$

which can be rewritten as

$$W_* = c\sqrt{W_*^T R W_*}\, R^{-1} P_q \,. \tag{6}$$

The crosscorrelation vector is defined as $P_q = E[X(n)d_q(n)]$, the variance of the intermediate (unquantized) output at convergence is $\sigma_{y_*}^2 = E[y_*^2(n)] = W_*^T R W_*$, and $c = \sqrt{\pi/2}$ is a constant. This expression defines the stationary points of the perceptron algorithm. Observe that (6) is a *nonlinear* function of W_*; because of this form, there are infinitely many solutions. However, if for fixed values of μ and α the convergence point $\sigma_{y_*}^2$ of $\sigma_y^2(n) = E[y^2(n)]$ is unique, then the weight vector W_* will also be unique according to (5). A complete derivation and further discussion of (5) and (6) are given in [3], [5].

2. MODELING THE DESIRED RESPONSE

Although the desired response $d_q(n)$ is constrained to be ± 1, we may view it as being a quantized version of some *underlying* process $d(n)$, i.e., $d_q(n) = \mathrm{sgn}(d(n))$. In general, $d(n)$ is correlated with $X(n)$, and it can often be represented as a function (possibly nonlinear) of the elements of $X(n)$. One interesting case that will be considered here is when $d(n)$ is a linear function of $X(n)$ according to $d(n) = F^T X(n)$, where F is an unknown weight vector defined in a manner similar to $W(n)$. As such, $P_q = P/(c\sigma_d)$, where $P = E[X(n)d(n)] = RF$ is a crosscorrelation vector, and $\sigma_d^2 = F^T R F$ is the variance of $d(n)$. Substituting these expressions into (5), we have that

$$W_* = \frac{\sigma_{y_*}}{\sigma_d} R^{-1} P = \frac{\sigma_{y_*}}{\sigma_d} F \,, \tag{7}$$

i.e., the optimal weights are directly proportional to F, where the proportionality constant is a nonnegative scalar.

For illustration purposes, consider a simple example for $N = 2$. Assume that the bias term b is zero (see Fig. 1), and let the network be trained as follows. The input signals $x_1(n)$ and $x_2(n)$ are independently assigned values from a zero-mean, Gaussian distribution having a variance of 1 (i.e., $R = I$, the identity matrix). If $x_2(n) \geq x_1(n)$, $d_q(n)$ will be set equal to $+1$; otherwise $d_q(n) = -1$. The input samples will be presented to the network, and the perceptron algorithm will adjust the weights according to the time update in (1). For this simple example, it is straightforward to model the underlying process as the following linear combination of the input signals: $d(n) = x_2(n) - x_1(n)$. As such, the desired response model is simply $F = [\,-1 \;\; 1\,]^T$. Since the input samples are assumed to form a Gaussian vector, then $d(n)$ is necessarily a Gaussian process. This example will be used later in the computer simulations.

3. CONVERGENCE OF THE OUTPUT VARIANCE

The output variance of the linear combiner can be written as

$$\sigma_y^2(n, n) = E[W^T(n)X(n)X^T(n)W(n)] \,, \tag{8}$$

where the inner product $y(n) = W^T(n)X(n)$ has been substituted. For notational clarity, we have added a second time argument; together these arguments correspond to those of the weight vector and the input signal vector, respectively, under the expectation. After the weights are updated, the *a posteriori* variance can be expressed as

$$\sigma_y^2(n+1, n) = E[W^T(n+1)X(n)X^T(n)W(n+1)] \,. \tag{9}$$

Substituting the weight recursion from (1), we have that

$$\begin{aligned}\sigma_y^2(n+1, n) = {} & (1+\alpha)^2 E[W^T(n)X(n)X^T(n)W(n)] \\ & + 4\mu(1+\alpha)E[X^T(n)X(n)X^T(n)W(n)e(n)] \\ & + 4\mu^2 E[X^T(n)X(n)X^T(n)X(n)e^2(n)] \\ & - 2\alpha(1+\alpha)E[W^T(n)X(n)X^T(n)W(n-1)] \\ & - 4\mu\alpha E[X^T(n)X(n)X^T(n)W(n-1)e(n)] \\ & + \alpha^2 E[W^T(n-1)X(n)X^T(n)W(n-1)] \,,\end{aligned} \tag{10}$$

which can be written more compactly as

$$\begin{aligned}\sigma_y^2(n+1, n) = {} & (1+\alpha)^2\sigma_y^2(n, n) + 4\mu(1+\alpha)a(n) \\ & + 4\mu^2 b(n) - 2\alpha(1+\alpha)\gamma(n, n-1) \\ & - 4\mu\alpha a(n-1) + \alpha^2\sigma_y^2(n-1, n) \,,\end{aligned} \tag{11}$$

where, for convenience, we have defined the following scalar quantities:

$$\gamma(n, n-1) = E[W^T(n)X(n)X^T(n)W(n-1)] \,, \tag{12a}$$

$$a(n) = E[X^T(n)X(n)X^T(n)W(n)e(n)] \,, \tag{12b}$$

$$b(n) = E[X^T(n)X(n)X^T(n)X(n)e^2(n)] \,. \tag{12c}$$

In contrast to that in (8) and (9), the arguments of γ and a are defined only according to those of W under the expectation, and the argument of b is determined by that of e. To continue, we also need a recursion for $\gamma(n+1, n)$, as follows:

$$\begin{aligned}\gamma(n+1, n) &= E[W^T(n+1)X(n)X^T(n)W(n)] \\ &= (1+\alpha)E[W^T(n)X(n)X^T(n)W(n)] \\ &\quad + 2\mu E[X^T(n)X(n)X^T(n)W(n)e(n)] \\ &\quad - \alpha E[W^T(n)X(n)X^T(n)W(n-1)] \\ &= (1+\alpha)\sigma_y^2(n, n) + 2\mu a(n) - \alpha\gamma(n, n-1) \,,\end{aligned} \tag{13}$$

where again (1) has been substituted.

Near convergence the weights approach W_*, a stationary point, and we have for small μ that $\sigma_y^2(n+1, n) \approx \sigma_y^2(n, n) \approx \sigma_y^2(n-1, n) \to \sigma_{y_*}^2$, $\gamma(n+1, n) \approx \gamma(n, n-1) \to \gamma$, $a(n) \approx a(n-1) \to a$, and $b(n) \to b$, which are all independent of time. Therefore, we can replace (11) and (13) by the follow-

ing coupled pair of *deterministic* equations:

$$\sigma_{y_*}^2 \approx (1 + 2\alpha + 2\alpha^2)\sigma_{y_*}^2 + 4\mu a - 2\alpha(1 + \alpha)\gamma + 4\mu^2 b \quad (14a)$$

$$\gamma \approx (1 + \alpha)\sigma_{y_*}^2 + 2\mu a - \alpha\gamma . \quad (14b)$$

By eliminating the common terms from these two expressions, we have the following condition that defines the output variance *near convergence*:

$$(1 - \alpha)a + \mu b \approx 0 . \quad (15)$$

Notice that this condition depends on the parameters μ and α. By examining a near convergence, we can approximate it as follows:

$$a \approx W_*^T E[X(n)X^T(n)X(n)e_*(n)] = W_*^T S , \quad (16)$$

where W_* is the weight vector in (5), which has been factored from the expectation because we are assuming that the weight fluctuations near convergence are negligible. The vector S is given by $S = E[X(n)X^T(n)X(n)e_*(n)]$. By substituting (5) and (16) into (15) and solving for σ_{y_*}, we find that

$$\sigma_{y_*} \approx \frac{-b\mu}{c(1 - \alpha)\, P_q^T R^{-1} S} = \frac{\mu}{c(1 - \alpha)} k , \quad (17)$$

where we have defined the positive scalar $k = -b/(P_q^T R^{-1} S)$. Substituting (17) into (5), the following expression is obtained, which represents the properties of the perceptron weight vector near convergence:

$$W_* \approx k\left[\frac{\mu}{1 - \alpha}\right] R^{-1} P_q . \quad (18)$$

In general, it is difficult to determine closed-form expressions for P_q, S, b, and thus k. However, we are not so much interested in evaluating (18) as we are in the asymptotic relationship (i.e., near convergence) between the weights and the parameters μ and α. Notice that W_* is a *linear* function of μ; if μ is increased by a factor of 10, for example, then the weight values are also scaled by a factor of 10. On the other hand, W_* depends on α in a *nonlinear* way. Furthermore, it behaves differently for positive and negative values of α. If $\alpha > 0$, then the weights increase as $\alpha \to 1$, becoming extremely large as α approaches 1. However, for $\alpha < 0$, the weights decrease as $\alpha \to -1$, remaining relatively small.

Finally, if we assume that $d(n)$ is generated according to the model F, then (18) simplifies to

$$W_* \approx k'\left[\frac{\mu}{1 - \alpha}\right] F , \quad (19)$$

where (7) has been used and $k' = -b/(F^T S)$ is a positive scalar. Thus, the weights near convergence are proportional to F, as expected.

4. SIMULATION RESULTS

In the simulations presented here, a two-weight perceptron ($N = 2$) was examined with $b = 0$, $R = I$, and $F = [-1 \;\; 1]^T$. As such, $P = [-1 \;\; 1]^T$ and $\sigma_d^2 = 2$. The perceptron was trained starting from the zero weight vector. In all simulations, the weight trajectories were averaged over 100 independent computer runs to generate relatively smooth curves. We considered two cases: (a) one weight was fixed and the other was allowed to adapt, and (b) both weights were allowed to adapt.

Figure 2 shows the weight trajectories of $w_2(n)$ for four values of μ with $\alpha = 0$ and $w_1(n)$ fixed at -1. Since one weight is fixed, the stationary points are unique in this case, corresponding to $w_2(n) \to 1$ (note that we must have $w_2(n) = -w_1(n)$ because $d(n)$ was generated with $F = [-1 \;\; 1]^T$). Observe that $w_2(n)$ converges to 1 as expected, and that the rate of convergence increases as μ is increased. The steady-state weight variance is greater for larger values of μ; this result is similar to that observed for the LMS (least-mean-square) algorithm, and it is a form of misadjustment [6]. Figure 3 shows similar weight trajectories, except μ was kept fixed at 0.01 and α was varied for several positive and negative values. Observe that the rate of convergence increases as α is increased until it becomes unstable at $\alpha = 1$. On the other hand, the rate of convergence decreases as α becomes negative and it is again unstable when $\alpha = -1$. These results suggest that negative values of α would not be used even though the algorithm is stable.

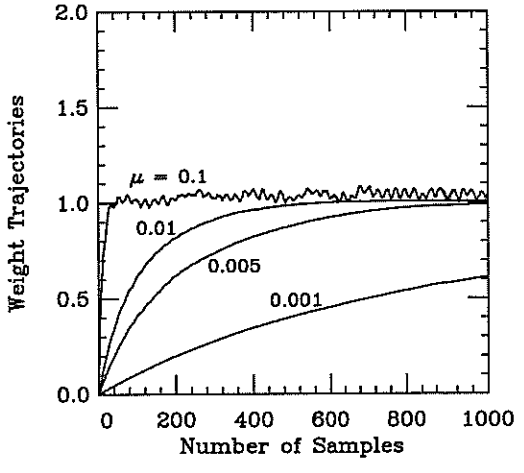


Fig. 2. Trajectories of $w_2(n)$ with $\alpha = 0$ and $w_1(n) = -1$.

Figures 4 and 5 show several weight trajectories of $w_2(n)$ for various values of μ and α, where $w_1(n)$ was also adapted. We show only the trajectories of $w_2(n)$ because we have found that $w_1(n) \approx -w_2(n)$ when we initialize them both to zero. In Fig. 4, observe that the weight trajectories

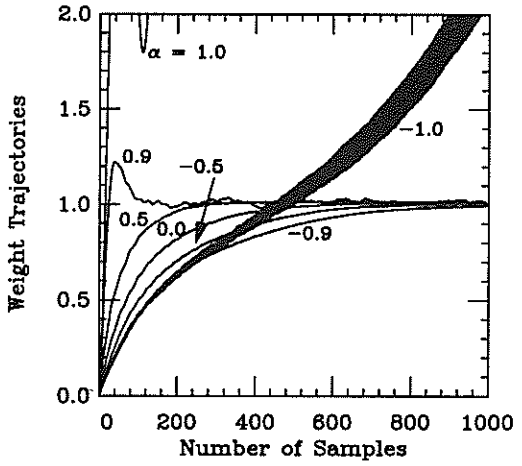


Fig. 3. Trajectories of $w_2(n)$ with $\mu = 0.01$ and $w_1(n) = -1$.

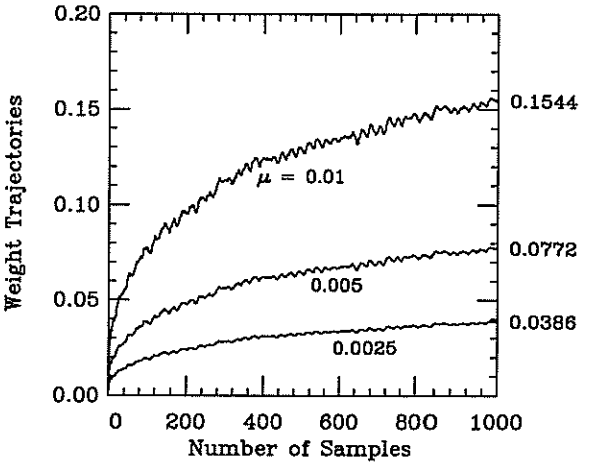


Fig. 4. Trajectories of $w_2(n)$ with $\alpha = 0$.

are directly proportional to changes in the step size μ, as predicted by the analysis, and observe in Fig. 5 that they depend on α in a nonlinear way. (The weight value at iteration 1000 for each curve is shown to the right of the figures.) For a value of $\alpha = 0.5$, $w_2(n)$ should be scaled up by a factor of 2, and this result is verified by the simulation. On the other hand, the weights should be scaled down by a factor of 0.667 for $\alpha = -0.5$; this result is also verified in the simulation. A similar property is evident for $\alpha = \pm 0.3$, and we have observed the relationship predicted by (19) for other values of μ and α.

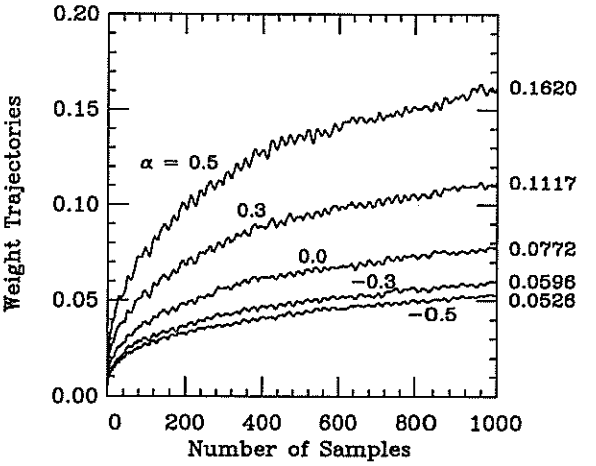


Fig. 5. Trajectories of $w_2(n)$ for $\mu = 0.005$.

5. CONCLUSION

The stationary points and weight trajectories near convergence of a perceptron learning algorithm with momentum updating have been examined for a Gaussian input vector. It was demonstrated that the stationary points are not unique, and that the convergence point of the algorithm depends on the step size μ and the momentum factor α, as well as the statistics of the underlying process $d(n)$. As μ is increased, the weight trajectories increase in direct proportion to changes in μ. On the other hand, the algorithm convergence properties depend on α in a nonlinear way, and it is unstable for $|\alpha| = 1$, as demonstrated by computer simulations.

ACKNOWLEDGMENT

This work was supported by the University of California MICRO Program, Rockwell International, Applied Signal Technology, Inc., and the University of Pennsylvania Faculty Initiation Program.

REFERENCES

[1] R. P. Lippmann, "An introduction to computing with neural nets," *IEEE ASSP Mag.*, vol. 4, pp. 4-22, Apr. 1987.

[2] B. Widrow, R. G. Winter, and R. A. Baxter, "Layered neural nets for pattern recognition," *IEEE Trans. Acoust., Speech, Sig. Proc.*, vol. 36, pp. 1109-1118, July 1988.

[3] J. J. Shynk and S. Roy, "Analysis of a perceptron learning algorithm with momentum updating," in *Proc. IEEE Int. Conf. Acoust., Speech, Sig. Proc.*, Albuquerque, NM, Apr. 1990, pp. 1377-1380.

[4] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, D. E. Rumelhart and J. L. McClelland, Eds. Cambridge, MA: MIT Press, 1986, pp. 318-362.

[5] J. J. Shynk and S. Roy, "Convergence properties and stationary points of a perceptron learning algorithm," *Proc. IEEE*, Special Issue on Neural Networks, vol. 78, Aug. 1990.

[6] B. Widrow and S. D. Stearns, *Adaptive Signal Processing*. Englewood Cliffs, NJ: Prentice-Hall, 1985.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

NONLINEAR PREDICTION OF STOCHASTIC PROCESSES USING NEURAL NETWORKS

Herbert Reininger and Dietrich Wolf

Institut für Angewandte Physik der Universität Frankfurt a. M.
Robert-Mayer-Str. 2-4, 6000 Frankfurt am Main, FRG

A new method for designing nonlinear predictors without explicit knowledge of the underlying statistics arises from the framework of artificial neural networks. In simulation experiments we evaluated the performance of nonlinear prediction based on neural networks on the basis of stochastic processes with different nonlinear statistical dependencies. The results show that neural networks represent a promising approach for realizing nonlinear predictors.

1. INTRODUCTION

The design of an optimum p-th order nonlinear predictor for predicting sample values of a stochastic process with nonlinear dependencies require the knowledge of the (p+1)-dimensional probability density. Even for the calculation of a predictor based on a polynomial approach higher-order moments are necessary. A new method for designing a nonlinear predictor without the explicit knowledge of the underlying statistics arises from the framework of artificial neural networks.

This paper describes the use of neural networks for nonlinear prediction of noisy time series with different nonlinear statistical dependencies. The performances of nonlinear predictors based on neural networks, evaluated in terms of prediction gains, is compared to those of optimum linear predictors and to the maximum values.

2. PREDICTION WITH NEURAL NETWORKS

Artificial neural networks are composed of interconnected nonlinear computational elements, so-called neurons. In a neuron j the inputs $\{x_i\}$ from other neurons weighted by the corresponding connection weights $W = \{w_{ij}\}$ are accumulated into an activation potential

$$h_j = \sum_i w_{ij} x_i + w_{0j} \quad ,$$

where w_{0j} denotes the threshold of neuron j. The output activation of neuron j is in general a nonlinear function $f(h_j)$ of the activation potential. In the specific type of artificial neural network called multilayer perceptron (MLP) only feed-forward connections are used [1]. The neurons in a MLP are divided into three different types of layers. As shown in Fig.1 the input layer consists of those neurons which transmit signals from outside into the net, the neurons whose output activations are observed from outside the net constitute the output layer, intermediate layers between the input and the output layer are called hidden layers. Information propagates in a MLP with L layers only from the input layer $l = 1$ via hidden layers $l = 2, \ldots, L-1$ to the output layer $l = L$.

Assuming that all activation functions $f_j^l(h_j)$ are continuous functions the MLP defines a continuous nonlinear mapping $F: R^{N_1} \rightarrow R^{N_L}$ between input vector $x^1 \in R^{N_1}$ and output vector $x^L \in R^{N_L}$ given by the

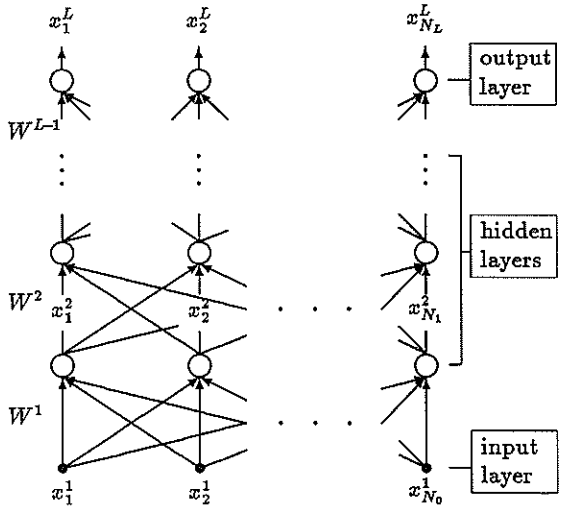


Figure 1: Multilayer Perceptron

recursion

$$x_j^{l+1} = f_j^{l+1}(\sum_{i=1}^{N_l} w_{ij}^l x_i^l + w_{0j}^l) \quad ,j = 1,\ldots,N_{l+1}$$

defined for $l = 1,\ldots,L-1$. It can be shown that any continuous nonlinear mapping can be realized with a MLP containing 2 hidden layers each with a sufficient number of neurons [2].

In order to perform a certain nonlinear mapping the connection weights of a MLP must meet specific values. Given the desired output vector $z \in R^{N_L}$ for a input vector x^1 a gradient descent method called error-back-propagation [3] can be derived for minimizing the quadratic error

$$E = \frac{1}{2}(z - x^L)^2$$

between desired output and actual output of the net. Using the chain rule for calculation of the derivatives $\partial E/\partial w_{ij}^l$ one obtains the weight changes

$$\delta w_{ij}^l = \Delta_j^l \cdot x_i^l \quad , \begin{matrix} i = 1, & \ldots & ,N_l \\ j = 1, & \ldots & ,N_{l+1} \end{matrix}$$

where Δ_j^l for the weights to the output layer is given by

$$\Delta_j^{L-1} = (z_j - x_j^L) \cdot f_j^{L\prime}(h_j^L) \quad ,j = 1,\ldots,N_L,$$

and for the weights of all other layers $l = 1,\ldots,L-2$ by

$$\Delta_j^l = f_j^{l+1\prime}(h_j^{l+1}) \cdot \sum_{k=1}^{N_{l+1}} \Delta_k^{l+1} w_{jk}^{l+1} \quad ,j = 1,\ldots,N_l.$$

The weights are changed in an iteratively using the weight changes multiplied with a small factor η.

A MLP learns to predict an actual vector $x(n)$ from a vector sequence $\{x(n)\}$ given p preceding vectors $x(n-1),\ldots,x(n-p)$, i.e. realizes a predictor of order p, if in the training procedure p preceding vectors are chosen as input and the actual vector as desired output.

3. SIMULATION EXPERIMENTS AND RESULTS

In simulation experiments we evaluated the performance of nonlinear prediction based on MLP using stochastic processes with different nonlinear statistical dependencies. Random number sequences $\{y(n)\}$ with nonlinear dependencies were generated by filtering of statistical independent gaussian distributed random numbers $\{x(n)\}$ with the different nonlinear recursive filters given in the lines 2-4 of Table 1 .

Table 1: Recursive filters for generating random number sequences with nonlinear dependencies

No.	Difference Equation
1	$y(n) = x(n) + 0.9 \cdot y(n-1)$
2	$y(n) = x(n) + 0.21 \cdot y^2(n-1)$
3	$y(n) = x(n) + 3 \cdot \tanh(3 \cdot y(n-1))$
4	$y(n) = x(n) + 3 \cdot \tanh(3 \cdot y(n-1)) + 3 \cdot \sin(y(n-2))$

For comparison also the linear filter of line 1 was included in the simulations. From each of the sequences $\{y(n)\}$ a training sequence consisting of 1000 input/output data vectors was extracted in order to train the weights of MLP with different configurations by means of the error-back-propagation algorithm. For all neurons in the layers $l = 2,\ldots,L-1$ the sigmoid function

$$f(h) = \frac{1}{1+e^{-h}}$$

was used as activation functions, while for the neurons in the output layer $l = L$ linear activation functions were used. The performances of the resulting nonlinear predictors were measured in terms of the prediction gain in dB

$$G_{net} = 10 \cdot \log \frac{\langle x^2(n) \rangle}{\langle [x(n) - \tilde{x}(n)]^2 \rangle} \quad ,$$

where $\tilde{x}(n)$ denotes the prediction of $x(n)$. G_{net} was compared to the values G_{lin} obtained with the optimum linear predictors and also to the maximum achievable values

$$G_{opt} = 10 \cdot \log \frac{\langle y^2(n) \rangle}{\langle x^2(n) \rangle} \quad .$$

The results are shown in Table 2, where the number N_i of input neurons is equivalent to the predictor order. All predictors considered here were scalar predictors, i.e. the number of output neurons is equal to 1. It can be seen that in the case of only linear dependencies in $\{y(n)\}$ the MLP nonlinear predictor (MLPNP) achieves about the same prediction gain as

Table 2: Prediction gains in dB

Eq.	N_i	N_h	G_{lin}	G_{net}	G_{opt}
1	1	1	6.120	6.020	6.120
2	1	1	0.091	0.595	0.826
2	1	2	0.091	0.829	0.826
3	1	1	6.788	9.550	9.550
4	2	8	2.822	9.070	11.577

an optimum linear predictor. Minor differences are due to insufficient training data or number of training iterations. In all other cases the MLPNP outperforms the linear predictor and in some cases it even provides optimal prediction.

The dependency of the performance of an MLPNP on the number N_h of hidden neurons is illustrated in Fig. 2 for equation No. 2 of Table 1. It shows the input data $y(n-1)$ together with the desired output data $y(n)$, i.e. the training sequence, by means of a 2-dimensional point distribution as well as the mappings realized by the corresponding predictors. The straight line resulting from the optimum first order linear predictor obviously is a very poor approximation of the underlying quadratic mapping. A MLPNP with $N_h = 1$ hidden neuron is only able to reconstruct one part of the quadratic map. Thus, it achieves only about one half of the maximum prediction gain. However, $N_h = 2$ hidden neurons are sufficient for the MLPNP to reproduce the quadratic map and therefore to approache G_{opt}. Fig. 3 illustrates how the quadratic map is constructed by the MLPNP with the trained weight matrices

$$W^1 = \begin{pmatrix} 1.22 & 1.42 \\ \cdots\cdots & \cdots\cdots \\ -3.20 & 2.81 \end{pmatrix}$$

$$W^2 = \begin{pmatrix} -2.50 \\ 3.58 \\ \cdots\cdots \\ 2.19 \end{pmatrix}$$

augmented with the threshold values in the last row. The responses of the 2 hidden neurons with sigmoid activation functions are shown in Fig. 3a and 3b. One neuron changes its output with respect to negativ input values, while the other hidden neuron reacts on positive input values. Multiplication with the weights w^2_{11} and w^2_{21} connecting the hidden neurons with the output neuron leads to functions shown in Fig. 3c and 3d. The magnitudes of the weights are greater than 1 leading to a magnification of the output of the hidden neurons and, since w^2_{11} is negative, the output of the first hidden neuron is negated. Due to the linear activation function of the output neuron the curves in Fig. 3c and 3d are added und shifted by the threshold w^2_{01}, which results in the nearly quadratic map of Fig. 3e.

A more complicated example is given by equation No. 4 of Table 1, which is a two-dimensional nonlinear mapping with periodicity in one component. As demonstrated with the 5-level greyscale-plot in Fig. 4 an MLPNP of order $N_i = 2$ with $N_h = 8$ hidden neurons learns the rather complicated mapping from

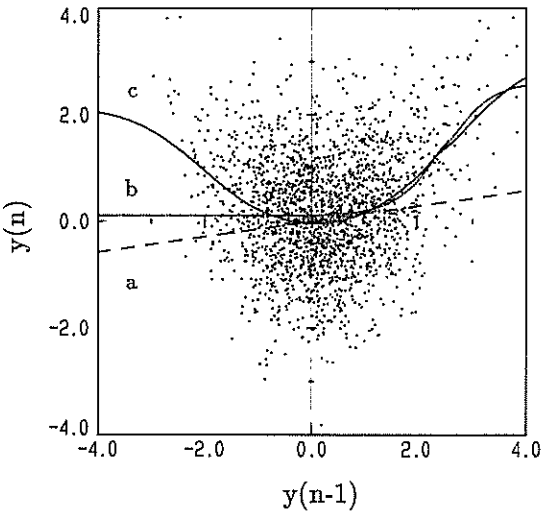


Figure 2: Training sequence and mappings realized by a linear predictor (a), a MLPNP with 1 hidden neuron (b), and a MLPNP with 2 hidden neurons (c)

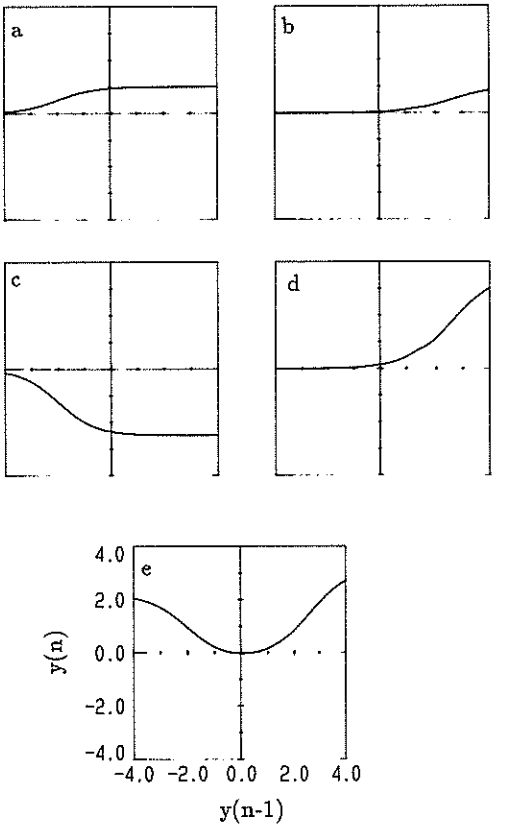


Figure 3: Construction of a quadratic map with a MLPNP

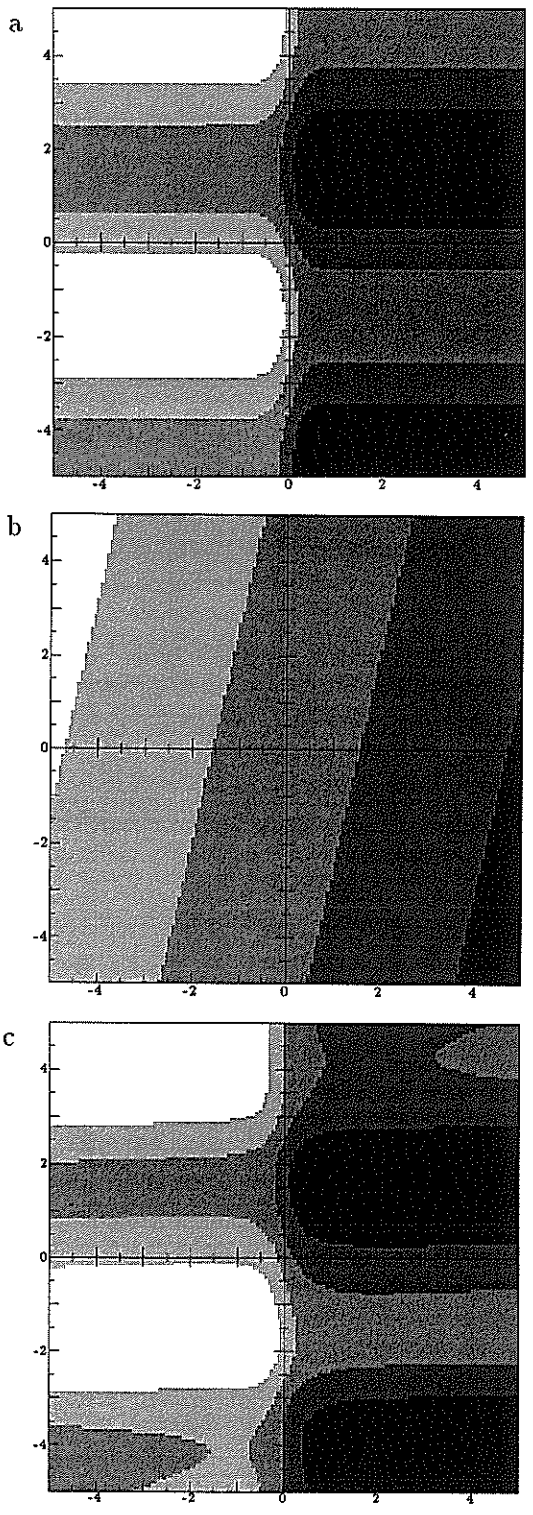


Figure 4: Desired 2-dimensional mapping (a), mapping of a linear predictor (b), and of a MLPNP (c)

noisy sample values using the error-back-propagation algorithm.

Presently, experiments are performed to reveal whether it is possible to realize nonlinear vector prediction with MLP.

4. CONCLUSIONS

The results show that nonlinear predictors can be effectively realized by means of MLP. Furthermore, the nonlinear predictors based on MLP achieve prediction gains close to the maximum values, provided that a sufficient number of neurons is available. This indicates that MLP is able to learn the nonlinear statistical dependencies of stochastic processes from noisy sample values by means of the error-back-propagation algorithm.

REFERENCES

[1] Lippmann, R. P., An Introduction to Computing with Neural Nets, IEEE ASSP Magazine, vol. 4, 1987, pp. 4-22.

[2] Lapedes, A. S., and Farber, R. M., Nonlinear Signal Processing using Neural Networks: Prediction and System Modelling, Los Alamos preprint LA-UR-87-2662, 1987.

[3] Rumelhart, D. E., and McClelland, J. L., Parallel Distributed Processing, Exploration in the Microstructure of Cognition, vol. 1: Foundations, MIT Press, 1986.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

MULTILAYERED PERCEPTRONS FOR NARROWBAND DIRECTION FINDING

D. Goryn & M. Kaveh

Department of Electrical Engineering
University of Minnesota
Minneapolis, Minnesota 55455
U.S.A

ABSTRACT

A technique for using Multilayered Perceptrons in passive narrowband direction finding is presented. The network is trained with a statistic based on received array data. After training the network is subjected to array data and responds almost instantaneously with the source locations present in the received signal. The network is also presented with data corresponding to directions not present in the original training set, in order to study the generalization capabilities.

1. INTRODUCTION

Artificial Neural Networks (ANN's) have recently found a large number of possible applications in a wide range of signal processing areas such as speech processing, image processing, prediction and signal classification problems. A general model of a ANN is a network consisting of a large number of simple processing units (neurons) interconnected with adjustable weights. The network function is determined by the connection topology, the values of the interconnection weights and the function performed at each processing unit. The main advantage of ANN's lies in their inherent parallelism, which should result in high computational capabilities.

This paper addresses the problem of localizing radiating sources using an arbitrarily placed array of sensors. Typical application areas are seismology, sonar and radar. Over the past decades several methods have been studied for the direction finding problem and one of the earliest methods examined was the beamformer. However, a disadvantage with this method is its limited capability to resolve closely spaced sources. In order to improve resolution capabilities many high-resolution techniques have been proposed. Techniques such as maximum-likelihood estimation and signal subspace methods utilizing the eigen structure of the data correlation matrix, are some of the most prominent. A drawback with these approaches is that they depend on computationally expensive algebraic techniques and this can often limit their applicability.

In this paper a novel technique for direction finding using a Multilayered Perceptron (MLP) is presented. The network is trained on a set of array data corresponding to different directions of arrival and different signal-to noise ratios. In order to obtain data reduction for the network training set we use a *statistic* based on the received array data as training data for the network. The desired outputs from the network are the direction of arrival angles. After training, the network is subjected to array data from directions not previously seen by the network in order to evaluate how well the network generalizes. The network is trained using a conjugate gradient algorithm in order to improve convergence rates compared to standard backpropagation algorithms.

This paper is organized as follows. In section 2 we introduce the MLP and the conjugate gradient training algorithm. Section 3 describes the direction finding problem and develops the MLP estimator model. In section 4 simulation results are presented and section 5 summarizes the results.

This work was supported in parts by the SDIO/IST, managed by the Office of Naval Research under contract # N00014-86-k-0410, and the Minnesota Super Computer Institute

In our notation throughout this paper we will denote vectors by bold lower case letters and matrices with bold upper case letters. Transpose by superscript T and conjugate transpose by superscript H.

2. MULTILAYERED PERCEPTRONS

Multilayered Perceptrons form a class of ANN's in which the processing units are arranged in a feed-forward layered structure. The first layer (input layer) receives signals from the environment and propagates them through the network to the last layer (output layer). MLP's can have one or more intermediate layers (hidden layers) between input and output. Each individual processing unit output is produced by computing the inner product of its signal and weight vector. The output is then "fanned-out" to as many as desired other units in the next layer.

The MLP belongs to a class of networks whose information processing function is to approximate a mapping or function. Thus given the network input vector, $\mathbf{x}$, and the network weights, $\mathbf{w}$, the network output can be expressed as $\mathbf{y}=\mathbf{h}(\mathbf{w};\mathbf{x})$. It has recently been shown [1]-[3] that MLP's with as few as one hidden layer, linear output units and sigmoid activation functions at the hidden layer are universal approximators, i.e they are capable of arbitrarily accurate approximation to an an arbitrary mapping, say $\mathbf{f}(\mathbf{x})$, given that there are sufficiently many hidden units available. The desired network mapping is accomplished by adjusting the network connection weights in a supervised fashion until a given error criterion is minimized. Usually this error criterion is taken to be the sum of squared errors

$$E(\mathbf{w}) = \sum_{k=1}^{N} \left\| \mathbf{h}(\mathbf{w};\mathbf{x}_k) - \mathbf{f}(\mathbf{x}_k) \right\|^2 \qquad (1)$$

where N is the total number of given input-output pairs $(\mathbf{x}_k,\mathbf{f}(\mathbf{x}_k))$ presented to the network in the training phase. The minimization problem at hand is an unconstrained non-linear least squares problem. Several training algorithms have recently been introduced and the most frequently used is the backpropagation (BP) algorithm [4], [5]. This algorithm and its momentum version is an iterative technique that finds the weights, $\mathbf{w}$, that minimize (1). It is a gradient descent optimization technique that propagates gradient information from the output layer backwards through the network. The weights are changed proportionally to the negative gradient of the error criterion

$$\Delta \mathbf{w}(k) = -\mu \frac{\partial E(\mathbf{w}(k))}{\partial \mathbf{w}(k)}, \qquad (2)$$

where μ is a fixed step size and k denotes iteration number.

BP as other simple gradient descent methods suffer from a slow rate of convergence. The convergence is linear and this can make the algorithm unattractive for many problems. In this paper we will use a conjugate gradient type of algorithm for the minimization of (1). This type of algorithm takes implicitly into consideration information about second order partial derivatives, i.e the Hessian matrix $\mathbf{H}=\nabla^2 E(\mathbf{w})$, in order to obtain search directions along which substantial decrease in the error criterion can be made. However, it does not require the explicit calculation, storage and update of the Hessian matrix as is necessary with several other higher order methods, e. g. Quasi-Newton methods.
The weight and search vector update equations are given by

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \alpha(k)\mathbf{p}(k+1), \qquad (3)$$

$$\mathbf{p}(k+1) = \mathbf{g}(k) + \beta(k)\mathbf{p}(k), \qquad (4)$$

where $\mathbf{g}(k)=\partial E(\mathbf{w}(k))/\partial \mathbf{w}(k)$ and $\alpha(k)$ and $\beta(k)$ are two scalar parameters. $\mathbf{p}(k)$ is a search direction which satisfies the so called conjugacy property, $\mathbf{p}^T(k)\mathbf{H}\mathbf{p}(j)=0$ for $k\neq j$. The algorithm starts out with $\mathbf{p}(1)=-\mathbf{g}(0)$. There are several choices for the scalar parameter $\beta(k)$ and in this paper we will use the Fletcher-Reeves approach [6], where $\beta(k)$ is given by

$$\beta(k) = \frac{\mathbf{g}(k)^T \mathbf{g}(k)}{\mathbf{g}(k-1)^T \mathbf{g}(k-1)}. \qquad (5)$$

The scalar $\alpha(k)$ is the exact step length to the minimum of $E(\mathbf{w}(k)+\alpha(k)\mathbf{p}(k+1))$ given $\mathbf{w}(k)$ and $\mathbf{p}(k+1)$, and is usually found by performing an line search. The line search part of this algorithm can become costly, since it can involve several function evaluations in order to find $\alpha(k)$. However, we use a modified conjugate gradient technique which uses inexact line searches and modifies the gradient and search vectors obtained by this inexact line search. Further details regarding this algorithm and benchmark test can be found in [7]. The conjugate gradient algorithm

described here offers significant advantages in convergence and speed-up compared to standard BP algorithms.

The network considered in the remainder of this paper will be a three-layer network (one hidden layer), fully connected, with sigmoid activation functions in the hidden layer and linear output units.

3. SENSOR ARRAY PROCESSING

Arrays of sensors are used to localize sources which are radiating energy. The sensor array is used to estimate the direction of the source relative to the location of the array. We will consider an array composed of L sensors with arbitrary locations and assume that d narrowband, uncorrelated, sources located in the far-field and centered around a known frequency, f_0, impinge on the array from locations θ_1, ..., θ_d. The signal vector received by the array can be modelled as

$$\mathbf{q}(t) = \sum_{k=1}^{d} \mathbf{a}(\theta_k) s_k(t) + \mathbf{n}(t) , \qquad (6)$$

where $s_k(t)$ is the received signal from the k^{th} source at a given reference point, $\mathbf{n}(t)$ represents complex valued additive white gaussian noise present at the sensor output and $\mathbf{a}(\theta_k)$ is the (Lx1) steering vector corresponding to the source direction θ_k and given by

$$\mathbf{a}(\theta_k) = [a_1 e^{-j2\pi f_0 \tau_1(\theta_k)}, .., a_L e^{-j2\pi f_0 \tau_L(\theta_k)}]^T ,$$

where $\tau_i(\theta_k)$ is the propagation delay between the reference point and the i^{th} sensor for source direction θ_k and a_i is the corresponding sensor gain. The received vector can also be written as

$$\mathbf{q}(t) = \mathbf{A}(\theta)\mathbf{s}(t) + \mathbf{n}(t) , \qquad (7)$$

where $\mathbf{A}(\theta)$ is the (Lxd) matrix of steering vectors. The second order statistic for our data model is given by

$$\mathbf{R} = \mathbf{A}(\theta)\mathbf{P}\mathbf{A}(\theta)^H + \sigma^2\mathbf{I} , \qquad (8)$$

where $\mathbf{R}$ is the (LxL) spatial correlation matrix, $\mathbf{P}$ is a (dxd) diagonal matrix containing source powers and σ^2 is the noise intensity. We assume that $\mathbf{R}$ is estimated by the sample correlation matrix, $\hat{\mathbf{R}}$, based on T independent received signal vectors,

$$\hat{\mathbf{R}} = \frac{1}{T}\sum_{i=1}^{T} \mathbf{q}(t_i)\mathbf{q}(t_i)^H . \qquad (10)$$

Our aim is to train a three-layer MLP on a pre-determined set of array data corresponding to different combinations of directions of arrival. We will assume that we have a preliminary estimate in which spatial region the sources are located and consider directions from this region as candidates. We will consider the case of two closely spaced radiating sources.impinging on the array and the network training data will consist of the upper triangular part of the estimated sample correlation matrix (10) evaluated at different combinations of directions of arrival. The network input vector can be expressed as

$$\mathbf{x}_k = [\hat{r}_{11}, \mathrm{Re}(\hat{r}_{12}), \mathrm{Im}(\hat{r}_{12}), ..., \hat{r}_{LL}]^T \qquad (11)$$

where $\mathrm{Re}(\cdot)$ and $\mathrm{Im}(\cdot)$ denote the real and imaginary parts of the upper triangular entries of (10). The desired mapping for the network to learn is the mapping between the vector $\mathbf{x}_k$ and the direction of arrival angles, $\theta_k = \mathbf{f}(\mathbf{x}_k)$. During the training phase the network is presented with a set of input-output pairs $(\mathbf{x}_k, \theta_k)$ until a given error critrion is met, $E(\mathbf{w}) < \varepsilon$, where $E(\mathbf{w})$ is given by (1). After training the network is subjected to data from the array and responds almost instantaneously with a estimate of the angles.

4. SIMULATIONS

Case 1

In our simulations we will consider a 4-sensor uniform linear array with sensor spacing, p, of half a wavelength ($\lambda/2$). The first sensor is taken to be the reference point and the propagation delay between the reference point and the i^{th} sensor is given by

$$\tau_i(\theta_k) = \frac{(i-1)p}{\lambda f_0}\sin(\theta_k), \quad i = 1,...,L \qquad (12)$$

The sensor gains, a_i, are assumed to be equal to unity for all signal arrivals. We assume that two sources are located in the spatial region 10^o-25^o. The training is performed on noise-free data and with combinations of different angles in the

spatial region mentioned above. The training set is shown in fig. 1. The input layer size is 16 and there are 24 processing units in the hidden layer. The network is trained until the final sum of squared errors is < 10-3. Fig. 1 also shows the response of the network to the training data after training is completed. We test the performance of the network for noisy data, i.e. for different signal-to-noise ratios. We consider two equi-powered sources at θ_1=13.5° and θ_2=22.5°. The sample correlation matrix is estimated from 500 array snapshots. Fig. 2 and 3 show the variance and bias of the estimated source locations vs. SNR for 100 independent trials. From these simulation results we see that when presented with noisy data the direction estimates tend to be highly biased for low SNR's. In order overcome this problem the network must be trained with noisy data.

Case 2

We now train the network with noisy data under the same conditions as in case 1. We train and evaluate the performance of the network at the following SNR's: 20dB,10dB, 5dB and 0dB. The results are presented in fig. 2 and 3. We notice that the bias in the estimates decreases if we allow for training with noisy-data.

5. CONCLUSIONS

A Multilayered Perceptron architecture was introduced for the direction finding problem. The network was trained (off-line) with a statistic based on array data. After training, the network was subjected to data not present in the original training set and the performance was evaluated. Our preliminary simulation results show that the network has generalization capabilities, i.e. it responds quite well to directions not present in the training set.

REFERENCES

[1] K-I. Funahashi, "On the Approximate Realization of Continous Mappings by Neural Networks", Neural Networks, vol.2, no. 3, 1989.

[2] K. Hornik, M. Stinchcombe, H. White, "Multilayer Feedforward Networks are Universal Approximators.", Neural Networks, vol.2, no. 5, 1989.

[3] G. Cybenko, "Approximation by Superpositions of a Sigmoidal Function.", Math., Control, Signals Sys., vol. 2 1989.

[4] P.J Werbos, "Beyond Regression: New tools for prediction and analysis in the behavioral sciences.", Ph. D. dissertation, Harvard University, 1974.

[5] D.E Rumelhart, J.L McClelland, "Parallel Distributed Processing: Explorations in the Microstructure of Cognition", vol. 1, MIT Press, 1986.

[6] R. Fletcher, "Practical Methods of Optimization", vol.1, John Wiley, 1980.

[7] D. Goryn, M. Kaveh, "Conjugate Gradient Training Algorithms For Multilayered Perceptrons." 32nd MidWest Symposium on Circuits and Systems, Urbana, Illinois, 1989.

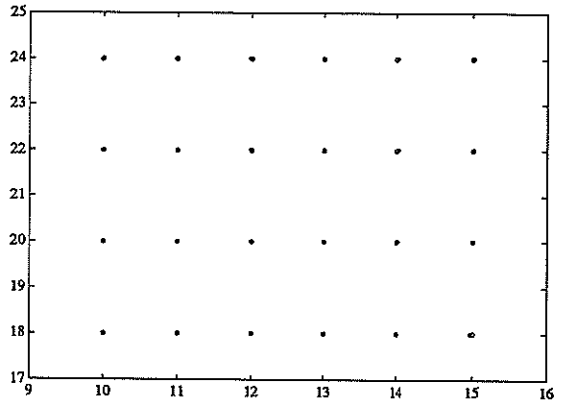


Fig.1 o - Training angles for MLP network.
* - Recalled angles after training (noise free data).

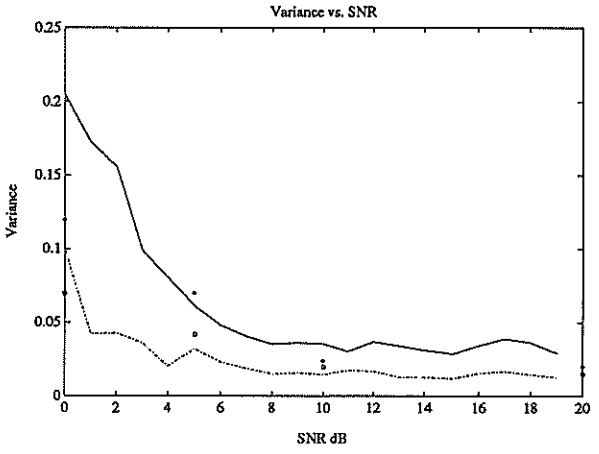


Fig.2 Variance vs. SNR:
----- 22.5° noise free training data, case 1.
-.-.- 13.5° noise free training data, case 1.
* 22.5° noisy training data, case 2.
o 13.5° noisy training data, case 2.

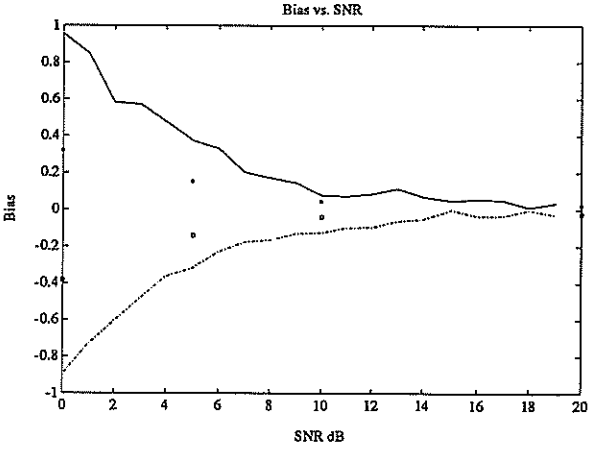


Fig.3 Bias vs. SNR:
----- 22.5° noise free training data, case 1.
-.-.- 13.5° noise free training data, case 1.
* 22.5°, o 13.5° noisy training data, case 2.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SHAPES CLASSIFICATION BASED ON HOMOTHETIC ANALYSIS

A. HOURI and G. MICHEL

Laboratoire ORL/GBM
PCEM Av. J. Vallot Nice 06034 FRANCE

In this paper, a new method for the morphological analysis and classification is proposed. Based on the homothetic ratio between two signals, this is an easy to implement but reliable and fast procedure; it seems to fit well for the study of physiological signals. Thought this algorithm seems to be elementary, it is as efficient as the other mathematical methods.

1. METHODS

The characterisation of shape differences between two signals $S_1(n)$ and $S_2(n)$ is introduced by the homothetic ratios, samples to samples, of the two signals.

$$R_{12}(i) = S_1(i) / S_2(i) \quad i = 1, N \qquad (1)$$

R_{12} defines the set of homothetic ratios of signal S_1 towards S_2.

$$R_{12}(i) = 1 \text{ for all i if } S_1(i) = S_2(i) \qquad (2)$$

The shape similarity is expressed by the number values $R_{12}(i)$ which are included in a reliable interval centered on the value 1. We then deduce a likeness index:

$$C = P/N \quad C <= 1 \text{ and } C = 1 \text{ if } S_1 = S_2$$
$$\text{with} \qquad (3)$$
$$P = \text{Card } \{R_{12}(i)\ i = 1,N \ / \ 1-ß < R_{12} < 1+ß\}$$

The ß parameter fixes the margins on the shape differences between signals S_1 and S_2. If we take ß close to zero we will get a very high discrimination of the two signals.

In order to make a shape analysis of a set of signals, it is necessary to classify them in terms of the likeness indexes. The set of K signals are compared one with the other, giving K*(K-1)/2 indexes C_{ij}, i=1, K-1, j=i+1, K.

We use the algorithm 1 to determine the different classes.

```
l = 1
do i=1, K-1
    if Cl(i) = 0 then          /* element i not assigned */
        Nt(l) = Nt(l) + 1
        Nc(l, Nt(l)) = i       /* element i assigned */
        Cl(i) = l              /*    to class l      */
        do j=i+1, K
            if C_ij > 1-µ then  /* i and j are similar */
                if Cl(j) = 0 then
                    Nt(l) = Nt(l) + 1
                    Nc(l, Nt(l)) = j
                    Cl(j) = l
                endif
            endif
        enddo
    endif
    l = l + 1
enddo
end
```

Algorithm 1

The value of the parameter µ controls the selectivity of the classification. Signals of the same class will be as much similar as the value of µ will be chosen low. In the opposite view, a too high value for µ may induce a single class.

So the sharpness of the shape analysis of the set of signals will be determined by the good choice of the parameters ß and µ. As a general rule, for high signal to noise ratio, the value of ß is taken low (0.01 to 0.1) to get a significant index; depending on µ, the signals will be regrouped in classes of more or less similar shapes.

2. SIMULATIONS AND RESULTS

We have tested this procedure on different kinds of signals. The choice of the test shapes was made according to the classification of the P waves of an electrocardiogram.

We have simulated the P waves with gaussian signals of variable variances. The accuracy of the shapes classification was performed in terms of different signal to noise ratios.

We have then studied the influence of the modulation on the shapes analysis. Indeed, for the treatment of electrocardiogram waves, it is necessary to take account of the action of the respiratory on the shape of the studied signal [1].

The results we obtained are compared with an other method of shapes analysis: the Repartition Function method [2-3] associated to the classification by a modified K-mean algorithm [4].

2.1. Gaussian simulations.

The detection of the phenomena linked to the auricule of the heart, such as the activity of the His bundle, needs efficient algorithms to extract signal from noise. Indeed the recorded His potential at the surface of the body is not directly measurable; a synchrone averaging procedure must be used in order to improve the signal to noise ratio. But it is necessary to sum similar waves to get efficient results.

The temporal relation between the His bundle and the auricular activities being known [5-6], the synchrone averaging procedure could be used on the P waves to extract the His potential from the noise.

We then tested our shapes analysis procedure on simulated P waves (the simulations were based on gaussian waveforms).

Considering the two gaussian waveforms:

$$\text{Type 1: } S_1(i)=\exp(-0.5*(i-128)^2/v_1^2)$$
$$\text{Type 2: } S_2(i)=\exp(-0.5*(i-128)^2/v_2^2) \quad (4)$$
$$i = 1, 256 \quad \text{and } v_2 = v_1*1.15$$

On figure 1 we have represented S1 and S2 for different values of S/N. The lowest value (10 dB) was choosen relative to the signal to noise ratio of a P wave in High Amplification electrocardiography.

On a set of 10 signals made of gaussian waveforms of each type, the discrimination of shapes was perfectly obtained by our algorithm, so far as we were over the minimal S/N of 10 dB already defined. The arbitrary repartition of the signals was:

if we note E the set of the 10 signals and E_i one element i of E:

$$E = \{E_i, i=1, 10\} \text{ with } \begin{matrix} E_i = S_1 \text{ for } i=1, 5 \\ E_i = S_2 \text{ for } i=6, 10 \end{matrix} \quad (5)$$

Table 1 summarises the different cases we tested.

S/B (dB)	Tol. ß	Sel. µ	Nb of classes	Comments (type of classes)
30	0.1	0.1	2	one of type 1 one of type 2
21	0.1	0.4	2	one of type 1 one of type 2
17	0.2	0.3	3	two of type 1 one of type 2
14	0.3	0.2	4	three of type 1 one of type 2

Table 1

The results obtained by the two methods we present are quite identical but the comparative study with the Repartition Fonctions algorithm, which results tend to lower themselves under 15 dB, shows off some important points in favour of the Homothetic analysis:

- gain of computation time as much significant as the number of signals to analyse increases. The Homothetic analysis is based on sample to sample ratios of the signals when the Repartition Function needs to compute the integral of each signal and then the integral of their difference.

- quantification of the difference of shape between two signals by the use of the C index, equation (3), which gives a percentage of similarity between two curves.

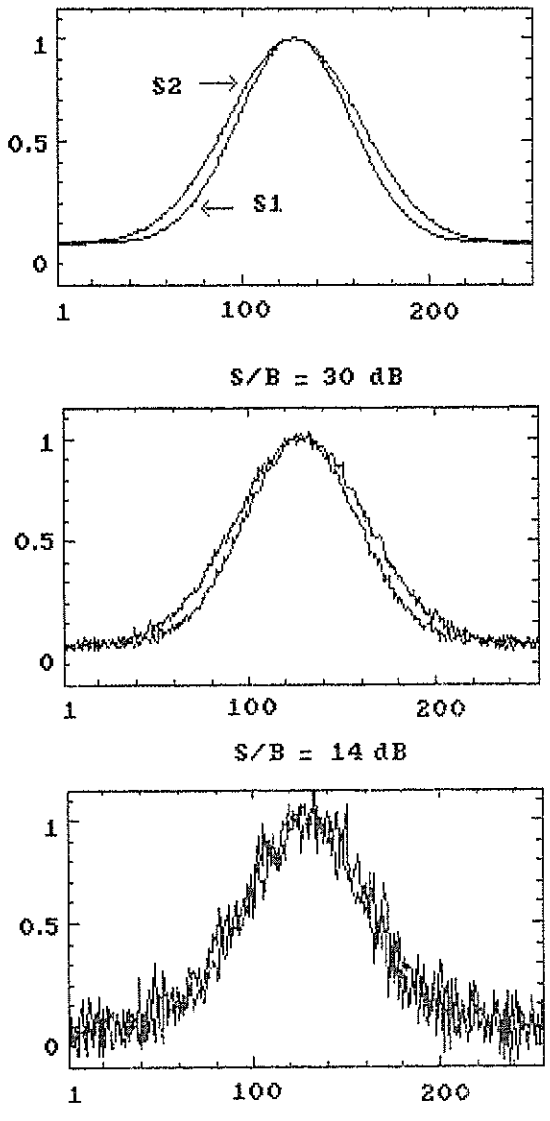


Figure 1: Simulation signals
(arbitrary units)

- a classification result which does not depend on an arbitrary choice of the number of classes as the K-mean algorithm does, but only function of a selectivity test.

2.2. Effect of modulation

The variation of the baseline of an electrocardiogram is in part due to the action of the breathing which can be approximed by an additive modulation. Its influence being not a specific characteristic of the electrical activity of the heart, it is important to verify that it does not modify the results of a morphological classification of the ECG signal.

Thus we used two kinds of simulation signals:

a) one set of identical triangles,

b) one set of triangles of same amplitude but with 10% difference of width.

Each simulation was modulated by an additive sinusoide which period was not proprotionnal to the width of the triangles and which amplitude was either 1/5 of the test signal, for the high modulation case, or 1/15 for the low modulation case.

Both algorithms of shape analysis were tested and the results are presented on tables 2 and 3.

In practical cases, such as the analysis of the P waves of an electrocardiogram, it is very important to notice that the demodulation could not be made through analog filters. The frequencial repartitions of the P waves and of the respiratory modulation being comparatively close, the non linear phase shift due to the demodulation analog filters may introduce artificial modifications of the shape of the P waves and therefore the morphological analysis will not be significant.

Method	Case a	Case b
F.R.	Max and Min of modulation	Max and Min of modulation
Homot.	One classe	Two classes one of each type

Table 2: High Modulation

Method	Case a	Case b
F.R.	Max and Min of modulation	Two classes one of each type
Homot.	One classe	Two classes one of each type

Table 3: Low Modulation

If the modulation amplitude is to high, it is possible to make a partial demodulation before the computation of the homothetic ratios. This demodulation is equivalent to the normalisation of the maximum of the signals. Thus equation (1) becomes:

$$R'_{12} = S_1(i) / S'_2(i) \quad i = 1, N$$
$$\text{with} \quad S'_2(i) = S_2(i) - (S_{M2} - S_{M1}) \qquad (6)$$
$$\text{and} \quad S_{Mj} = \text{Max} \{ S_j(k), k=1, N \}$$

Considering tables 1 and 3, we can establish that the accuracy of our algorithm is absolutly not altered by the action of an interference modulation on the analysed signals. Then this procedure is less dependent on baseline drift and modulation than the F.R. method is.

3. CONCLUSION

We have introduced a new algorithm based on the notion of homothetic ratios between two curves which allows the morphological analysis and classification of a set of signals. Its performances, low computation time, indifference to the variation of the baseline of the signal and relative robustness to noise, give a well adapted tool for the study of physiological waveforms shuch as electrocardiograms.

At the present time, we work on the use of this procedure as a pretreatment to the synchrone averaging of the P waves of High Amplification ECG, with the object of detecting the electrical activity of the His bundle by a noninvasive way.

REFERENCES

[1] A. Houri and G. Michel "Pretreatments by class of synchrone averaging in electrocardiography." Signal Processing IV, EUSIPCO 1988

[2] H. Rix "Detecting small variations in shape." Trans. Syst. Man. Cyber., 1980

[3] S. Jesus "Estimation d'un signal répétitif bruité par sommation synchrone et lissage adaptatif: application à la structure fine du signal cardiaque." Thèse de Docteur en Sciences, Université de Nice Juin 1986

[4] J.T. Tou and R.C. Gonzalez "Pattern recognition principles." Addison Wesley Publish Comp. Massachusetts 1974.

[5] L. Siegel, E.B. Mahoney, J.A. Manning and S. Stewart "Conduction cardiograph bundle of His detector." IEEE vol BME-22 n°4 July 1975

[6] N.C. Flowers, V. Shvartsman, B.M. Kennelly, G.S. Sohi and L.G. Horan "Surface recording of the His Purkinje activity on an every beat basis without digital averaging." Circulation 63 n°4 1981

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

HANDWRITER IDENTIFICATION BASED ON ACCELERATION OF HANDWRITING MOTION

Takenobu Matsuura

Department of Communications Engineering ,Tokai University
Hiratsuka-shi,Kanagawa,259-12, JAPAN

ABSTRACT

This paper discusses a handwriter identification method using impulse response of the finite impulse response (FIR) system characterizing acceleration of cursive script writing motion. The acceleration can be estimated by differentiating twice analytically the handwriting motion approximated by piecewise-linear function. In that case the acceleration components in the horizontal and vertical directions of handwriting motion are regarded as the input and output, respectively, of the system characterizing the accelerations .
With a resulting impulse response the system including the essential information on the individual handwriting motion can be fully described.
This method has an advantage that the difference of handwriting motion can be represented as that of the system characterizing the acceleration of handwriting motion.

1. INTRODUCTION

The methods on the automatic writer recognition have been reported [1,2].
In the first method [1], the timing information is used to extract the features of handwriting. The second method [2] analyzed the handwriting images with a scene analysis method and a nonlinear transformation.

In this paper an online handwriter identification method based on acceleration of handwriting motion for a cursive script is presented. The proposed method recognizes unknown writer by using impulse response of a system characterizing the acceleration in the horizontal and vertical directions of his handwriting motion. It is considered that the impulse response of the system includes the distinct features of handwriting motion for a particular writer. With such an impulse response it is possible to represent the difference of handwriting motion as that of the system characterizing the acceleration of the handwriting motion.

The impulse response can be determined here from the acceleration components in both directions by regarding one component and the other as the input and output of the system, respectively.

2. REPRESENTATION OF HANDWRITING MOTION IN TERMS OF PIECEWISE-LINEAR FUNCTION

The horizontal and vertical components of the handwriting motion are denoted here as $x(t)$ and $y(t)$,respectively. Then the components $x(t)$ and $y(t)$ are normalized here as

$$\tilde{x}(t)=\{x(t) - x_{min}\}/\{x_{max} - x_{min}\} \tag{1}$$

and

$$\tilde{y}(t)=\{y(t) - y_{min}\}/\{y_{max} - y_{min}\} \tag{2}$$

where x_{min} (y_{min}) and x_{max} (y_{max}) are the minimum and maximum values of $x(t)(y(t))$,respectively.

The piecewise-linear function $\hat{x}(t)$ approximating $\tilde{x}(t)$ is defined as

$$\hat{x}(t)= \begin{cases} \hat{x}_1(t)= a_1 t + a_0 & \text{for } t\in[t_0, t_N] \\ \hat{x}_{k+1}(t)= a_{k+1}(t-t_{kN}) + \hat{x}_{kN}(t) & \text{for } t\in[t_{kN}, t_{(k+1)N}] \end{cases} \tag{3}$$

$$(k=1,2,---,L-1)$$

and the coefficients a_i's can be determined in the least square sense:

$$\text{minimize} \sum_{i=kN}^{(k+1)N} \{\tilde{x}(t_i) - \hat{x}(t_i)\}^2. \quad (4)$$

Differentiating $\hat{x}(t)$ in eq.(3) twice analytically with respect to t ,the acceleration in the horizontal direction of the writing motion can be obtained as

$$\hat{x}''(t) = \sum_{i=0}^{L} b_i \delta(t-t_{Ni}) \quad (5)$$

where $b_0 = a_1$, $a_{L+1} = 0$, $b_i = a_{i+1} - a_i$ $(i=1,---,L)$, and $\delta(t)$ is Dirac's delta function.
Similarly, the piecewise-linear function approximating $\tilde{y}(t)$ is defined as

$$\hat{y}(t)= \begin{cases} \hat{y}(t)= c_1 t + c_0 \ , \ t\in[t_0,t_N] \\ \hat{y}_{k+1}(t)= c_{k+1}(t-t_{kN}) + y(t), \\ \qquad t\in[t_{kN},t_{(k+1)N}] \end{cases} \quad (6)$$

Then the acceleration in the vertical direction of the writing motion is given as

$$\hat{y}''(t)= \sum_{i=0}^{L} d_i \delta(t-t_{Ni}) \quad (7)$$

where $d_0=c_1$, $d_{L+1}=0$, $d_i = c_{i+1}-c_i$ $(i=1,---,L)$.

3. IMPULSE RESPONSE OF SYSTEM CHARACTERIZING ACCELERATION OF HANDWRITING MOTION

Fig.1 is a block diagram of an FIR system identification model. $b(n)$ (b_n in eq.(5)) and $d(n)$(d_n in eq.(7)) , which denote the acceleration components in the horizontal and vertical directions of the writing motion,are the input and output sequences of the system describing the acceleration of the writing motion, where n is an integer index. It is desired to model the unknown system as an FIR filter, assumed to be of duration M_{i+1} samples, so that impulse

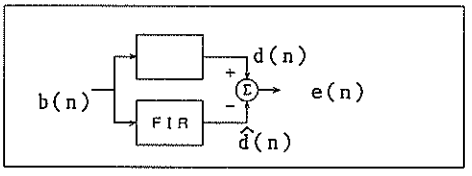


Fig.1

response $h_i(n) = 0$ for n < 0 and $n>M_i$. The order of the FIR system is defined here as the highest index for which $h_i(n)=0$. For the case above, the order is M_i.

The feature extraction of a cursive script writing motion can be performed by regarding the horizontal and vertical components of the acceleration of writing motion as the input and output, respectively, of the identification model. It is assumed that the input and output are of duration of N_L and then system characterizing the acceleration of handwriting motion consists of L FIR system identification models.

The i-th FIR system output estimate is given by

$$\hat{d}(n) = \sum_{m=0}^{M_i} h_i(m)b(n-m) \quad (8)$$

and the resulting estimation error is then

$$e(n) = d(n)- \hat{d}(n) \quad (9)$$

The subscript M_i and $h_i(m)$ denote that these coefficients are the ones calculated for order M_i. The solution to the discrete system identification problem can be obtained by minimizing the least-square error at order M_i,

$$\text{minimize} \quad P_i = \sum_{n=N_{i+1}}^{N_{(i+1)}} \{e(n)\}^2 \quad (10)$$

The optimal solution,

$$h_i = [h_i(0),---,h_i(M_i)]^T \quad (11),$$

can be obtained by the following equation:

$$R_i h_i = g_i \quad (12)$$

where

$$R_i = (r(i)_{km}) \ ;$$

$$r(i)_{km} = \sum_{n=N_{i+1}}^{N(i+1)} b(n-m)b(n-k) \quad (k,m=1,---,M_i) \quad (13)$$

$$g_i = [g(i)_1,---,g(i)_{Mi}]^T;$$

$$g(i)_k = \sum_{n=N_{i+1}}^{N(i+1)} [d(n)b(n-k)] \quad (14)$$

If h_i's (i=1,---,L) are known, then the system is fully characterized, i.e, the

response to an arbitrary input can be found in terms of h_i's (i=1,---,L).

4. SIMILARITY MEASURE

The similarity measure S_{pq} between the genuine writer(p) and the forger(q) is defined here as

$$S_{pq} = \sum_{i=1}^{L} w_i f(h_i^{(p)}, h_i^{(q)}) /L \qquad (15)$$

$$f(h_i^{(p)}, h_i^{(q)}) = \begin{cases} 1; (h_i^{(p)}, h_i^{(q)}) > K_i \\ 0; (h_i^{(p)}, h_i^{(q)}) < K_i \end{cases} \qquad (16)$$

where $h_i^{(p)}$ and $h_i^{(q)}$ are nomalized,
K_i; Threshold value,
L; Total number of FIR system,
w_i; weighting parameter.

5. WRITER IDENTIFICATION ALGORITHM

It is assumed that an impulse response of the system characterizing a standard acceleration of handwriting motion for a genuine writer(p) is estimated in advance. Under the assumption the writer identification algorithm for a cursive script is given below:

(1) Calculate impulse response $h^{(q)}_i$(i=1,---,L) of the system characterizing acceleration of handwriting motion for a cursive script of a writer to be identified.
(2) Calculate the similarity measure S_{pq} between the genuine writer(p) and the forger(q).
(3) The writer is recognized as
(i) the genuine writer if $S_{pq} > K$,
(ii) the forger if $S_{pq} < K$.
(K;a prescribed threshold value)

6. EXPERIMENT

A digitizing graphic tablets was used to enter cursive scripts into a personal computer.
A writer identification experiment involving 7 writers (20 samples each) were attempted. A standard pattern in each class was determined by using 10 samples in the class.

Fig.2 shows a sample of cursive scripts written by a particular person. Fig.3 shows the components , $x(t)$ and $y(t)$,of his handwriting motion for a script "shop".Fig.4 is $\hat{x}(t)$ and $\hat{x}(t)$, the piecewise-linear approximation, of $x(t)$ and $y(t)$.
And the x and y components of the acceleration , $\ddot{x}(t)$ and $\ddot{x}(t)$ are shown in Fig.5 (a) and (b), respectively. The impulse response of the system which yields the acceleration sequences, b(n) and d(n)(n=N_i,---,N_{i+1} ;i=0,1,---,L-1; N_0=0) in Fig.5 as the input and output respectively, were determined.
The respective impulse response is shown in Fig.6.

The experimental results for a cursive script "shop" are shown in Table 1.

Table 1.Error rates

Type I	0 %	5.0 %
Type II	7.5 %	0 %

where L=10,M_i=2,N_i=13,w_i=1(i=1,---,L)
K=0.8, K_i=0.8 (i=1,---,L)

7. CONCLUSION

A handwriter identification method using an impulse response of FIR system characterizing the acceleration of handwriting motion for a cursive script has been proposed.

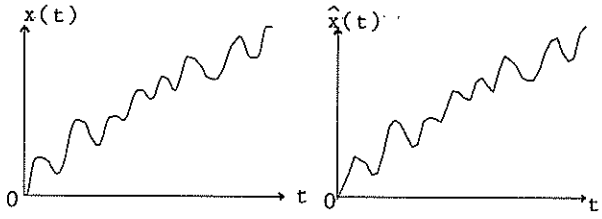
An impulse response of the system describing an acceleration of handwriting motion was used as the features of handwriting . With such an impulse response , the difference of handwriting motion could be represented as that of system characterizing the accelerations.

And the present method should be available for signiture verification.

ACKNOWLEDGEMENT

The author wish to thank K.Ikeda and T.Nakamura for help in accomplishing the experiment.

Fig.2



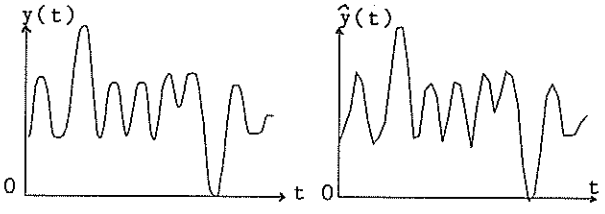


Fig.3

Fig.4

REFERENCES

[1]K.P.Zimmermann and M.J.Varady,"Handwriter identification from one bit quantized pressure patterns", Pattern Recognition Vol.18,No.1,pp.63-72,1985.

[2]K.Steinke,"Recognition of writers by handwriting images",Pattern Recognition Vol.14,Nos.1-6,pp.357-364,1981.

[3]S.L.Marple,Jr."Efficient least squares FIR system identification",IEEE Trans. Vol.ASSP-29No.1,Feb.1981.

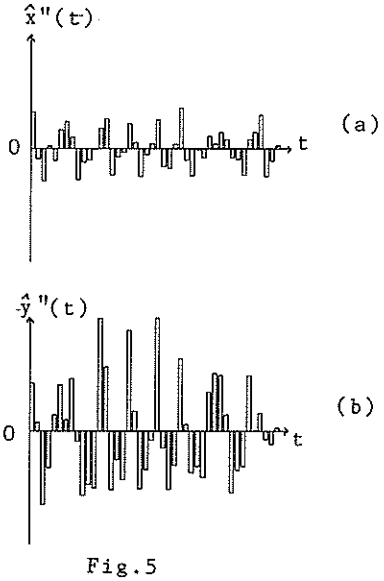


Fig.5

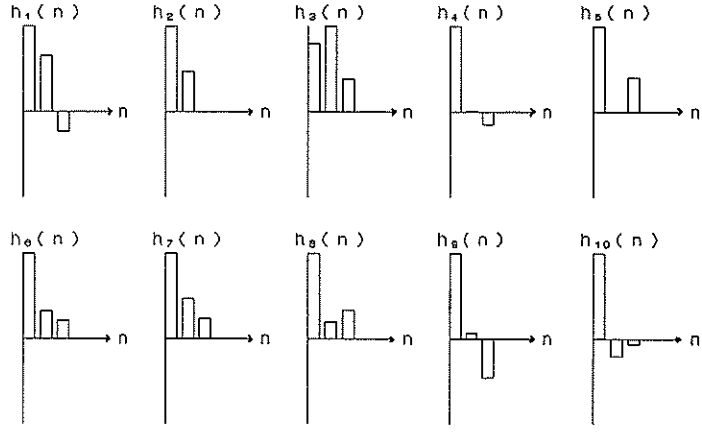


Fig.6

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

An Improved 2D Polar Separable Filter for Texture Analysis

R. C. ZHAO
Dept. of Computer science and Engineering,
Northwestern Polytechnical University,
Xi'an, Shaanxi, China.

J. Kittler, J. Illingworth and I. Ng
Dept. of Electronic and Electrical Engineering,
University of Surrey, Guildford GU2 5XH,
United Kingdom.

Abstract

A 2D Improved Polar Separable Filter (IPSF) is introduced in this paper. The filter is polar seperable in the frequency domain, with the radial dependence based on a Fourier transform of a prolate spheroidal sequence and the orientational depndence given by a quadrature function pair composed of exponential attenuation functions. The new filter is near optimal as its frequency characteristics approximate a 2D Cartesian separable filter composed of prolate spheroidal sequences. The new filter has been successfully used for texture analysis and is shown to be an efficient tool for the estimation of the frequency and orientation parameters of local image texture.

1 Introduction.

Texture analysis using a set of filters distributed over the frequency plane i.e. multichannel texture analysis, has been an active area for research in the last decade [1]-[8]. Gabor filters are most commonly used as they can be easily implemented and have the interesting property that they achieve optimal joint resolution in space and spatial frequency. However, the shape of the Gabor filter may not be optimal for tesselating the frequency plane when strongly directional textures are considered. To achieve effective discrimination of directional textures using Gabor filters requires many narrow bandwidth filters. Knutsson[4] has recently considered this problem and has suggested a filter whose shape is more suited to these situations. This means that directional textures can be discriminated using a small number of wider bandwidth filters. However, Knutssons filter is not optimal in terms of energy loss. In a previous paper[2] we have developed Knuttsons approach further and suggested a QPS (Quadrature Polar Separable) filter which is near optimal in energy loss. The filter consists of two parts. The first part is a Fourier transform of a prolate spheroidal sequence that is dependent on the polar radius. The second part is a cosine function of the polar angle. In the present paper we consider the use of exponential attenuation functions as the orientational weighting function. This modification produces a filter which more closely approximates the 2D Cartesian prolate spheroidal filter, and hence is nearer optimal in energy loss than our previously suggested QPS filter.

Section 2 briefly overviews the design considerations which led to the development of a QPS filter[2]. The new orientational weighting function is then considered in Section 3. Section 4 shows how this function is used to derive a new IPSF. The filter is applied to the texture problem in Section 5 and brief conclusions are offered in Section 6.

2 A QPS filter

The characteristic frequency response of the QPS filter suggested in [2] is composed of two parts. The radial part is a Fourier transform of a prolate spheroidal sequence which is an eigenvector corresponding to maximum eigenvalue of the matrix E, that is a solution of the following equation system

$$(\mathbf{E} - \lambda \mathbf{I})\mathbf{h} = 0, \tag{1}$$

where $\mathbf{h}$ is a vector

$$\mathbf{h} = [h_0, h_1, \ldots, h_{N-1}], \tag{2}$$

and $\mathbf{E}$ is $N \times N$ matrix. After simplification, elements of $\mathbf{E}$ are given by [9]

$$e_{mn} = \begin{cases} \frac{1}{2}m(N-m), & n = m-1 \\ (\frac{N-1}{2} - m)^2 \cos 2\pi\epsilon, & n = m \\ \frac{1}{2}(m+1)(N-1-m), & n = m+1 \\ 0. & |n-m| > 1 \end{cases} \tag{3}$$

Hence the first part of the QPS filter frequency response is just

$$V(\rho) = \psi_0(\rho), \tag{4}$$

$$\psi_0(\rho) = \mathcal{F}[\psi_0(x)]. \tag{5}$$

The second part of the QPS filter frequency response is a quadrature function pair based on a cosine function of the angle i.e.

$$V_e(\varphi) = \cos^{2A}(\varphi - \varphi_k) \tag{6}$$

$$V_o(\varphi) = V_e(\varphi) * \mathrm{sign}[\cos(\varphi - \varphi_k)], \tag{7}$$

where φ_k is the primary phase of the filter i.e. it specifies the orientation of the filter in the frequency domain. A is a parameter which controls the shape of the filter. Choice of A is discussed more fully in [2]. e and o denote even and odd function respectively. Thus, the frequency response of the 2D QPS filter may be represented as follows

$$V_i(\rho,\varphi) = V(\rho)V_i(\varphi), \quad i = e,o \tag{8}$$

3 Orientational weighting function

In this paper we suggest a set of exponential attenuation functions for the orientational part of a QPS filter i.e.

$$V_e(\varphi) = \begin{cases} \exp(-k(\varphi-\varphi_k)^2), & 0 \le \varphi \le \pi+\varphi_k \\ \exp(-k(\varphi-\varphi_k-\pi)^2), & \pi+\varphi_k < \varphi \le 2\pi \end{cases} \tag{9}$$

$$V_o(\varphi) = \begin{cases} \exp(-k(\varphi-\varphi_k-\pi)^2), & \varphi_k \le \varphi \le 2\pi \\ \exp(-k(\varphi-\varphi_k+\pi)^2), & \varphi < \varphi_k \end{cases} \tag{10}$$

where $V_e(\varphi)$ and $V_o(\varphi)$ represent the even and odd functions of the quadrature pair respectively. φ_k is the primary phase and k is the attenuation coefficient which controls the orientational bandwidth of the filter. To get an optimal frequency response so that the energy of the filter in the spatial and frequency domains achieves maximum concentration simultaneously, the filter must be circularly symmetric. This can be done by making function $V_e(\varphi)$ approximate the radial weighting function i.e. a prolate spheroidal sequence which corresponds to the maximum eigenvalue. Zhao et al.[2] give an asymptotic representation of the Fourier transform of the order one prolate spheroidal function as follows

$$\psi_1(c,\omega T/2\omega_c) = (c/2)^{3/4}(\omega T/\omega_c)exp(-c\omega^2T^2/8\omega_c^2), \tag{11}$$

We can obtain an optimal value for coefficient k using a least square method by making function $V_e(\varphi)$ approximate function $\psi_1(\omega)$. The error is taken as

$$\Phi = \Sigma_{i=0}^{N-1}[V_e(\Delta\varphi_i - \pi/2) - \psi_1(\omega_i)]^2, \tag{12}$$

where N is both the number of samples in the radial direction and the number of samples over the angular range $[0,\pi]$. $\Delta\varphi_i$ (i=0,1,...,N-1) is the i^{th} sample of the angle difference and $\psi_1(\omega_i)$ is the i^{th} sample of the first order prolate spheroidal function. If the derivative of the error function Φ with respect to parameter k is set equal to zero then

$$\Sigma_{i=0}^{N-1}[V_e(\Delta\varphi_i-\pi/2)-\psi_1(\omega_i)]*\exp(-k(\Delta\varphi_i-\pi/2)) *(\Delta\varphi_i-\pi/2)] = 0 \tag{13}$$

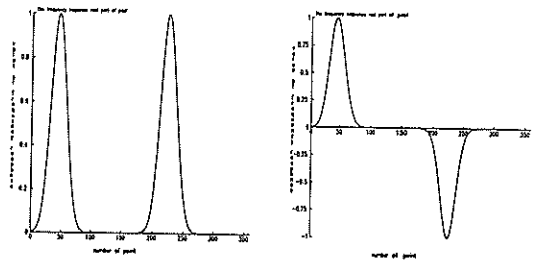


Figure 1: The function plot of $Ve(\varphi)$, $V_o(\varphi)$

In order to obtain a circularly symmetric filter, variables ω and $\Delta\varphi$ can be regarded as the same as each other. Solving equation (13) gives coefficient k. The selection of the parameters, ω_c,T and c, is dependent on the specifications of the filter [2].

Different orientational filters can be obtained by varying the two parameters φ_k and k. The larger the value of k the higher the localization of the filter response in the orientation. Fig. 1 shows the shape of the curves $V_e(\varphi)$ and $V_o(\varphi)$ with $N = 48$, $k = 0.003$ and $\varphi_k = 45°$.

4 A new IPSF

An improved QPS filter is obtained by simply taking the product of functions $V(\rho)$ and $V_i(\varphi)$ (i=e,o) as shown in equations (5),(9) and (10) i.e.

$$V_{Ni}(\rho,\varphi) = V(\rho)V_i(\varphi) \qquad i = e,o. \tag{14}$$

It is apparent that the filter is polar separable and hence its design is quite easy. However to use it in the spatial domain, the inverse Fourier transform of the function $V_{Ni}(\rho,\varphi)$ must be computed and hence it is necessary to change to Cartesian coordinates. Fig. 2 shows the steps to design the impulse response of the new filter.

The new filter has the following main properties:

- It can be designed to have near optimal smallest energy loss, because its frequency characteristic is similar to the 2D cartesian separable prolate spheroidal filter.
- The filter frequency characteristic is easy to adjust because it is only controlled by parameter N, centre frequency ρ_0 and orientation angle ψ_k when parameters c, ω_c or T are given.
- It is easy to design highly orientational filters if large values are used for attenuation coefficient k.

Fig. 3 shows plots of the functions $\psi_1(\rho)$, $V(\rho)$, $V_e(\varphi)$ and $V_o(\varphi)$ with ($\omega_c = 0.125, N = 48$, $k = 0.003, \psi_k = 45°$ and $\rho_0 = 1/5$). Fig. 4 gives a 3D plot of the new

filters.

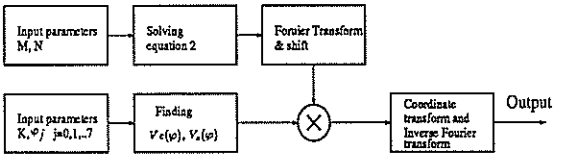


Figure 2: The design steps of the IPSF

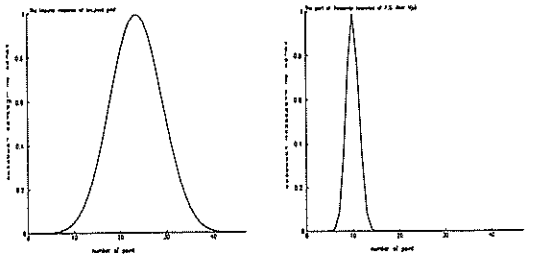


Figure 3: The function plot of $\psi_0(\rho)$ and $V_N(\rho)$. ($\omega_c = 0.125$, $\rho_o = 1/5$ and $N = 48$)

5 Experimental Results

For experimental purposes, a set of synthetic textures was created by filtering uniform noise images with narrow bandwidth IPSF filters. A synthetic texture image which consists of a collage of five such textures is given in fig. 5.

These synthetic texture images were segmented as follows:

- Filter the synthetic texture image using a set of eight filter pairs. This set comprises two distinct groups. Four pairs have $\omega_1 = 0.25$ and $\rho_{10} = 0.20$ while the other four pairs have $\omega_2 = 0.50$ and $\rho_{20} = 0.40$. Within each group the filters are distinguished by primary phase angle. The four values used for φ_k are 0, 45, 90 and 135 degrees.
- For each pixel form a vector whose components are energy values computed from the filter outputs i.e.

$$\mathbf{f}(x,y) = [f_0(x,y), f_1(x,y), f_2(x,y), f_3(x,y)]^T, \quad (15)$$

with

$$\begin{aligned} f_i(x,y) &= ((g(x,y) * v_{Nei}(x,y))^2 + \\ &\quad (g(x,y) * v_{Noi}(x,y))^2)^{\frac{1}{2}}; \\ &\quad i = 0,1,\ldots,7 \end{aligned} \quad (16)$$

where an asterisk denotes convolution.

- Segment the synthetic image using the non parametric classification method proposed by Spann and Wilson[10]. This includes quadtree smoothing, local centroid clustering and boundary estimation.

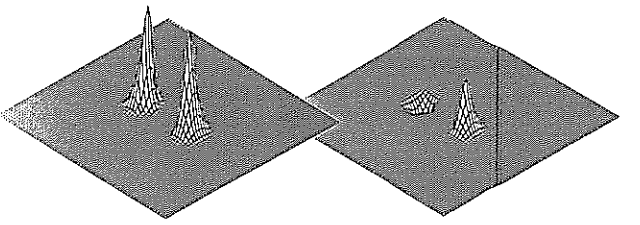


Figure 4: The 3D figures of $V_{Ne}(\rho,\varphi)$, $V_{No}(\rho,\varphi)$.

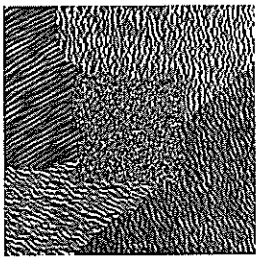


Figure 5: A synthetic texture image which consists of five different directional textures.

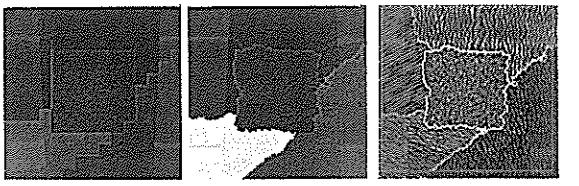


Figure 6: Result of texture segmentation.

The final results of this process are shown in Fig.6. It can be seen that the results are extremely good even though only a few wide bandwidth filters have been used. Further work is in progress to test the filters on real image data.

6 Conclusions

An improved polar separable filter has been proposed in this paper. The suggested filter is near optimal in energy loss, due to the introduction of prolate spheroidal sequences into the design of the filter. The choice of

this angle function ensures filter symmetry. The filter characteristic is easy to control as it is only dependent on the bandwidth and orientation requirements. In addition, the design of the filter is simple and it has low computation cost. It has shown promise for use in texture analysis.

Acknowledgement

The work was partially supported by BP Exploration.

References

[1] L. V. Gool, P. Dewaele, and A. Oosterlinck, "Texture Analysis anno 1983", *Computer Vision, Graphics, and Image Processing*, Vol. 29, 1985, pp. 336-357.

[2] R. C. Zhao, J. Kittler, J. Illingworth and I. Ng, "A New Quadrature Polar Separable Filter and its Application to Texture Analysis", *IEEE Proc. of the Int. Symposium Circuits and System*, May.1-3, 1989.

[3] G. H. Granlund, "Description of Texture Using the General Operator Approach", *Proc. 5th Int. Conf. on Pattern Recognition. Miami Beach, Florida*, Dec. 1-4,1980, pp. 776-780.

[4] H. Knutsson, "Filtering and Reconstruction in Image Processing", *Ph.D. dissertation*, Linkoping Univ., 1982.

[5] M. Kass and A. Witkin, "Analyzing Oriented Patterns", *Computer Vision, Graphics, and Image Processing*, Vol. 37, 1987, pp. 362-385.

[6] A. R. Rao and B. G. Schunck "Computing Oriented Texture Fields", *IEEE Proc. of The Int. Conf. Computer Vision Pattern Recognition, San Diego, CA*, June 4-8, 1989, pp. 61-68.

[7] M. Vetterli, "Multi-dimensional Sub-band Coding: Some Theory and Algorithms", *Signal Processing*, Vol. 6, No 2 1981, pp. 97-113.

[8] A. Ikonomopoulos, and M. Kunt, "High Compression Image Coding via Directional Filtering", *Signal Processing*, Vol. 8, No 2, April 1985, pp. 179-203.

[9] D. Slepian and H. O. Pollak, "Prolate Spheroidal Wavefunction, Fourier Analysis and Uncertainty-I,IV and V", *Bell Syst. Tech.J.*, Vol. 40, 1961, pp. 43-46, Vol. 43, 1964, pp. 3009-3057, and Vol. 57, 1978, pp. 1371-1430.

[10] R. Wilson and M. Spann, "Finite Prolate Spheroidal Sequences and Their Applications I and II: Generation and Properties ", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. PAMI-9, No. 6, 1987, pp. 787–795, and Vol. PAMI-10, No. 2, 1987, pp. 193–203.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SPECTRAL SIGNATURE RECOGNITION WITH A VIEW TO COUNTING ACOUSTIC EVENTS

TROUILHET J-F. - BABANI E. - GUILHOT J-P.

Laboratoire d'Acoustique de Métrologie et d'Instrumentation
38, Rue des 36 ponts 31400 TOULOUSE - FRANCE - Tel: 61 55 65 33

ABSTRACT: This study, which was originally undertaken for a particular case: the recognition of a characteristic sound produced by an animal, in order to make a statistical census of its population, has proved to have more general significance. It has led to the setting up of polyvalent methods of instrumentation taking advantage of the new techniques of signal processing and artificial intelligence.
In this article, the signals to be recognized occur at random times, have noise added to them by the environment, and have characteristics that fluctuate from one observation to another.
After a brief description of the numerical processing software tools suited to the analysis of the signal we are concerned with, we propose, results in hand, a spectral model. This enables us to design and evaluate instrumentation suitable for recognizing the specified signal. We tackle the problem of setting up this instrumentation on an IBM AT type computer equipped with a signal processing card controlled by an ADSP 2100 processor.
We finish by indicating the most encouraging solutions for optimizing the results from this device.

1. INTRODUCTION

This study, initiated by the applied ichthyology laboratory of E.N.S.A.T. is being carried out at L.A.M.I. with the help of a grant from A.N.V.A.R.

Our work [1] comes under inter-university cooperation between L.A.M.I. and the applied ichthyology laboratory of E.N.S.A.T.
Since 1980, this team has been attempting to protect the shad population (migratory fish) of the river Garonne. The most urgent need being a better knowledge of the parameters determining how frequented the spawning-grounds are, a study of these is under way.
Shad spawning is a nocturnal phenomenon which can be detected by the characteristic noise (made by the spawning fish swimming on the surface). During the spawning period (from midnight to four o'clock in the morning during May and June) the signal is recorded at present by a tape recorder at each place where eggs are laid. A research worker then listens to the tape and counts the noises to quantify how frequented the spawning grounds are.
To free research workers from this huge task, our study should allow a system to be made which will carry out the analysis of the tapes in real (or even accelerated) time without the intervention of an operator.

2. ANALYSIS OF THE PHENOMENON AND SPECTRAL CHARACTERISTICS

For an entity to be recognized, it must be compared with a reference which may, or may not, be invariant with time. The choice of this reference depends on the problem under consideration.
In the case of acoustic signatures and, more generally, of analog signals, the most commonly used visual reference is undoubtedly the power spectral density (short term for non- stationary signals). After having experimentally determined the stationarity interval of the signal and the upper frequency limit for the spectrum (80 ms and 5 kHz), we analyzed a large number of recordings of the phenomenon under study.

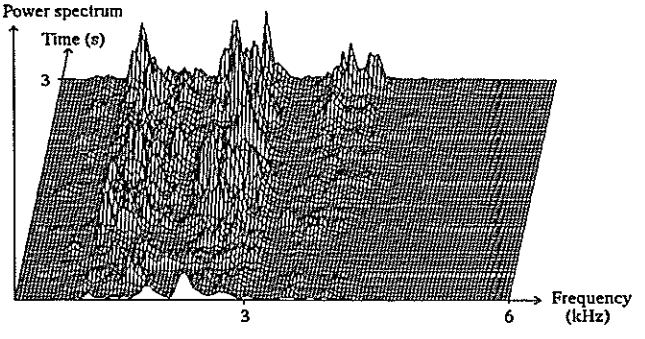


Figure 1

E.N.S.A.T.: Ecole Nationale Supérieure d'Agronomie de Toulouse
A.N.V.A.R.: Agence Nationale pour la Valorisation de la Recherche

We show above the most representative graph, which we have chosen to present the spectral evolution of the signal with time.
This graph was plotted from 63 power spectral density functions, using the modified spectral estimator [2] with a Blackman weighting window.
The principle of this estimator consists of dividing the signal, observed over a length of K samples, into L sections Xl(k) each having a length of M samples, and weighting each section with a window function W(k).
The estimator expression is:

$$R_{xl} = \frac{1}{MP} \left| \sum_{k=0}^{M-1} X_l(k).W(k).e^{-j2\pi fk} \right|^2$$

With l = 1, ... ,L
and

$$P = \frac{1}{M} \sum_{k=0}^{M-1} W^2(k)$$

The bias and variance are given by:

$$E\left[\bar{R}_x(f) \right] = \int_{-1/2}^{+1/2} \Phi x(g).\Phi w(f-g).dg$$

and

$$Var\left[\bar{R}x(f) \right] \approx \frac{1}{L} \Phi^2 x(f)$$

With:

$$\Phi_w(f) = \frac{1}{MP} \left| \sum_{k=0}^{M-1} W(k).e^{-j2\pi fk} \right|^2$$

It can be seen that the bias and variance of this estimator are inversely proportional to the number of samples per section (M= 256 samples, i.e. 21.25 ms) and the number of sections used for calculating the mean (L = 4 sections, i.e. 85 ms) respectively. It should be noted that the dilemma of bias versus variance, a feature of non-parametric estimators, reappears here.

An examination of the time-frequency representations enabled us to demonstrate the distribution of energy in frequency bands. The position of these bands remains the same for the whole duration of the phenomenon, but they may be shifted slightly from one recording to another, depending on the strength of the spawning fish and the current at that particular place. This specificity simplifies the problem as it allows the signal to be considered as stationary, at least from a qualitative point of view. Here are the results obtained for an observation time of 2.7 seconds (K = 32768 samples) using 64 spectra taken from 40.5 ms sections (M = 512 samples) composing the observation time.

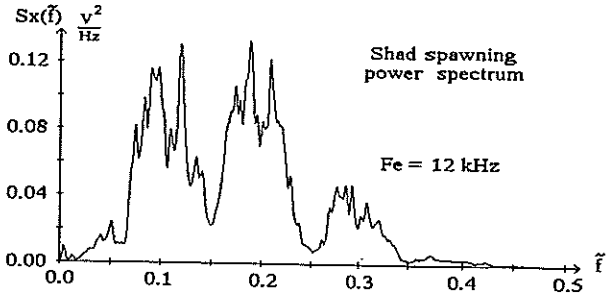


Figure 2

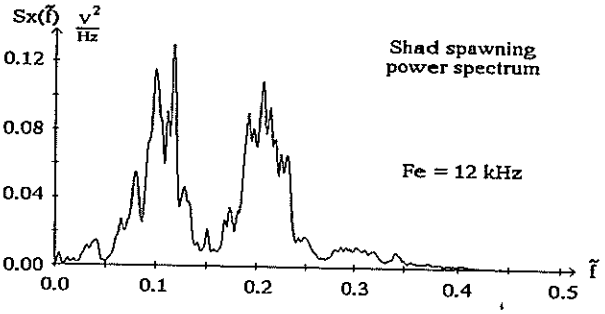


Figure 3

To keep our account short and concise, we have deliberately omitted showing the spectra of the various noises around the spawning ground, i.e.:
- artificial noises:
  lorries, cars, motorcycles, mopeds, trains, etc.
- natural noises:
  birds, crickets, river, wind, rain, etc.

The spectra of all these sounds partly overlap the shad spectrum but certain peculiarities of the energy distribution with frequency make it possible to differentiate them.

3. REAL TIME RECOGNITION

The simplicity of the final system is very important but, in the development phase, we opted for polyvalent, and therefore more complex, instrumentation. We used an IBM AT type computer fitted with a card for numerical processing of the signal, controlled by an ADSP 2100 processor. Tasks were shared as follows: the ADSP 2100 card performed sample acquisition and processing while the IBM AT performed the display and storage of the results.

The chosen solution is shown diagrammatically in the figure below:

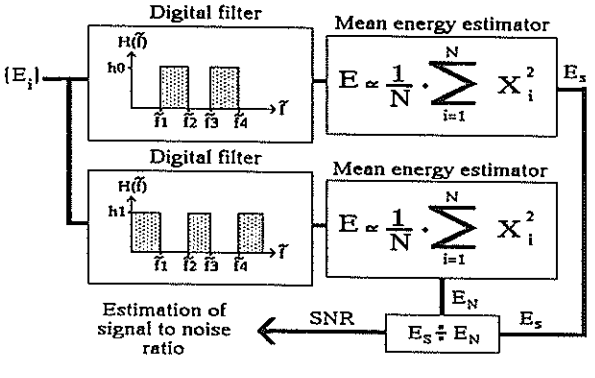


Figure 4

Let ES be the energy (taken to be due to the shad) seen at the output of a filter favouring frequency bands characteristic of the shad spectrum. The solution of estimating the noise energy as the difference between the total energy and ES was rejected because it does not ensure a positive result. We therefore added a filter complementary to the first for noise estimation. It is then possible to calculate:
SNR = 10.LOG(ES) - 10.LOG(EN),
which is the signal to noise ratio as we define it. The system then only needs to be completed by a threshold detector to obtain Boolean information on the presence of the signal.
This solution, although simple, can give results that are not bad at all if care is taken in adjusting the parameters of the device:

- Filter transfer function,
- Number of samples for energy estimation,
- Detection threshold,
- Observation time between decisions.

These parameters were optimized by studying the following 4 probabilities:

- Probability of detecting a shad knowing that there is a shad;
- Probability of not detecting a shad knowing that there is no shad;
- Probability of detecting a shad knowing that there is no shad;
- Probability of not detecting a shad knowing that there is a shad.

Initially, we chose a lattice filter structure. The coefficients were those of the auto-regressive model, characteristic of the shad spectrum, calculated using the Levinson-Durbin algorithm [3]. But since the shad spectrum is relatively flat, the results remained disappointing, and we decided to use a more selective filter (i.e. a multiple band filter).

The filters were synthesized [4] using software developed in our laboratory. From the coefficients of a Tchebycheff digital filter, defined by a template, we obtain the desired coefficients through a low pass to band pass frequency transformation, the expression for which is given below:

$$Z^{-1} \Longrightarrow -\frac{Z^{-2} - \frac{2.\alpha.k}{k+1}.Z^{-1} + \frac{k-1}{k+1}}{\frac{k-1}{k+1}.Z^{-2} - \frac{2.\alpha.k}{k+1}.Z^{-1} + 1}$$

With:

$$\alpha = \frac{\cos\left[\frac{\Omega_h + \Omega_l}{2}\right]}{\cos\left[\frac{\Omega_h - \Omega_l}{2}\right]}$$

and

$$k = \cot\left[\frac{\Omega_h - \Omega_l}{2}\right] \tan\left[\frac{\Omega_c}{2}\right]$$

We chose a second order, cascade, cell type structure for programming the filtering in ADSP 2100 assembly language. The Z transfer function can be written in the form of products:

$$H(Z) = \prod_{i=1}^{N} \frac{a_{0i} + a_{1i}.Z^{-1} + a_{2i}.Z^{-2}}{1 + b_{1i}.Z^{-1} + b_{2i}.Z^{-2}}$$

When the previous transformation is applied, we obtain a transfer function, the order of which has doubled:

$$G(Z) = \prod_{i=1}^{N} \frac{c_{0i} + c_{1i}.Z^{-1} + c_{2i}.Z^{-2} + c_{3i}.Z^{-3} + c_{4i}.Z^{-4}}{d_{0i} + d_{1i}.Z^{-1} + d_{2i}.Z^{-2} + d_{3i}.Z^{-3} + d_{4i}.Z^{-4}}$$

With:

$c_{4i} = \alpha_0.(a_{0i}.\alpha_0 - a_{1i}) + a_{2i}$
$c_{3i} = -\alpha_1.(2.a_{0i}.\alpha_0 - a_{1i}.(1 + \alpha_0) + 2.a_{2i})$
$c_{2i} = (a_{0i} + a_{2i}).(2.\alpha_0 + \alpha_1^2) - a_{1i}.(1 + \alpha_0^2 + \alpha_1^2)$
$c_{1i} = -\alpha_1.(2.a_{0i} - a_{1i}.(1 + \alpha_0) + 2.a_{2i}.\alpha_0)$
$c_{0i} = \alpha_0.(a_{2i}.\alpha_0 - a_{1i}) + a_{0i}$

$d_{4i} = \alpha_0.(\alpha_0 - b_{1i}) + b_{2i}$
$d_{3i} = -\alpha_1.(2.\alpha_0 - b_{1i}.(1 + \alpha_0) + 2.b_{2i})$
$d_{2i} = (1 + b_{2i}).(2.\alpha_0 + \alpha_1^2) - b_{1i}.(1 + \alpha_0^2 + \alpha_1^2)$
$d_{1i} = -\alpha_1.(2 - b_{1i}.(1 + \alpha_0) + 2.b_{2i}.\alpha_0)$
$d_{0i} = \alpha_0.(b_{2i}.\alpha_0 - b_{1i}) + 1$

and

$$\alpha_0 = \frac{k-1}{k+1} \qquad \alpha_1 = \frac{2.\alpha.k}{k+1}$$

It only remains to factorize this ratio and pair the roots of the numerator and denominator to constitute cells of order 2. This improves the performance of the filter whilst minimizing the poles zero distance of each cell.

4. RESULTS

Using recordings on magnetic tape which have already been analyzed by a human expert, we can evaluate the performance of our system by comparing our results with those of the expert.
Performance principally depends on noise level. On sites with little noise, e.g. in the country, the system holds its own against the human expert and it is difficult to say whether it is the human expert or the now expert automat who makes the most mistakes.
Unfortunately, as the noise level rises relative to the signal, the system's success rate falls considerably faster than that of the human expert. It was predictable that a system of reasonable complexity would not be able to match the fine auditory perception of the human being.

5. CONCLUSIONS AND PERSPECTIVES

The current solution has the enormous advantage of being possible using an analog technique. But the low cost of fabrication cannot wholly compensate for the system's poor performance in a noisy environment.
It seems - and we intend to work on this in the near future - that parametric modelling should obviate this problem, at least partially. Indeed the considerations developed above, based purely on the energy, disregard the frequency, all the spectral information being contained in the parameter vector of such a model.

REFERENCES:

[1] J-F. TROUILHET, Etude d'un dispositif de reconnaissance de signature acoustique, Rapport de stage de D.E.A. A.I.I.T.S. 1989, I.N.P.Toulouse.

[2] P.D. WELCH, The use of fast fourier transform for the estimation of power spectra, IEEE Trans. Audio Electroacoustics, vol. AU-15, June 1967, pp70-73.

[3] J. MAX, Méthodes et techniques de traitement du signal, tome II, éditions MASSON, pp140-142.

[4] L.R. RABINER B. GOLD, Theory and application of digital signal processing, Prentice-Hall, pp230-235, pp258-263, pp323-325.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

ADVANCED SIGNAL ANALYSIS AND INTERPRETATION OF QUALITY VARIATIONS IN CROSS DIRECTION OF PAPER MACHINES

Karl HOLMSTRÖM and Risto RITALA

The Finnish Pulp and Paper Research Institute, P.O. Box 70 SF-02151 Espoo, FINLAND

We have developed an analysis system which allows the user to combine powerful basic signal analytic tools into **ad hoc** applications which are specific to the analysis of cross direction variations in paper machines. The analysis environment consists of five modules: the basic signal analysis, the applications, the connection to the data sources, the representation of results and the graphical application generator. The system is UNIX-based, mainly written in LISP using presently the Flavors objects. The system has proved extremely powerful in analysing quality variations in papermaking.

1 Introduction

Papermaking is large scale industrial operation carried out on machines which run typically at speed of 20 m/s and which are up to 9 m wide. The final product is only 50 - 200 μm thick. A typical capital investment in a paper machine is 100 million $ thus setting strict requirements on the runnability of the production line. The uniformity of the quality is extremely important when competing for customers. Thus studying the process and controlling the uniformity are an integral part of the operation of a paper mill and investments in research and equipments supporting these activities are well justified.

The control of variations in the running direction of the paper machine is quite well established. Complicated MIMO control of cross direction (CD) variations has been developed during the last 10 years. However, due to the finite number of actuators (at most 100), the slow scanning of sensors across the web (typically 30 s/scan) and different control actions needed for different quantities significant variations still occur. These variations often reflect false operation or structure of some part of the process. Such variations include those due to malfunctioning CD control, narrow stable streaks and semistable variations either travelling back and forth or appearing and disappearing in a rather irregular manner.

These observations lead us to develop the FLEXPRO system. The system is eventually aimed at process engineers. Presently, the FLEXPRO is used as a service combined to laboratory scans of the CD variations. The main reason for not having on-line applications so far is that the sensors do not have high enough resolution in CD to allow for the study of narrow streaks. This, however, is changing rapidly and we expect on-line applications within a year or two.

2 FLEXPRO analysis system

The specifications for our analysis system of the cross direction variations are as follows:

- collect to the system such basic signal analytic tools which may be of any practical use in the CD analysis; note also that the signal to be studied is non-causal;
- divide the variations into components of different physical origin;
- find out at which stage of the process the variations appear by using the facts that different properties of the web are "formed" at different phases of the process and that there exists information on the CD variations of a given property at different stages;
- because information available and problems vary greatly from process to process make the system flexible and allow *ad hoc* analysis;
- make the system modular in order to allow continuous upgrading in terms of data sources and analysis methods;
- make the system easy to use.

Figure 1 displays the five modules of the FLEXPRO and their communication. The user sees the analysis system entirely through the graphical user interface which governs the whole operation. The data source module connects to the measurements. We prefer connection to a history database or files and do not support on-line data collection. The data source module is easily expandable and thus allows the system to be connected on request to automation systems, to laboratory scanners and so on.

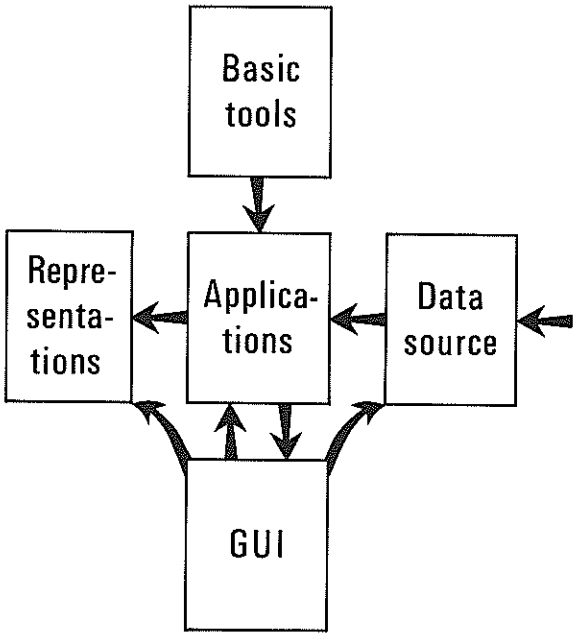


Figure 1. The structure of the FLEXPRO system.

The basic tools and application examples are discussed in Chapter 3. The graphical user interface is described in Section 4.

The results of the analysis will be described as pop-up figures and tables. The representations of results are Flavors objects. Thus the representation module offers, for example, an object-oriented interface to the Starbase graphics.

A part of the data source module is written in C. The rest of the system is in LISP to ensure maximal flexibility. The most time-demanding operations on data may have to be rewritten in C.

3 Basic tools and application examples

The data structures are the following:

- set of measurements; different quantities measured at the same stage of the process or a given quantity measured at the different stages;
- a measurement; a set of scans of a given quantity;
- a profile; a single scan of a quantity;
- a result; divided into sub classes; capable of describing all the results not in form of the other data structures.

The data source provides objects belonging to either set-of-measurements class or a-measurement class. The representation module handles objects either in a-profile class or in a-result class.

All the basic tools are functions belonging to $M_1 \times M_2 \times ...M_k \rightarrow P$, where M_i and P are data structures (the result objects cannot be inputs to a basic tools). Furthermore, the tools may have any number of parameters belonging to the following classes:

- a file name;
- a function name;
- a list of numbers;
- a number;

Thus basic tools are functions with any number of inputs or parameters and which have one output. Tools implemented in the present version of FLEXPRO include linear and nonlinear filters, edge detectors, non-local linear models, causal and non-causal "time" series modelling, data selectors, basic statistical tools and so on. Updating and expanding the basic tools module is simple: new functions are defined in LISP and they are added to the list of available methods (which also defines their functional type) in the graphical user interface module. Note that any analysis which is describable in terms of existing basic tools is not "a basic tool" but "an application" and can be described graphically as explained in Chapter 4.

Applications are collections of basic tools mainly defined to solve a CD variation related problem. These applications serve two purposes: firstly, they are a means to distribute to non experts a process analyst's knowledge on how CD problems and process stability should be analysed; secondly the applications serve as well-documented complicated routine analysis with the possibility to check intermediate results.

We describe the results of two applications. Figure 2 shows the original profile and it's division into four components each with different origin. The first component shows large scale variations mainly due to imperfections in the CD control. The second component consists of variations correlated in CD; these are due to the machine structure. The third component shows variations which do not belong to the first or second component and which are statistically exceptional in the rest. The interpretation of this variation depends on whether the profile is a single scan or an average of several scans. Finally, the fourth component is the rest of the variations and its origin is in the small scale turbulence applied to the pulp slurry during the formation of paper, and also in measurement errors.

The second example shows how the analyst can decide whether the variations originate from a given unit process or are they generated at an earlier stage. Paper is often coated on-line. The coating mass per unit area varies after the coating and papermakers need to know whether the variations are due to the base paper or the coating process. The quantities measured before coating are, for example, mass per unit area and thickness. After coating one measures the total mass per unit area. The stable coating mass per unit area profile is the difference between stable (averaged) mass per unit area profiles after and before coating. The variations in this quantity are first modelled non locally with the base paper properties. As the base paper properties are also dependent on each other we first subtract the part of the thickness variations explainable by the mass per unit area variations in base paper the latter being a quantity formed earlier in process than the former. The variations not explainable by base paper properties are again divided into components.

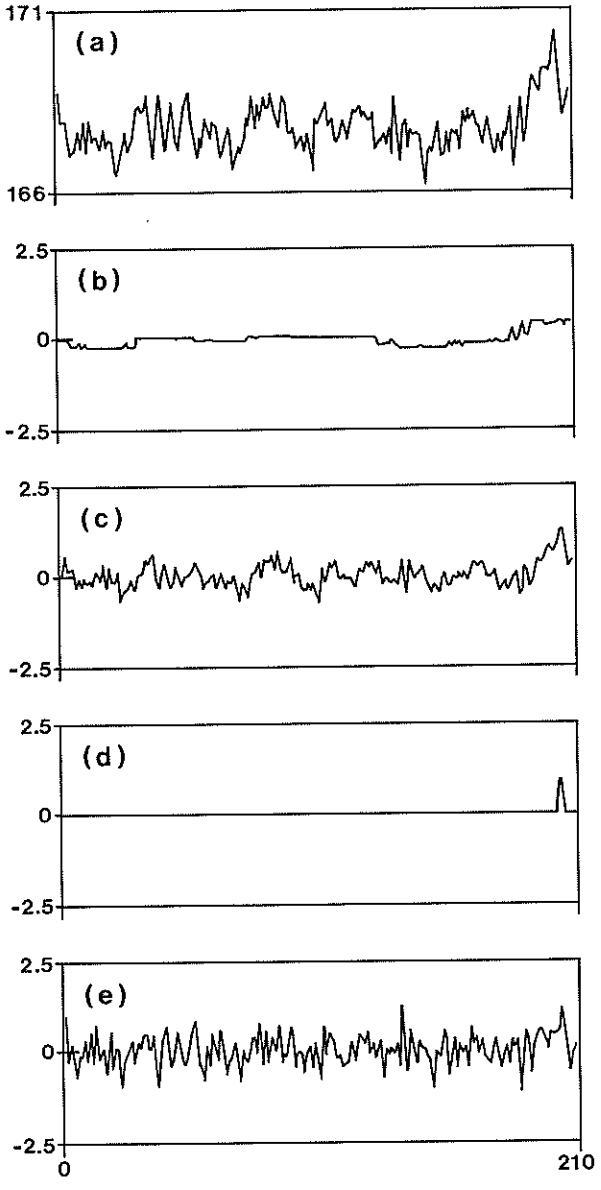


Figure 2. Original profile (a) and the four components as explained in the text.

An example of results is showin in the Table:

Factor of variations	Machine 1	Machine 2
Base, mass per unit area	63 %	22 %
Base, reduced thickness	3 %	2 %
Large scale variations	8 %	43 %
Correlated variations	2 %	7 %
Uncorrelated variations	23 %	20 %

On Machine 1 the efforts should be directed upstream to establish where the variations in the base paper originate from. On Machine 2 the CD control of coating should be examined. Note that the percentages on the Table do not add up to 100 %; this is because the different factors are not strictly uncorrelated.

4 Graphical user interface

The need to build ad hoc applications rapidly was the main reason that lead us to build the FLEXPRO system. The graphical user interface (GUI) combines the construction of new applications to the management and use of existing applications. An application is defined in three steps:

- the user selects the basic tools for the application and places them on a window by using mouse and pop-up menus; at this stage the user may fix the parameters of the basic tools or leave them unfixed so that they will appear as free parameters of the application;
- the user defines the data flows between the basic tools, again by using mouse and menus; the system checks that connections are legitimate i.e. that the output of the tool serving as a source is of the same type as the input of the sink;
- the user chooses the system to create and load the LISP code corresponding to the data flow chart.

The data flow graph can be stored into a file in the form of LISP code for a later recall. The application can also be included amongst the basic tools but then the user looses the possibility to inspect intermediate results and only one output will be available.

When the application is carried out the system asks the user to give the parameters and/or the input data structures. This is carried out using fill-in forms. Then the corresponding function is evaluated. The user has access to all of the outputs of the basic tools belonging to the data flow graph. When the user clicks the tool with mouse and selects the "Representation" option from the menu the system will pop up a window describing the result, either a graph, a table or a number.

Figure 3 displays an example of data flow chart.

The graphical user interface is LISP-based. Each node and connection in the graph is a Flavors object. Each basic tool is a LISP structure containing the information on input, output and parameter types. When the node object, i.e. a realization of a basic tool, is created, it will be sent properties, in fact another strucutre, according to the chosen basic tool structure. The code is

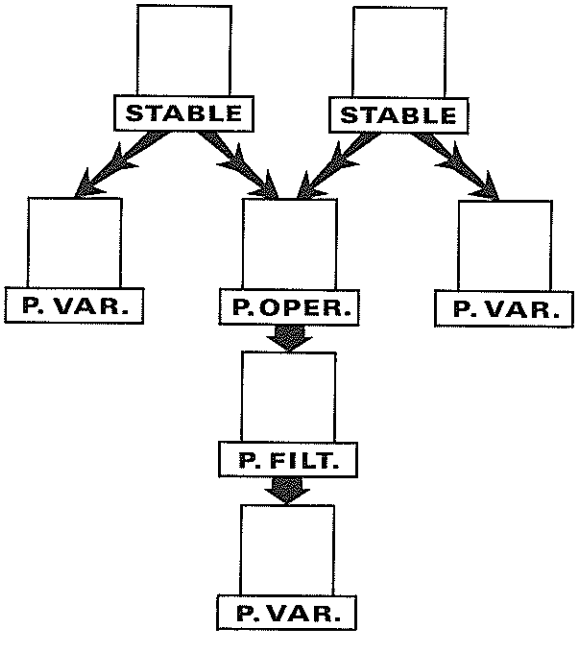


Figure 3. An example of data flow graph used to generate a LISP application and to run it.

generated on the basis of these structures and the list of all connection objects. The code generated consists on send statements which put the output of analysis into the properties of the node objects. Thus the graph of Fig. 3. will produce the following function.

```
(defun analysis (measurement1 measurement2 function)
  (send node1 :set-output
   (stable-profile measurement1))
  (send node2 :set-output
   (stable-profile measurement2))
  (send node3 :set-output (profiles-operated
   (send node1 :get-output)
   (send node2 :get-output)
   function))
  (send node4 :set-output (profile-filtered
   (send node3 :get-output)
   '(1 -2 1)))
  (send node5 :set-output (profile-variance
   (send node1 :get-output)))
  (send node6 :set-output (profile-variance
   (send node2 :get-output)))
  (send node7 :set-output (profile-variance
   (send node4 :get-output)))
```

The parameter in the profile-filtered, the list (1 -2 1) is fixed while the parameter in the profiles-operated is free. The code generator checks that no loops exist in the graph. Recursive functions cannot be generated with the system.

5 Conclusions

The FLEXPRO system is a very powerful tool for both process analysis specialists and process engineers. Modularity makes the system easily expandable and it can be tailored for various purposes. We have chosen LISP as our main programming language because of flexibility and functionality. Graphical user interface offers support and documenting for non specialists as well as fast generation of ad hoc applications for the specialists.

FLEXPRO has been tried out for practical problems for almost a year while still developing new features to it. The system has already paid back all the investments and the research work on developing the system.

FLEXPRO is a rather general purpose system and can thus be applied to problems outside the range of CD variations in papermaking.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

WRITING-ASSISTANCE SYSTEM FOR DISABLED PERSONS IN A MAN-MACHINE COMMUNICATION

Ph. BOISSIERE, D. DOURS

IRIT, Université Paul SABATIER, 118 Route de Narbonne,
31062 Toulouse Cedex, FRANCE

The aim of the VITIPI project is to provide writing assistance to disabled people. Our project consists in constructing a program which automatically provides either the ending of words or some parts of them. In the first part of the talk, we will see how the application is modelled by a transducer and how this transducer is created and minimized. In the second part, we will see how the system can make inferences to deal with new and mis-spelled words. This point is the main point of our talk. In the last part, we will give some results obtained for french vocabularies.

1. INTRODUCTION

The aim of the VITIPI [Boissiere 85b,86,89] project is to provide writing assistance to disabled people. For such cases as tetraplegics,it takes a long time to write a text [Boissiere 85a], even with special devices eventually available [Bonastre 82], [Gabus 86]. For example, it could take about one minute to write from one to eight characters while a typist writes 40 words.

Our project consists in constructing a program which automatically provides either the ending of words or some parts. Indeed the problem of such a software is that it must not be dependant on the lexicon. This is very important because each user refers to a typical vocabulary. Hence it is difficult to make a list of words [Gouguenheim 64], [Catach 84] which are common to all users. Our system has been made for disabled people who make a lot of typing mistakes. Generally, the systems which exist, (like WriteNow) when we are typing words which do not belong to the lexicon or contain a mistake, is blocked and can no longer help to write. To keep the system being blocked, VITIPI adapts itself to all unknown situations. This is why the concept of self-adaptability is central in our work.

We also have to consider that when the system has detected a new word or a mistake in a word and has given a correct answer (i.e the good string of characters) to the user, VITIPI stores the answer for future uses. We can consider that a training has been made.

In the first part of the talk, we will see how the application is modelled by a transducer and how this transducer is created and minimized. In the second part, we will see how the system can make inferences to deal with new and mis-spelled words. In the last part, we will give some results obtained for French vocabularies.

2. VITIPI SYSTEM CREATION

Before talking about transducer creation, we must explain more completely how the system works and give examples in French. When a user starts writing the first letters of a word, the system displays either the end of the word or part of a word. As long as the word remains incomplete, what comes next in the word is provided to the user, the system continuously tries to write as many letters as possible. Whenever a character is entered into the system, VITIPI has to provide and display an output (which can be one or more characters). That is why we have chosen a transducer to model our problem.

2.1. VITIPI modelling

If T denotes the system transducer [Harrison 78] then

$T = <Q, X, Y, \Gamma, \delta, q_0, q_f>$

where

Q is the set of the states of the transducer
X is the input alphabet
Y is the output alphabet
δ is the transition function mapping $Q \times X \rightarrow X$
Γ is the output function mapping $Q \times X \rightarrow Y$
q_0 is the start state
q_f is the set of final states.

example :

Let $Z_1, Z_2, \ldots, Z_5$, five words of French vocabulary (Letters in italics are automatically displayed by the system.)

Z_1 ca*che*
Z_2 ce*rtain*
Z_3 ch*aise*
Z_4 ch*ance*
Z_5 ci*gale*

If we want to write the word Z_3 = ch*aise*, we first have to input the first character *c* which is displayed on the screen. Then the following input *h* is entered by the user and displayed with the letter *a*. We just have to enter *i* which is also displayed with the string *se*.

Theoretically, we can consider that, if Z_i is the word we want to write, X_i ($X_i = x_{i1}.x_{i2} \ldots$) is the input string, Y_i ($Y_i = y_{i1}.y_{i2} \ldots$) is the output string.

For example if i = 3 then
Z_3 = ch*aise*

X_3 = chi ; x_{31} = c, x_{32} = h, x_{33} = i
Y_3 = ch*aise* ; y_{31} = c, y_{32} = h*a*, y_{33} = i*se*.

2.2. VITIPI Creation

The VITIPI creation principle is based on the determination of the X_i and Y_i strings. Thanks to five constraints, bearing on X_i, Y_i, Z_i, δ and Γ, an algorithm has been made which automatically produces the transducer. Algorithm and constraints will be shown in [Boissiere 90].

The transducer thus created will then be minimized but the minimization will not be optimal and will not bring out new strings (or words) other than those present in the training sample (or vocabulary).

Our minimization consists in merging transducer states which are equal.

We say two states q_i and q_j of Q are equal if and only if for all x_k of X such as $\delta(q_i,x_k)$ not indefinite **and** $\delta(q_j,x_k)$ not indefinite

$1° \ \Gamma(q_i,x_k) = \Gamma(q_j,x_k)$

$2° \ \delta(q_i,x_k) = \delta(q_j,x_k)$

Notice that, if similar states are merged, new words will be brought out after minimization procedure, (as shown in [Boissiere 90]).

Remember that two states q_i and q_j of Q are similar if and only if for all x_k of X **either** $\delta(q_i,x_k)$ not indefinite **or** $\delta(q_j,x_k)$ not indefinite. If $\delta(q_i,x_k)$ **and** $\delta(q_j,x_k)$ are definite then

$1° \ \Gamma(q_i,x_k) = \Gamma(q_j,x_k)$

$2° \ \delta(q_i,x_k) = \delta(q_j,x_k)$

Minimization algorithm will be given in [Boissiere 90]

3. HOW VITIPI WORKS

When we provide the system with an input word that is not present in the training vocabulary it cannot normally produce an output. To avoid its being blocked when faced with an unknown word, we have realized a system which allows the transducer to adapt and modify itself automatically in order to provide an output string that satisfies the user. The system will then have to infer a solution.

A word which is not present in the learning vocabulary could either be a new word, or an altered word (misspelled words and/or typing mistakes). In the former case, we have to use a single analogy inference whereas in the latter case we need several string alteration inferences.

When we are faced with a new or altered word, we have to choose "the good" inference. Various strategies are explained at the end of this paragraph.

When we provide VITIPI with a word which is not present in the training vocabulary, the transducer is faced with an undefined transition. How to determine this transition is the central problem of our inference procedures.

The parsing of a new word brings an undefined transition.

Let W = w1.x'.w2 the new word

Hypothèses :

$$\langle q_0,w_1\rangle \xrightarrow{*} \langle q_i,y_i\rangle$$
$\langle q_i , x'\rangle$ --------> undefined

3.1. Analogy inference

The goal is to find the q_j state which is nearest (i.e most similar) to q_i such that $\delta(q_j,x')$ is not undefined.

We have defined a similitude function S between two states.

We have built the state set Qx' such that for all q_j that belong to Qx' the transition $\delta(q_j,x')$ is not undefined.

Let q_j a Qx' state such that

$S(q_i,q_j) = \mathbf{Max}\ \{S(q_i,q_k) \mid q_k \in Qx'\}$

If q_j is such that $\langle q_j,x'\rangle \rightarrow \langle q\ell,y\rangle$,

then $\langle q_i,x'\rangle \rightarrow \langle q\ell,y\rangle$

example :

S is mapping Q x Q into [-1 , +1].

Let : q_i, q_j two states of Q.

Ident : Number of transitions which have the same output **and** the same future state for the same input.

Ident_out : Number of transitions which have the same output **and** not the same future state for the same input.

Diff_out : Number of transitions which have not the same output **and** not the same future state for the same input.

Thus

$$S(q_i,q_j) = \frac{\text{Ident} + \text{Ident\_out} - \text{Diff\_out}}{\text{Ident} + \text{Ident\_out} + \text{Diff\_out}}$$

Application :

If we want to type *directrice* which does not belong to the vocabulary, we start writing *directr* and we are faced with an undefined transition. The nearest state with a defined transition for the input *r* has got rice as output. The *rice* string is inferred and displayed on the screen.

3.2. String alteration inferences

We can find four kinds of word alterations namely : substitution, elision, insertion, orthographic mistakes. An inference procedure has been made for each kind of string alterations.

Substitution

This inference principally deals with typing mistakes. The goal is to find the x_i input which is nearest to x_i such that $\delta(s_j,x_i)$ is not undefined. In he same way of analogy inference, a mathematical distance has been defined between two keyboard characters. The input x_i which is nearest to x' is chosen and the transition is defined.

For instance, if we write the word *trpisième* instead of the French word *troisième* then thanks to our mathematical distance, we find that character p has been substituted to character o (o & p are quite near on the keyboard).

The VITIPI system has been made for disabled people, thus substitution mistakes were carefully treated. Special characters could be substituted to alphabetical letters if the word we are writing is not finished. Typical French stressing (accentuation) mistakes are also taken into account [Boissiere 90]

Elision

The goal is to find which letter has been omitted (i.e. which letter could appear before the x_i input). When an input is entered into the system, VITIPI can produce or not other letters. Depending on the case, we have to search either all possible future states to see if a single transition which has x_i as input exists, or (all the output string of the actual state) to see if there is single transition such that the x_i input occurs in second position.

The first solution can be illustrated by the French word ***quiproquo*** if the following mis spelled word is entered ***quproquo***

We find an undefined transition on the ***p*** input. We are going to search all future states to see if there is a single transition which has ***p*** as input. In the actual state, if ***i*** is entered, a transition leads us to a future state and in this future state there is a transition which has got ***p*** as output. Thus we infer that ***i*** has been omitted.

The second solution can be illustrated by the French word ***trottoir*** if the following mis-spelled word is entered ***trotoir***

We are faced with an undefined transition on the second ***o*** output. In the actual state some of the output transitions produce more than one character. When all outputs are analysed we can see only one of them with letter ***t*** as input has got letter ***o*** in second position. Thus we infer that ***t*** has been omitted.

Insertion

This is the easiest inference procedure because we have to suppose that the character has been inserted so we just have to skip it and wait for the next one. We do not give any example.

3.3. Spelling mistakes

We now have to consider the case when the undefined transition we have got is due to a spelling mistake. This mistake can be due either to an orthographic substitution of the last letter or to a group of letters badly spelled. To make a correction, we have built a small expert system with about 200 French orthographic rules. Two kinds of actions can be started by rules : Substitution of the last input with an equivalent orthographic character, or rewriting a part of a word with an equivalent letter group.

We have partitioned the set of rules such that each rule packet is assigned to a letter. If, in packet of rules, one of them can be applied, the correction is made following the way the rule works. As long as there are letters in the word and no rules can be applied, we go back in the word.

Examples can not be given because they need to understand French rules and predicator rules explanations (it would be too long to explain).

3.4. Strategies

When this transducer is faced with undefined transitions he has got five ways of making inferences; how will it choose?

For some typical cases, the couple formed with the last but one letter and the actual letter can orientate to the "good inference" choice. If the couple never appears in a language (or with a weak probability), we can suppose that the word has been altered. In the same way, if we find the same letter three times, then we can suppose that an insertion has been made and call the insertion inference.

If the couple probability is not weak, we first of all try the analogy inference unless the similitude function is equal to -1. It has been noticed that in this case, the user does not accept the inference solution. String alteration inferences are then orderly proposed to the user depending on various strategies such as the Last Recently Used (L.R.U), the most frequently used or some more sophisticated strategies shown in [Boissiere 90].

If the proposed solution does not satisfy the user, he will tell the system. Depending on the strategy, the system will then propose another solution and consequently modify its choice criterion. Hence the system adapts itself to all unknown situations. If no solution is satisfactory the word is added by the user to the lexicon; a training is made.

4. RESULTS AND CONCLUSION

VITIPI has been checked on French vocabulary. We first of all evaluate the ratio of letter output by the system. A transducer has been built with a 4,236 French word vocabulary.

12,091 letters were displayed by the system
20,307 letters were entered into the system.

The ratio of letters displayed is 37.32%. It is much better than one third which is HUNICUTT'S ratio [Hunicutt 85]. HUNICUTT uses syntactical information

while VITIPI does not. We plan to add this kind of information in our next versions.

Neither in HUNICUTT's work nor in WriteNow word processing do we find the ability to give assistance for unknown words. We do and we evaluate it. A transducer made up with 1,617 French words was created. A minimization of this transducer deleted 1,563 states out of 2,063 (i.e 75.76%). 641 new words were presented to VITIPI and could facilitate the writing of 570 words (i.e. 88.92%) giving 1,979 letters out of 5,939 letters. Thanks to analogy inference 32 letters were also provided.

VITIPI project is not limited to FRENCH vocabulary. Users will not be only disabled people but other people too which are not typists and have to use keyboards.

We do hope that our inference procedures will be applied to other subjects (may be speech recognition).

REFERENCES

[Boissiere 85a] BOISSIERE Ph. : *«Version Intégrant le Traitement Informatique pour les Personnes Invalides»*, Rapport de DEA Laboratoire CERFIA, Juin 85.

[Boissiere 85b] BOISSIERE Ph. : *«V.I.T.I.P.I.»*, Colloque Recherche emploi handicap A.P.I.H.M.S Toulouse 28-29 Novembre 1985.

[Boissiere 86] BOISSIERE Ph. : *«Système d'aide à l'écriture destiné aux handicapés»*, Actes du Congrès INFORSID. Fontevraud, Mai 1986.

[Boissiere 89] BOISSIERE Ph et al : *«Modèle structurel adaptatif pour le dialogue Homme-machine dans les systèmes d'aide aux handicapés»*, Journées d'informatique médicales, Toulouse, Mai 1989.

[Boissiere 90] BOISSIERE Ph. : *«VITIPI : Un système auto-organisationnel pour faciliter le dialogue écrit Homme-machine»*, Thèse à paraître.

[Bonastre 82] BONASTRE J. : *«A la recherche d'une pédagogie spécifique pour les I.M.C»*, Les Cahiers de la fondation Fredrick R. Bull, n°4, Informatique et Handicaps, Décembre 1982.

[Catach 84] CATACH N. : *«Les listes orthographiques de base du Français (LOB)»*, NATHAN 1984.

[Gabus 86] GABUS J.C. : *«La communication chez les personnes sans langage verbal»*, I.M.C. défi n°8 Décembre 1986.

[Gougenheim 64] GOUGENHEIM G. et al. : *«L'élaboration du français fondamental»*, Edition DIDIER 1964.

[Harrison 78] HARRISON M.A. : *«Introduction to formal language theory»*, ADDISON WESLEY Series in Computer Science, 1978.

[Hunnicutt 85] HUNNICUTT S. : *«A lexical prediction for a text-to-speech system»*, Rapport du Dept. of speech communication, Stockholm STL-QSPR 1985.

[Perez 88] PEREZ J.Cl. : *«De nouvelles voies vers l'intelligence artificielle : Pluri-disciplinarité, Auto-organisation, Réseaux neuronaux»*, MASSON Editeur, 1988.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A COMPLETE AND STABLE SET OF FOURIER DESCRIPTORS OF 2D SHAPES FOR INVARIANT ANALYSIS AND RECONSTRUCTION OF 3D OBJECTS

BURDIN V., GHORBEL F., de BOUGRENET de la TOCNAYE J.L., ROUX C.

Groupe Traitement d'Images
Département Mathématiques et Systèmes de Communication
E.N.S.T. de Bretagne BP 832 29285 BREST Cédex France

Abstract: A general theoretical framework for the description of 3D structure is stated in this paper. The invariance of the description under some elementary 3D geometrical transformations is closely related to the invariance properties of a complete and stable set of Fourier Descriptors of 2D closed contours. Application to long bone structures allows a cylindrical parametrisation to be performed on 3D objects. We present a method which gives a 3D shape segmentation and a 3D reconstruction from compressed data.

1. INTRODUCTION

3D contour representation and analysis are problems of great importance in fields such as robotic vision or computer graphics. Some properties like geometrical invariancy are often required, which are well known to be non trivial in the 3D case with respect to 2D invariant shape analysis. Fortunately the analysis is generally improved by taking into account special considerations resulting from the applications required. It is obvious for instance that (arm, leg) bones can be roughly seen as cylinders, therefore introducing considerable simplifications and possible solutions to the above conjecture.
Shape analysis can therefore be performed following diverse approaches. Either prior or learned *primitives* exist or no knowledge about the shape is assumed, like here where one of the aims is to define precisely what the bone shape primitives are. In the case where the contour plays a major part (here closed slices resulting from scanner sections), Fourier descriptors are known to be well suited to shape representation, including invariant properties because of algorithmic simplicity and data compression efficiency. The aim of the work is to show that the problem of 3D bone shape representation (including a wide variety of bones) can be reduced to the standard 2D case using Fourier descriptors which are endowed with complete 3D invariant properties.
Furthermore, the problem of pattern analysis generally goes beyond the problem of pattern reconstruction and synthesis, using for instance, truncated descriptor developments, leading us directly to the definition of shape *primitives*. This is permissible because the proposed descriptors are endowed here with mathematical properties of stability and completeness.

2.SURFACE SIMILARITY INVARIANT DESCRIPTION

The method consists in the description of 3D anatomical structures (e.g. ulna or radius bone). We have a set of 2D longitudinal scanner sections and we reorganize it to obtain a set of cross-sectional slices which are more suitable to the theoretical problem described below. These grey-level images are processed to extract the contour of each slice [1]. We perform successively segmentation in three areas (air, bone, medulla) [2], contour extraction of two areas, bone and medulla, using morphological filters and a storage of boundary point coordinates in order. Finally we interpolate on boundary points to obtain a parametrisation with a normalized arc length [3].

Therefore, a surface in a 3D space can be written in the following parametric form :

$$\begin{aligned} x &= f(s,t) \\ y &= g(s,t) \\ z &= h(s,t) \end{aligned} \qquad (1)$$

The choice of cylindrical parametric form seems to be natural here because of the morphology of the shape, thus equations (1) become:

$$\begin{aligned} x &= \rho(s,t)\cos[\theta(s,t)] \\ y &= \rho(s,t)\sin[\theta(s,t)] \\ z &= t \end{aligned} \qquad (2)$$

where $\rho(s,t)$ is a radius function which measures the length of the line connecting the boundary of the slice on the plane $\{z = t\}$ to its centroid, $\theta(s,t)$ is a polar angular function in the same plane, and s is the arc length of the 2D curve. Such a shape parametrisation enables us to give a reduced representation which is directly related to 2D shape information. A slice is here represented by its boundary which is itself described by the planar closed curve expressed by :

$$\begin{aligned} x &= \rho(s,t)\cos[\theta(s,t)] \\ y &= \rho(s,t)\sin[\theta(s,t)] \end{aligned} \qquad (3)$$

The radius function, in the case of the bone is sufficient to describe the contour because the sign function c(s,t) from which the $\theta(s,t)$ angular function can be deduced [4] is constant. Fourier Descriptors (FD's) are then obtained from this representation [5].

$$c_k(t) = \frac{1}{L(t)} \int_{\{z=t\}} \mathrm{Log}[\rho(s,t)] e^{\frac{2i\pi ks}{L(t)}} ds \qquad (4)$$

where L(t) is the length of the curve when z = t.

Invariancy in relation to rotation, scale, translation and starting point will be achieved by considering the combination of the FD's [4] and [6], and is given as :

$$I_k(t) = \frac{c_k(t)}{c_1^k(t)} |c_1(t)|^{k+1} \qquad (5)$$

The set $\{I_k(t), k \neq 0\}$ is shown to make up a complete and stable set of invariant features to represent the 2D contour [6] which is, in this case, the 2D slice of the bone at z = t.

Note: because the bone can be introduced into the scanner from both its extremities, the set $\{I_k(t)\}$ becomes $\{I_k^*(L-t)\}$ where L is the length of the longest axis of the bone and I^* is the conjuguate complex of I. For reasons of invariancy, we consider the two following sets :

$$J_k^1(t) = A_k(t) + i|B_k(t)| \qquad (6)$$
$$J_k^2(t) = |C_k(t)| + i|D_k(t)|$$

with :

$$A_k(t) = \mathrm{Re}\{I_k(t) + I_k(L-t)\}$$
$$B_k(t) = \mathrm{Im}\{I_k(t) - I_k(L-t)\}$$

$$C_k(t) = \mathrm{Re}\{I_k(t) - I_k(L-t)\}$$
$$D_k(t) = \mathrm{Im}\{I_k(t) + I_k(L-t)\}$$

where Re{u} and Im{u} are respectively the real and imaginary parts of the complex number u.

The set $\{J_k^1(t), J_k^2(t)\ k \text{ integer}\}$ added to the sign information of $B_k(t)$, $C_k(t)$ and $D_k(t)$ gives the property of completeness to the shape representation, therefore this property is sufficient to reconstruct the bone shape fully. Hence we notice that :

$$A_k(0) = A_k(L) \qquad B_k(0) = -B_k(L)$$
$$C_k(0) = -C_k(L) \qquad D_k(0) = -D_k(L)$$

This implies that :

$$J_k^1(0) = J_k^1(L) \qquad J_k^2(0) = J_k^2(L)$$

Thus, $J_k^1(t)$ and $J_k^2(t)$ are L-periodic functions and they can be expanded in a Fourier series:

$$F_{k,l}^1 = \frac{1}{L} \int_0^L J_k^1(t)\, e^{-\frac{2i\pi lt}{L}} dt$$

$$F_{k,l}^2 = \frac{1}{L} \int_0^L J_k^2(t)\, e^{-\frac{2i\pi lt}{L}} dt \qquad (7)$$

where $F_{k,l}^j$ (j = 1,2) give a discrete and stable set of invariant features for our application.

3. APPLICATION TO BONE STRUCTURE

3.1. Extraction of Primitives

From the above-mentioned presentation we built a matrix of invariants of equation (5) containing M columns and N rows where M is the number of invariants and N is the number of slices. Let us consider one row of the matrix; we compute the Fourier descriptors C_k from I_k in equation (5), and the Rate of Energy Information Conserved RI:

$$RI(j) = \frac{\sum_{i=1}^{j} |C_i|^2}{\sum_{i=1}^{N/2} |C_i|^2} \qquad (8)$$

We truncate the signal at the first index j_0 which give $RI(j_0) \geq 70\%$. After the computation of j_0 for all the slices n, on a bone, we get:

$$m = \max_{n \in \{1,\ldots,N\}} \left\langle \mathrm{Arg}\{RI[j_0(n)]\} \right\rangle \qquad (9)$$

For a bone like an ulna, we found generally m equal to 5. Thus, the matrix contains 5 columns and N rows as shown in figure 1 which we use to determine a 3D segmentation.

$$M = \begin{pmatrix} I_1(t_1) & \ldots\ldots & I_m(t_1) \\ I_1(t_2) & \ldots\ldots & I_m(t_2) \\ \vdots & & \vdots \\ \vdots & & \vdots \\ I_1(t_N) & \ldots\ldots & I_m(t_N) \end{pmatrix}$$

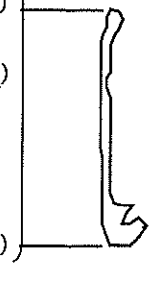


Figure 1

Figure 2 shows the modulus of invariants of rank 1 to 5 in the y-coordinate and the rank of slice in the x-coordinate.

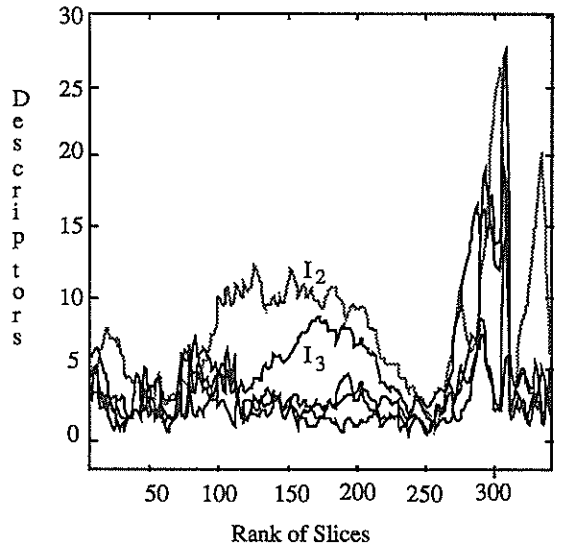


Figure 2

We define a global threshold denoted s and the sequences $\{m_i(z)$ with i=1,5$\}$, where m_i is the modulus of I_i and z is the slice's rank. We compare each parameter of sequences to s in order to determine the required invariants for a smoother representation of the bone. The thresholded sequences are lacunary because a modulus m_i smaller than s is removed from the sequence and replaced by the value s. Thus, we define a new sequence providing information on the presence or not of invariants in successive slices, which is called a *primitive*:

$$\{ m_1(z), m_2(z), m_3(z), m_4(z), m_5(z), n_1 \le z \le n_2 \}$$

$$\text{with } m_i(z) = s \text{ if } m_i(z) < s \text{ for } z \text{ in } [n_1, n_2]$$

where n_1 and n_2 indicate the location of the sequence on the bone. Figure 3 shows the sequences when we consider for example only two invariants.
The change in the *primitive* is related to the sequence permutation of $\{ m_i, s \}$. Thus, we define a finite and small number of primitives to describe the bone. In case of a very high threshold, we can get a generalized cylinder. On the other hand, a low threshold will keep the whole information. Therefore, the fineness of the segmentation, the number of *primitives* and the threshold are interdependent.

The eight sequences obtained are:

$S_1 = \{m_2, m_3, [0,20]\}$ $S_2 = \{m_2, s, [20,65]\}$
$S_3 = \{m_2, m_3, [65,100]\}$ $S_4 = \{m_2, s, [100,115]\}$
$S_5 = \{m_2, m_3, [115,230]\}$ $S_6 = \{m_2, s, [230,250]\}$
$S_7 = \{s, s, [250,260]\}$ $S_8 = \{m_2, m_3, [26,340]\}$

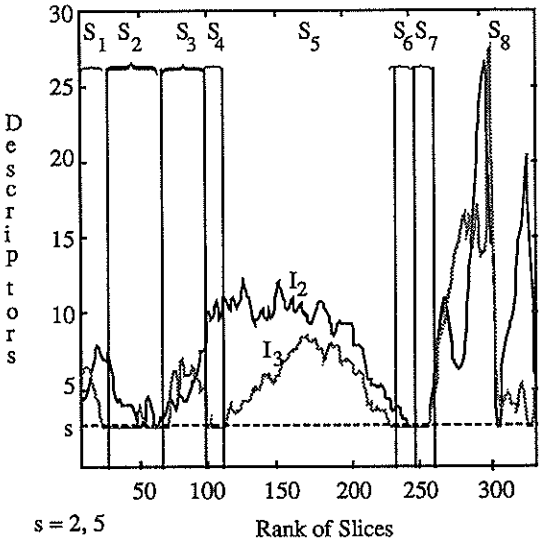


Figure 3

3.2. Synthesis of Primitives

Taking into account the remark in part II, we built two new matrices as explained above to compare different bones under the longest axis invariancy: they will be the same if they have similar coefficients in the two matrices.
Another use of matrices of invariants concerns 3D synthesis. The complete and stable set of Fourier descriptors of 2D shapes is the basic tool of this work. It allows us to reconstruct a 3D synthetic shape from five invariants in each slice. We compute the radius function from the parameters of two invariant matrices. As said above, it can be proved that the information of the radius function is sufficient to compute the angular function. Therefore, we have the full information to reconstruct all slices, then the bone.
We check the reconstruction validity displaying both curves, original and synthetic. The validity of the reconstruction can be appreciated by looking at figure 4.

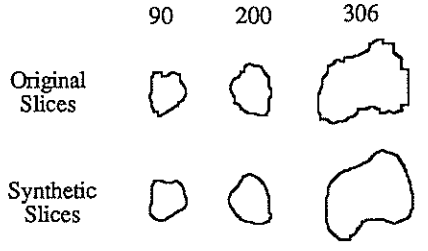


Figure 4

Finally, we have found a description which allows both 3D compression data and 3D shape reconstruction.

4. DISCUSSION

Besides its inherent properties of orientation invariancy in the 3D surface space and because of completeness and stability, the proposed shape analysis and representation method allows us to pass from elementary shape *primitives* such as the perfect cylinder (descriptor I_0 only) to more complex ones reproducing fine details of the surface. Possible continuity conditions following the z-coordinate, resulting from invariant series representation, can be deduced, avoiding the inherent difficulty of shape jointing and overlapping when using known geometric *primitives* [7].
Moreover, all standard pattern recognition extensions and applications, such as different kinds or families of bone shape classifications are possible. Coupled with a statistical analysis, it is also possible to give the representative values of *primitives* for the species, i.e. in considering the invariant as a random variable and then extracting an average template of the bones or areas of bone. This can be used successfully to deduce information about sex or age class scattering.

ACKNOWLEDGEMENTS

The authors thank Janet Ormrod for her kind assistance with the English of this paper.

REFERENCES

[1] BURDIN V., de BOUGRENET de la TOCNAYE J.L., ROUX C., LEFEVRE C. "Une représentation des structures osseuses tridimentionnelles par un diagramme polaire plan de descripteurs de Fourier" 5ème Forum Jeunes Chercheurs (Paris 1990).
[2] DELPRAT J.M. "les moments invariants comme descripteurs 3D - Application à la morphométrie osseuse". Rapport interne - ENST Br (1988).
[3] BIDEGARAY A., SILVA M., ARCANGELI M., APPRATO M. "Splines paramétrées - Application aux courbes paramétrées". Rapport interne, Département de Mathématiques - Université des Sciences de Pau (1988).
[4] GHORBEL F. and de BOUGRENET de la TOCNAYE J.L., "Similarity-invariant analysis of handwritten zip code using Fourier descriptors: a statistical approach". to appear in International Journal of Research & Engineering.
[5] GHORBEL F., CAZUGUEL G., de BOUGRENET de la TOCNAYE J.L., "Similarity-invariant analysis of handwritten zip code using Fourier descriptors". International Journal of Research & Engineering, Inaugural Issue, 1-5, (1989).
[6] GHORBEL F., de BOUGRENET de la TOCNAYE J.L., HILLION A. "A complete and stable set of invariant Fourier descriptors". Submitted to IEEE PAMI.
[7] OZAKI Y., SATO K., INOKUCHI S. " Rule-driven processing and recognition from range Image". 9th international conference on Pattern Recognition, 1988, Rome, 804-807.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

CNV PATTERN RECOGNITION: STEP TOWARD A COGNITIVE WAVE OBSERVATION

Liljana Bozinovska
Institute for Physiology, Medical Faculty, University of Skopje
G. Stojanov, M. Sestakov, S. Bozinovski
Electrical Engineering Faculty, University of Skopje, Yugoslavia

A closed loop CNV paradigm, which enables registration of a cognitive waves in the human brain representing phenomena analogous to classical conditioning paradigm in animal learning theory, is designed and experimentaly proven. The regression angle computed in the interstimulus interval is used as a CNV recognition parameter.

1. INTRODUCTION

1.1. Learning, expectation, and the CNV

The problem of learning in physiological systems is the primary concern of our research. The learning as a phenomenon has several features, the expectation being one of the crucial among them. We understand the expectation as a hypothesis about a future event. We understand the learning as an increase of an expectation about a responce to a certain stimulus, due to a previous experience.

For many years the expectation was a notion of primary concern in psychology. In 1964 Walter et al. [1] have pointed out an electrophysiological evidence of the expectation phenomenon, named Contingent Negative Variation (CNV) of the biosignals measured from the human head during certain experimental paradigm. Since then, the CNV paradigm has been widely investigated among the researchers dealing with the biomedical evoked potentials. Many aspects have been considered [1-4]. However a new, closed loop CNV paradigm was recently proposed [6-8], as a new research direction relevant to investigation of the human learning and cognition.

1.2. Open-loop CNV paradigms

In the classical CNV paradigm, two stimuli, S1 and S2, are presented to a human repeatedly, S1 being considered as a warning stimulus, and S2 as a stimulus on which the subject is to react by a motor action. After several times of S1-S2 presentation, a specific, CNV wave, is produced as an event related brain activity. In other words, it is an open loop control paradigm, having the scheme

```
Classical CNV Paradigm: Procedure
Repeat alternate(S1,S2) until CNV=TRUE
                        End.
```

There have been several modifications of this paradigm, mainly concerning the random appearence of the S2 signal:

```
Random S2 CNV Paradigm: Procedure:
S2=random, distribution=GIVEN
Repeat alternate(S1,S2) until CNV=TRUE.
                        End.
```

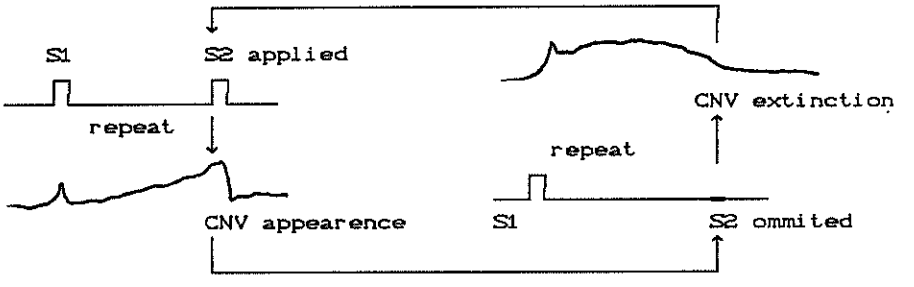


Figure 1. The CNV oscillatory paradigm

2. THE CNV OSCILLATORY PARADIGM

Working with both the above described CNV paradigms [5], we have hypothesized that, using a computer control, it is possible to design a closed loop, biofeedback paradigm with the scheme

CNV Oscillatory paradigm: Procedure:
Repeat
 Repeat alternate(S1,S2) until CNV=TRUE
 Repeat alternate(S1,NIL) until CNV=FALSE
until endcondition=TRUE.
End.

The Figure 1 shows the idea. After recognizing the appearence of the classical CNV wave, the control computer disables the further appearence of the S2 signal; that will in turn cause the extinction of the appeared CNV wave; recognizing that event the control computer again enables the S2 signal; and so on, an oscillatory behaviour is taking place.

3. THE CNV RECOGNITION PROCEDURE

3.1. ERP extraction

The CNV, as an event related potential (ERP), is burried in the louder EEG signal. To extract it we use weighted average procedure

$$ERP(n) = (1-k)*ERP(n-1) + k*EEG(n)$$

where n is the trial number. For the weight factor we used k=0.1 in our experiments.

3.2. CNV recognition

There are several possible features of the CNV signal which could be used as its identification parameter, examples being its amplitude and its integral. As identification parameter in our research we used the regression angle within the interstimulus interval. Using the angle as an identification parameter, we set some

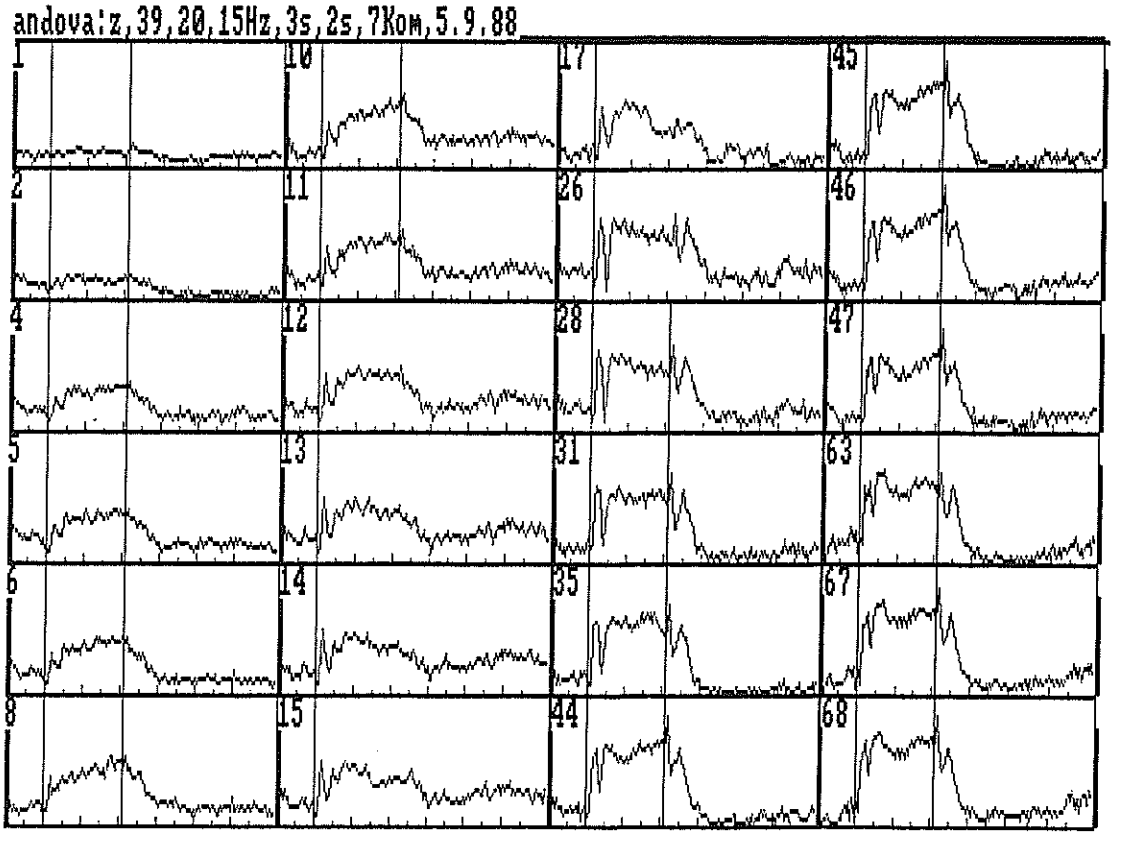


Figure 2. A series of CNV signal samples in a CNV oscillatory paradigm.

threshold angle to show whether the CNV is formed. If the computer confirms in p successive trials (e.g p=2 or 3) that it is the case, then the event CNV=TRUE has happend; if, for p successive trials, we have the regression angle bellow the threshold angle (e.g. 20°) then CNV=FALSE has happened.

4. EXPERIMENTS

The hypothesis on possibility to obtain oscillatory CNV paradigm was experimentaly proven already in the first series of experiments, on all 42 considered subjects. Within 100 trials on each subject, S2 has been turned on and off in average 4 times. Figure 2 shows a typical series of trials within a session with a human subject.

During the experiment the regression angle has been recorded. It controls the oscillatory behavior of the CNV appearence and dissapearence. An example is given on Figure 3, showing the regression angle and its threshold level, simultaneously with the series of onset/ofset of the CNV, where S2_ommitance=CNV_presence.

5. THE CNV OSCILLATOR BEHAVIOR

The results stated above suggested a hypothesis about the asymptotic behavior of the CNV oscillator: it was hypothesised that, exposed to the oscilatory CNV paradigm for relatively long period (e.g. 100 times), the subject expresses a specific pattern of activity concerning the length of the periods of the CNV presence. Pursuing that idea we computed average behavior of the CNV oscillator for the considered subjects. Figure 4 shows the obtained result.

The Figure 4 shows that, for this experimental group (38 students of the medical department, each exposed to a 100 trials experiment), the first time the CNV is registered by the computer is in average at the 18-th trial, the first time the computer registers the CNV extinction is in average on the 30-th trial, and so on.

We did not experimented far beyond the 100 experimental trials border due to the subject fatigue. For this series of

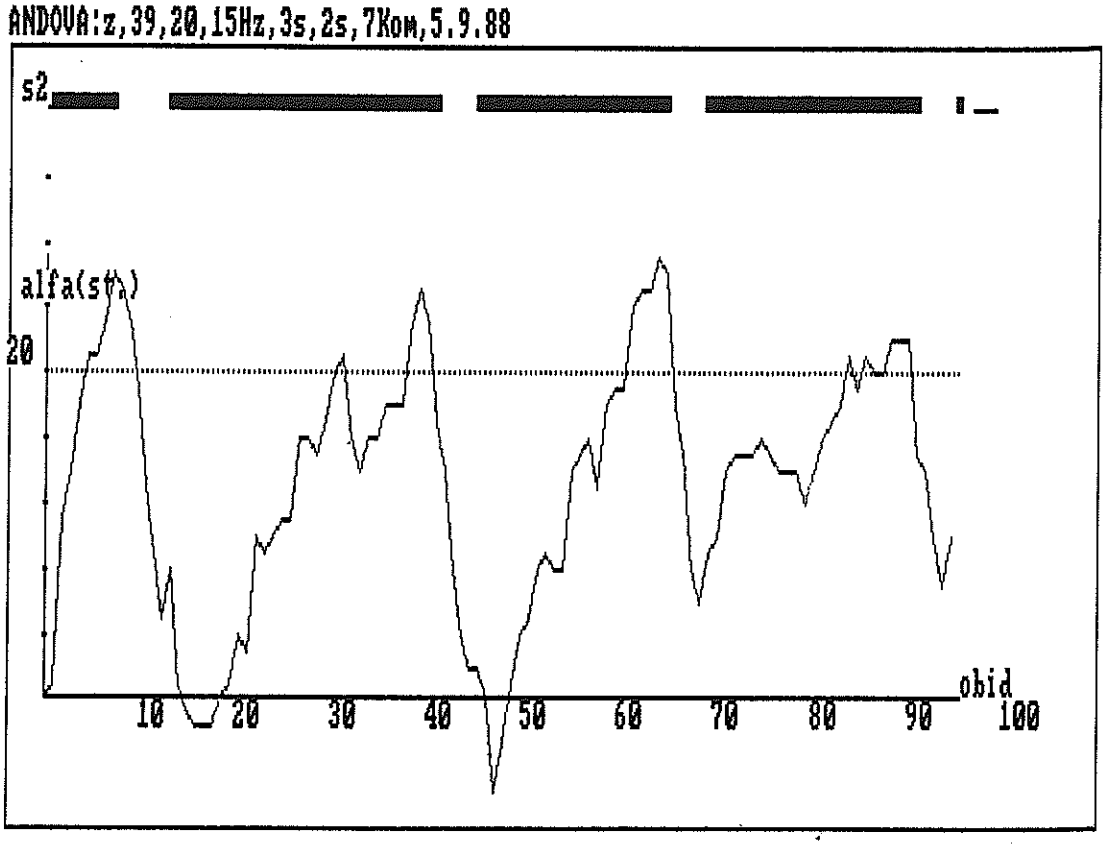


Figure 3. Regression angle changes during a CNV oscillatory paradigm

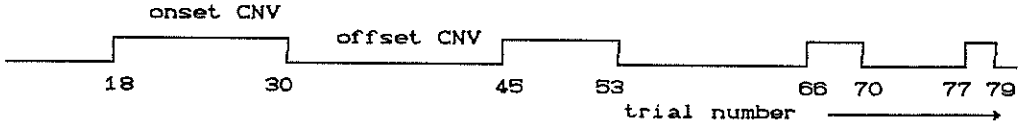


Figure 4. A wave pattern of the CNV oscillator

experiments, the CNV oscillator wave pattern shows that the its frequency increases with the number of trials, i.e. the period of existence of the CNV wave becomes shorter and shorter.

6. THE CNV OSCILLATORY PARADIGM AND THE CLASSICAL CONDITIONING

From the begining of our research we understood that the CNV oscillatory paradigm could give some insight in the processes analogous to the classical conditioning, investigated in the animal learning theory. We recognize the analogy CNV_paradigm(S1,S2,CNV) = classical_conditioning(CS,US,CR), where CS is the conditioned stimulus, US is an unconditioned stimulus, and CR is the conditioned responce for a given classical conditioning experiment. For the famous Pavlov experiment we had CS=bell, US=food, CR=salivation. The CNV oscillatory paradigm could be interpreted in terms of conditioned refex establishing and conditioned reflex extinguishing, as in classical conditioning theory.

7. THE CNV OSCILLATOR WAVE AS A COGNITIVE WAVE

The CNV oscillator shown on Figure 4 is obtained as a feature of a human biosignal activity during the process of expectation of an event. When the subject realizes that an event is to be expected, he gives a strong evidence of that by a CNV signal. That signal is used as a biofeedback that disable the cause of the CNV appearence, the S2 signal. The process involves learning and cognition that the event is to be expected or not. Thus, we could point out that what is obtained during the CNV oscillatory paradigm, is a cognitive wave about a situation related to the S2 appearence.

8. CONCLUSION

Using signal processing methods, we have shown a phenomenon interpreted as CNV oscillator wave. For obtaining it, we first used the signal processing technique of weighted averaging to obtain the CNV signal, and then the signal pattern recognition technique based on linear regression. The new waves obtained are cognitive waves, and has never been studied before. Actualy they represent the human behavior in tasks analogous to tasks of classical conditioning in animals.

REFERENCES

[1] Walter G., Cooper R., Aldridge J., McCallum C., Winter A. "Contingent negative variation: An electric sign of sensomotor association and expectancy in the human brain" Nature, 1964

[2] Tecce J. "CNV and physiological process in man" Physiological Bulletin, 1977

[3] Gailard A., Perdok J. "Slow brain potentials in CNV paradigm" Acta Physiologica 44, 147-163, 1980

[4] Bozinovska L. "The CNV paradigm: Electrophysiological evidence of expectation and attention" Psychological physiology term paper, Psychology department, University of Massachusetts at Amherst, 1981

[5] Bozinovska L., Izgum V., Barac B. Electrophysiological and phenomenological evidence of expectation process in the reaction time measurements" Yugoslav physiol. and pharmacol. acta, 1985

[6] Bozinovska L., Sestakov M., Stojanov G., Bozinovski S. "CNV potential intensity change during the personal computer guided biofeedback training" (In Serbocroat) Neurologia, 1988

[7] Bozinovska L., Bozinovski S., Stojanov G., Sestakov M. "Introducing feedback in the CNV paradigm" (in Serbocroat) Proc. Conf. ETAN, Vol. Biomedical Engineering, Novi Sad, 1989

[8] Bozinovska L., Stojanov G., Sestakov M., Bozinovski S. "CNV oscillator: A hypothesis and its experimental verification" Int. Symp. on Sensory EEG, Ljubljana, 1989

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

REAL-TIME MONITORING OF EMG VARIABILITY USING FAST STATISTICAL FILTERING

Hannu Nieminen, Risto Suoranta* and Kari-Pekka Estola*

Medical Engineering Laboratory, Technical Research Centre of Finland
P.O. Box 316, SF-33101 Tampere, Finland
**Machine Automation Laboratory, Technical Research Centre of Finland*
P.O. Box 192, SF-33101 Tampere, Finland

In this paper a new method is applied for the continuous real-time monitoring of median value and other percentile values of EMG (electromyography) signals. Results of the EMG analysis are used for the assessment of the physical loading on the musculo-skeletal system during manual work. The algorithm used is based on the multiresolution estimation of the moving histogram. The method takes advantage of the finite wordlength representation of the digitalized data resulting in an extremely fast realization.

1 INTRODUCTION

EMG-signals are electrical signals caused by the neural activation of the muscle. Average amplitudes of EMG-signals have been found in some situations to correlate to the force produced by the muscle during contraction. These force predictions have been used e.g. in ergonomic evaluations for the assessment of the long-term physical loading on the musculo-skeletal system during manual work. EMG-signals have been used for the evaluation of both short-term and long-term local muscular strain. Main aims of the analysis have been twofold: estimation of the force produced by the muscle and the analysis of local muscular fatigue. Many problems affect the interpretation of the EMG especially during dynamic muscular work, because EMG-signals are nonstationary in nature and thus difficult to analyze.

EMG-signals have mainly been analyzed either using the values directly or calculating e.g. various kinds of amplitude distributions for full-wave rectified and averaged signals [3]. The accumulation of localized muscular fatigue has been monitored using spectral analysis of the EMG. EMG amplitude probability distribution function (APDF) has [3] been used for the analysis and monitoring of muscular loading and the corresponding percentile values derived from the APDF have been used for the parametrization of signal variability.

Median filtering has also been used for EMG smoothing and noise rejection [5]. Although the median operation is nonlinear, theoretical studies and experiments have revealed some of its properties. Median filtering technique is known for preserving sharp changes in signals and for being effective in removing impulsive noise. Median filtering has also been used for the analysis of long-term trends in EMG amplitudes connected to static stress on the muscles caused by prolonged muscle contractions [5].

It would be valuable to be able to combine median filtering of EMG to the continuous estimation of the variability of the signals using the APDF analysis. Physical loading could then be characterized through calculating and refining continuously percentiles or through calculating the trends in various percentiles of the APDF during long-term work. Real-time operation would be needed for practical studies conducted during normal working situations. Typical methods used for calculating the median and other percentile values are far too slow for real-time applications

especially if the number of data samples included in the median operation is high. The computational complexity of median filters increases rapidly with the number of data samples. The computational complexity can be decreased using moving median filters which compute the median within a window sliding over the data points. Several authors have developed fast methods for computing the median in real-time applications. These include for example the radix method introduced by Ataman et al. [1] and the histogram method by Huang et al. [4].

In this paper, we are applying a new multiresolution histogram method for the continuous real-time monitoring of median value and other percentile values of EMG. The algorithm used is based on the multiresolution estimation of the moving histogram [2]. The method used takes advantage of the finite wordlength representation of the digitalized data. The worst case complexity of the filter is of order O(L), i.e. linearly proportional to the wordlength L. The complexity of the filter does not depend on the number of samples included in the median computation. Thus because the size of the moving window can be arbitrarily long, the variance of the estimates to the percentiles can be made low. The main difference between this method and the previously used methods [1,4] is the use of a hierarchical variable depth tree structure in sorting the data. The control structure is also very simple and it is easily implementable in VLSI. VLSI implementation of the algorithm would produce a practical, portable tool for the analysis of EMG measured during normal work.

The basic algorithm computes the median and other percentiles exactly but the algorithm can also be used to compute approximations to the true percentiles, which is faster. This can be done simply by decreasing the number of bits in representing the data samples within the tree structure.

2 THE MULTIRESOLUTION HISTOGRAM METHOD

The algorithm used [2] is based on the multiresolution estimation of the histogram of the data in a specified window of length M. The estimated histogram forms an *n*-ary tree having L/d levels excluding the root. The relation between n and d is d=log2n. The total number of memory elements needed increases with decreasing *d* such that the maximum $R_{max} = 2L+1-2$ is obtained with the binary tree, i.e. d=1. Figure 1 illustrates an n-ary tree with n=4, d=2 and L=8.

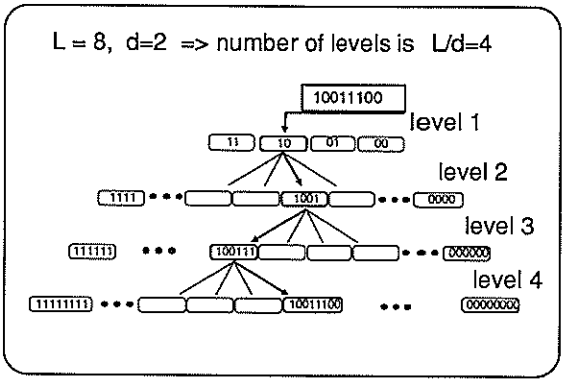


Figure 1. An example of the n-ary tree for the multiresolution histogram.

The tree contains M data samples at one time. The tree is implemented in such a way that each node in the tree contains the total number of the data samples in its descendants. Thus the nodes at a certain level comprise the histogram estimates and the histogram at that level is quantized with *ilev* times *d* bits, where *ilev* is the specified level. This means that several histograms are computed at the same time from the same set of data with different class interval widths. In the bottom level the interval width corresponds to the maximum resolution.

As the window slides over the data the contents of the nodes are updated. This is accomplished by incrementing the contents by one for an incoming new sample and decrementing the contents by one for the outgoing sample. In order to achieve a fast realization, the approach uses the values of the data samples as the addresses for the nodes to be updated. The address for the node at the first level is the d most significant bits of the data and at the second level the 2d most significant bits and so on.

The search for the various percentiles is started from the top of the tree with the low resolution

histogram. When the top level interval is found the search is continued in the next level. At each level the search is performed from the lower addresses towards the higher addresses. The search is accomplished by summing the contents of the nodes and comparing the sum to the value indicated by the specified percentile. When the value is exceeded the search continues on the next lower level. In the lowest level, the address of the node is the searched percentile. Depending on which level the search is stopped, different resolution estimates for the percentiles can be obtained.

The multiresolution histogram method has been found [2] to be superior to the reference methods [1,4] in terms of the computation complexity. The method is twice as good as the radix method of Ataman et al. [1] for a decimation ratio of 1 and becomes even better as the decimation ratio increases. The required number of memory elements is, however, somewhat higher for the method at small values of d but at large values of d the difference is insignificant.

3 EMG ANALYSIS USING THE MULTIRESOLUTION HISTOGRAM

Multiresolution histogram method is a suitable method also for the analysis of EMG in ergonomic evaluations, where the aim is to characterize the long-term physical strain on the musculo-skeletal system. Strain measurements can be used for example in field studies conducted during normal manual work, where the effects of different work postures and methods on the musculo-skeletal strain are compared. Amplitude probability distribution function derived from the EMG activity of the muscles exposed to potentially harmful physical loading can in these studies be used for the statistical characterization of the total physical strain [3]. Calculation of the APDF should be done in real time, because the aim is to compare the results from different postures and work methods and rapid feedback is thus needed.

Nature of the physical strain depends on the type of the muscular contractions involved. Particularly long-term static work has been considered harmful, because prolonged tension causes both muscle fatigue leading to pain and immobilization of the joint and mechanical irritation of tendons. Static work involves prolonged contractions of muscles corresponding to approximately constant levels of muscular activation and average EMG amplitude, while dynamic work involves rhytmical contractions and relaxations of working muscles. Muscular activation is changing rapidly during dynamic work. During normal dynamic work the peak tissue loading is typically within the physical capability of the tissues. However, prolongation of tissue loads and frequent dynamic exertions produce similar effects as in the case of static work.

Many problems affect the interpretation of the EMG especially during dynamic muscular work, because EMG-signals are nonstationary in nature and thus difficult to analyze. The characteristics of the measured signals are also affected by many physiological factors and depend strongly on the measurement site and on the orientation of the active muscle fibres with respect to measuring electrodes. Because of this, the average EMG amplitudes have to be connected to force using calibration protocols, such as normalizing the amplitudes according to the maximum voluntary contraction of the muscle.

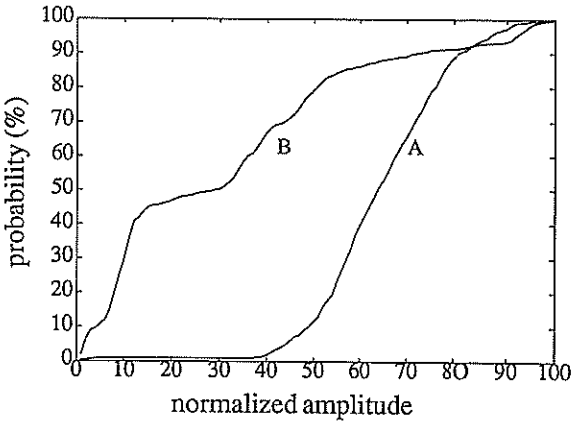


Figure 2. An example of two APDF's calculated from EMG-signals. A: static work, B: dynamic work.

Percentile values derived from the APDF, such as median, 10 % and 90 % values, have been used for the parametrization of signal variability [3]. The extreme variations in signals due typically to inherent nonstationarities can then be ruled out

from the analysis. By using the measured EMG-force relationship in reverse the APDF of an EMG-signal can then be transformed into an APDF of the relative force of contraction, and compared directly to the external load on the muscle. Shape of the measured APDF also bears information about the nature of the loading. For example heavy static loading at some force level corresponds to APDF rising sharply at that level.

Figure 2 depicts an example of the APDF analysis of EMG-signals. APDF's have been calculated using the multiresolution histogram method. EMG-signals have first been full-wave rectified and filtered with a moving averager (window length 50 ms). Amplitudes of the EMG-signal have been calibrated according to an MVC test done prior the measurement. Analyzed EMG sequence was measured from the trapezius muscle during two different types of work sequences. Curve A corresponds to a sequence, where the static "base level" of activation was high and curve B corresponds to dynamic work.

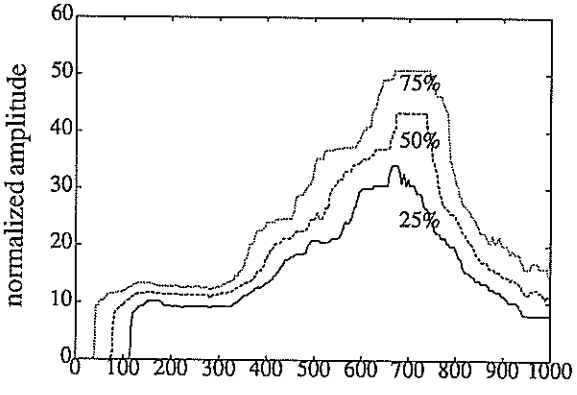


Figure 3. 25, 50 and 75 % percentile values from EMG probability density functions.

Multiresolution histogram method can also be used for the real-time median filtering of EMG-signals. Median filtering is an efficient method for the filtering of EMG-signals, because the signals are nonstationary in nature and contain many sharp level changes and transients. Median filtering can also be used for decomposing the signal into components describing separately the static and dynamic aspects of muscular strain. Static components are then connected to long trends in the signal, which correspond to an approximately constant level of muscular activation. Median filters with long window lengths can be used for the extraction of these long trends from the signals. Multiresolution histogram method enables the use of these long window lengths even in real time. Median filter banks can also be used for the extraction of different dynamic ranges from EMGisignals.

Using the multiresolution histogram method it is also possible to build filters utilizing besides the median value also other percentile values derived from the APDF in real time. Analysis of the long trends in EMG signals can be supported with the variability analysis of EMG using for example the 25 % and 75 % values of the APDF together with the median. An example of this is depicted in figure 3, where a static work sequence is followed by a dynamic sequence.

4 REFERENCES

[1] Ataman, E., Aatre, V.K., Wong, K.M., A Fast Method for Real-Time Median Filtering, IEEE Trans. Acoust., Speech and Signal Processing, Vol. ASSP-28, pp. 415-421, Aug. 1980.

[2] Estola, K-P., Suoranta, R., A Fast Probabilistic Median Algorithm for Integer Arithmetics, in: Proceenigs of the Douzieme Colloque sur le Traitement du Signal et des Images, Juan-Les-Pins, France, 1989, pp. 61-64.

[3] Hagberg, M., On Evaluation of Local Muscular Load and Fatigue by Electromyography (Arbetarskyddstyrelsen, Sweden, 1981).

[4] Huang, T.S., Yang, G.T., Tang, G.Y., A Fast Two-Dimensional Median Filtering Algorithm, IEEE Trans. Acoust., Speech and Signal Processing, Vol. ASSP-27, pp. 13-18, Feb., 1979.

[5] Nieminen, H., Haemeenoja, S., Analysis of Static and Dynamic Aspects of Muscular Strain Using Nonlinear Trend Analysis of Surface EMG, in: A. Mital (ed.), Advances in Industrial Ergonomics and Safety I (Taylor & Francis, New York, 1989), pp. 783-790.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Object-Based Information Modeling for Pattern Recognition and Motion Analysis

V. Cappellini, Dipartimento di Elettronica,Via di S. Marta 3, 50139 Firenze, Italy
R. Cecchini, Dipartimento di Fisica, L.go E. Fermi 2, 50125 Firenze, Italy
A. Del Bimbo, Dipartimento di Sistemi e InformaticaVia di S. Marta 3, 50139 Firenze, Italy
P. Nesi, Dipartimento di Sistemi e InformaticaVia di S. Marta 3, 50139 Firenze, Italy

In this paper we describe a system architecture for an image understanding system, with motion analysis support, based on the object-oriented paradigm. The choice of this paradigm allows to greatly facilitate reality modeling in addition to reduce the "impedence mismatch" problem at the database interface. In our intention the system proposed should be a first step towards a truly "generic" image understanding system. The first prototype of the system described has been implemented in C++.

1. Introduction

Most of the existing image understanding systems are small systems geared to a specific task with a very limited data repository. The efforts to build a *general purpose* image understanding system, inevitably lead to the problem of managing very large information databases. Thus the problem of *knowledge structuring* becomes fundamental for these systems. An ideal system would be one in which the real world is mimicked in the most natural way, so as to capture in its database(s) a model of *real world knowledge* as exact as possible. It is obvious that for such a system the usual problems of updating and maintenance would be greatly reduced [1].

The requirements of a "good" modeling of reality are even more pressing if we want to perform some *motion analysis*. Think, for example, to a system for wood fire monitoring, where the behavior of the clouds of smoke is fundamental for the strategy to be followed by the fire extinguishers.

A large part of the work done in the Artificial Intelligence field is devoted to the study and simulation of the mechanisms of human knowledge and learning (e.g. first order logic, semantic networks, frame systems [2,3]). We think however that all the proposed systems lack the capability to interface *very large* databases, which, we believe, is fundamental to surpass the *prototype* stage of the implementations. The answer to this need might be found in the Object-Oriented Paradigm (OOP), a methodology which tries to answer to the requirements — often contrasting — from the Software Engineering and Artificial Intelligence fields [4,5].

In this paper we describe a system architecture for an image understanding system, with motion analysis support, based on the object-oriented paradigm. The choice of this paradigm allows to reduce the "impedance mismatch" problem at the database interface, in addition to greatly facilitate reality modeling [6]. In our intention the system proposed should be a first step towards a truly "generic" image understanding system, a goal which is still very far from completion. The first prototype of the system described has been implemented in C++ [7]. Work is in progress to support a multiprocessor (transputer) architecture.

2. Information Modeling

As we said, if we want to define a not too specific image understanding system, we have to face the problem of the enormous quantity of information that has to be memorized and processed to perform recognition. Moreover it is well known that "generic" recognition systems need general information about the problem domain.

Thus the database of the system has to contain two totally different kinds of information:

1. those of strict pertinence to the pattern recognition task, like information on textures and shapes (Visual_models);
2. those about the general aspects of the problem domain, like the general characteristics of the objects which can be found in the environment under analysis (Non_Visual_models).

The second kind of information is used to pilot the recognition process, helping to prune the tree of the possible alternatives. (If, for example, we are analyzing aerial pictures for crop classification, information on the local climate is of great help in ruling out

unfeasible hypotheses).

The object-oriented approach, owing to the strict correspondence it allows to create between the "real" and the "software" world, is of great help in overcoming the difficulties to represent the previous kinds of information. Of course, to avoid the "impedance mismatch" problem at the application-database interface, this latter too must be object-oriented. In this way every entity in the problem domain is simulated in the system by a corresponding software *object*, with the same attributes and properties found in the real world.

Moreover, making use of the *inheritance* mechanism (IS_A), it is possible to define the multiple levels of abstraction at which the entities can be seen. (For example, at increasing levels of abstraction, an *Alfa Romeo* could be seen as a *car*, a *vehicle* and a *moving object*). The definition of the abstractions for a particular recognition problem is a very important and delicate task, as from it depends the capability of the system to classify efficiently. In fact, a good abstraction hierarchy allows a quick selection of the possible candidates at the first stages of the recognition process, with a minimum of elaboration time. Tab.1 summarized the main characteristics of the OOP paradigm for knowledge representation.

	OOP: KNOWLEDGE REPRESENTATION
OBJECT	* real world object * entity characterized with form and behavior
CLASS	* conceptual abstraction to group objects with similar characteristics * type definition support
INHERITANCE	* tool for logical structuring by specialization relationships

Tab.1.

The information modeling possibilities offered by the object-oriented paradigm are by no means limited to the ones above described. What it is more important, perhaps, is the capability to model the *human recognition process*. This is of fundamental importance if we think that we have to find a way to deal quickly and efficiently with the enormous amount of data produced by the sensors. (For example a video camera typically produces 30 frames per second of 512×512 bytes). The mechanism used by the humans is a clever combination of low-resolution processing for the whole image with high-resolution processing for the more interesting parts [8]. This corresponds to the two zones of the retina, with different types of sensors. To simulate such a behavior, we defined a hierarchy of classes with different levels of detail. Each simulated object is thus instantiated more than one time, each one at a typical abstraction level. In this way the recognition process can start at a low resolution (i.e. with low consumption of CPU time), taking into account only the more fundamental characteristics (like e.g. a simple including geometrical form), to continue — only for the more promising cases — at greater detail but on more restricted areas. Fig.1 shows the organization of vision system according to the previous considerations, directions of searching are also explicited.

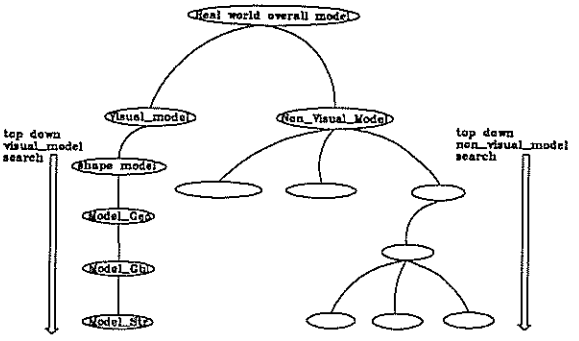


Fig.1-Links shown are IS_A (specialization) links.

3. Motion Modeling

Motion analysis becomes of fundamental importance in processing a sequence of images when:

- the shapes of the objects are not sufficient for their recognition (e.g. a system which tries to discriminate between different kinds of vehicles in a nocturnal environment, when their shapes are not much more than light blobs);
- more information can be derived from the motion of the identified object (e.g. the motion of the clouds of smoke can give precious information on the strength of the wind and the possible evolutions of fires);
- it is important to predict the future positions of already identified objects.

In all of the previous cases the object-oriented paradigm, with its modeling capabilities, can be of great help. In fact, the motion properties can be easily added to the other features of the classes which the designer has chosen to define. In this way, the functions and attributes which describe the *dynamic behavior* of the object can be used both in the recognition process — to discriminate between object with similar shapes — and in the prediction of future behaviors

after the recognition is complete.

In the first case, the yet unknown object grabbed by the sensor is simulated by an instance of a "dynamic_object" class. Among the methods of this class there is a parametric motion function — typically of the form, having verified that a second order motion equation is adequate for most of the cases:

$$(1) \qquad s(t) = a_1 s(t-\Delta t) + a_2 s(t-2\Delta t)$$

whose coefficients are calculated from the sensor data (by a feedback error correction mechanism) so as to describe the motion of the object for the observation period as exactly as possible. The values calculated are then compared with those of the reference models so as to make hypotheses on the object nature.

In the second case the same motion equation, this time with well defined values for the coefficients, allows to make predictions on the object future behavior.

At each frame sample, the system computes the predicted position s_p (see Fig.2) according to equation (1). The s_p is hence compared with the observed position at time t, $s(t)$.

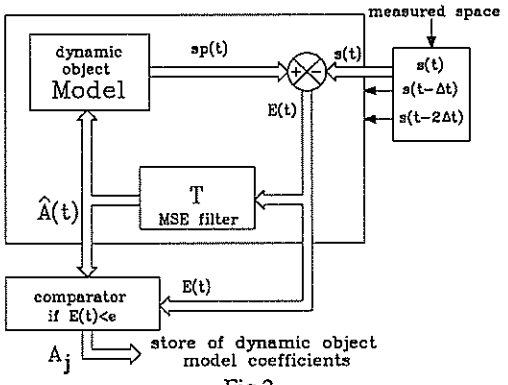


Fig.2.

The estimated error $E(t)$ and the motion history allow to evaluate the values of the motion coefficients for the following steps using a minimum square error filter:

$$(2) \qquad \hat{A}(t) = \{\hat{a}_i \mid i = 1,\ 2\} \qquad with$$

$$\hat{a}_i(t) = \frac{det_i G}{det\ G} \qquad i = 1,\ 2$$

where G is a $2x2$ matrix of the coefficients g_{ij} defined according to:

$$g_{ij} = \sum_{k=0}^{n_o - 1} s(t-(k+i)\Delta t)\ s(t-(k+j)\Delta t)$$

for $i=1,2; j = 1, 2$.

n_o being the number of observations, and $det_i G$ is the determinant of the matrix G where the column ith has been replaced with the column vector V with elements v_i in according to:

$$v_i = \sum_{k=0}^{n_o - 1} s(t-(k+i)\Delta t)\ s(t-(k+n+1)\Delta t)$$

for i = 1,2;

In most cases, few steps are sufficient to stabilize the coefficients.

Generally, observing some moving entity in a frame sequence, several sets of coefficients of equation (1) that fit the behavior of real world object at different time intervals result. Therefore some general model M of the behavior of an entity will be defined by:

$$M = \{A_j \mid j = 1, m\} \qquad with: \qquad A_j = \{a_i \mid i = 1, 2\}$$

where m is the number of observations and the order of the motion modeling equation is 2. In practice only the bounds of the parameter space of the motion modeling equation are stored. Dynamic recognition is performed by comparing the estimated motion coefficients with those stored in the non-visual entity models in the database. In Fig.3, as example typical figures of coefficients of the autoregressive equation, drown by experimental measurements are shown: the figures are related to two kinds of vehicles, car and truck respectively, truck parameter space is included in the one of car.

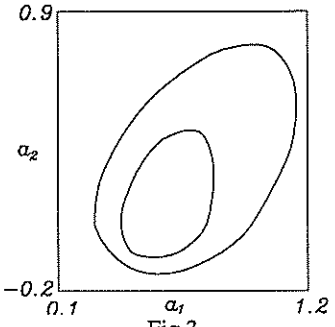


Fig.3

4. Conclusions

In this paper we have presented a system architecture for an image understanding system based on the object-oriented paradigm.

We believe that the use of this paradigm is of mandatory in the building of a "general purpose" recognition system, where the problems of representation of

real world knowledge become fundamental. Moreover, the modeling capabilities offered by OOP allow close modeling of the problem domain and of *the recognition process itself* and in addition including in the system support for motion analysis (i.e. recognition by behaviour and/or prediction of the behavior of the identified objects).

Further advantages offered by the object-oriented paradigm can then be summarized as:

- greater system modularity and flexibility, the resulting architecture is easily adaptable to different environments;
- the use of an object-oriented database for the data repository allows a reduction of the "impedance mismatch" problem at the application-database interface, with great advantages from the point of views of system extendibility and maintainability;
- the resulting architecture is very suitable for an implementation on a multiprocessor system, which is of fundamental importance in all those situations in which there are time constraints on the recognition process.

References

[1] Rao, A.R. and Jain, R., *IEEE Expert* spring 88 (1988) 64.

[2] Barr, A. and Feigenbaum, E., *The Handbook of Artificial Intelligence* (William Kaufmann, Los Altos, 1982).

[3] Rich, E. *Artificial Intelligence* (McGraw-Hill, New York, 1981).

[4] Stefik, M. and Bobrow, D., *AI Mag.* **6** (1986) 40.

[5] Shriver, B. and Wegner, P. (eds.), *Research Directions in Object-Oriented Programming* (MIT Press, Cambridge, 1987).

[6] Andrews, T. and Harris, C., "Combining Language and Database Advances in an Object-Oriented Development Environment", in: *Proceedings OOPSLA '87* (Springer-Verlag, Berlin, 1987).

[7] Cecchini, R., Del Bimbo, A. and Nesi, P., "An Object-Oriented Approach to Object Recognition Systems", *Report n. 25/89* (Department of Systems and Informatics, University of Florence, Florence, 1989).

[8] Burt, J.P., in: Freeman, H. (ed.), *Machine Vision* (Academic Press, New York, 1988).

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

Extraction of Straight Lines in Aerial Images †

V. Venkateswar and R. Chellappa

Signal and Image Processing Institute, Department of Electrical Engineering–Systems
University of Southern California, Los Angeles, CA 90089-0272, U. S. A.

We describe a straight line extractor that can produce high quality line descriptions from aerial images. The input to the line extractor is in the form of an edge image, where the contrast and direction of each edge pixel is specified. The system first scans the edge image left to right, top to bottom (LRTB) and assigns a line label for each scanned edge pixel, thereby generating a *label image*. At the end of this process each edge pixel has a line label associated with it and edge pixels that belong to the same line will be assigned the same line label. Also, with each line label, a record that stores the end points, the average contrast and the pixel support of the line is generated. The *label image* is used as a spatial index to further link fragmented lines. We also describe techniques to eliminate a large portion of physically insignificant lines from the database. This leads to time and memory savings in the higher level modules. The heuristics used at different stages of the line extraction process were derived after observation over several aerial images. A real image example is shown to illustrate this system and comparison with the Nevatia-Babu Line Finder is given.

1 Introduction

The amount of data contained in a typical aerial image is large. The purpose of the low level module is to reduce this data by extracting a small number of primitives. These primitives can be regions, curves, straight line segments etc. In this paper we consider the extraction of straight line segments (lines, for short) from aerial images. A few methods are known in the literature [1, 2] for the extraction of lines from images. In the Nevatia-Babu method [1], edge pixels are linked by marking the locations of the predecessor and the successor of each edge element in a predecessor and a successor file respectively. The boundary segments are then computed from these files. Each boundary segment is then approximated by an iterative end point fitting method. In Burns et al. [2], edge pixels are first determined by convolution with two simple 2×2 masks. The pixels are then grouped into line-support regions of similar gradient orientation. Then the intensity surface associated with each line-support region is approximated by a planar surface. Lines are then extracted by intersecting this fitted plane with a horizontal plane representing the average intensity of the region weighted by a local gradient magnitude.

In this paper we present a system for the detection of lines in aerial images. This line extractor has its roots in a preliminary version presented in [3]. However, it is more sophisticated and the results are significantly better in quality. The initial stage of line extraction can be viewed as a connected components algorithm. The edge pixels are the nodes in the graph. Connections between the nodes (2 or 3 at a time) are specified with heuristic templates based on the edge pixel directions. A labelling algorithm assigns line labels to the nodes and thereby generates a *label image*. At the same time the algorithm builds an internal description of the lines. Subsequent stages of the system attempt to link fragmented lines. These stages use the *label image* as a spatial index to efficiently search for proximate and collinear lines. Section 2 illustrates the labelling algorithm. Sections 3 and 4 describe the stages of the system that attempt to link fragmented lines. Section 5 describes techniques to get rid of noisy lines that have no physical significance.

2 A Scan and Label Process

The input to the system is in the form of an edge image. At each edge pixel in the image the contrast as well as the quantized direction must be specified. We use the Canny

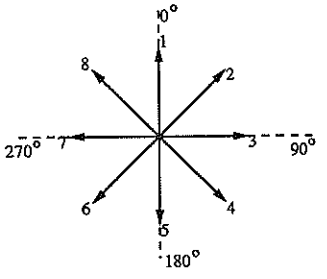


Figure 1: Quantizing the edge pixel directions.

edge detector [4] to generate the edge image. The direction of each edge pixel is quantized into one of eight directions, as shown in Figure 1. Figure 6 shows a typical aerial image (320×320 pixels). Figure 7 shows the result after applying the Canny operator (edge pixels are shown as black dots).

†Partially supported by the Joint Services Electronics Program, at the University of Southern California through the Air Force office of Scientific Research under Contract F-49620-88-C-0067.

The labelling algorithm scans the image LRTB and assigns a line label to each edge pixel. At the same time it incrementally builds an internal description for each line label. This is a record that has slots for the start and end points, the average contrast along the line and the pixel-support of the line (number of pixels assigned this line label). The idea of this scan-and-label process is to assign the same line label to edge pixels that fall along a straight line contour. So when an edge pixel is visited the algorithm checks to see if it inherits the line label of a neighboring pixel based on a continuity criterion. The continuity criterion is based on the directions of the edge pixels. Consider Figure 2(a). Three

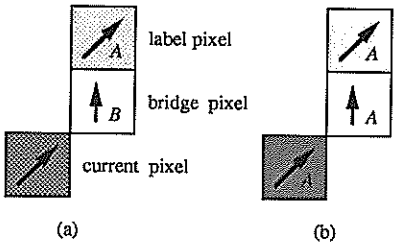


Figure 2: (a) Pixel types. (b) Result after labelling the current pixel. The line labels are shown in the lower right corner of the pixel boxes. The arrows represent the edge directions.

kinds of edge pixels that participate in this continuity test can be defined. The *current-pixel* is the edge pixel that is to be labelled. The *label-pixel* is the edge pixel whose line label the current-pixel can inherit. The *bridge-pixel* is the edge pixel that bridges any gap in continuity between the label-pixel and the current-pixel. Before the current-pixel is assigned any label, the label-pixel and the bridge-pixel have already been assigned line labels by the scan-and-label algorithm (Figure 2(a)). The algorithm checks the following in sequence:

1. The current-pixel must have the same direction as the label-pixel.
2. It then ensures that the line corresponding to the bridge-pixel has unitary support. This is to ensure that lines do not intersect.
3. The bridge-pixel is assigned the same label as the label-pixel. Its previous line label is discarded along with the descriptions associated with it.
4. The current-pixel is also assigned the line label of the label-pixel. The end-point slot of the line is set to the coordinates of the current pixel. The pixel-support and average-contrast slots of the line are appropriately updated.

The effect of this procedure on Figure 2(a) is shown in Figure 2(b).

Continuity is tested for by heuristic templates that were derived after observation over several aerial images. The set of templates for the edge direction 1 are shown in Figure 3. This template matching can be done in parallel. A random choice can be made in the case of a conflict.

At the end of the LRTB scan, corresponding to the edge image, there is now a *label image*. This *label image* will prove useful in the later stages of the linking process, mainly as a spatial index for the lines. A list of all the line labels assigned (*line-label-list*) is also accumulated. The scan-and-label procedure when applied to the edge image in Figure 7 resulted in the detection of 3747 lines shown in Figure 8.

3 Linking single-pixel-support Lines

The edge detectors that are in popular use are step edge detectors and do not work well at corners where two edges meet. The reason is that, at the corners, the image is not

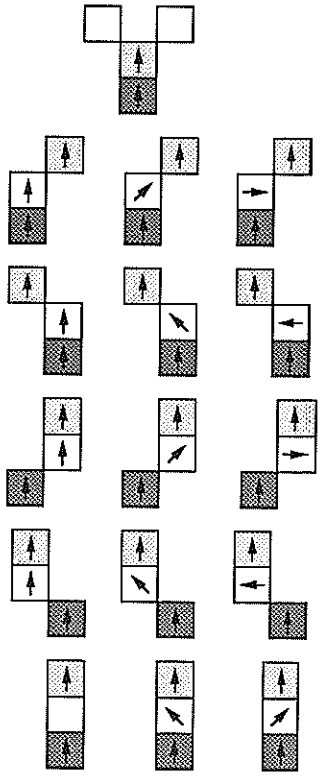


Figure 3: Templates for edge direction 1.

a perfect step and this leads to a low response from the edge operator. The effective result is that the scan-and-label process often misses linking the edge pixels at the ends of lines. These edge pixels are frequently fragmented into single-pixel-support lines (pixel-support of only 1). So in the next stage the line extractor threads these single-pixel-support lines together by observing their relative patterns. The patterns for testing for the directions 1, 2 and 3 are shown in Figure 4. The number of lines in Figure 8 is reduced to 3448.

Even after this merging operation some single-pixel-support lines remain. Some of these lines have no physical significance whatsoever and are pure noise. However, a small percentage of them are natural extensions of some longer line. So we examine each line in the line-label-list that has

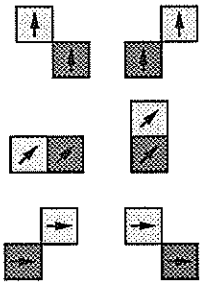


Figure 4: Templates for linking single-pixel-support lines.

a pixel-support greater than 1. At the end points of these lines, based on the line direction we identify the pixel coordinates within a 5×5 window that are natural extensions to the line. We then examine to see if there are any single-pixel-support lines at these coordinates. If the edge pixel direction is within (± 1 mod 8) of the quantized line direction, then the long line is extended and merged with the single-pixel-support line. At the end of this stage any remaining single-pixel-support lines are treated as noise and removed from the line-label-list. The *label image* is appropriately updated. With these procedure the number of lines is further reduced to 1773 (shown in Figure 9).

4 Merging Collinear Lines

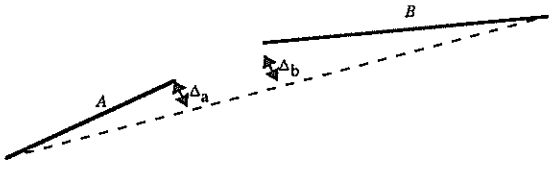


Figure 5: Merging collinear lines.

It should be clear from the above description and the templates in Figure 3 that the decisions to link are made locally and conservatively. Consequently, there are virtually no wrongly linked lines in the generated description. However, weak lines are frequently fragmented. So we link significantly collinear line fragments. For this, we again examine each line in the line-label-list. To identify collinear segments we first look for another line within a 7×7 window at each end point. We then join the farthest end points of the lines and ensure to see that the sums of the maximum absolute deviation of each line from this hypothetical line is less than a threshold. This is illustrated in Figure 5, where the total deviation $\Delta_a + \Delta_b$ must be less than a collinearity-deviation-threshold (τ_d) for the lines to be merged. Merging proceeds recursively. This stage reduces the number lines in Figure 9 to the 1484 shown in Figure 10 ($\tau_d = 1.2$).

5 Suppressing Noisy Lines

All the lines that are extracted by the line extractor will be asserted in a database for manipulation by the higher level modules. The smaller the number of lines, the more efficient the higher level modules will be. Consider the systems that detects objects in aerial images by first identifying their boundaries. Ideally, the low level module must only extract the line segments that correspond to physical object edges. However, we find that a significant amount of the lines extracted (greater than 50%) is just pure clutter. Based on our observation of aerial images, we have developed two procedures to get rid of noisy line segments. In the first procedure we retain only those line segments that meet atleast one of two criteria. All line segments whose average contrast is greater than a threshold τ_c are retained. The idea is that physical object edges exhibit a reasonable contrast. Also those lines with average contrast less than τ_c, but whose length exceeds τ_l are also retained. Since long edges have strong physical significance, this will help retain some object edges that failed the first criteria. In the second procedure we delete all those line-segments that are less than certain length τ_{l_1} and do not have a long line segment (length $> \tau_{l_2}$) within a 7×7 neighborhood of both the end points. The idea is that physically significant edges occur in close proximity. Also short line segments have a high probability of being clutter. So any short edge that is not close to a long edge has a weakened evidence of being physically significant. The *label image* is again used as a spatial index to efficiently search for proximate line segments. These two procedures when applied to the lines in Figure 10 reduce the number of lines to 836 ($\tau_c = 41$, $\tau_l = 3$, $\tau_{l_1} = 4$ and $\tau_{l_2} = 6$). This order of reduction is typical and leads to significant savings in memory as well as computational time for the higher level modules. The final result is shown in Figure 11. As a comparison to an existing method we present the line segments that were detected by the Nevatia-Babu Line-Finder [1] in Figure 12. The line extractor described in this paper has been successfully used as the front end of a system that detects buildings in aerial images [5].

Figure 6: A typical aerial image.

References

[1] R. Nevatia and K. R. Babu, Linear feature extraction and description, *Computer Graphics and Image Processing*, 13, pp. 257–269, 1980.

[2] J. B. Burns, A. R. Hanson, and E. M. Riseman, Extracting straight lines, *IEEE Trans. Patt. Anal. and Mach. Intell.*, PAMI–8, pp. 425–455, 1986.

[3] Y. T. Zhou, V. Venkateswar, and R. Chellappa, Edge detection and linear feature extraction using a 2-D random field model, *IEEE Trans. Patt. Anal. and Mach. Intell.*, PAMI–11, pp. 84–95, January 1989.

[4] J. F. Canny, A computational approach to edge detection, *IEEE Trans. Patt. Anal. and Mach. Intell.*, PAMI–8, pp. 679–698, 1986.

[5] V. Venkateswar and R. Chellappa, A framework for interpretation of aerial images, In *Proceedings International Conference on Pattern Recognition*, June 1990.

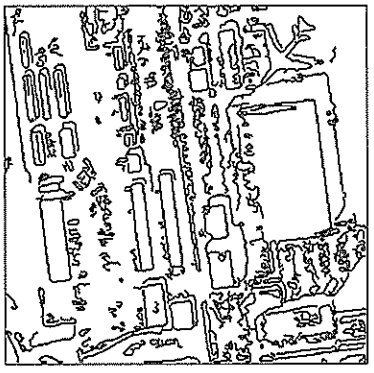


Figure 7: Edges detected by the Canny operator.

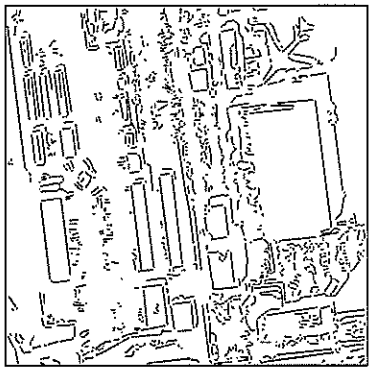


Figure 8: Set of lines after the scan and label process.

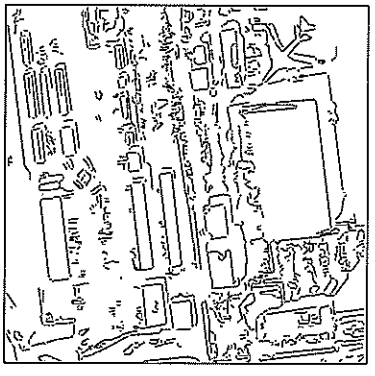


Figure 9: Set of lines after linking single-pixel-support lines.

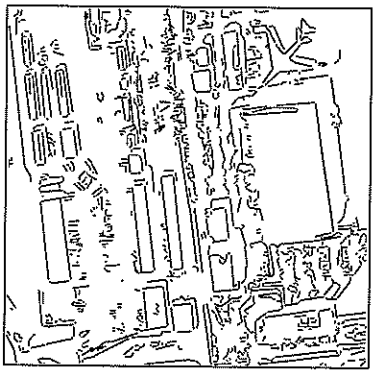


Figure 10: Set of lines after merging collinear lines.

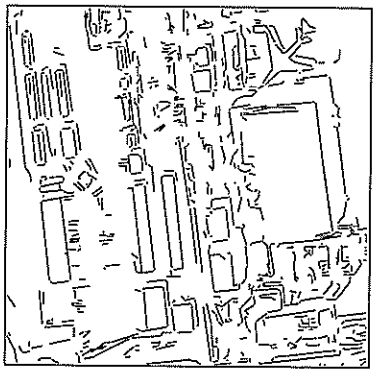


Figure 11: Set of lines after suppressing noisy lines.

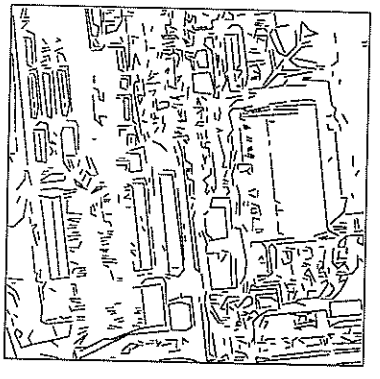


Figure 12: Set of lines obtained by the Nevatia-Babu Line Finder.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A NEW IMPROVEMENT ON LINEAR ASSOCIATIVE MEMORIES

Sheng-Wei Zhang Zhong-Kai Yang Li-He Zou

Department of Information and Control Engineering, Xi'an Jiaotong University, Xi'an, P. R. China*.

In this paper, based on the regression theory, a new method is proposed by introducing the Ridge estimator into the associative memories. Biased associative matrices for rejecting white noise and colored noise are then obtained. Theoretical analysis and computer simulations show that this model has a better error reduction than many others'.

I. INTRODUCTION

An associative memory can be defined as a mapping from a finite set of vectors to another. The principle of the associative memory resembles that of human brain [1-2], in which information is stored distributely in the whole memory areas and retrieved on its content rather than its address. Recently, associative memories have been attracting more and more attention in signal processing and pattern recognition fields [3-4].

In finite dimensional linear space, the problem can be formulated as to design a transform matrix M of dimensionality MxN on two given sets of K vectors $\{\underline{x}_i \in R^N, 1<i<K\}$, $\{\underline{y}_i \in R^M, 1<i<K\}$. When the memory is stimulated with a noisy vector $\underline{x}_i+\underline{n}_i$, its output $M(\underline{x}_i+\underline{n}_i)$ will be the closest to $\underline{y}_i$. By least Square criterion, it is

$$\min_{M} E\{\|Y-M(X+N)\|^2\}$$
$$=\min_{M} E\{\mathrm{trace}[Y-M(X+N)][Y-M(X+N)]^t\} \quad (1)$$

where X, Y, N are NxK, MxK and NxK matrices defined by the following eqs.

$$Y=[\underline{y}_1\ \underline{y}_2\ \cdots\ \underline{y}_k] \quad (2)$$

$$X=[\underline{x}_1\ \underline{x}_2\ \cdots\ \underline{x}_k] \quad (3)$$

$$N=[\underline{n}_1\ \underline{n}_2\ \cdots\ \underline{n}_k] \quad (4)$$

X, Y and N are called the key matrix, data matrix and noise matrix, respectively. E is the expectation operator. In this paper, boldfaced upper case letter denotes matrix, underline letter denotes vector, t represents the transposition of vector. For autoassociative memories (AAMs), Y=X and the output is an input key vector reduced noise. For heteroassociative memories (HAMs), the output denotes the class of the input vector. In general, AAMs are employed for noise reduction and error correction and HAMs are used to provide decisions.

The optimal linear associative memory proposed by T. Kohonen [1] is a solution of (1), which has the form:

$$M=YX^+ \quad (5)$$

where + is the Moore-Penrose inverse of a matrix. It is one of the most interesting associative models because it is very simple and it has close connection with neural networks and learning. However, the often occurred ill-conditioning of X limits Kohonen solution to be used in many practical applications. Various improvements have been proposed. Among those is that proposed by Murakami [5], which is based on the space reduction, and can easily perform a stable association even when input key vectors are degraded by noise. But its association error is not always minimum.

In this paper, we regard the construction of M as a problem of parameter estimation, and propose a new associa-

*This Research was supported by the Natural Science Foundation of China.

tive model by introducing the Ridge estimator from regression theory. By using the singular-value decomposition (SVD), the parameter in the new model can be estimated. Ridge estimators for rejecting white noise and colored noise are analyzed in section II and section III, respectively. Computer simulations show in section IV that our model is superior to Murakami's.

II. REJECTING WHITE NOISE

The associative memory is a system that allows the recall of data vector or its approximation from a corresponding (or degraded) key vector $\underline{\hat{x}}_h$, h=1, 2,.., K, by the operation,

$$\underline{z}_h=\mathbf{M}\underline{\hat{x}}_h \tag{6}$$

It is assumed that the degraded input key vector $\underline{\hat{x}}_h$ is a noise added key vector,

$$\underline{\hat{x}}_h=\underline{x}_h+\underline{n}_h \tag{7}$$

where the additive noise $\underline{n}_h$ is an N-dimensional random vector. In the white noise case, the noise matrix $\mathbf{N}$ with $\underline{\hat{n}}_h$ as its column satisfies the following property:

$$E[\mathbf{N}\mathbf{N}^T]=K\sigma^2\mathbf{I}, \quad \text{and} \quad E[\mathbf{N}]=\mathbf{0} \tag{8}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{0}$ is the zero matrix.

We introduce the ridge regression estimator $\mathbf{X}^*=(\mathbf{X}^t\mathbf{X}+\beta\mathbf{I})^{-1}$ in place of the Moore-Penrose inverse $\mathbf{X}^+$ in eq. (5) and get

$$\mathbf{M}=\mathbf{Y}(\mathbf{X}^t\mathbf{X}+\beta\mathbf{I})^{-1}\mathbf{X}^t \tag{9}$$

If $\beta=0$, then $\mathbf{M}=\mathbf{Y}\mathbf{X}^+$, the proposed associative memory model is identical with Kohonen's. Selecting the value of β, we have to make a compromise between the noise rejection effects and bias introduced. By a reasonable selection of β, the ridge estimator is in general considered better than the least square one [8]. Estimation of the parameter β in this model is based on the following theoretical analysis.

By using singular value decomposition (SVD), the key matrix $\mathbf{X}$ of (3) can be expressed as follows:

$$\mathbf{X}=\sum_{i=1}^{L}\alpha_i^{\frac{1}{2}}\underline{p}_i\underline{q}_i^t \tag{10}$$

where L is the rank of $\mathbf{X}$ and α_i (i=1, 2,...L) are the nonzero eigenvalues of $\mathbf{X}\mathbf{X}^t$ (or $\mathbf{X}^t\mathbf{X}$) such that

$$\mathbf{X}\mathbf{X}^t\underline{p}_i=\alpha_i\underline{p}_i \tag{11}$$

$$\mathbf{X}^t\mathbf{X}\underline{q}_i=\alpha_i\underline{q}_i \tag{12}$$

where $\underline{p}_i$ and $\underline{q}_i$ are left and right singular vectors of $\mathbf{X}$, respectively. It is assumed that the eigenvalues α_i are ordered so that $\alpha_i\geq\alpha_{i+1}$

Substituting $\mathbf{X}$ in (9) by (10), we have,

$$\mathbf{M}=\mathbf{Y}\sum_{i=1}^{L}\alpha_i^{\frac{1}{2}}(\alpha_i+\beta)^{-1}\underline{q}_i\underline{p}_i^t \tag{13}$$

The association error, denoted as E_n, is defined as

$$E_n=E\{\|\mathbf{Y}-\mathbf{M}(\mathbf{X}+\mathbf{N})\|^2\} \tag{14}$$

By the eq. (8), it turns out

$$E_n=\|\mathbf{Y}-\mathbf{M}\mathbf{X}\|^2+E\{\|\mathbf{M}\mathbf{N}\|^2\} \tag{15}$$

Let the first and the second terms of (15) be E_{n1} and E_{n2}, respectively. The E_{n1} does not contain a random variable. From (10)-(13), it can be found that

$$E_{n1}=\sum_{i=L+1}^{K}\|\mathbf{Y}\underline{q}_i\|^2+\sum_{i=1}^{L}\beta^2(\alpha_i+\beta)^{-2}\|\mathbf{Y}\underline{q}_i\|^2 \tag{16}$$

where, q_{L+1}-q_K are K-dimensional column vectors that are orthonormal to q_1-q_L.

In the same way, the E_{n2} can be obtained as

$$E_{n2}=\sum_{i=1}^{L}K\sigma^2\alpha_i(\alpha_i+\beta)^{-2}\|\mathbf{Y}\underline{q}_i\|^2 \tag{17}$$

Combining the (16) and (17), we get

$$E_n=E_{n1}+E_{n2}$$

$$= \sum_{i=L+1}^{K} \|Yg_i\|^2 + \sum_{i=1}^{L} (\beta^2+K\sigma^2\alpha_i)(\alpha_i+\beta)^{-2}\|Yg_i\|^2 \quad (18)$$

When $\beta \to 0$, the association error, the error of proposed model of (9), is identical with the association error E_k of Kohonen's model [5].

$$E_k = \sum_{i=L+1}^{K} \|Yg_i\|^2 + K\sigma^2 \sum_{i=1}^{L} \alpha_i^{-1}\|Yg_i\|^2 \quad (19)$$

From (18) and (19), it is understood that parameter β should be chosen properly such that

$$(\beta^2+K\sigma\alpha_i)(\alpha_i+\beta)^2 \le \alpha_i^{-1}K\sigma^2 \quad (20)$$

From (20), we can show that the sufficient condition for this model to be superior to Kohonen's is

$$\beta \le 2K\sigma^2(1-K\sigma^2\alpha_i^{-1}) \text{ for all } \alpha_i > K\sigma^2 \quad (21)$$

There are always some $\alpha_i > K\sigma^2$ as long as the noise is not overwhelming. Suffice it to choose β by

$$\beta = 2K\sigma^2(1-K\sigma^2/\alpha_1) \quad (22)$$

Since the right hand side of eq. (21) monotonically decreases with respect to α_i and α_1 is the largest eigenvalue. Furthermore, due to the inequality

$$\text{trace}(X^tX) = \sum_{i=1}^{L} \alpha_i > \alpha_1 \quad (23)$$

a simpler formula which does not require any eigenvalue computations but sacrifices a little noise rejection performance, is

$$\beta = 2K\sigma^2[1-K\sigma^2/\text{trace}(X^tX)] \quad (24)$$

Above discussions are also valid for autoassociative recall.

III. REJECTING COLORED NOISE

In this section, we first state a theorem.

Theorem: The unbiased estimator M that minimizes $f(M)=||Y-M(X+N)||^2$ when the noise matrix N satisfies

$$E(N)=0 \quad \text{and} \quad E(NN^t)=R \quad (25)$$

is

$$M=Y[X^tR^{-1}X]^{-1}X^tR^{-1} \quad (26)$$

The proof of this theorem is easy but may not be contained in such a limited paper.

This theorem is a different form of the theorem 1 in [3]. However, By introducing ridge estimator to (26), an biased estimator M can be constructed as:

$$M=Y[X^tR^{-1}X+\beta I]^{-1}X^tR^{-1} \quad (27)$$

We can show that it is superior to unbiased M of (26) by the following condition of β

$$\beta = 2x(1-1/\alpha_1)^{-1} \quad (28)$$

where α_1 is the largest eigenvalue of $X^tR^{-1}X$. A simpler form is

$$\beta = \frac{2}{1-[\text{trace}(X^tR^{-1}X)]^{-1}} \quad (29)$$

IV. COMPUTER SIMULATIONS

Computer simulations of the association recall performance of our model and Murakami's [5] (which is better than Kohonen's) are carried out. The dimensions of the key and the data vectors are set as 20. The number K of the stored pairs of vectors is varied from 5-20. The value of elements of the key vectors and the data vectors are determined by uniform random numbers in the range between -1 and +1. The noise is made by the Gaussian random numbers with N(0, 0.09). For the given stored pairs of vectors $(\underline{x}_k, \underline{y}_k)$, k=1-K, and σ^2, the parameter β of our model is calculated by use of (24) and (29), respectively. Then, the linear operator M of (9), (27) are constructed respectively. By varying the noisy input key vector $\hat{\underline{x}}_h$ from $\hat{\underline{x}}_1$ to $\hat{\underline{x}}_K$, the association outputs of the proposed model are obtained, and the following association error E_r is calculated:

$$E_r = \sum_{h=1}^{K}\sum_{j=1}^{M} (y_{hj}-z_{hj})^2/KM \quad (30)$$

where y_{hj} is the jth element of the data vector $\underline{y}_h$ to be recalled and z_{hj} is the jth element of the actually recalled association output $\underline{z}_h$ of the proposed model.

Fig. 1. shows the heteroassociative recall and autoassociative recall errors E_r of our model and Murakami's by two doted lines signed by (1) and (2), respectively.

The results show that, even by use of the simplified formulas (24) and (29), the recall error of our model is smaller than that of Murakami's.

V. CONCLUSION

We have proposed a method for improving the linear associative memories by means of ordinary ridge estimator. Other estimators from regression theory , for instance the generalized ridge estimator, may also be suggested, although its theoretical analysis, perhaps, is complicated.

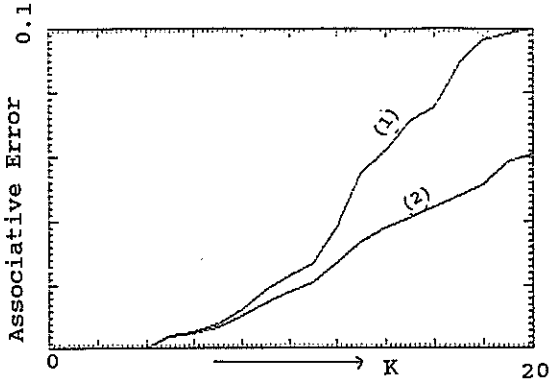


Fig. 1a Association error of heteroassociative recall versus number K of memorized pairs of vectors. (1) the model in [5]. (2) proposed model.

REFERENCES

[1] T. Kohonen, SELF-ORGANIZATION AND ASSOCIATIVE MEMORY. New York: Springer-Verlag, 1984.

[2] J. J. Hopfield, 'Neural networks and physical systems with emergent collective computational abilities', Proc. Natl. Acad. Sci. USA, vol. 79, pp.2554-2558, 1982.

[3] P. D. Olivier, "Optimal Noise Rejection in Linear Associative Memories," IEEE Trans. Syst., Man, Cybern., vol-18, pp.814-815, 1988.

[4] G. S. Stiles and D. Denq, " A quantitative comparison of the performance of three discrete distributed associative memory model," IEEE Trans. Comput., vol. C-36, pp.257-263, 1987.

[5] K. Murakami and T. Aibara, "An Improvement on the Moore-Penrose Generalized Inverse Associative Memory," IEEE Trans. Syst., Man, Cybern., vol.SMC-17, pp.699-707, 1987.

[6] H. D. Vinod and A.Ullah, RECENT ADVANCES IN REGRESSION METHODS, Marcel Dekker Inc., New York, 1981.

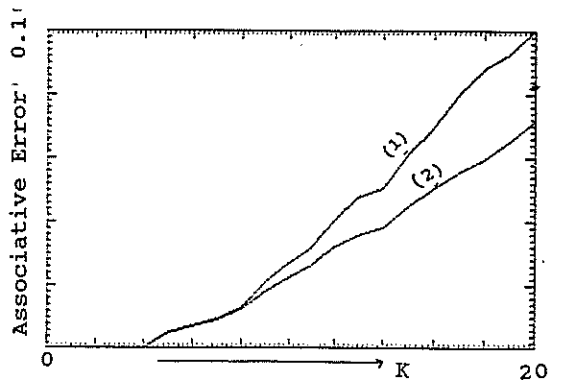


Fig. 1b Association error of autoassociative recall versus number K of memorized pairs of vectors. (1) the model in [5]. (2) proposed model.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

NUMERIC-SYMBOLIC SIGNAL PROCESSING WITH APPLICATIONS TO RADAR TRAJECTORY SMOOTHING

Mille Millnert and Peter Nagy
Department of Electrical Engineering, Linköping University,
S-581 83 Linköping Sweden

An environment for numeric-symbolic signal processing is presented. It is also show how this environment can be used to implement heuristics typically needed for signal processing tasks where human interaction is required.

1 INTRODUCTION

A fundamental issue in science is the modeling of various phenomena and observations. In signal processing a key issue is how to construct and use relevant models for signals and systems.

The focus in signal processing has traditionally been on models that can be expressed in the terms of differential or difference equations together with stochastic processes.

However, it is a well-known fact that much of the code in computer programs for typical signal processing applications is concerned with handling logical choices, exceptional cases and the like. These parts of application programs are more seldom based on "hard-core" mathematical models, but more frequently on experience and heuristic reasoning.

To develop conceptual frameworks and software environments to handle this type of problem is thus an important and challenging task. We will here report on work which attacks the problem by the means of *numeric/symbolic* signal processing, i.e., simultaneous processing of both traditional numeric information and symbolic information. With symbolic information we here mean the relations between "named" quantities, e.g., higher/lower concerning peaks in spectra, signals being more/less oscillatory and so forth. Symbolic information is easiest conceived as quantities and rules in a rule-based expert system.

A framework for numeric/symbolic signal processing gives several advantages:

- The possibility to include heuristic information.
- The possibility to make the knowledge on which the implemented algorithms are based more explicit (declarative programming).
- The possibility to automate tasks where reasoning-based decisions has to be made.

We will here describe both a software environment for numeric/symbolic signal processing, MaMiC, and an application in which the environment has been used, smoothing of radar trajectories.

Work has been conducted in the area of numeric/symbolic signal processing for two decades with different degrees of sophistication. One of the first attempts was the speech-understanding project HEARSAY-II [7, 6] at the Carnegie-Mellon University. One of the main spinoffs of this project was the black-board architecture [5, 16, 15]. The black-board technique provides a method to connect losely coupled knowledge-sources in expert system applications. Another important early system is the ocean surveillance system HASP/SIAP [17] from Stanford University.

The above mentioned systems for "symbolic/numeric" signal processing can be said to represent the AI community approach to the topic. A signal processing approach to the problem was taken by the group of Allen Oppenheim and Randal Davis at MIT. Their work is reported in for instance [10, 4, 13].

2 THE MAMIC ENVIRONMENT

MATLAB [12] has become a widely used tool for signal processing and automatic control applications. MATLAB contains the traditional numerical facilities to accomplish these tasks. In complicated signal processing and control engineering, human interaction is demanded. Such problems are hard to program entirely in MATLAB and a there is a need for a more advanced programming tool, containing parts such as expert systems. To be able to handle these problems the MaMiC programming environment has been developed [14]. The design of the programming environment was governed by two main goals:

- The numeric capabilities of the new system must not be less than that of MATLAB.
- The use of the system should be similar to the use of MATLAB (at least for numerical calculations).

These goals were achieved by interfacing Common Lisp with MATLAB. Through this interface, Common Lisp has full access to the capabilities of MATLAB. The Common Lisp programming language together with the flavors object package on the other hand is a modern and powerful tool for symbolic computation. Moreover the system was extended by inclusion of the YAPS expert system shell, see [1].

The interface to MATLAB contains the following types of functions:

- Functions for sending standard MATLAB expressions to MATLAB for execution, with facilities for exchanging data between MATLAB and Common Lisp.
- Functions for data conversions.
- Functions for controlling the MATLAB interface.

The first group of commands are important for the MaMiC system and will therefore be described below. The other two groups of commands are rather technically involved with Lisp programming and are not discussed further here, see the report [14] for details.

Calls to MATLAB are written as ordinary MATLAB expression surrounded by curly brackets. To send data from MATLAB to Common Lisp, the data are assigned to the variable @ ('commercial at' sign). Expressions in a MATLAB call that are surrounded by | (pipe sign) are Lisp expressions. These expressions are first evaluated and the result is then sent to MATLAB. Some examples:

```
<cl> {x=0:0.1:10; plot(x,sin(x))}
<cl> (setq x {@=0:0.1:10;})
<cl> {plot(|x|,sin(|x|))}
```

In the first line above $\sin(x)$ is plotted for values of x in the interval [0 10]. In the second line the lisp variable x is bound to data from MATLAB and in the third line the lisp variable x is used in a call to MATLAB.

The YAPS expert system shell [1] is a rule based expert system shell similar to OPS5 [3]. The knowledge in rule based expert systems is represented by IF–THEN statements. If the conditions in the IF part are all true, actions in the THEN part can be performed.

A typical rule looks like:

```
(p rule-1 (a -a)
          (b -b)
    test (> -a -b)
     --> (msg "a=" -a " is larger than
b=" -b N))
```

The rule checks if data in a are larger than data in b, if so this is printed on the screen.

The formula manipulation package MACSYMA [8, 9] is also added to the MaMiC system for handling symbolic mathematics. Formulas created in MACSYMA can be evaluated in MATLAB. For further information see the report [14].

3 SMOOTHING OF RADAR TRAJECTORIES

Filtering, prediction and smoothing of radar trajectories is representative for many signal processing tasks. To some aspects of the problem "optimal" methods can be applied. However, the methods require that parameters are chosen and that the methods are combined in an intelligent way. Let us here briefly review a possible solution to the problem of smoothing a radar trajectory to see what role symbolic processing can play in this context.

Figure 1 shows a radar recording of an aircraft flight. In an off-line situation the noise can be reduced by use of a Kalman smoothing filter, see [2]. Application of a

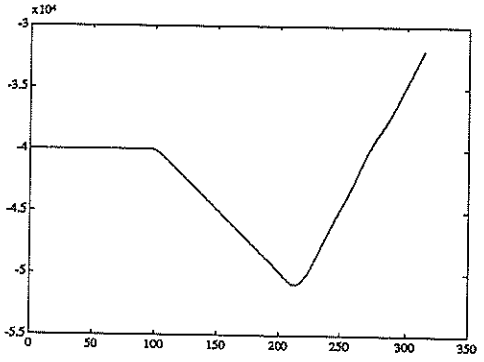


Figure 1. A recording of a aircraft radar trajectory.

smoothing filter requires a signal model. A simple but appropriate model for the signal in figure 1 is a double integrator in state space form:

$$\begin{aligned} x(t+1) &= Fx(t) + G(u(t) + w(t)) \\ y(t) &= Hx(t) + e(t) \end{aligned} \tag{1}$$

where the state vector $x(t)$ contains the positions and velocities of the two spatial coordinates. $w(t)$ and $e(t)$ are white noises-sequences accounting for random forces acting on the aircraft and measurement errors respectively. Denote their covariance matrices with R_1 and

R_2. $u(t)$ is the forces acting on the aircraft due to pilot maneuvers. This input is of-course not known to us.

The design parameters in this case are the covariance matrices R_1 and R_2 (or the ratio [1] between them). An increase in R_2 will give a smoothing filter which is of a more pronounced low-pass character. This will give a better noise reduction but it also tend to "smear" the parts of the trajectory with maneuvers. A higher value of R_1 on the other hand will give an unnecessary low noise reduction in "straight-flight" parts of the trajectory. One should observe that the best compromise for R_1 and R_2 is *not* given by their "true" values, i.e., the values that are obtained from analyzing the measurement error, the external forces due to wind turbulence etc. The reason is that the noise in (1) must model *both* those stochastic forces *and* the unknown force due to pilot maneuvers. In on-line situations it possible to handle situation by using some adaptation technique which will increase the R_1 value during maneuvers.

The quality of the smoothing can be significantly improved if the input $u(t)$ was known. In an off-line situation a natural idéa is therefor to estimate the unknown input signal. A procedure for this can consist of the following steps:

1. Smooth the signal y using a Kalman smoothing filter based on model (1). Use a value of R_2/R_1 sufficiently large to account both for the process noise $w(t)$ and the unknown input signal $u(t)$. Denote the smoothed signal $\hat{y}(t)$.

2. Make an inverse filtering through the system (1). Since the system is a double integrator this corresponds to differentiating $\hat{y}$ twice. This gives an estimate of $u(t) + w(t)$. Examples of the resulting signal for different values of R_2/R_1 are shown in figure 2.

3. Separate the segments containing noise only form the segments which contains the estimated input signal u (the "bell-shaped" peaks). Denote the signal obtained by putting the noise segments equal to zero by $\hat{u}$.

4. Use the estimated input signal in an other pass with the Kalman smoother. Since we now have an estimate of the input signal a larger R_2/R_1-value can be used.

Let us summarize the above discussion by pointing out some of the choices that has to be made.

- The ratio R_1/R_2. The choice depends on.
  - The signal to noise ratio.
  - The intended use of the smoothed signal.
  - How good an estimate of the input signal that is known
- What is noise and what is "deterministic" signal i figure 2.

[1] We will speak intuitively about R_1 and R_2 being smaller or larger although they are matrices

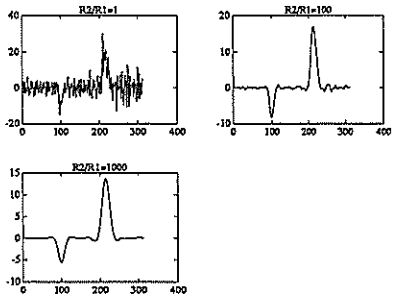


Figure 2. An crude estimate of the input signal obtained by smoothing and inverse filtering

4 AN EXAMPLE

Let us now see how the environment described in section 2. can be used to implement the heuristics needed for making the choices in the "manual" algorithm described in the previous section. We will here concentrate on the task of chosing a ratio R_2/R_1 suitable for separating the bell-shaped input signal from the noise in the crude input estimates depicted in figure 2. For details and a description of the other aspects of problem see [11].

The rules in the expert system is based on a number of features in the signals, e.g., the number of "jumps" in the differentiated $\hat{y}$-signal, the width and smoothness of the "bells" etc.

The rule base contains rules for evaluating the result of the signal processing based on the features and rules for chosing new R_2 and R_1 values.

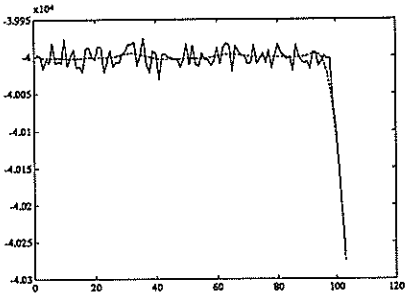


Figure 3. The trajectory and the trajectory smoothed with a Kalman smoother using an estimated input signal.

An example of a rule for chosing a new ratio R_2/R_1 is:

for smoothing

```
;;; rule for increasing R2/R1 when the current
;;; bell  is corrupted and there is a smooth
;;; one in the database
;;;
;;; IF  "current bell is corrupted"
;;; AND "there is a smooth bell in the
;;;      database"
;;; THEN "set R2/R1 to the geometric mean of
;;; the R2/R1 of corrupted one and the smooth
;;; one"
(p bell-30
        (r2/r1 -r2/r1)
        (bell corrupted - - -r2/r1x)
        (bell smooth - - -r2/r1y)
        (~ (bell corrupted - - -r2/r1z) with
           (<> -r2/r1z -r2/r1))
   test (= -r2/r1 -r2/r1x)
        (<> -r2/r1 -r2/r1y)
    --> (yaps-remove 1)
        (setq *tune-r2/r1-result*
        (sqrt (* -r2/r1 -r2/r1y))))
```

5 CONCLUSION

In the paper we have shown how an environment for numeric/symbolic signal processing can be used for automating tasks where subjective, heuristic based decisions has to be made. In the environment the heuristics are represented as rules in an expert system shell. This together with the numerical capabilities of the system provides a flexible and powerful tool for programming signal processing applications. An open and important question is however how the type procedures that conveniently are implemented in the environment can be analyzed in a systematic way.

REFERENCES

[1] E. M. Allen. Yaps: Yet another production system. Technical Report TR - 1146, Department of Computer Science, University of Maryland, 1983.

[2] B. Anderson and J. Moore. *Optimal Filtering.* Information and System Sciences Series. Prentice Hall, New Jersey, 1979.

[3] L. Brownston, R. Farrell, E. Kunt, and N. Martin. *Programming Expert system in OPS5.* Addison-Wesly Publishing Company, 1985.

[4] W. Dove. *Knowledge-Based Pitch Detection.* PhD thesis, MIT, Cambridge, Mass., 1986.

[5] R. Englemore and A. Morgan. *Blackboard Systems.* Addison-Wesley, Reading, Massachusetts, 1987.

[6] L. D. Erman, F. Hayes-Roth, V. R. Lesser, and D. R. Reddy. The hearsay-ii speech understanding system: Integrating knowledge to resolve uncertainty. *Computing Surveys*, 12(2):213–254, 1980.

[7] V. R. Lesser and L. D. Erman. A retrospective view of the hersay-ii architecture. In *Proc. 5th Int. Jt. Conf. Artifical Intelligence*, pages 790–800, Cambridge, Mass., 1977.

[8] MACSYMA. *MACSYMA Ref Guide.* Symbolics, 1985.

[9] MACSYMA. *MACSYMA User Guide.* Symbolics, 1987.

[10] E. E. Milios. *Signal Processing and Interpretation using Multilevel Signal Abstraction.* PhD thesis, MIT, Cambridge, Mass., 1986.

[11] Mille Millnert and Peter Nagy. Smoothing of radar trajectories using numeric-symbolic signal processing. Technical report, Dept of EE. Linköping University, 1990.

[12] C. Moler, J. Little, and S. Bangert. *PRO–MATLAB.* The Math Works, Inc. Sherborn, MA, 1987.

[13] C. Myers. *Numeric and Symbolic Representation and Manipulation of Signals.* PhD thesis, MIT, Cambridge, Mass., 1986.

[14] P. A. J. Nagy. Mamic : A programming environment for numeric/symbolic data processing. Technical Report LiTH-ISY-I, to appear, Dept. of Electrical Engineering, Linköping University, Sweden, 1989.

[15] H. P. Nii. Blackboard systems, blackboard application systems, blackboard systems from a knowledge engineering perspective. *The AI MAGAZINE*, (August):82–106, 1986.

[16] H. P. Nii. Blackboard systems: The blackboard model of problem solving and the evolution of blackboard architectures. *The AI MAGAZINE*, (Summer):38–53, 1986.

[17] H. P. Nii, E. A. Feigenbaum, J. J. Anton, and A. J. Rockmore. Signal-to-symbol transformation: Hasp/siap case study. *The AI MAGAZINE*, 3:23–35, Spring 1982.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

MORPHOLOGICAL RANGE IMAGE DECOMPOSITION

I.Pitas, A.Maglara

Department of Electrical Engineering, University of Thessaloniki,
Thessaloniki 54006, GREECE

In this paper a method is proposed for analyzing range images using Mathematical Morphology. More precisely, the objects contained in the range images are decomposed into simpler parts by using the morphological decomposition algorithm of grayscale images. Finally the decomposition results are used in a recognition algorithm suitable for range images.

1. INTRODUCTION

In recent years, several object recognition algorithms have been proposed [1]. At the same time the use of range images for object recognition is widely spread. However, most of the well known object recognition methods cannot be applied directly to range images. Therefore, new methods have been proposed for range image analysis. Such a method is described in [2]. Range images are described by using differential geometry leading to surface primitives that are used to build a model to be used in a 3-D object recognition algorithm.

In this paper a method will be introduced for range image analysis and coding that results to data reduction. This method is based on mathematical grayscale morphology [3-4]. It is an extension of a binary shape description and recognition scheme called morphological shape decomposition [5-6]. Furthermore, the resulting range object description will be used in object recognition in range images.

A basic notion in grayscale morphology is the so-called structuring function g(x). The symmetric structuring function $g^s(x)$ with respect to the origin is given by:

$$g^s(x) = g(-x) \tag{1}$$

Let $I_i(x)$ be a function having value 1>0 and region of support $L_i \subset \mathbb{R}^n$ such that:

$$I_i(x)=1>0 \quad x \in L_i \tag{2}$$

A function $f_i(x)$ is called morphologically *simple*, if it has the form:

$$f_i(x) = [\, I_i \oplus rg \,](x), \tag{3}$$

In the following we proceed to the description of the discrete morphological signal decomposition. The basic properties of the decomposition are presented in section 3. A range object recognition method is briefly examined in section 4. Finally simulation examples are shown in section 5 and conclusions are drawn in section 6.

2. DISCRETE MORPHOLOGICAL SIGNAL DECOMPOSITION

Mathematical morphology is a very useful instrument for decomposing an object into simpler, easier to describe elements. If f(x) is a 2-D discrete function, $x \in D \subset \mathbb{Z}^2$ and g(x) a discrete structuring function, f can be decomposed into discrete simple functions f_i of the form:

$$f_i(x) = [\, I_i \oplus ng \,](x), \quad n \in \mathbb{N} \tag{4}$$

As ng(x) we define a structuring function such that:

$$ng(x) \triangleq [g \oplus \ldots \oplus g](x), \quad n \in \mathbb{N} \quad \text{(n-times)} \tag{5}$$

Before continuing, some assumptions are made that will be used throughout the paper. Without loss of generality it is assumed that the signal f(x) is non negative. The structuring function g(x) is assumed to have g(0)>0, so that erosion is an antiextensive operation [3,4]. Also the region of support D' of the erosion $f \ominus g^s$ is limited to those points $x \in D$ for which $f \ominus g^s \geq 0$. The

notion of a *maximal* function in a discrete signal f(x) is a discrete function ng(x) for which:

$$f(x)\ominus(n+1)g^s(x)<0 \quad \forall\, x\in D \tag{6}$$

After having given the above definitions, the recursive formula for the discrete morphological signal decomposition of f(x) into its components:

$$f_i(x) = (f-f'_{i-1})_{n_i}g(x) \tag{7}$$

$$f'_i(x)=\sum_{j=1}^{i} f_j(x) \qquad i=1,2,\ldots \tag{8}$$

$$f'_0(x) = 0 \tag{9}$$

where f_i is the i-th component of f(x). Every structuring function $n_i g(x)$ has a size n_i such that $n_i g(x)$ is maximal in $f-f'_{i-1}$. This fact defines the stopping condition of the above recursive formula, which is:

$$\left[(f-f'_{i-1})\ominus(n_i+1)g^s\right](x)<0 \quad \forall\, x\in D \tag{10}$$

The approximation $\tilde{f}$ of a signal f using k clusters is given by:

$$\tilde{f}(x)=\sum_{j=1}^{k} f_j(x) \tag{11}$$

Decomposition (7-9) has finite components and thus the decomposition describes f(x) with finite detail. The derivation of the error function $e=\|f-\tilde{f}\|$ involved is not a straightforward procedure. Qualitatively, the error depends on the relative size of the structuring function and the signals details. An alternative way for the definition of a structuring function of size n could be the following:

$$ng(x)\triangleq\left[g_1 \oplus\ldots\oplus g_n\right](x),\ n\in\mathbb{N} \quad \text{(n-times)} \tag{12}$$

where $g_1, g_2, \ldots, g_n$ may be n different structuring functions. The usefulness of (12) is that by the suitable selection of the individual structuring functions the decomposition describes the function f(x) in a more appropriate manner and $\tilde{f}(x)$ converges faster to f(x). Decomposition (7-9) is equivalent to:

$$l_i(x) =\left[(f-f'_{i-1})\ominus n_i g^s\right](x) \tag{13}$$

where $n_i g(x)$ is the maximal function g(x) in $(f-f'_{i-1})(x)$. Since the step by which g(x) erodes $f-f'_{i-1}$ is finite, $l_i(x)$ is not a single-valued function of the form (2). Formula (13) is equivalent to:

$$l_i(x)=\left[\left[(f-f'_{i-1})\ominus g^s\right]\ldots\ominus g^s\right](x) \quad \text{(n-times)} \tag{14}$$

As a result, the spines $l_i(x)$ can be calculated by eroding $f-f'_{i-1}$ repeatedly. This process is terminated by (10). The component f_i can be obtained from l_i by applying n_i dilations to it, using the same structuring function g. It is clear from this discussion that $\{l_i, n_i\}$, i=1, ...,k are enough to describe an object. Thus, the above method can be very useful to an object recognition algorithm. In most practical cases, this description also leads to a significant data reduction.

3. PROPERTIES OF THE DECOMPOSITION

The decomposition and its components f_i have the following properties. Their proofs are relatively simple and are presented in [7].

1. They are antiextensive:
   $f_i(x) < f(x), \quad f'_i(x) < f(x), \ \forall\ i\in\mathbb{N}$
2. They are translation and scale invariant but not necessarily rotation invariant.
3. The signal representation $f_i, f_{i-1}, \ldots, f_1$ increases monotonically.
4. The components f_i of a signal f are simple according to (2)-(3) only for continuous signals ($D\subset\mathbb{R}^2$). In the discrete case they are not simple.

In the case of 2-D signals the rotation invariance depends on how circular the structuring function is.

The convergence of f'_i to f depends on the resemblance of f to the structuring function g. If the signal is complex then a large number of components is needed in order to attain a satisfactory representation of f by its components.

In the case of discrete signals ($D\subset\mathbb{Z}^2$) their spines l_i are not equivalued in their regions of support L_i. If the spine $l_i(x)$ is multivalued, then the calculation of $f_i(x)$ from $l_i(x)$ by using n_i dilations is a time consuming operation. This fact means that the object reconstruction

from its components by using (8) is tedious and time consuming. Fortunately a fast reconstruction method has been derived and is described subsequently. In the following, we shall drop the index i from $l_i(x)$ and we shall use the notation $l(x)$ for convenience.

Let $l(x)$ be a k-valued function in $L \subset \mathbb{Z}^2$. It can be decomposed into k two-valued functions of the form:

$$l_j(x) = \begin{cases} j & \text{if } l(x)=j \\ c & \text{elsewhere} \end{cases} \qquad (15)$$

where c is a suitably chosen negative number in $\mathbb{Z}^2$. This choice arises from the assumption that our signals are non negative. The function $l(x)$ can be reconstructed from these two-valued functions as follows:

$$l(x) = \max_{x \in L,\, l_i \in M} (l_i(x)) \qquad (16)$$

where

$$M = \left\{ l_1(x), l_2(x), \ldots l_k(x),\ x \in L \right\} \qquad (17)$$

An example of decomposing a 3-valued function with a finite domain into 3 two-valued functions, is the following:

$$\{2,4,5,5\} = \max\left[\{2,c,c,c\},\{c,4,c,c\},\{c,c,5,5\}\right]$$

The dilation of the function (16) by a single-valued structuring function is a simple and fast procedure and can be proven to be equal to:

$$[l \oplus g](x) = \max_{l_i \in M} \left[[l_i \oplus g](x) \right] \qquad (18)$$

The calculation of each of the dilations $l_i \oplus g$ is very simple. In the case of a single-valued square structuring function g the dilation of a point function l_i is equivalent to the union of shifted squares. Thus the dilation of a finite domain function l by a square structuring function is easily derived by first finding each individual point dilation and then merging them according to (18).

4. OBJECT RECOGNITION IN RANGE IMAGES

A method for object recognition in range images will be described in this section. The mathematical methods used are the morphological decomposition algorithm and the morphological correlation M_{fh} [8] or classical correlation C_{fh} given respectively by:

$$M_{fh}(\vec{k}) = \sum_{\vec{n} \in \mathbb{Z}^2} \min\left[f(\vec{n}+\vec{k}), h(\vec{n})\right] \qquad (19)$$

$$C_{fh}(\vec{k}) = \sum_{\vec{n} \in \mathbb{Z}^2} f(\vec{n}+\vec{k}) \cdot h(\vec{n}) \qquad (20)$$

where $\vec{k}=(k_1,k_2) \in \mathbb{Z}^2$ in both cases.

The recognition procedure consists of two phases, the learning phase and the recognition phase. In the first phase a pattern library is built. It consists of records each describing the morphological decomposition components of a range object F and a few quantities that identify an object, such as the area that it occupies [1]. Each library record is a structure in C programming language terminology. In the second phase a reference prototype f of the range object F is reconstructed by using the information stored in the reference library. The objects h and f are then submitted to the matching process. The matching is carried out by using the correlation function of (19) or (20). The outcome of the correlation M_{fg} or C_{fg} is used to determine if the two objects are or are not the same according to a decision rule.

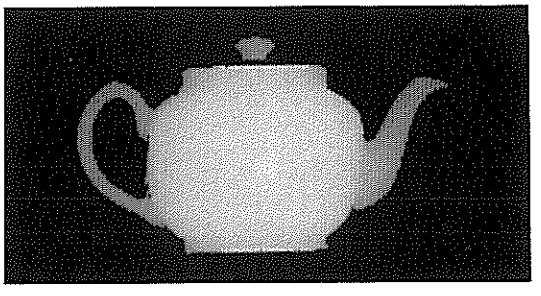


Figure 1. Original range image POT.

5. SIMULATION EXAMPLES

The morphological signal decomposition algorithm has been programmed in C language. All range objects, used, have undergone preprocessing such as median filtering, background removal etc. [4].

The result of preprocessing the range object POT in Figure 1 is shown in Figure 2.

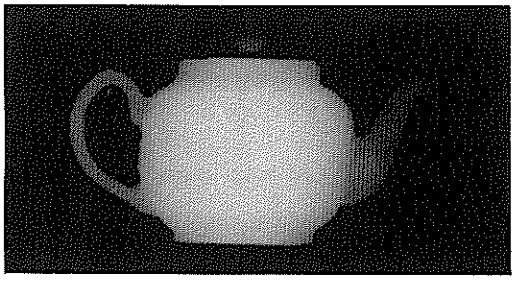


Figure 2. Range image POT after preprocessing.

The decompositions of POT have been tested towards scaling, translation and rotation, using two kinds of structuring functions. The structuring functions used were 2-D functions with square or circular cross-sections. In the first case the decomposition was found to be scale and translation invariant, while in the second case it was also rotation invariant. The decomposition of POT using five components and a structuring function having a circular cross-section is shown in Figure 3. While the algebraic difference between POT and its decomposition (Figure 3) is given in Figure 4.

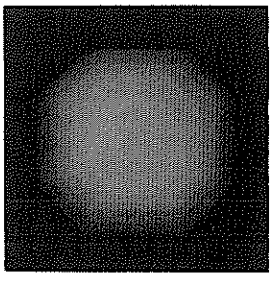


Figure 3. The first five components of the decomposition of POT.

The decompositions were used to build a reference library for use in the object recognition procedure. The recognition algorithm has been programmed in C language and tested using several different objects from range images.

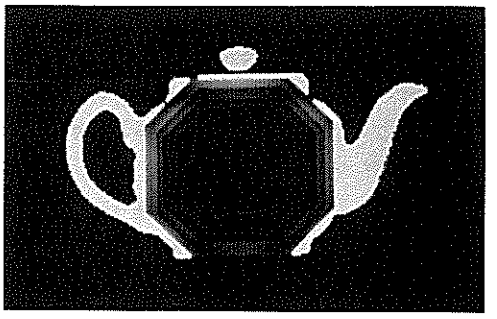


Figure 4. Algebraic difference between POT and its decomposition (Figure 3).

6. CONCLUSIONS

In this paper a new method for range image analysis is introduced based on the discrete morphological signal decomposition algorithm. The proposed decomposition is translation and scale invariant and in some cases rotation invariant. By using the decomposition the objects may be expressed with as much detail as needed. The individual object components are directly described by their spines and thus allowing data reduction. The results of the decomposition were used in range object recognition.

REFERENCES

[1] Levine,M.D., Vision in Man and Machine (Mc Graw-Hill,1985).

[2] Besl,P.J, Surfaces in Range Image Understanding (Springer-Verlag, 1988).

[3] Serra,J., Image Analysis and Mathematical Morphology (Academic Press, 1982).

[4] Pitas,I., Venetsanopoulos,A.N., Nonlinear Digital Filters: Principles and Applications (Kluwer Academic, 1990).

[5] Pitas,I., Venetsanopoulos,A.N., Morphological shape decomposition, in: IEEE Transactions on Pattern Analysis and Machine Intelligence (IEEE, Jan. 1990).

[6] Pitas,I., Sidiropoulos,N.D., Pattern Recognition of Binary Image Objects using Morphological Shape Decomposition, under review: Computer Vision, Graphics and Image Processing.

[7] Pitas,I., Morphological signal analysis, submitted to IEEE Transactions on Acoustics, Speech and Signal Processing, 1989.

[8] Maragos,P., Optimal Morphological Approaches to Image Matching and Object Detection, in: Proc. IEEE International Conference on Computer Vision (IEEE, 1988) pp. 695-699.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

AN ULTRASONIC ROBOT EYE FOR THREE-DIMENSIONAL OBJECT RECOGNITION USING NEURAL NETWORK

Sumio Watanabe and Masahide Yoneyama

RICOH Research and Development Center
16-1, Shin-ei-cho, Kohoku-ku, Yokohama, 223 Japan.

For the purpose of constructing an ultrasonic robot eye in air, we have developed a new system which combines acoustic imaging with neural network. By this system, using simple shape objects for teaching patterns, images of unknown objects can finely be reconstructed. The structure of the system and an experimental result are reported.

1. INTRODUCTION

Three-dimensional object recognition is an important technique to develop an intelligent robot eye. Despite some researches into the use of television cameras, no definite method has emerged. We are developing a new system based on acoustic imaging, and in this paper, report a recent result.

The method using ultrasonic waves has three advantages to object recognition. First, since sound travels slowly, the phase information can easily be measured, resulting in the direct calculation of an object's three-dimensional image. Second, the object's shape can be recognized despite differences in color, transparency or luminescence, allowing identification of glass or metal objects. Last, objects can be recognized even in dark or smoky environment.

The practicability of ultrasonic recognition has been limited, however, by its low image resolution. This is a result of a combination of factors, including long wavelength, a limited number of receivers, a small aperture, and attenuation due to the propagation though air. To overcome these problems and increase the practicability of acoustic imaging, we devised a new system to combine acoustic imaging with neural network [1],[2]. Experimental results showed that three-dimensional objects can be identified and locations of them can be estimated.

In this paper, it is reported that, using simple shape objects for teaching patterns, images of unknown objects can finely be reconstructed by a small set of receivers.

2. THREE-DIMENSIONAL ACOUSTIC IMAGING

Let us consider a plane wave with an angular frequency ω

$$p_{in}(\vec{r},t) = \Theta(ct-|\vec{r}-\vec{r}_0|)\exp(j\vec{k}_{in}\cdot(\vec{r}-\vec{r}_0)-j\omega t) \quad (1)$$

$$\vec{k}_{in} = (k\sin\theta, 0, -k\cos\theta)$$

illuminating an object at an angle θ, where $\vec{r}_0$ is a place of a transmitter, c is a sound velocity, k is a wave number, and $\Theta(x)$ is 0 if $x<0$, or 1 if otherwise. The places of a receiver and an object are denoted $\vec{r}=(x,y,z)$ and $\vec{r}'=(x',y',z')$. When a reflection coefficient of the object is $\xi(x',y')$ and a surface function is

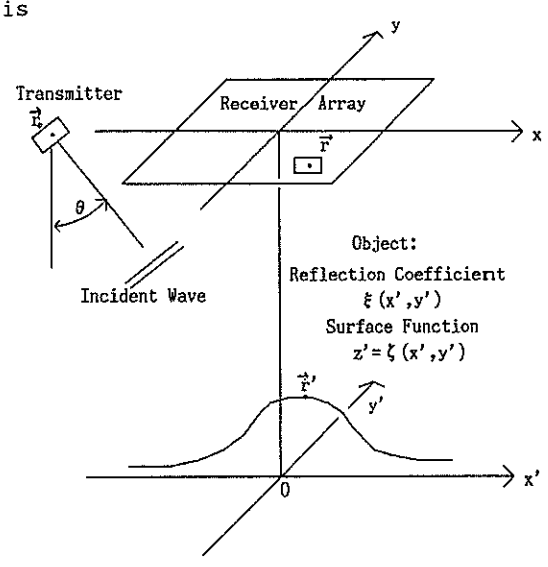


Fig.1 Arrangement of Object, Receiver Array, and Transmitter

$$z' = \zeta(x',y') \quad (2)$$

the Length $L(\vec{r},\vec{r}')$ from the transmitter, $\vec{r}_0$, to the receiver, $\vec{r}$, via object's surface point $\vec{r}'$ is

$$L(\vec{r},\vec{r}') = |\vec{r}'-\vec{r}_0|+|\vec{r}'-\vec{r}| \quad (3)$$

The sound pressure of the scattered wave at the receiver's location, $\vec{r}$, and time t is given by

$$P(\vec{r},t) = \frac{jexp(jkr)F(\vec{r})}{4\pi r}\int_{-\infty}^{+\infty}dx'\int_{-\infty}^{+\infty}dy'\exp(j\vec{V}\cdot\vec{r}') \times \xi(x',y')\Theta(ct-L(\vec{r},\vec{r}')) \quad (4)$$

where

$$\vec{r}' = (x',y',\zeta(x',y'))$$

$$\vec{V} = (V_x,V_y,V_z)$$

$$V_x = -k(\frac{x}{r} - \sin\theta)$$

$$V_y = -k(\frac{y}{r})$$

$$V_z = -k(\frac{z}{r} + \cos\theta)$$

$$F(\vec{r}) = \frac{|\vec{V}|^2}{V_z}$$

Focusing on the origin causes $|\vec{r}'|$ to be small enough that $L(\vec{r},\vec{r}')$ can be approximated by $L'(\vec{r},\vec{r}') = r+r_0-(1+\cos\theta)\zeta(x',y')$. Then, by replacing t by $T+(r+r_0)/c$, and using inverse Fourier transform, it follows that

$$\xi(x',y')\Theta(cT+(1+\cos\theta)\zeta(x',y'))\exp(-jk(1+\cos\theta)\zeta(x',y'))$$

$$= \frac{(kz)^2}{\pi}\exp(-jkx'\sin\theta)\int_{-\infty}^{+\infty}dx\int_{-\infty}^{+\infty}dy\frac{P(\vec{r},T+(r+r_0)/c)}{jF(\vec{r})r^3\exp(jkr)} \times\exp(j\frac{k}{r}(xx'+yy')) \quad (5)$$

The absolute value of the left hand side of eq.(5) is equal to the reflection coefficient $\xi(x',y')$ if $cT+(1+\cos\theta)\zeta(x',y')>0$, or 0 if otherwise. Therefore, by the relation $z' = -cT/(1+\cos\theta)$, three-dimensional image of the object is obtained.

3. NEURAL NETWORK

Using the above method, a 3-dimensional acoustical image of object is calculated. However, this image is distorted because of long wavelength, limited number of receivers, and small size of aperture. To reconstruct a fine image from this image, a neural network is used.

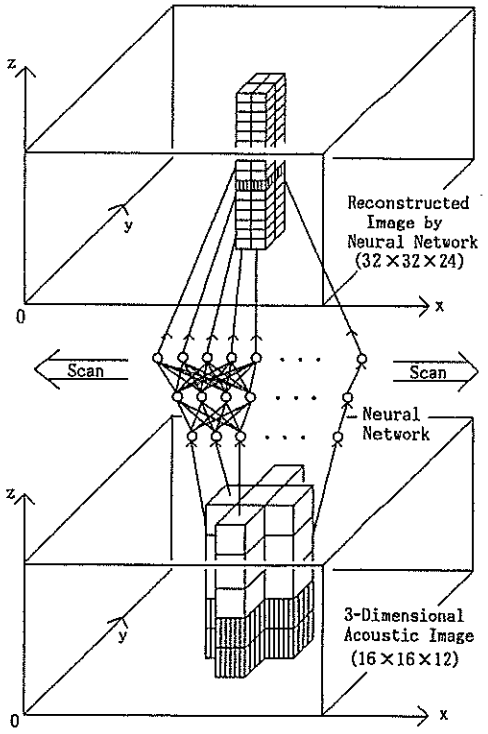


Fig.2 Neural Network Structure

A 3-layered feedforward neural network is used for image reconstruction.

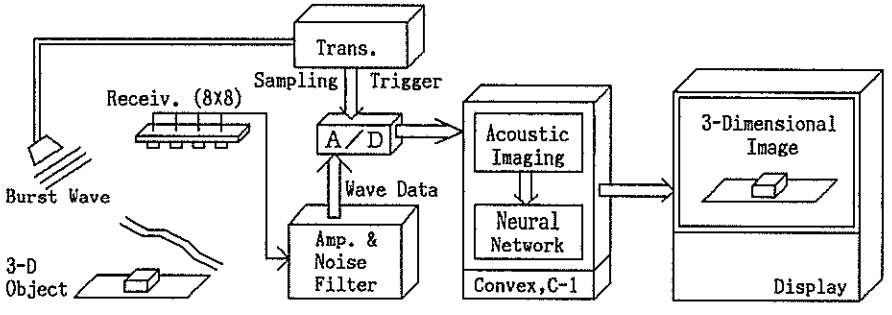


Fig.3 System Block Diagram

Fig.2 shows the structure of the neural network used for image reconstruction. It is a three-layered feedforward neural network, which can learn the correspondence between input patterns and teaching patterns by the error backpropagation algorithm. The network with 60 input units, some hidden units, 96 output units scans an acoustic image(16×16×12) and reconstructs a fine image(32×32×24). In the teaching mode, teaching value is given by 1.0 if the surface crosses the corresponding pixel, or 0.0 if otherwise.

4. SYSTEM STRUCTURE

Fig.3 shows the block diagram of an ultrasonic robot eye system. Fig.4 is its photograph. An object is placed on the surface z'=0mm, and the receiver array at z'=H=245mm. The array consists of 8×8 receivers, and the distance between each receiver is d=20mm. A transmitter sends a burst wave made of 10 cycles of sine waves onto an object at an angle, 0.278 rad. Sampling sound pressures at each 1.0 microsecond, the complex sound pressures $P(\vec{r},t)$ for 12 cycles are calculated by the inner product of scattered waves and referential cosine or sin waves.

From these 8×8×12 sound pressures, 16×16×12 acoustic images are calculated by extrapolating 0 at the outside of the array. The reconstructed area is 104mm ×104mm, by calculating $(2\pi H)/(dk)$. The reconstructed region of height is 52mm by calculating $12\lambda/(1+\cos\theta)$, where λ= 8.5mm is a wavelength.(40KHz wave.)

Neural network is simulated by software on a mini-computer,Convex,C-1.

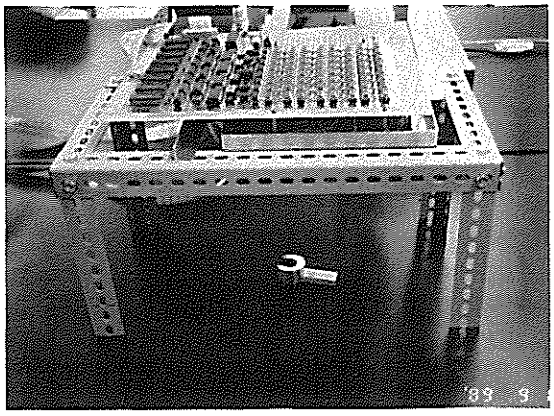


Fig.4 Photograph of System

The array has only 8 by 8 receivers.

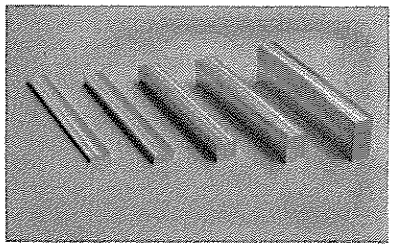


Fig.5 Photograph of Objects

These were used for teaching patterns.

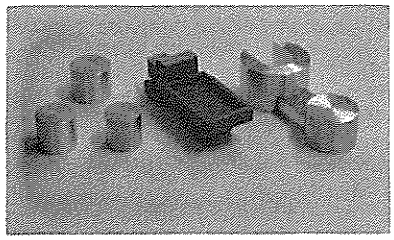


Fig.7 Photograph of Unknown Objects

Spanner, Car, Three Cylinders.
These were used for unknown objects.

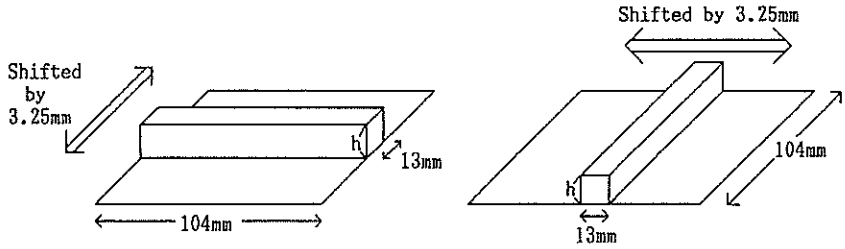


Fig.6 Collection of Teaching Patterns.

Reconstructed area of acoustic image is 104mm×104mm.
Height of object was taken as h = n×2.16mm (n = 0,1,2,...,20).

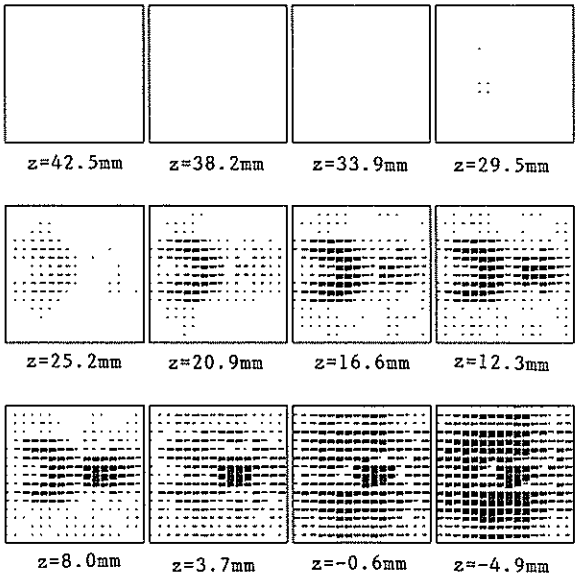


Fig.8 Acoustic Image of Spanner

Three-dimensional image is represented on each horizontal surface, z=constant.

5. EXPERIMENTAL RESULTS

Fig.5 is a photograph of objects, (rectangular blocks), which were used for teaching neural network. The size of them are 104mm×13mm×h (h=n×2.16mm, n=0, 1,2,...,20.). They are parallel shifted by 3.25mm (Fig.6), and constitute teaching patterns. Fig.7 is a photograph of unknown objects, a spanner, a car, and three cylinders.

Fig.8 is a three-dimensional acoustic image of the spanner in fig.7.

In the teaching mode, the neural network was taught 100 learning cycles. Fig.9 shows neural network outputs for learned patterns.

In Fig.10,(1.a),(2.a) and (3.a) show object's shapes obtained from acoustic images by comparing a threshold, and (1.b),(2.b), and (3.b) show neural network outputs for unknown objects. By Fig.10, it is clear that fine images can be obtained by neural network even for unknown objects.

6. Conclusion

By combining acoustic imaging with neural network, an ultrasonic robot eye was constructed. Using simple shape objects for teaching patterns, three-dimensional images of unknown objects can finely be reconstructed.

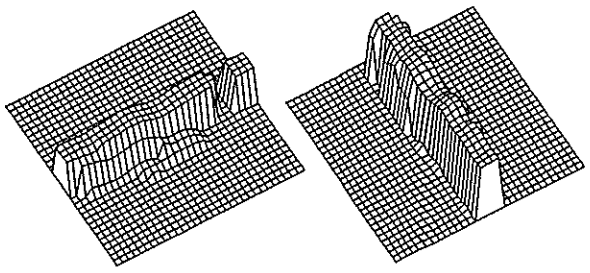
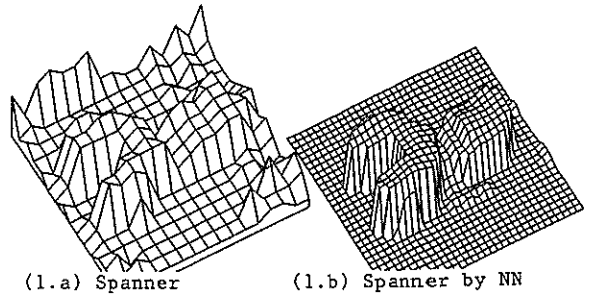
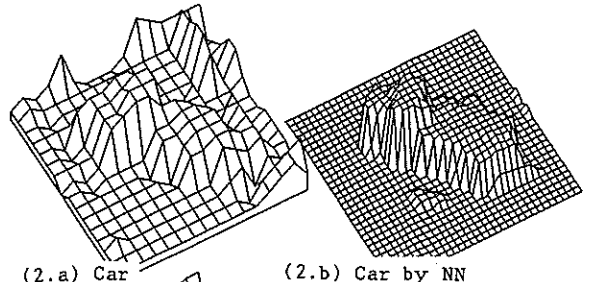


Fig.9 Neural Network Outputs for Learned Patterns





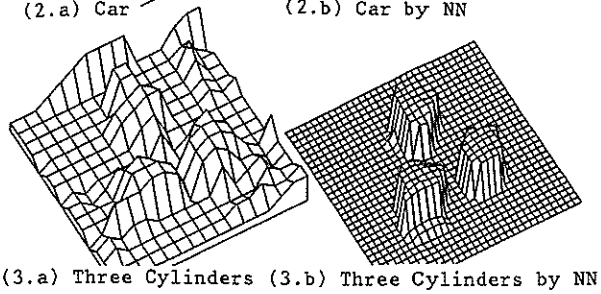


Fig.10 Reconstructed Images for Unknown Objects

(a) Objects' images by comparing threshold.
(b) Neural Network(NN) outputs for unknown objects.

REFERENCES

[1]S.Watanabe,M.Yoneyama,"An Ultrasonic Robot Eye for Object Identification Using Neural Network", IEEE 1989 Ultrasonics Symposium.

[2]S.Watanabe,M.Yoneyama,"The Ultrasonic Robot Eye Using Neural Network", Acoustical Imaging 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A Layered Neural Net for the Recognition of Image Symmetry

G. Corsini - G. Marola

Istituto di Elettronica e Telecomunicazioni, University of Pisa, Pisa, Italy

This paper presents a method for the detection of symmetry of an imag by using an analog electronic neural network. The procedure is base on the search of the global symmetry axis by means of the computatic of the local symmetry coefficient relevant to some preferentia directions of the used retina.

1. INTRODUCTION

The problem of recognizing an object in a scene is of particular interest in several fields of application as industrial automation, robotics, radar imaging and so on. A possible approach for locating objects is based on the symmetry detection of images obtained by means of suitable preprocessing techniques [1]. It consists of building up a planar figure by means of direct superposition of the model on the image under test, or also by convolution of model and image. The so obtained figure is symmetric only if the tested image is equal, even if rotated, to the model.

Clearly an unperfectly symmetric figure correspond to a partial resemblance of model and image. Hence its degree of symmetry, which may be quantified by introducing a suitable coefficient of symmetry, may be assumed as similarity coefficients between image and model.

Generally, the algorithm for finding the above symmetry coefficient, even if simple, is time consuming and is sensible to the noise or other disturbances which may corrupt the given image.

An other difficulty in finding the symmetry coefficient of a symmetric or almost symmetric planar figure arises from the fact that the tested image appears in the scene with an unknown orientation so that the position and slope of the symmetry axis may first be detected. However layered neural networks can be designed with invariance to rotation as discussed, for example, by B. Widrow [2] and others. Using Widrow's nets we may build up a layered net which is insensitive to both translation and rotation so that knowledge of position and slope of the symmetry axis is ininfluent. However, the network proposed by Widrow requires a very large number of elements in order to obtain invariance to rotation with respect to a small angular increments.

In this paper, a layered neural net having a structure particularly suitable for detecting image symmetry, is proposed. This net is composed by an input layer (hexagonal retina) and six hidden layers of neurons which perform the computation of the symmetry coefficient relative to the preferential direction of the retina. An interpolating neural net followed by a *maximum selector*, is devoted to the localisation of the symmetry axis and related coefficient of symmetry.

2. SYMMETRY, COEFFICIENT OF SYMMETRY AND LOCAL SYMMETRY

Assume we are given a planar intensity image whose gray level may be represented by the function $f(x,y)$. By denoting with $f(x,y)$ and $f(\bar{x},\bar{y})$ the intensity of two points in symmetric position about a given axis, we may introduce the following coefficient of symmetry [1]:

$$\sigma = 1 - \frac{\int_A [f(x,y) - f(\bar{x},\bar{y})]^2 \, dxdy}{2 \int_A [f(x,y)]^2 dxdy} = \frac{\int_A [f(x,y)]\,[f(\bar{x},\bar{y})]\, dxdy}{\int_A [f(x,y)]^2 dxdy} \quad (1)$$

where A is the area covered by the image.

Clearly the above symmetry coefficient is not simple to find, due to the fact that the position and slope of the symmetry axis is not known a priori. In addition a symmetrical image may be partially occluded by an unsymmetrical one, giving rise to a compound image which even if does not satisfy to the symmetry condition, preserves some evident characters of symmetry in a limited area.

For this reasons, it may be useful to introduce the concept of local symmetry i.e. a symmetry limited to a reduced area surrounding a given point or a given axis. Thus an extended area of symmetry may be detected simply by finding all points having local symmetry and by verifying that they are aligned along a straight line coinciding with their local symmetry axes, or conversely, by verifying if there is an axis whose local coefficient is close to one.

3. BUILDING UP SYMMETRIC IMAGES BY CONVOLUTION

The first step of the recognition procedure consists of creating a figure, starting from the image under test and a template, that is symmetrical if the model and the image are the same irrespective to a shift or a rotation.

There are many methods for building up symmetrical figures starting from two equal and mutually rotated images[1]. Convolution has the advantage to be independent of the centre of mass and therefore we will use it in this work.

If T is a planar (not symmetric) figure, let us denote with T* its mirror image about a vertical reference axis "a". By rotating T, θ degree clockwise and convolving it with T*, we obtain a symmetric image, as shown in Fig.1.

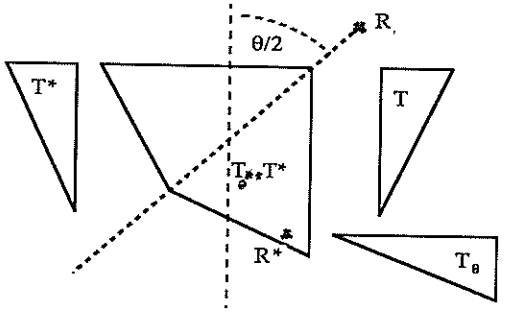


Fig.1: Symmetrical figure obtained by convolution.

From Fig.1 it is evident that the figure T_θ**T* obtained by convolution is symmetrical about an axis having slope equal to $\theta/2$ with respect to the vertical axis "a". Moreover notice that if T*, during convolution, is referred to an arbitrary point R* then the symmetry axis of T_θ**T* passes through the point R_θ of T_θ placed in the same position with respect to it. This property can be used for locating the exact position of the searched object in the image. Clearly for images generated from partially occluded objects , we obtain partially symmetric figures, for which it is necessary to use local symmetry, as introduced in the previous section.

4. FINDING THE SYMMETRY AXIS AND COEFFICIENT OF SYMMETRY BY INTERPOLATION

Let us consider a symmetrical figure with respect to an axis having slope θ_o and passing through a given point P. If we compute the symmetry coefficient of the same image making reference to an axis (also passing through the point P) with variable slope θ we obtain the plot of Fig.2.

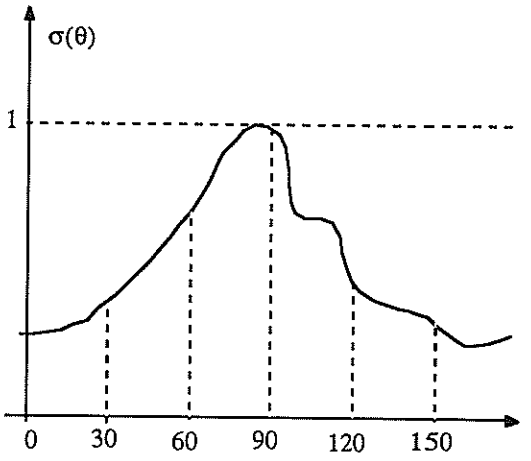


Fig.2: Symmetry coefficient relative to an axis with variable slope.

In general in a quantized environment the exact computation of the symmetry coefficient for all possible slopes is not possible save for some privileged directions i.e., for example, that having slope 0°, 45°, 90° and 135° for a square grid of pixels or that having slope 0°, 30°, 60°, 90°, 120° and 150° if we use a hexagonal grid. In this last case, and bearing in mind that in general σ is a smooth function of θ, the

knowledge of six points can allow a satisfactory reconstruction of the curve $\sigma(\theta)$ by means of a suited interpolation procedure.

Hence the procedure for detecting the symmetry can be summarized in the following steps:

a) first, we find the value of the symmetry coefficient $\sigma(\theta)$ for all privileged directions;

b) by using an interpolation technique we find the value of coefficient of symmetry for an extended set of values of the slope.

c) finally the maximum value of the coefficient of symmetry is found and compared with a given threshold in order to decide if the image under test is symmetrical or not.

5. USING LOCAL SYMMETRY AND NEURAL NET FOR THE EVALUATION OF THE GLOBAL SYMMETRY COEFFICIENT $\sigma(\theta)$.

Consider now the problem of finding the symmetry axis of a symmetrical figure as that shown in Fig.1, using a neural net. Assume we use a retina having hexagonal structure as shown in Fig.3.

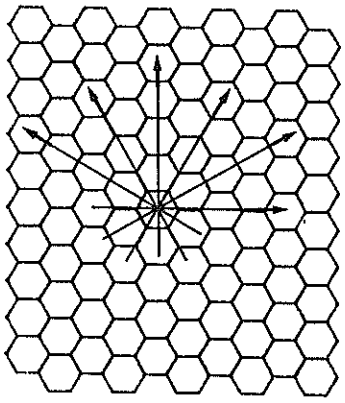


Fig.3: Retina with hexagonal structure and preferential directions

This retina is particularly convenient because it allows the test of local symmetry with respect to axes having slope equal to 0°, 30°, 60°, 90°, 120° and 150° in a very simple way. In fact, as it is shown in Fig.3, it is possible to arrange all pixels in columns parallel to each of the above six preferential directions so that the corresponding coefficient of symmetry may be found in a simple way.

From an heuristic point of view the value of $\sigma(\theta_i)$, i=1, ... ,6 can be found by selecting the maximum of the local symmetry coefficients of all the parallel columns. In fact, if the figure has the symmetry axis coinciding with one of the above columns, the corresponding local symmetry coefficient must be maximum. The most efficient way to find the column having maximum local symmetry coefficient is based on the use of a *maximum selector* implemented by an analog electronic neural network [3] shown in Fig.4.

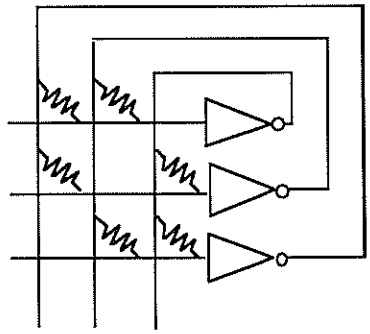


Fig.4: Neural network implementing a maximum selector

The slope of the actual symmetry axis is found, first, by reconstructing by interpolation the entiere plot of $\sigma(\theta)$ and then by selecting the maximum value of the so obtained curve. The interpolation can be accomplished by using the network shown in Fig.5.

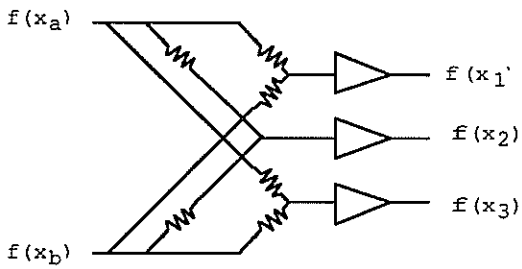


Fig.5: Example of a neural architecture for intepolation.

In this example the two inputs of the net are the two known values $f(x_a)$ and $f(x_b)$ and the three outputs are three unknown values $f(x_1)$, $f(x_2)$, $f(x_3)$ for $x_a < x_1 < x_2 < x_3 < x_b$.

Of course the maximum is found, even in this case, by using the aforementioned *maximum selector*. Finally this value is compared with a suited threshold in order to decide if the image under test is symmetrical or not.

6. CONCLUSIONS

In the paper we have presented a method to verify if an image represented in a hexagonal retina is symmetrical or almost symmetrical. The procedure is implemented by using a suitable analog electronic neural network. This network consists of an input layer (the retina), six hidden layers devoted to the computation of local symmetry coefficients along the privilegiate direction of the retina and finally a neural net which finds the maximum value of the coefficient of symmetry and compares it with a suited threshold in order to test if the image is symmetrical.

REFERENCES

[1] G.Marola" Using Symmetry for detecting and locating Objects in a Picture", *Computer Vision, Graphics and Image Processing* ,46, 179-195, May 1989.

[2] B.Widrow, R.G. Winter and R.A. Baxter" Layered Neural Nets for Pattern Recognition", *IEEE Trans. on ASSP*, Vol. 36,No. 7, 1109-1118, July 1988.

[3] H.P. Graf and L.D. Jackel, "Analog Electronic Neural Network Circuits," *IEEE Circuits and Devices Magazine*, pp. 44-49, July 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

PC-BASED SYSTEM FOR HANDWRITTEN CHARACTERS RECOGNITION WITH MULTILAYER PERCEPTRONS

Claudio Furlan, Enzo Mumolo* and Francesco Pazienti
ALCATEL FACE Research Centre, Via Nicaragua 10, 00040 Pomezia,Italy

In this paper we will describe a real-time handwritten characters recognizer based on Multilayer Perceptrons with back error propagation learning algorithm. The system has been implemented on a Personal Computer. The necessity of evaluating the performance of neural networks for handwritten characters recognition in real environment brought us to the implementation of a complete system available for end users. This approach forces to consider the variables and uncertainties not foreseen in simulation phase: the final performance will take into account the recognition efficiency, the adaptation capability to different applications and the facilities provided for user friendly operations. The network design has been made with the aid of a recently published theorem [2] that establishes a relationship between the number of regions separable by the network (M) and the number of hidden units (H). The recognition accuracy, obtained by testing the system with a data-base of digits handwritten by 6 people, was 80 %. Experimental results are reported in terms of recognition accuracies and speed of convergence of the learning phase in the different examinated conditions. Finally, the efficient implementation of the system (computational time very close to real-time on a PC AT compatible) opens to practical applications in the area of improved man-machine communication; this aspect will be briefly explored in the paper.

1. INTRODUCTION

Neural networks have demonstrated very interesting capabilities in Pattern Recognition. In this area, many different applications have been successfully developed, mostly in the speech and image domains. Research done so far has shown that neural networks can be attractive, with respect to the conventional pattern classifiers, in terms of computational requirements and robustness to noise. The paper describes a system, based on a PC with two plug-in boards for image acquisitions and for speech synthesis, that recognizes handwritten characters by using a multilayer perceptron neural network and outputs the result by means of vocal messages. The main contributions of this work have been to show that the network design can be effectively improved by using the theorem described in [2] and briefly illustrated in this paper and that a low-cost character recognizer system with good accuracy can be realized by using a PC as computational unit.
A particular characteristic of the system is that the recognized character code is sent to a speech synthetizer, based on a DSP and plugged into the PC, so that the results can be heard. This type of output allows also for a vocal guidance to the system operations. The interest to realize a system supplied with the ability of vision and speech consists of the possibility to introduce this kind of performance in a large range of practical problems. In particular the vocal assistance by synthetized words can be useful for both a feedback of the machine status and an aid to the use of the system. The network learning phase was realized on a SUN-386 workstation; the obtained weights were then downloaded on the PC for the final network operation.
The paper is organized as follows: in section 2, the general organization of the system is described and in section 3 the image processing subsystem is explained. Sections 4 and 5 deal with the neural network design and implementationa. Finally, in sections 5 and 6 the experimental results and some possible applications will be discussed.

2. OVERVIEW OF THE SYSTEM

The recognition system described in this paper has been realized in order to acquire handwritten characters via a high resolution video camera, convert them in binary images, perform a classification by neural network and produce a vocal output. The system is based on a Personal Computer containing two additional boards for input/output data processing: a frame grabber for image acquisition and a speech synthetizer for voice production. The system is functionally described in fig. 1.
The acquired image (512 x 512 pixels) is first simply processed by applying an amplitude threshold to the gray levels and then compressed by averaging to 16 by 16 pixels for reducing the amount of inputs to the neural network. The pixels are then transformed to binary patterns by using a second threshold. The compressed image is fed into the neural network which recognizes the character. The weights are computed by a back propagation algorithm running on a SUN workstation and then downloaded to the PC. The number of neural

* Current address: SINCROTRONE TRIESTE, Padriciano 99, 34012 Trieste, Italy

network outputs is equal to the number of characters to be recognized. A decision logic selects the output with maximum value and passes the information to a speech synthetizer that produces the corresponding vocal output. Speech is synthetized by using LPC-12 algorithm running on a TMS32010 processor.
All the processes, except speech synthesis, are implemented via software using "C" language.

3. IMAGE PROCESSING

In fig.1, where a general block diagram of the system is depicted, it can be noted a first block called "Image Processor".
This block takes the input signal from a video camera and pass it to the subsequent block, the "Neural Network" block. The task that this block does is to sample, acquire and process the image. The processing of the image is needed to filter-out the interference noise basically produced by the varying luminosity conditions during acquisition.
The image processing subsystem was formed by a high resolution video camera and a PC board (by Image Technology Inc.). As it will be pointed out in the following, in this system it was not made any work for reducing the errors due to possible translations, rotations or scale variations of the image, and in fact this is the main drawback of the system. In other words, the character must be carefully centered before acquisition.
The acquired image is first processed in a very simple way just setting a suitable chosen threshold to the pixel gray levels. This operation filters out several noise contributions to the image due to wrong luminosity conditions. The image is then compressed to 128 by 128 pixels by averaging four contiguous pixels. Finally, the pixels are converted to binary patterns (with size of 256 bits) through the application of a second threshold. The processing operations cannot be very complex because of the limited available computational power and the need of real-time response.

4. NEURAL NETWORK

The automatic recognition of handwritten characters has been approached, so far, by "classical" pattern recognition techniques, ranging from syntactical [3] to statistical [4] analysis. Recently, neural network pattern classifiers have been proposed as an alternative solution to the handwritten characters recognition problem [6]. By using neural network pattern recognition algorithms some advantages with respect to the classical solutions can be achieved. First of all, the computational power needed for running the recognition process by feed-forward neural network is much less than that of the classical cases. Second, the used memory is lower. The recognition accuracy, on the other hand, is about the same in the two cases. A common problem in all the written characters recognizers in the dependency from translations and rotations of the image. Even though work has been done toward the independence from distortion factors [7] (typically accomplished via suitable image processing algorithms) in this work we didn't take care of that, because our goal was to verify the network design using the results of the theorem [2] and to demonstrate the feasibility of a low-cost system for character recognition that uses only a PC as computational unit.

4.1 Network design

This phase started with the choice of the network architecture to be used. This decision was made on the basis of the networks capability and the complexity of their learning algorithms. Several architectures (i.e. linear models, single layer non linear models, Hopfield models, Hamming models and Boltzmann models, all of them having several constraints in terms of limitation of pattern types or complexity of learning procedures) have been analyzed. The multilayer feed-forward perceptron model with back error propagation [1] has been chosen because it has no limitations of pattern types, its simulation is rather simple and the learning algorithm is very powerful.
Given this choice, several parameters remain to be decided namely the number of layers in the network, the type of connections between units and the number of hidden units.
A three layer structure, with a single hidden layer, and global connections between units has been chosen on the basis of a literature analysis.
A big problem that remained open was the choice of the number of hidden units.
The choice of the number of hidden units is typically guided by intuition, experience and simulation results. A recently proposed theorem [2] sets a relationship between the number of hidden nodes, the input space dimension and the number of linearly separable regions which is related to the number of training patterns.
This relationship gives a lower bound of hidden units and training patterns to be used. In this work, the theorem results have been used in the decision of the number of hidden nodes; even if a complete verification of the theorem was not made because it was not the scope of this work, the experimental results, described in section 5, are not inconsistent with them.
The theorem states that:

In d-dimensional space, the maximum number of regions that are linearly separable using H hidden nodes is given by

$$M(H,d) = \sum_{k=0}^{d} \binom{H}{k} \quad where \binom{H}{k} = 0 \quad for\ H < k$$

For H < = d, from the above relation it can be obtained that M = 2**H.
This result agrees with the empirical observations already made by other authors and described in previous works [5].
The number of separable regions identifies the minimum number of training patterns.
The number of training patterns is usually not exactly equal to the number of separable regions because the back error propagation algorithm cannot guarantee to

reach a global minimum. In our case, we have to recognize handwritten digits (0-9) and the image is quantized with 16 by 16 binary pixels. The input space dimension is therefore 256 and at least 10 regions must be distinguishable.
With this values, it comes out that the minimum number of hidden nodes must be not less than 4.
In practice the optimum number of hidden nodes will be greater than 4, and therefore the number of training patterns greater than 16 but at least we have a minimum value of hidden units to investigate.
The important conclusion that can be drawn from the theorem result is that a relation between the number of hidden nodes and the number of separable regions, does exist.

5. EXPERIMENTAL RESULTS

During the network learning and the network operation, several results have been collected in order to find the optimum parameters of the network itself and to analyze its characteristics. A data-base formed by handwritten digits has been built; the data-base is made-up of 400 characters written manually by 6 people. The data-base has been divided into two parts: one of 300 digits used in the training phase and one of 100 digits used in the test phase. In fig. 2 is reported an example of the digits written by 6 people.
The learning phase was carried out on a SUN-386 workstation; after each learning the weights configuration was loaded on the PC where the tests were ran. As the theorem states, for d = 256 and 10 minimum separable regions, the minimum condition to be tested was H = 4 (therefore M = 16). This first trial, with a training set (T) of 20 digits, was not successful because of convergence problems. The second trial was with H = 5 (therefore M = 32) and T = 40. In this case, the accuracy was about 45 %.
The next test was made with H = 8 (M = 256) and 300 training patterns. In this case it was noted an accuracy close to 80 %. Other tests, with H = 12 and H = 20 (in both cases the training patterns remained equal to 300) shown only very little changes of the recognition accuracy. From these data, we can say that the optimum parameters lay in between H = 5 and H = 8, at least for the size of our training set; this is not inconsistent with the theorem result which predicts a number of hidden nodes greater or equal to 4. The accuracy rates in the four cases are shown in fig.3.
The following remarks can be done:

- the classification performance cannot be significantly improved by increasing the number of hidden nodes (as result of the theorem) without increasing the size of the training set
- the number of regions to be considered in the input space to solve a specific problem (and then the size of the training set) is strongly dependent on the separability of samples in different classes.

The theorem is then useful to avoid an excessive number of hidden nodes (with waste of memory) or an insufficient set of training examples.
An interesting aspect of the learning procedure is related to its convergence. The mean squared errors versus the number of learning cycles are reported in fig.4. As shown in the figure, as the number of hidden nodes increases, the number of learning cycles decreases. But this is partially compensated by the fact that as the number of hidden nodes increases, the number of connections increases and the time requested for the single learning cycle increases.
In our network 5 hidden nodes involve 1330 connections and a learning cycle time of 3.5 s whereas 20 hidden nodes increase the connections to 5320 and the cycle time to about 90 s. This is confirmed by the curves reported in fig.5 which show the mean squared error versus time. As said before, the total time needed to reach a certain mean squared error is roughly the same for different network configurations.

6. APPLICATIONS

The applications of the proposed system can be found in three main areas:

- scanning of pre-produced documents and recognition of the textual parts (handwritten characters)
- translation of handwritten documents in electronic form, allowing interactive editing
- reading of input documents by a synthetized voice.

This last field seems to be very useful for aid to blind people, as new version of the OPTACON (Optical to Tactile Converter) [8] where the output is the vocal message instead of the matrix of vibrating dots.
In its operative form the machine should include a scanner as input block: if properly designed it can represent a simple and flexible sight prosthesis.
In this direction the main goal is the optimization of the man/machine dialogue, in order to obtain good performance in terms of reading speed, ease of use, special forms of interaction and portability.
Specific interest for machine recognition of handwritten digits has been shown also for ZIP codes on pieces of mail and in general for automatic routing of parcels.
Other considered applications include multifunction workstations for office automation.
Of course, the method described in this paper is only a basis for the analysis of the general problem of cursive handwriting.
It was decided to concentrate initially on digits just to evaluate the computational proprieties of the neural networks in this area.
The progression of this work will cover the development of algorithms for recognition of lower case unconnected characters and the extension to the analysis of cursive script.
The throughput rates for classification of isolated digits in the test phase allow real time performance with software implementation of the neural network on Personal Computer: an implementation on digital signal-processing hardware authorizes the hope of exceedingly fast results of neural networks as complementary approach to the symbolic processing paradigm.

7. CONCLUSIONS

A PC-based system for handwritten characters recognition has been described. The main contributions of this paper was to shown that a recent theorem can be successfully used in the network design and that a low cost system can be developed and used in several applications. Further work should be done in the image processing stage for improving the system independence from distortion factors such as translations and rotations of the image.

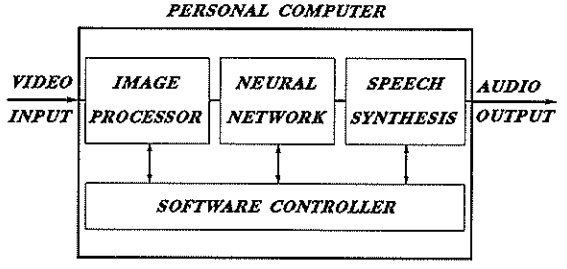


Fig.1 Block diagram

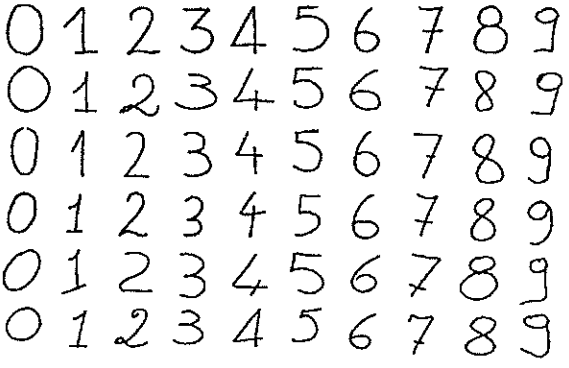


Fig.2 Examples of handwritten digits

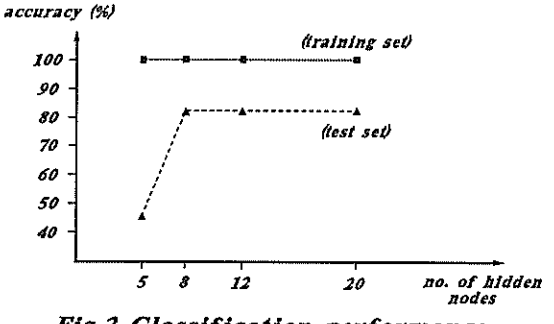


Fig.3 Classification performance

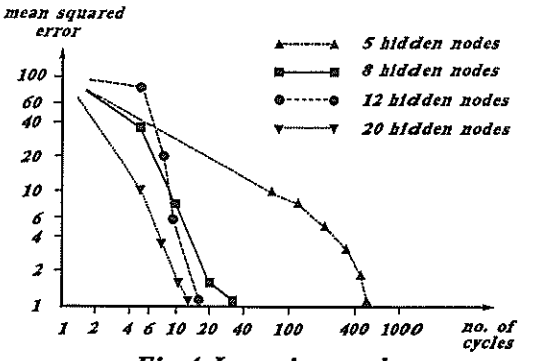


Fig.4 Learning cycles

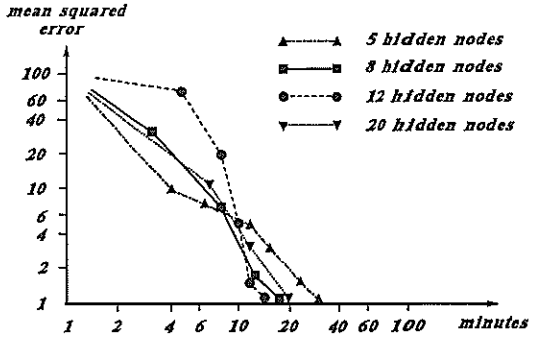


Fig.5 Learning time

REFERENCES

[1] McClelland, J.L. and Rumelhart, D.E., Parallel Distributed Processing, Vol.1 (MIT Press, Cambridge, 1986).

[2] Mirchandani, G. and Cao, W., IEEE Transactions on Circuits and Systems (1989) pp. 661-664.

[3] Zhou, H.B. and Yuan, B.Z., Knowledge Based Parallel Recognition of Handwritten Alphanumerics, in: Proc. ICASSP (Glasgow, 1989) pp. 1807-1810.

[4] Vlontzos, J.A. and Kung, S.Y., Hidden Markov Model for Character recognition, in:
Proc. ICASSP (Glasgow, 1989) pp. 1719-1722.

[5] Gorman, R.P. and Sejnowski, T.J., Neural Networks (1988) pp. 75-89.

[6] Denker, J.S. et al., Neural Network Recognizer for Handwritten Zip Code Digits, in:
Touretzky, D. (ed.), Advances in Neural Information Processing Systems (Morgan Kaufmann, 1989) pp. 323-331.

[7] Khotanzad, A. and Lu, J.H., Neural Networks (1987) pp. 625- 632.

[8] Da Ronch, A. and Spinabelli, R., Project for an Optical Character Recognizer for the blind, in:
2-nd Int. Conf. on Rehabilitation Eng. (Ottawa, 1984) pp. 587 -589.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

ANALYSIS OF EVOKED POTENTIALS BY ADAPTIVE NEURAL NETWORK

A. Uncini, M. Marchesi, G. Orlandi, F. Piazza*

Dipartimento di Elettronica e Automatica, Università di Ancona
via Brecce Bianche, 60131 Ancona, Italy
(*) Dipartimento INFOCOM, Università di Roma "La Sapienza"
via Eudossiana, 18, 00183 Roma, Italy

Networked structures have shown good capabilities for filtering non gaussian processes. Based on this approach, in the present paper the Multi-Layer Perceptron (MLP) neural network model is used for adaptive non linear filtering. The resulting structures have the advantage that they are able to learn the representation by examples, which is of great benefit when the nature of the process is unknown or is difficult to characterize.
The purpose of this paper is to analyse the possibility of using the MLP neural network for the processing of the Evoked Potentials (EP). In this case the process can be conceived as deterministic low amplitude signal (damped sine waves), corresponding to the brain response to stimuli, embedded in strongly coloured noise, the EEG background activity. Typical values of the signal-to-noise ratio are less than 0dB.

INTRODUCTION

A variety of approaches to adaptive waveform estimation have been developed in various disciplines. The linear adaptive systems such as the Adaptive Linear Combiner (ALC) and the Kalman filter have actracted the interest of many researchers because of their favourable properties. However, their use is limited to applications for which linear descriptions are appropriate. This assumption can be too restrictive in many cases where the process cannot be considered linear.

The recent resurgence of research activity in neural networks has shown the actractive properties of these systems for nonlinear processing. Moreover, the neural network can be considered a promising metaphor for the structure suggested by Palmieri and Boncelet [1] for nonlinear adaptive filtering, where an intermediate mapping function is introduced after a delay line and before a linear combiner, in order to try to linearize the input signal. The neural network approach has the further advantage that it can learn the representation of the process also when the nature of the nonlinearity is difficult to characterize, or is unknown.

The purpose of this paper is to analyse the possibility of using the MLP neural network [2] for the processing of the electrical responses of the brain to stimuli or Evoked Potentials (EP).

EP can be conceived as deterministic low amplitude signals embedded in coloured noise, the EEG background activity, which has temporal and spectral characteristics similar to the EP waveforms.This fact increases the difficulty of detecting and estimating the parameters of the EP themselves. Typical values of the signal-to-noise ratio are less than 0dB.

This characteristic of the process requires to repeatedly stimulate the subjects and to improve the low SNR by averaging a large number of trials in order to extract the response of interest. The average is a special kind of filter, the so-called "comb" filter which improves the signal to noise ratio between EP and EEG background activity by the factor $\sqrt{M}$, where M is the number of averaged trials (ensembles). A crucial assumption implicit with averaging is:

1) EEG background activity, as a stochastic signal, and EP are uncorrelated and additive. This means for the single ensemble s(t) with noise EEG n(t) and EP x(t) that:

   $s(t) = n(t) + x(t)$,

   $E[\, n(t_1)\, x(t_2)\,] = E[\, n(t_1)\,]\, E[\, x(t_2)\,] = 0$, because $E[\, n(t)\,] = 0$.

2) The EP is stationary in phase, form, latency and amplitude.

This work has been supported in part by Consiglio Nazionale delle Ricerche of Italy and in part by Ministero della Pubblica Istruzione of Italy.

The validity of hypothesis 2) is rarely verified when M increases. Since normally many averages are required and the troubles for the patient have to be minimized, many powerful signal processing techniques have been employed [3,4] in order to rapidly improve the SNR reducing the number of trials. Alternative approaches, based on minimization of the Mean-Square-Error (MSE) between the signal and the output of a filter, have recently been developed. In particular, the use of Wiener filtering [5] and optimal filters derived taking into account the nonstationarity of signal and noise [6] have been proposed. Both these methods require extensive complex calculations of covariance or correlation matrices, which presupposes knowledge of signal characteristics (such as power spectra) of a large number of tests.

In this paper a nonlinear adaptive processing technique using a MLP is proposed for processing the brain EEG evoked potentials. Experiments are performed on both synthetic and real signals.

THE MLP FILTER

The proposed MLP architecture is characterized by linear input nodes, sigmoidal hidden nodes and a single linear output node (Fig. 1). The linear output layer operates as a linear combiner and allows to circumvent dynamic range limitations. This structure can be viewed also as the cascade of a nonlinear mapping and a linear combiner.

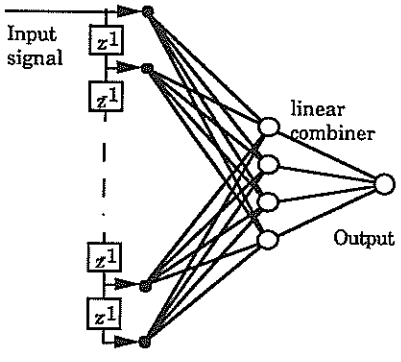


Fig.1 The MLP with a single linear output node.

The learning of the MLP is obtained by the Back-Propagation Algorithm (BPA) [2]. The proposed adaptive processing scheme is shown in Fig. 2. The processing is carried out in the following way. The input signal is one of a set of stimulated responses of a subject and the target signal is another response of the same set. The network is trained by iteratively presenting the EP ensembles of the available set. Each ensemble consists of N samples which are successively fed into the network through a sliding window wide as the number of input nodes. No average is required before the processing. After the training phase the output signal is a filtered version of the available EP ensemble. Averaging very few output ensembles improves dramatically the quality of the EP estimation.

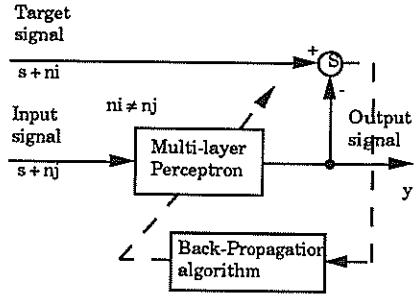


Fig.2 Scheme of the proposed adaptive architecture for EP processing.

In order to evaluate the performance of the proposed technique a set of synthetic EP signals were generated. These signals consist of one cycle of a sine wave followed by a half cycle of attenuated sine wave added to various uncorrelated noise realizations [7]. The noise samples were generated by filtering sequences of random numbers with uniform distribution through the following 11-point smoothing filter:

$$y_n = (-36x_{n-5} + 9x_{n-4} + 44x_{n-3} + 69x_{n-2} + 84x_{n-1} + 89x_n + 84x_{n+1} + 69x_{n+2} + 44x_{n+3} + 9x_{n+4} - 36x_{n+5}) / 429.$$

Each obtained ensemble consists of 96 samples (Fig. 3a, dashed line). The SNR, defined as the ratio Signal Power / Noise Variance, was assumed equal to -6 dB. The following parameters of the neural network were chosen for the experiment:

units in the input layer: = 10;

units in the hidden layer: = 6;

samples processed in the learning phase : = 576,000;

learning rate constant : = 0.01;

momentum constant : = 0;

sigmoid :f(x)=Gain*[2/(1+exp(-x*Slope)-1].
Gain = 2;

Slope $= \frac{1}{2}$,

Fig. 3a shows the filtered signal (solid line) compared with the true synthetic EP (dotted line) and Fig. 3b shows the filtered signal, averaged on 12 ensembles, compared to the signal obtained by simply averaging the input signals on the same number of ensembles.

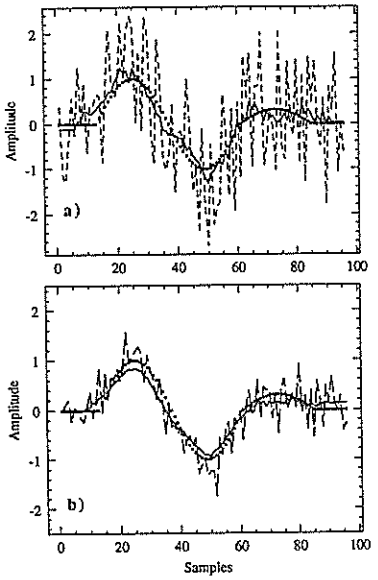


Fig.3 a) An ensemble of the measured EP (dashed line) compared to the filtered EP (solid line) and the true synthetic EP (dotted line);
b) the averaged filtered signal (solid line) compared to the signal obtained by the simple average method (dashed line) and true synthetic EP (dotted line).

To quantify the performance of the network, the mean of the MSEs over each available ensemble was calculated after the learning phase (Fig. 4). Moreover the after training output ensembles were averaged and the corresponding MSE was compared with those obtained by the averaging technique (Fig. 5). The proposed method provides good results with very few EP ensembles and without the necessity of a-priori knowledge of the signal characteristics.

In Fig. 6. a comparison between the proposed NN filter and an adaptive linear combiner with no hidden units is reported.

Another experiment on real evoked potentials was made using a neural network equal to that used with the synthetic data. A set of four evoked potentials of 96 samples (sampling frequency = 64 Hz) was used. The evoked potentials were obtained by a "simple reaction time" (SRT)-experiment, where the probands have to push a

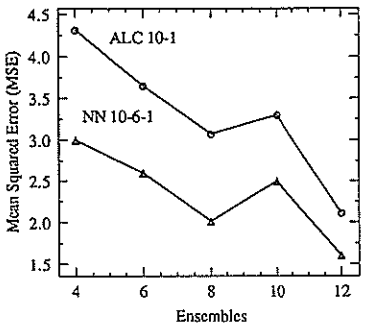


Fig. 4 Mean MSE obtained averaging over various ensembles.

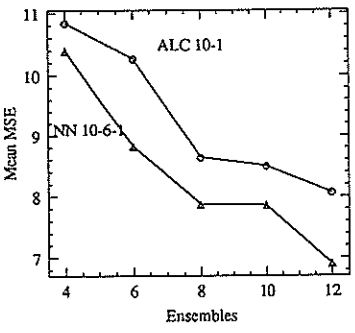


Fig. 5 MSE obtained after averaging over various ensembles.

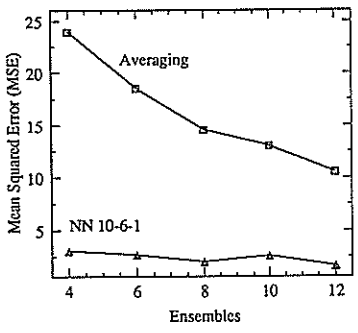


Fig. 6 Comparison between the proposed NN filter and the adaptive linear combiner (MSE computed as in Fig. 5).

button as fast as possible at every appearance of a 2.54 cm^2 quadratic flash. The stimuli onset are presented in a room with reduced luminosity for 54 ms in the center of the proband's visual field. They appear at regular intervals of 4 seconds.

The results are reported in Fig. 7, where the effectiveness of the filtering procedure on each single EP is proved by filtering the EP's through respectively a 10-6-1 NN and a 10-10-10-6-1 NN.

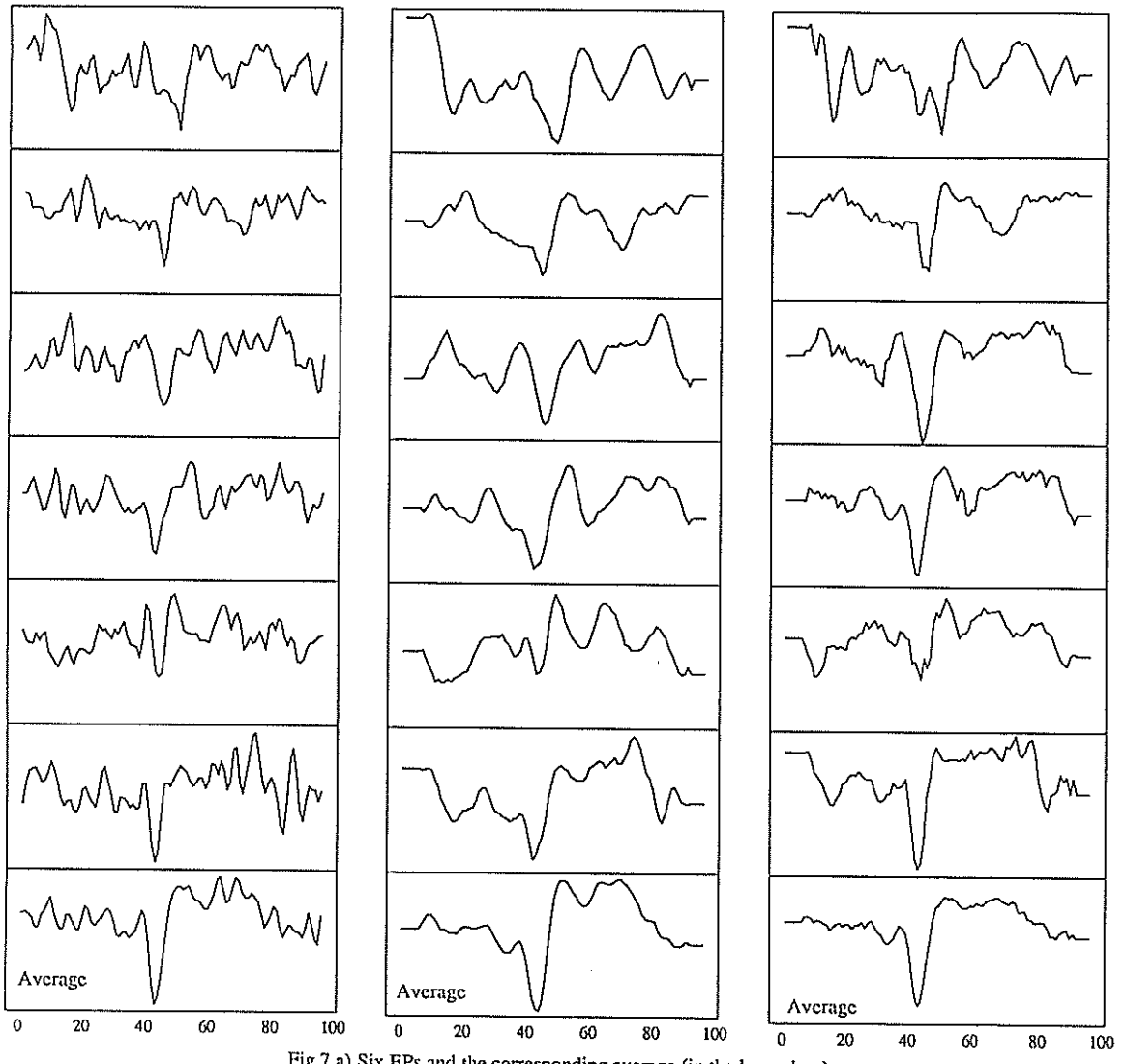


Fig.7 a) Six EPs and the corresponding average (in the lower box); b) the filtered EPs by a 10-6-1 NN; c) the filtered EPs by a 10-10-10-6-1 NN.

REFERENCES

[1] F. Palmieri and C. G. Boncelet Jr., "A Class of Nonlinear Adaptive Filters",Proc. IEEE ICASSP 88, Vol. D, pp. 1483-1486, 1988.

[2] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, "Parallel Distributed Processing, Explorations in the Microstructure of Cognition", Vol 1: Foundations. Cambridge, MA: MIT Press, ch. 8, 1986.

[3] J. I. Aunon, C. D. McGillem, and D. G. Childers, "Signal processing in evoked potential research: Averaging, principal components, modeling", Critical Reviews in Biomedical Engineering. Cleveland, OH: CRC, pp. 5:232-5:367, 1981

[4] C. D. McGillem, J. I. Aunon, and D. G. Childers, "Signal processing in evoked potential research: Applications of filtering and pattern recognitions", Critical Reviews in Biomedical Engineering. Cleveland, OH: CRC, pp. 225-265, 1981.

[5] J. P. C. de Weerd and W. J. L. Martens, "Theory and practice of a posteriori Wiener filtering of averaged evoked potentials", Biol. Cybern., Vol. 30, pp. 81, 1978.

[6] K. Yu and C. D. McGillem, "Optimum filters for estimating evoked potential waveforms", IEEE Trans. Biomed. Eng., Vol. BME-30, pp. 730-737, 1983.

[7] N. V. Thakor, "Adaptive Filtering of Evoked Potentials", IEEE Trans. on Biomed. Eng., Vol. 34, pp. 6-12, 1987.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

AN INVESTIGATION INTO THE INTEGRATION OF NEURAL NETWORKS AND HIDDEN MARKOV MODELS FOR REAL-TIME AUTOMATIC SPEECH RECOGNITION

Y. Arriola , R.A. Carrasco
Staffordshire Polytechnic, U.K.

This paper presents the results of an investigation into the integration of the Multilayer Perceptron (MLP) and the Hidden Markov Model (HMM) in order to solve both the time-domain and the spectral variations present in any automatic speech recognition (ASR) environment.

The introduction of specialised hardware devices and a flexible and efficient software organization for the implementation of the system is also presented.

Graphic and numerical results, conclusions and future research orientation are also expounded in this paper.

1. INTRODUCTION

A new speech recognition system based on the integration of three semi-independent blocks is presented: the Acoustic Processor (AP), which converts the speech signal into a set of robust acoustic features; the Multi-layer Perceptron (MLP), that maps the acoustic feature sequences into phonemes, discriminating the spectral variations from the real phonetic information; and the Hidden Markov Model (HMM), which produces a final identification of the entire utterance as a consequence of the computations of the probabilistic phonetic observations that are output by the MLP.

The integration of the MLP and the HMM is developed in order to solve both the time-domain and the spectral variation problems present in the automatic speech recognition (ASR) environment and this presents a new solution to the problem.

As a continuation of a previous paper [1] in which the main features of the system were introduced, this paper makes a relevant description of the system implementation, showing a general view of the sophisticated software and hardware main directives oriented toward a multi-processor real-time solution. In section two an overall description of the system with its theoretical background is introduced. Main system features for a flexible and efficient system are introduced in section three. Section four analyses the main configuration, organization and inter-processor communication algorithms. The main hardware features and the introduction of two specialised microprocessors are also presented in this paper. Experimental results in a real recognition system are shown in section six. The final section discusses the conclusions obtained and suggests a means by which the system can be improved in the future.

2. SYSTEM DESCRIPTION

Three main blocks can be distinguished in the system (fig. 1):

- The Acoustic Processor (AP) [1][2]
- The Multi-layer Perceptron (MLP) [1][3]
- The Hidden Markov Model (HMM) [1][4]

The three blocks are interconnected and have very different functions. Each module can be modified separately without altering the remainder due to their particularly different nature.

The AP will realize the acoustic analysis of the speech signal. The signal is input and filtered by the acoustic processor to make a perfect reconstruction of the signal so as not to lose information.

The MLP consists of simple processing units arranged in layers, connected together via weighted links. The output of each unit is the weighted sum of the outputs of the units in the previous layer. The MLPs applied to ASR have proven their efficiency in the extraction of the phonetic information from the training data, and its incorporation into the network.

The MLP can acquire very complex patterns and can extract linguistic and extralinguistic knowledge from speech, overcoming the spectral variation problem of the speech signal, despite this it cannot cope properly with time alignment variations and is largely unable to store syntactic and semantic knowledge. In order to tackle this problem, the MLP is integrated into a higher level system, the HMM.

The HMM is a doubly stochastic process with an underlying stochastic process that is not directly observable, but can be observed through another set of stochastic processes that produce the sequence of observation symbols.It can integrate syntactic and semantic information into the system and also overcome the time aligment problem. The probabilistic phonetic outputs of the MLP will become the observation inputs to the HMM, so that the model with maximum probability will be the recognised utterance.

More information and theoretical background concerning these three functional blocks can be found in the references.

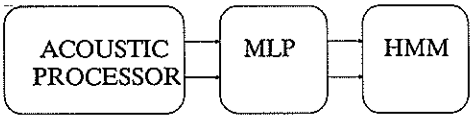


Fig. 1: Main block representation of the system

3. SYSTEM FEATURES

These are the main requirements introduced in the system:

Hard real-time process: Speech recognition is labelled as a Hard Real-Time process, that is, all activated tasks must be completed in prefixed time intervals.

Hardware / Software flexibility: Knowledge relating to the speech and speech recognition processes is still to be fully appreciated, so neither the hardware nor the software of the system can be inflexible and deterministic, because several modifications can be expected. For this reason a modular sys-

tem is proposed in this system. All modules are intercommunicated and distributed with an increased level of parallelism. Software is optimally partitioned throughout the processors allowing the programmer to modify the code corresponding to a specific processor without affecting the other processors in the system. A set of specialized processors adopting very different architectures completes a homogeneous hardware system. The whole network of processors is interlinked and organized under the supervision of the manager processor.

Multiprocessor system : It is not possible to consider the implementation of such a complex and flexible speech recognition system by using a single processor; a more complex notation must be introduced: the multiprocessor system.

All processors are configured under the supervision and management of a main processor which distributes all the processes into the processor network in the most convenient and economical way. The manager processor will always be able to ascertain the *state* of each processor and it will execute distribution plans whenever this is necessary.

All processors will be intercommunicated and interfaced with the manager processor. Data/command information is communicated through *channels* following a minimum distance criteria.

This internal mini-operative system facilitates a halcyonic and organized way of scheduling all the different processes throughout the network, and a convenient and simple interface with future external processes can be realized.

Self-organization: The definition of the recognition system will be made by the user. The vocabulary size, number of speakers, models for each vocabulary word, layers in the MLP, units per layer, and the number of tasks in the Acoustic Processor (AP) will be some of the specifications of the system. The manager processor will automatically choose the correct set of processors, activate the necessary tasks for each processor and consequently activate the optimum communication paths between them.

4. SOFTWARE IMPLEMENTATION

The system consists of two main functional blocks:

- *The low level processor*: This will enhance the speech signal and its acoustic analysis. The speech signal is input to the low-pass filter, which is interfaced to the analogue-to-digital converter. The digital representation of the speech signal is modelled in a parametric form and then is sent to the high level processor.

- *The high level processing network:* This is interfaced to the low level processor. The output of the low level processor is received as input and as a result of the application of several high level algorithms a word, phoneme or sentence will be recognized. As was previously mentioned, two different subnets of processors are distinguished: the Multi-layer Perceptron and the Hidden Markov Model system, but both are integrated into the same block for implementational reasons as will be discussed later in the paper.

4.1 Low level processing organization

When low level processing of speech is considered, it is usual to refer to its acoustic processing. The main job of the acoustic processor is to input and analyse the speech signal in a given interval of time.

The analysis techniques will be distributed throughout processes, and each process will be activated when an *event* (interruption, clock, request from MLP or HMM, ...) is produced. The AP will realize all the types of analyses over the speech signal but will only output the requested parameters. Each process will have its own *local memory (LM)* and will interchange information through the *global memory (GM)*.

The processes may be defined as *concurrent real-time processes [5]*, since all of them share the same processor.

The call to each process is placed in two priority groups, *foreground* and *background* groups. Foreground processes are high priority processes (e.g. sampling), while background processes are low priority processes(e.g. the FFT). These are the main processes for the actual AP:

FOREGROUND	BACKGROUND
Clk 1 every125 microsec. Clk 2 every 16 milisec. Sampling of the speech Input data storage	Energy computation Zero-crossing processing Autocorrelation analysis LPC analysis FFT analysis Cepstral analysis Mel-scale power spectrum

4.2 High level processing organization

The remaining two blocks, the MLP and the HMM, due to their computational nature can be easily decomposed into a parallel structure.

A set of possible tasks is allocated to each processor, but they are only in an active state when the manager processor demands this in each processor.

4.3 Architecture

In relation with the general architecture of the system, four main blocks have been defined: The acoustic processor, the MLP , the HMM and the Manager Processor with the external Interface.

The Acoustic Processor is placed into an individual processor due to its independent functional nature with respect to the rest of the processes. However, the MLP and the HMM due to their similar nature have been implemented on similar parallel networks of processors. The system is interfaced to the standard input/outputs of a personal computer and a data base where all information is stored before the system is decommisioned.

The MLP and HMM networks are considered independent since their constraints and requirements are different. Thus, any of them can be expanded while the other remains the same.

4.4 Communications

The features that are required for the communication process are:

- Absence of deadlock and lack of allocated computational

load in the processors;
- Independence of network topology or size;
- High data rate;
- Distributed routing density;
- Implementation in VLSI;

The system architecture is highly consistent with the communication specifications, but in addition, an optimum algorithm is needed to improve the communication efficiency.

The communication is established through physical bidirectional links called channels. The communication unit is referred to as the *message,* and in this particular case the format of the message is unique. This makes communication much faster avoiding any data conversion or transformation while distributing the data throughout the network.

4.5 Distribution

The distribution of the MLP and HMM processes over the network of processors is a very important problem to be solved.

For the HMM system the problem of distribution is not difficult, due to the constant repetition of the matching process for each vocabulary word. It is logical to reason that a proportioned partition of the task over the processor network is a sufficiently efficient method.

For the MLP system the problem of distribution appears more difficult; the computation process for each neural unit is based upon the output of other neural units, and thus an inefficient distribution of the neural network can have serious consequences in terms of processing time.

The waitting time T can be defined as,

$$T = \sum_{k=1}^{K} ts(k) + \sum_{k=1}^{K} \sum_{j=1}^{K} tc(k,j) \qquad (2)$$

where K is the number of processors in the network, ts(k) is the time the processor k is halted and tc(k,j) the communication time between processors k and j.

If each layer is subdivided into K sub-layers, K being the number of processors, and the computation of one sub-layer from each layer is processed by each processor, then all the processors run during the whole process time. All the processors will be synchronized at the same time and the value of ts(k) will be minimum, provided that the processors are ready to send and receive at the same time.

The drawback of this distribution algorithm is that any neural unit needs to have information concerning all output values from the previous layer, and each processor must communicate all the output values to all the remaining processors. Thus, the value of the second summation is increased.

4.6 Structure of the program

For simplicity reasons, it is assumed that all the processes can be run at any moment in any processor. This assures that the redistribution of the processes throughout the network never requires software modifications. The internal scheduler of each processor is at any moment waiting to receive a message from any of the processes activated at a particular moment. As a consequence, an action will be taken.

The scheduler, on the other hand, will receive information concerning the computational load of each process. This information will be sent to the manager processor to facilitate a better distribution of tasks.

Although the processes are functionally very different, they all can be included in this general organization algorithm. Following are the most important processes of the whole network:

- Process to generate a distribution plan.
- Send message.
- Receive message.
- Internal scheduler.
- Interface with data-base.
- MLP computation process.
- MLP training process.
- HMM computation process.
- HMM training process.

5. HARDWARE IMPLEMENTATION

The following are the main features of the hardware system :

*Due to the variability of the speech recognition tasks, reconfigurable hardware support is necessary.

*Modular system, where the components are independently programmed and linked using a highly sophisticated interface system.

*Special-purpose microprocessors:

-The TMS 320C25 DSP for the acoustic analysis of the speech signal [6]:
  -Advantageous machine cycle length (100 ns.).
  -Efficient instruction set provided.
  -Fully oriented toward DSP applications.

-INMOS TRANSPUTER as the processing element of the MLP and the HMM networks [7].
  -High degree of parallelism and concurrency
  -Flexible and robust communication system
  -Fast processing technology
  -Efficient internal scheduling organization
  -Effective software support

6. RESULTS

The experiment was to check the performance of the AP. Speech was low-pass filtered to a 3.8 KHz. cut-off frequency and sampled at an 8Khz. sampling rate. Every 16 msec., 256 sample frames were computed. The AP fulfilled the necessary ASR real-time requirements.

Two more diagrams are included (fig. 2) to show the potential of the software system. The first one shows the segmentation and labelling of the speech signal and the second one a graphic representation of certain of the neural units in the MLP. Weight values, partial error, global error learning rate, momentum, etc. are some of the parameters that can be graphically displayed and modified leading to a better recognition rate.

Table 1 shows the comparative recognition rates obtained with a simple HMM with vector quatization and an MLP-HMM model. It is noted that the recognition rate of the last one is marginally improved. The experiment was carried out in an isolated word recognition environment, using ten words vocabulary.

7. CONCLUSIONS AND FUTURE RESEARCH

Due to the inherent variability of the speech signal, current A.S.R. systems are not as efficient as desired. The implementation of a system combining the efficiency of the MLP to represent linguistic to extralinguistic information of the phonetic units of the speech signal and overcome the spectral variability of the speech signal, with the powerful syntactic and semantic representations offered by the HMM has been noted to be an applicable efficient solution.The possibility of a new modular speech recognition system has also been seriously cosidered and as result a flexible, specialized, parallel and highly organized system has been proposed.

Future research will be focused into the following main areas:

- Research into the optimum parameters of the MLP for each specific task.

- Optimum integration between the MLP and the HMM.

- Improvement of the existing communication and internal scheduling algorithms.

REFERENCES

1. Y. Arriola, R.A. Carrasco, "Integration of Multi-layer Perceptron and Markov Models for Automatic Speech Recognition", U.K. IT 1990 Conference IEE, pp.410-417, University of Southampton, March 1990.

2. L.R. Rabiner, R.W. Schafer, "Digital Processing of Speech Signals", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978.

3. N. McCulloch, W.A. Answorth, R. Linggard, "Multi-layer Perceptrons applied to Speech Technology", B. Telecom Technol. vol.6 ,no.2, pp. 131-139, April 1988.

4. S.J. Cox, "Hidden Markov Models for Automatic Speech Recognition, Theory and Applications", Br. Telecom Technol.vol. 6, no. 2, pp. 105-115, April 1988.

5. M. Ben-Ari, "Principles of Concurrent Programming", Tel-Aviv University, Prentice-Hall Internat.

6. Zbyshek, Gorzinski,"Real-Time Multi-tasking Speech Application on the TMS320", Microp. and Microsystems, vol. 11, no.3, pp. 149-156, April 1987.

7. Inmos, Transputer Reference Manual, October 1986.

TABLE 1 - Recognition rate: MLP-HMM versus traditional HMM

ITERATIONS	MLP-HMM	HMM
2000	76 %	
2500	80 %	
3000	83 %	82 %
3500	85 %	
4000	86 %	
4500	88 %	

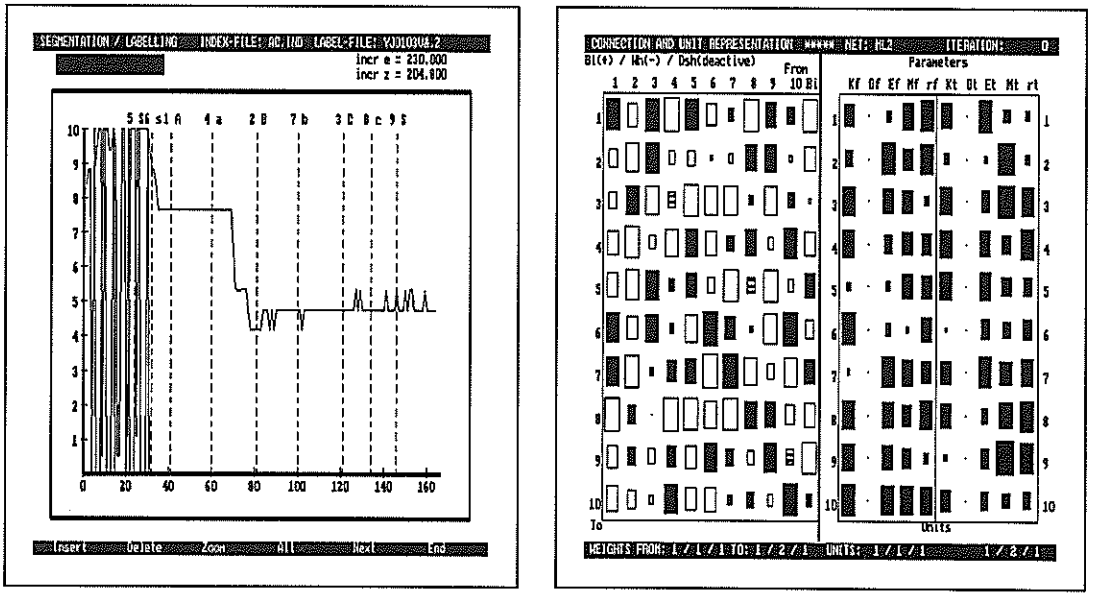


Fig. 2: Showing the power of the software: a) Speech segmentation; b) MLP representation.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

COARSE PHONETIC CLASSIFICATION OF CONTINUOUS SPEECH USING THE TEMPORAL FLOW MODEL

Karl-Heinz MAIER

INSIDERS GmbH * , Wilhelm-Theodor-Römheld-Str. 30, D-6500 Mainz 26, FRG

This paper describes some experiments using the Temporal Flow Model, a general connectionist model to segment and to label continuous speech with coarse phonetic categories. Experiments with several architectures show that the performance can be improved significantly. Temporal Flow Model is a powerful approach to automatic speech recognition.

1. INTRODUCTION

There is currently great interest in using connectionist or neural networks for speech recognition tasks. Several especially tailored models were proposed in order to capture the spectral as well as the temporal dependencies. The latter are particularly important for speech processing. Examples of such models are the TRACE Model by J. L. McClelland and J. L. Elman [4], the Time-Delay-Neural-Network by A. Waibel [5], the Jordan Model by M. J. Jordan [2] and the Temporal Flow Model by R. L. Watrous [6]. The Temporal Flow Model was used in the experiments described in this paper.

2. THE TEMPORAL FLOW MODEL

2.1. Definition

The Temporal Flow Model is an extended Multi-Layer Perceptron. The first extension allows delays which are attributed to the links (or connections) in addition to the weights. Through these variable delay links, units of a higher layer get information about various previous outputs of units in the lower layer. The second extension permits delayed recurrent links so that the activation of a unit depends on the previous output of that unit. Thus integration and differentiation of unit outputs may be performed. The data flows from input units to output units along the interconnection links, hence the name Temporal Flow Model.
The net input $net_j(t)$ into unit j at time t is a weighted sum of the outputs $a_i(t-d)$ of the units i at the time $t-d$ multiplied by the weight w_{jid} connecting the unit i at time $t-d$ and the unit j at time t:

$$net_j(t) = \sum_{i,d} w_{jid} a_i(t-d);$$

2.2. An Example Architecture

Figure 1 shows a trellis-like diagram of an example architecture. Time flows in the direction of x. The network lies in the yz-plane. From bottom to top the input, the hidden and the output layers are printed.

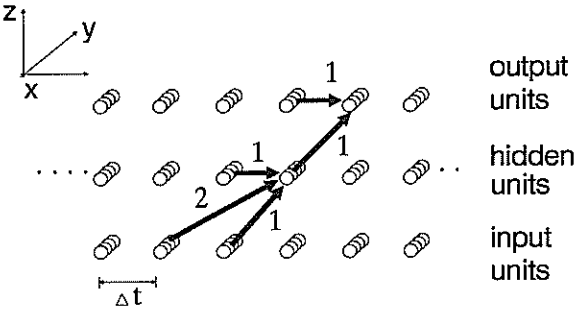


Figure 1: An example architecture

The horizontal distance of two units corresponds to the unit delay Δt, which is equal to the update time of the unit activations. The horizontal arrows show the recurrent links. The diagonal ones represent the links connecting the units of a lower layer to the units of a higher layer. The numbers next to the arrows are the delays the links are attributed to. In this example the activation of each hidden unit depends on its activation one unit delay ago and on the activations of the input units one and two unit delays ago.

2.3. Training

The delays are not learned but specified before training the network. The influence of the delays takes place only through the weights.
An extended back-propagation algorithm which takes the delays into account can be applied.
The activation of an output unit shows how much the network "decides for" or "is in favour of" the category the unit is attributed to. A target function defines the desired activation of each output unit for each time step. It is through these target functions that the desired transition from one category (i.e. phoneme) to the next can be specified during training.

[1]*The research reported here was done at SIEMENS CRS, Princeton, N.J., U.S.A. The author is now with INSIDERS GmbH

3. THE EXPERIMENTS

3.1. The Task

Based on the work of Watrous [6] who reached good results applying the Temporal Flow Model on phonetic discrimination tasks the experiments described in this paper were conducted to test the Temporal Flow Model on continuous speech under application oriented conditions. The task was to categorize (i.e. to segment and to label) continuous speech into 7 coarse phonetic classes [3]. The classes, their identification number and their phonem members were:[1]

silence (SI) 0	{/si/}
unvoiced plosives (UP) 1	{/h/, /k/, /kh/, /p/, /t/, /th/, /?/}
unvoiced fricatives (UF) 2	{/ch/, /f/, /rx/, /s/, /sch/, /x/, /z/}
voiced plosives (VP) 3	{/b/, /d/, /g/, /v/}
nasals (NA) 4	{/em/, /en/, /m/, /n/}
sonorant consonants (SO) 5	{/j/, /rj/, /l/, /r/}
vowels (VO) 6	{/a/, /ae/, /ai/, /au/, /ah/, /e/, /eh/, /eu/, /i/, /ie/, /o/, /oe/, /oeh/, /oh/, /u/, /ue/, /ueh/, /uh/}

3.2. The Speech Data

40 hand-labed German sentences out of the so-called Sotschek sentence database [2] spoken by a female speaker were used as data. The first 20 sentences served as a training set. The remaining 20 sentences served for testing the networks. The sampling frequency was 16 kHz. This speech data was preemphasized. For every 10 ms a melcepstal vector was computed over a 20 ms Hamming window. The first 16 melcepstral coefficients were used as components for the pattern vector of each window. Thus a sequence of pattern vectors was used as input for each neural network.

3.3. The network topology

Several three-layered architectures with a constant number of units were examined. The number of units in each layer was the same for every experiment. According to the 16 melcepstral coefficients the input layer consisted of 16 units. The only hidden layer was composed of 15 units. For each of the 7 coarse phonetic classes one unit in the output layer was needed.

3.4. Accounting for the pattern distribution

Preliminary experiments showed, that patterns of classes which were represented with a lot of tokens in the training data were better recognized than patterns of sparcely represented classes. To avoid these influences caused by the distribution of patterns in the training set, the error signal $z_j(t) - a_j(t)$, where $z_j(t)$ is the target value and $a_j(t)$ is the activaton of the j-th unit was multiplied by a normalization factor. Thus, during training, the movement in weight space was expected to be towards the overall class optimum of recognition, not the optimum concerning the well-represented patterns.

3.5. The experiments

Starting with only few delays the architectures were gradually designed more complex within the constraint of computation time, for the networks were simulated on serial machines. To speed up training a fast second order backpropagation algorithm was available [7].
According to the network architectures reported here the experiments can be devided into 4 groups:

1. Unit delay of all links (3.5.1)
2. Links with small delays, no recurrent links (3.5.2.)
3. Recurrent delayed links at the hidden units (3.5.3.)
4. Large delays for one component of the input vector (3.5.4.)

3.5.1. Unit delay of all links

A temporal flow network with unit delay of all links has the same mapping properties as a multi-layer perceptron if it has the same topology. Such a network is not capable of processing dynamical contextual data through means of propagation delay.
A recognition rate of 76 % was achieved.

3.5.2. Links with small delays, no recurrent links

Small delays are delays of three and less than three. Figure 2 shows the trellis diagrams of the three architectures. With the first architecture the net input into each hidden unit is computed out of three temporary adjacent pattern vectors. The second architecture is designed to delay the activations of the hidden units, i.e. the internal representation. The third architecture is a hybrid design of the first and second. The pattern the target value is attributed to is marked with a T.
In all three architectures information before and after the target token is processed. Thus context information of the future and the past relativ to the target is taken into account for the classification. With this choice of the architecture the network is able to process information of a 40 msec time segment. The recognition rates in the three experiments were nearly the same. The average rate for all testing patterns was about 77 %.

3.5.3. Recurrent delayed links at the hidden units

In this group one experiment was conducted. Figure 3 shows the trellis diagram. For the computation of the hidden units' activation their past activation is used. The internal representation is "stored", learning dynamic responses is thus supported. 78 % of the patterns

[1] /?/ means glottal stop
[2] The data was provided by SIEMENS ZFE, Munich

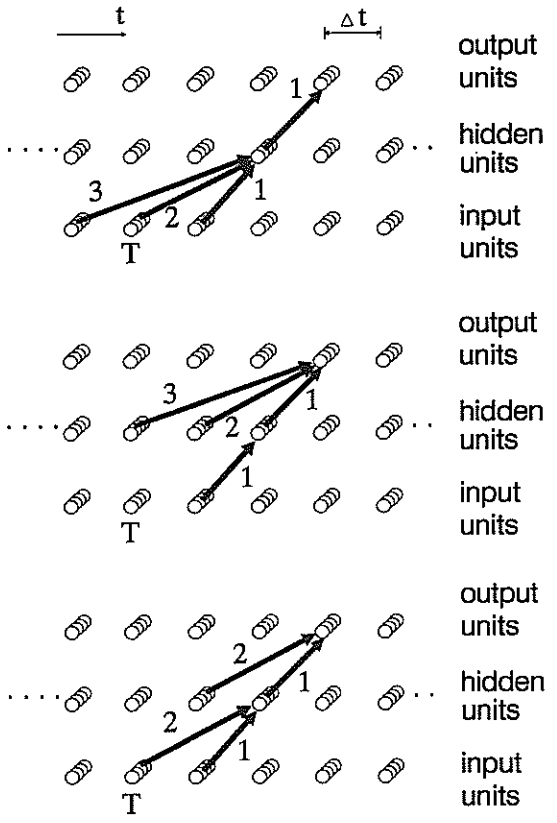


Figure 2: The three architectures with small delays

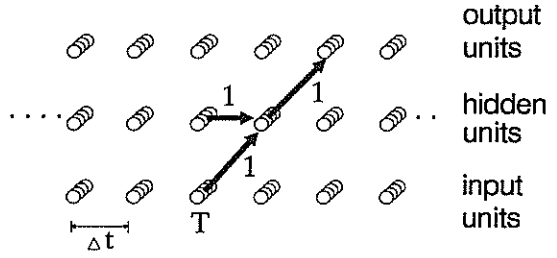


Figure 3: Trellis diagram of the architecture with recurrent links at the hidden units

of the testing data were categorized into the right classes.

3.5.4. Large delays for one component of the input vector

On one hand the hitherto described experiments show that the recognition rate can be raised significantly by using architectures which provide contextual information. On the other hand the complexity increases and hence the computation times for training. A more structured network was chosen to include more temporal or contextual information and to keep the learning times low.

This architecture is distinguished by a separation of temporal and spectral information: To supply enough temporal information only the first coefficient, which corresponds to the total energy within the window was processed for several patterns; to provide spectral cues only one whole melcepstral vector (16 coefficients) whic belonged to the target was used per time step.

Figure 4 shows this architecture. The thin arrows represent the connections from the first coefficient to all the hidden units. The thick ones symbolize a fully interconnected layer of links. With this architecture

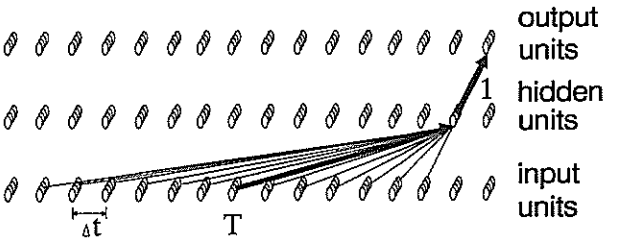


Figure 4: Large delays for one component of the input vector

input information out of a range of 140 ms (13 pattern vectors) is provided. An overall recognition rate of 81 % was achieved.

3.6. Comparison of the results

The recognition results were examined through recognition rates, confusions matrices and the network output, i.e. the activation of each output unit over time.

Figure 5 shows the maximum recognition rates for the test data on the first line of the table for each architecture. On the other lines the corresponding class specific rates are printed. Clearly the network with unit delay of all links performs badly on patterns of classes which are characterized through a dynamical stream of patterns, such as UP, VP or SO. The more contextual information is provided the more the recognition results improve for these classes.

Nevertheless even with the last architecture the recognition rate for the sonorant consonants is not acceptable.

recognition rates for	unit delay	small delays	recurrent links	large delays
overall	76 %	77 %	78 %	81 %
SI	94 %	93 %	92 %	96 %
UP	42 %	52 %	63 %	60 %
UF	63 %	61 %	59 %	67 %
VP	7 %	19 %	10 %	44 %
NA	80 %	84 %	84 %	83 %
SO	24 %	22 %	13 %	31 %
VO	92 %	92 %	93 %	92 %

Figure 5: The recognition rates for all patterns (overall) and for the patterns in the classes (VO etc.)

Figure 6 shows the net output for the utterance "Leise rollen wir aus dem Bahnhof", plotted as a function of time. The vertical lines mark the hand-labeled phoneme boundaries. Underneath the plot the labeled class numbers and the uttered phoneme string is indicated. In the upper part the course of the first melcepstral coefficient (C1) is displayed. The lower plots show the activations of the output units.

For the classes SI, NA and VO the transitions according

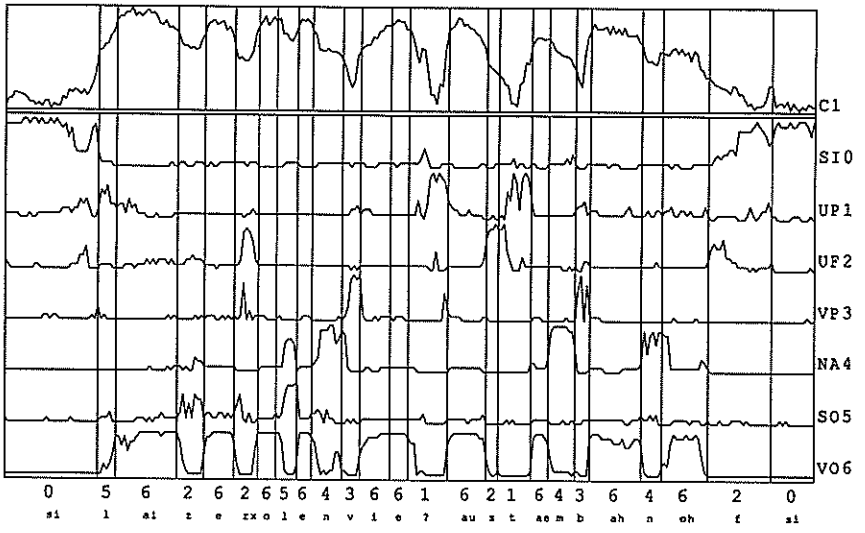


Figure 6: The output unit's activation over time during recognition of the sentence "Leise rollen wir aus dem Bahnhof"

to the given target functions were learned. For the other classes it is not adequate to use a step function as target function. Especially tailored target functions for the phonemes of these classes would improve the recognition.
Subsequent experiments (with the same networks) whic were conducted to get a comparison [1] of the Temporal Flow Model to other methods of automatic speech recognition showed that the performance of Temporal Flow Model is clearly superior to the Maximum Likelihood Classifier and comparable to the Hidden Markov Model.

4. CONCLUSIONS

It could be shown that the particular kind of architecture is crucial to improving the results. Especially the inclusion of enough context information is important. Thus knowledge about speech has to be incorporated. Furthermore the design of the architectures was governed by the need to keep the computation times low. Within these constraints the best architecture reached 81 % overall recognition rate. These first experiments with the Temporal Flow Model for continuous speech show that this model is a powerful means for acoustic phonetic speech recognition.

ACKNOWLEDGEMENTS

This paper is part of my diploma thesis at the TU München. The research was done at and supported by SIEMENS CRS, Princeton, N.J., U.S.A..
I would especially like to thank Wolfgang Feix, the former Group Leader of the Speech Group at SIEMENS CRS, for his encouragement, support and advice throughout the whole course of the project. Furthermore I am endebted to Raymond L. Watrous, Dave Lubensky and Bruce Ladendorf and the other members of the Speech Group for helpful discussions.

REFERENCES

[1] Aktas, A., Feix, W.H., Maier, K.-H. and Schmidbauer, O. *Classification of Coarse Phonetic Categories in Continuous Speech: Statistical Classifiers vs. Temporal Flow Connectionist Network*, to appear in Proc. ICASSP '90, Albuquerque, 1990.

[2] Anderson, S., Merrill, J.W.L. and Port, R. *Dynamic Speech Categorization with Recurrent Networks*, Technical Report No. 258, August 1988.

[3] Maier, K.-H. *Automatische Spracherkennung mit neuronalen Netzen*, Diplomarbeit, Technische Universität München, Mai 1989.

[4] McClelland, J.L.,Rumelhart, D.E., and the PDP Research Group *Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Volume 2: Psychological and Biological Models.* MIT Press, Cambridge, MA, 1986.

[5] Waibel, A., Hanazawa, T., Hinton, G., Shikano, K. and Lang, K. *Phoneme Recognition Using Time-Delay Neural Networks.* Technical Report TR-I-0034, ATR Interpreting Telephony Research Laboratories, Oktober 1987.

[6] Watrous, R.L. *Speech Recognition Using Connectionist Networks.* PhD Thesis, University of Pennsylvania, October 1988.

[7] Watrous, R.L. *Learning Algorithms for Connectionist Networks: Applied Gradient Methods of nonlinear Optimization.* Technical Report MS-CIS-87-51, University of Pennsylvania, June 1987.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Classification of Phonetic Categories in Continuous Speech with Connectionist Networks

Abdulmesih Aktas and Günther Ruske‡

Siemens AG, Corporate Research and Development, Otto-Hahn-Ring 6, 8000 München 83, FRG
‡Technische Universität München, Lehrstuhl für DV, Franz-Josef-Str. 38, 8000 München 40, FRG

This paper presents some experimental results of the application of connectionist models to phonetic recognition in continuous speech for two different tasks: (a) vowel classification and detection and (b) coarse phonetic category classification. Both tasks were investigated in a speaker-dependent mode. We used the 12 German vowels and seven coarse phonetic categories (CPC) which correspond to manner of articulation. For the classification of the vowels a TRACE model is used while the temporal flow model (TFM) is applied to the more difficult CPC recognition task.

1. Introduction

Appropriate modelling of the spectral and temporal characteristics of phonetic units in continuous speech is necessary for a robust recognition of a large vocabulary. The temporal structure of speech units, as well as their actual spectral properties, are inherently context-dependent and due to the speech generation process highly variant. Additional coarticulation and assimilation results in an overlap of adjacent phoneme cues. Consequently, the modelling of the time-varying characteristics for speech recognition is a crucial issue. Context sensitivity of the speech cues requires the choice of an appropriate way of modelling.

Hidden Markov Models (HMM) have shown to be effective in modelling temporal dependencies in speech. Artificial Neural Networks (ANN) represented by multi-layer perceptrons are well suited to deal with the spectral modelling problem and can form arbitrary nonlinear decision surfaces. Several ANN architectures like the TRACE model [1], the temporal flow model (TFM) [2] and the Time-Delay Neural Networks (TDNN) [3] have been introduced, which are appropriate to capture temporal dependencies of speech patterns. By taking left and right context into account, some coarticulation effects and the sequential process itself are quite well described. The more biological motivated approach of recurrent links and propagation delays as used in the temporal flow model improves the treatment of the time dependencies.

In order to cope with the time dependencies the current ANN architectures generally makes use of two basic concepts. (1) The spatialization: The time structure of speech is transformed into a representation in space. The input to the network is a time window which glides with a fixed step size over the utterance. In this sense the TRACE model is a spatialized model. (2) The temporalization: The temporal relationships are represented by propagation delays and/or recurrent links. The temporal flow model (TFM) makes use of both characteristics.

In our experiments both, the TRACE model and the TFM have been applied. The first architecture captures context information by considering several adjacent feature vectors as input. The training is performed with the well-known back-propagation algorithm. The TRACE model was used for the classification of the 12 German vowels. The TFM introduced by R. L. Watrous is a general connectionist network model that represents temporal relationships using delayed and recurrent links [2], and is thus capable of capturing the time-dependent statistical properties of continuous speech. The TFM handles the preprocessor output as a stream of parameter vectors. The temporal context used in training can be of arbitrary length. The TFM was chosen for the CPC task. All experiments were performed in a speaker-dependent mode. The latter task has been also described in a recent publication [4] in the context of a comparison of the TFM approach with statistical approaches like maximum likelihood classifiers and HMMs under identical conditions.

In order to compare the recognition results for the different tasks, confusion matrixes have been computed. It is shown that the performance can be improved by designing appropriate network topologies and using context information.

The outline of our paper is as follows: Section 2 deals with the vowel task. The network and the experiments are specified. In section 3 the CPC task is described. Results are discussed in section 4.

2. Vowel Classification

Motivation and Task

Vowel classification in particular has to deal with the stationary characteristics of the vowels. Therefore the time-varying aspect of speech could be neglected. Vocalic segments are usually defined by four to ten consecutive frames. Due to the negligible variation of the spectral information in the center of the vowel, the stationary part is suited to characterize a vowel.

The German language contains 19 vowel classes.

Seven vowels distinguish between long and short versions. As we do not differentiate between short and long vowels 12 classes (9 vowels and 3 diphthongs) and a non-vocalic (nv) category are used in our experiments: [nv,y,u,oi,i,e,au,ai,a,ɛ,ə,ø].

For the training of the different networks vocalic segments were marked by an automatic algorithm in fluently uttered sentences. The recognition was performed again on a different recording of the sentences where pauses and consonants had to be classified as a non-vocalic (nv) category resulting altogether in 12+1 classes. Additionally all pauses at the beginning and the end of an utterance have been removed by an endpoint detection algorithm.

Database and Analysis

The speech data used for the vowel classification experiments consists of a corpus of 23 German sentences uttered eight times by a male speaker. The sentences represent all consonants in the German language as well as all vowels and the most important consonant clusters. Seven recordings of the 23 sentences were used for training, one version for test. This means the tests are performed in a vocabulary-dependent mode.

An auditory based preprocessing according to Zwicker's loudness model [5] is carried out in intervals of 10 ms. 22 critical bands (50 - 8500Hz) are selected, giving a 22-dimensionsal loudness spectra (1-22 Bark). A further applied transformation is based on the computation of 13 so-called loudness-cepstral coefficients (LCC) which is similar to the well-known mel-frequency-cepstrum analysis. The loudness-cepstrum is computed by applying the cosine-transform to the loudness-spectra.

As the frequency of occurrences vary highly within the vowels, the weakly represented vowel patterns were used multiple times during training to the networks. From the non-vocalic frames every fifth frame was taken in order to balance vowel and non-vowel occurrences. The total number of training patterns was about 15,000.

Network Topology

Several experiments were conducted. The number of hidden layers and hidden units has been varied. Different preprocessing on the input pattern has been investigated. The basic network topology used for the experiments is shown in *Figure 1*, where the input layer has been extended here to three frame input. The number of hidden units was 40 and the number of the outputs was 13 (12 vocalic classes + non-vocalic class). In case of two hidden layers 15 and 45 units have been built in for the first and second hidden layer. For all experiments the standard back-propagation training algorithm has been applied. For the TRACE architecture the number of consecutive time slices considered by the model was varied (3 or 5).

Experiments and Results

Six different experiments (referred to as VT1... VT6) are described [6]. Confusion matrixes have been computed for each test. As general results, the recognition rates for all experiments are listed in *Table 1*. The last column gives the size of each

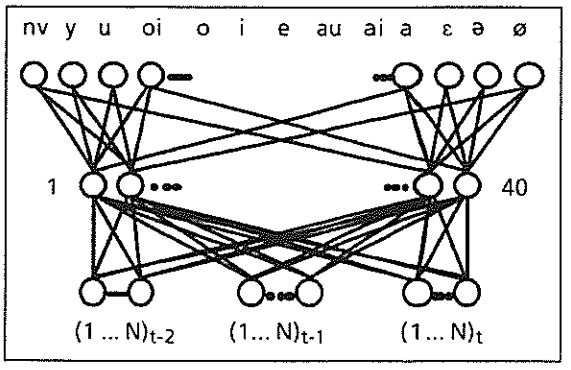


Figure 1: The TRACE model like network used for the vowel task (N = 22 or 13). The index t, t-1 and t-2 indicates the time shift.

network. The detection rate for correct vocalic and non-vocalic frames is given in column four.

TESTS	Recognition rates (%) vowel	total	vowel-Detection rate (%)	network size
VT1	76.5	81.7	-	22-40-13
VT2	75.5	81.3	-	22-60-13
VT3	73.8	81.8	-	22-15-45-13
VT4	76.4	82.6	-	13-15-45-13
VT5	73.4	84.1	98.1	3*13-40-13
VT6	77.4	86.3	97.8	5*13-40-13

Table 1: The recognition and detection rates for the vowel experiments.

VT1: The fully connected feed-forward network has in accordance to the extracted critical bands 22 (=N) inputs and 40 hidden units. As a result, we could observe that the vowel individual rates (not given in *Table 1*) were in the same range. But the individual results for the diphthong [ai] and the vowel [o] were poor (50 and 40%).

VT2: In this experiment the number of hidden units was extended to 60. This extension did not result in any improvement. It is obvious that the network is oversized and the effect of over-adaptation to the training data can be seen slightly from the results.

VT3: A further extension in form of a second hidden layer was applied to the basic network. 15 and 45 units were built in for the first and second hidden layer. Even this extension did not produce better results. As vowel recognition rate was slightly worse it is assumed that the network is oversized.

VT4: The number of the input units of the network from the last experiment was reduced to 13 input units by calculating 13 LCCs. This modification results in a slightly better overall performance.

VT5: The basic network with N=13 input units (LCCs) was extended to the TRACE model (*Figure 1*). One left and right adjacent frame was considered as context. The target value corresponded to the input frame index (t-1). Compared to the

	nv	y	u	oi	o	i	e	au	ai	a	ɛ	ə	ø
nv	5106	16	31	3	8	115	10	20	27	74	66	114	2
y	5	17				2				1			
u	29	2	62	3	14	8						9	1
oi	1			15									
o	10				23			7		2			
i	15	2				132	10				2	2	
e	3					13	45				16	13	
au	13			1	8			67	2	3			
ai	1								80	16	1		
a	19							5	11	206	10		
ɛ	22					3	3		8	13	210	12	3
ə	60	4			4	12	11		16	10	38	227	8
ø	2	8			2							1	11

Table 2: The vowel recognition results for experiment VT6.

SI (silence): [si]
UP (unvoiced plosive): [h,k,kh,p,t,th,?]
UF (unvoiced fricative): [ch,f,rx,s,sch,x,z]
VP (voiced plosive): [b,d,g,v]
NA (nasal-like): [em,en,m,n]
SO (sonorant-like): [j,rj,l,r]
VO (vocalic): [a,ae,ai,au,ah,e,eh,eu,i, ie,o,oe,oeh,oh,u,ue,ueh, uh]

Table 3: Allocation of phone labels to coarse phonetic categories (CPC): [?] indicates a glottal stop.

previous experiments the modification causes an improvement of the overall recognition rate. In this experiment the vowel detection rate was counted, i.e. the decision between vowel and non-vowel. The results show that vowels can be localized with an accuracy of 98.1%.
VT6: The TRACE model with five consecutive input frames has been used. The index considered was up to (t-4), whereby the index (t-2) defined the target value. Both, the vowel and the overall performance could be improved remarkably. The individual results are displayed in *Table 2*. The vowel detection rate was 97.8%.

A general discussion of these results is given in section 4.

3. Classification of Coarse Phonetic Categories (CPCs)

Database and Analysis

The experiments for this task were performed on manually labeled speech data incorporating two versions of the phonetically balanced "Berlin sentences". We used two recordings of 50 sentences both from one male speaker recorded under quiet conditions, filtered at 6.4 kHz, and digitized at a 16 kHz sampling rate. The computation of 16 cepstral coefficients which are presented as input to the TFM is based on an all-pole LPC analysis. 20 ms Hamming windows are used for analysis in 10 ms steps.

The seven coarse phonetic categories investigated in our recognition experiments correspond to the categories of manner of articulation, which are listed in *Table 3*. The CPCs used here are derived from manually labeled speech data by transforming 44 phone labels into seven coarse phonetic labels.

Figure 2 shows the frequency of occurrence of the specific categories in the given data. While SI and VO are represented well, only a small number of VP and SO are present. The unequal representation of the CPCs is important for an evaluation of the class specific recognition results. The total number of training frames within the 50 sentences was 13626.

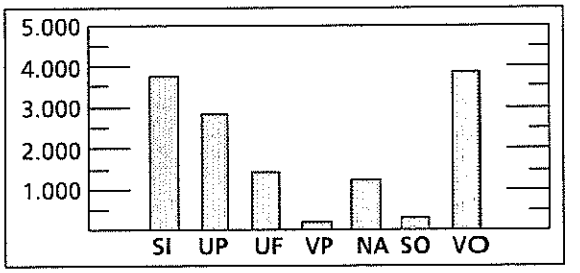


Figure 2: The frequency of occurence of the specific CPCs in the speech data.

Experiments and Results

Several TFM topologies have been tested in our experiments. All consisted of three layers. Beyond that, the influence of context and propagation delays have been investigated on the basis of recognition results. Training is accomplished by a second-order method of iterative nonlinear optimization using gradient descent which allows the computation of the complete gradient of recurrent networks [7]. The four different experiments with the TFM are subsequently referred to as CT1...CT4. For all four network topologies the number of input, hidden and output units was kept the same (16-15-7).

CT1: The network is a fully connected feed-forward network. No delays or recurrent links are used.
CT2: Recurrent loops at each hidden unit, with a propagation delay of 1 are added to the network from experiment *CT1*. By this means the network has information over 20 ms of information. Since the iterative computation of the gradient in the back-propagation was cut up after the delay of 1.
CT3: A propagation delay of 1 was added to each link between the input units and hidden units of the network in *CT2*. The time information is increased by this way. As a result, a second weight layer for a previous frame (t-1) was added between the input and the hidden layer.

CT4: In addition to the input with the index (t-6) left and right energy context is taken into account by an appropriate choice of the propagation delays. The chosen information for the context was the first cepstral coefficient (=energy) of the previous six and the following six frames.

Table 4 shows the frame recognition rates for the individual categories for the four experiments with TFM. The mean (average of the specific rates) and total frame rates are given in the last and the second to the last row. We can observe an improvement through the experiments. Categories with a low occurrence frequency (VP and SO) produce poor results. In this case the lack of the training data prevents any generalization. The best overall recognition rates for the TFM are achieved when energy context information is taken into account.

CPC	CT1	CT2	CT3	CT4
SI	90.5	89.8	89.6	90.9
UP	66.2	76.2	74.3	77.4
UF	86.1	88.6	90.5	90.1
VP	37.9	33.8	40.6	46.9
NA	68.6	61.7	61.8	63.9
SO	25.5	28.7	23.0	33.8
VO	89.5	89.4	89.6	92.1
Mean	66.3	66.8	67.0	70,7
TOTAL	82.3	83.1	83.1	85.2

Table 4: Frame recognition rates for the TFM experiments.

4. General Discussion

For the vowel task the influence of a greater number of hidden units could be neglected. Even the use of a second hidden layer did not cause a significant improvement. Our results indicate that the decision space formed by the vowel data can be separated by using only convex hyperplanes.

It is also argued that the network is able to perform some kind of normalization to the energy contour. This was verified by explicitly normalizing the energy of each frame. No differences could be observed in the performance of the networks.

The use of LCCs resulted in slightly better recognition rates. This implies that a suitable data reduction or preprocessing can simplify the task for the network because it can concentrate on optimizing the decision boundaries. The use of the TRACE model with five consecutive frames used as input pattern yields a significant improvement. The vowel detection rate was extremely high. Thus the network can favorably be used as syllable detector for syllabic speech processing.

Our results for the CPC task show that the performance of the TFM can be improved by taking context information into account. An overall frame recognition rate of about 85% was achieved with the TFM. The class-specific recognition rates differ, however. The influence of context can be best observed on the UP class, which represents transient sounds. Silence and vocalic categories achieve rates of more than 90%. The use of the recurrent links and propagation delays has not caused remarkable improvement. This may be due to the low delay times chosen (1 for our experiments). In order to test the robustness of the recognition, vocabulary-independent tests were performed on the network of experiment CT4. The result was, that the performance only slightly decreases (about 2%).

The TFM approach has been also used for the labelling task [8]. Except for difficulties in the classification of the sparsely represented categories like sonorants and unvoiced plosives, most of the sentences could be labeled correctly. From the activations it was evident, that the network tries to learn the hard transitions of the switching target function and fails on the segment boundaries. Improvements in the classification of the TFM can be achieved by selection of an appropriate target function during training.

Our preliminary results indicate that the TFM is a reliable means for automatic labelling of continuous speech. This is due to its ability to model temporal relationships well.

This work was partly supported by the German Federal Ministry for Research and Technology (BMFT) under the grant No. ITM 8801 B9 (SPICOS project).

References

[1] J.L. McClelland and J. L. Elman: "Interactive processes in speech perception: the TRACE model", in J.L. MacClelland and D.E. Rumelhard: PDP Group, Explorations in the Microstructure of Cognition", Vol. II, MIT Press, Cambridge, MA, 1986

[2] R.L. Watrous and L. Shastri: "Learning Phonetic Features Using Connectionist Networks: An Experiment in Speech Recognition", 1st Int. Conf. on NNs, San Diego, CA, 1987, pp. IV-381-388

[3] A. Waibel et. al.: "Phoneme Recognition Using Time-Delay Neural Networks", TR-1-0006, ATR Interpreting Telephony Research Laboratories, Japan, Oct. 1987

[4] A. Aktas, O. Schmidbauer, K.-H. Maier and W. H. Feix: "Classification of Coarse Phonetic Categories in Continuous Speech: Statistical Classifiers vs. Temporal Flow Connectionist Networks", Proc. of the ICASSP '90, Albuquerque/U.S.A. in print

[5] E. Zwicker, E. Terhardt and E. Paulus: "Automatic Speech Recognition Using Psychoacoustic Models", JASA, No. 65 (1979), pp. 487-498

[6] U. Arzt: "Anwendung Neuronaler Netze in der Spracherkennung", Diplom-Thesis, Techn. Univ. München, Lehrstuhl für DV, München (1989)

[7] R.L. Watrous: "GRADSIM: A Connectionist Network Simulator Using Gradient Optimization Techniques", RTL-88-TR-187, Siemens Corporate Research and Support, Princeton (1988)

[8] K.-H. Maier: "Automatische Spracherkennung mit neuronalen Netzen", Diplom-Thesis, Techn. Univ. München, Lehrstuhl für DV, München (1989)

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Decision with reject options

Bernard DUBUISSON

Université de Technologie de Compiègne
U.R.A. C.N.R.S. 817 Heuristique et Diagnostic des Systèmes Complexes
B.P. 649 - 60206 COMPIEGNE Cédex - FRANCE

This paper is devoted to the problem of assigning an observed signal to a class when one has not a complete knowledge about all the classes. So, we propose to use a decision with a reject option, with a distinction between ambiguity and distance reject. Parametric and non parametric cases are presented.

1. Introduction

Decison about the association of an observed set of signals to a class can be solved using statistical pattern recognition. Usually, d parameters are extracted from the set and the vector built up with these d parameters is represented as a point in a d-dimensional space, which we call the representation space R^d.
If this space is partitioned into as many areas as there are classes (let M the number of classes written $\omega_1, \omega_2, ...\omega_M$), the discrimination problem is identical to the knowledge of the belonging area of the observed vector. There are a lot of such discriminating method (linear discriminating rule, k nearest neighbor rule), the best one being the Bayes one which supposes the knowledge of the probability law in each class.
The main hypothesis of these methods remains the knowledge of all the possible classes to which a new vector may be associated. So, one supposes that the partition of the space or the knowledge of the different boundaries is complete. In many applications, this hypothesis is not verified. For example, for the diagnosis of a technological system, one knows often a class corresponding to the normal running of the system and, maybe, one or two other ones corresponding to some abnormal situations : if usual pattern recognition is applied, a new observed vector may correspond to a new situation and the proposed decision is an erroneous one. So, we propose, in order to solve this problem, to introduce a decision with a reject option. In the paper, we will call "learning set", the set of vectors for which we have a complete knowledge : the class of each vector in the learning set is known. But this set is not complete : all the possible classes are not represented by vectors. This set will be used in order to build up the decision rule.

A new observed vector can be either associated to one of the M known classes or rejected from any class. After some vectors have been rejected, they can be analysed and some new classes can be exhibited and, then, grouped with the previous ones.
By this method, first erroneous decision can be avoided, second the knowledge, about the system on which this decision rule is applied, increases.
In a first part, we introduce decision with reject in the parametric case. This work is based on Chow's work. But, we demonstrate that Chow's reject is not sufficient for our problem, and then we introduce a new reject called the distance reject.
In a second part, we present a decision with reject in the non parametric case using the well known k nearest neighbor rule.
At last, one has to analyse the points which have been rejected using a distance reject. A very simple method is proposed based on a clustering algorithm applied on these points. This point will be developed in the third part.

2. Parametric case

A vector from R^d follows in class w_i the probability law $f(x / w_i)$. In this part, this law is completely known with its parameters. Each class has an a priori probability $P(w_i)$ and we have the relation for the M possible classes : $\sum_{i=1}^{M} P(\omega_i) = 1$
In the usual classification problem, a new vector must be associated to one class among the M

possible ones. So, one has to define a decision rule d(x) so that d(x) = i means that x must be associated to class ω_i.
As indicated in the introduction, this solution is not satisfactory and Chow [1] has proposed to add a new class, he has called a reject class, ω_0. The decision rule is then called a decision rule with reject. Then, we have to decide among (M + 1) classes and build up the decision rule d(x) [2], so that :

$$\left.\begin{array}{l} d(x) = 1 \quad x \text{ is associated to } \omega_1 \\ d(x) = 2 \quad x \text{ is associated to } \omega_2 \\ \ldots\ldots\ldots \\ d(x) = M \quad x \text{ is associated to } \omega_M \end{array}\right\} \quad x \text{ is classified}$$

$$d(x) = 0 \qquad x \text{ is associated to } \omega_0 \qquad x \text{ is rejected}$$

In the usual Bayes rule, one has to choose classification costs $C(\omega_i / \omega_j)$. $C(\omega_i / \omega_j)$ is the cost associated to the decision $d(x) = i$ when x is a vector from ω_j ($i = 0, 1, \ldots, M$; $j = 1, \ldots, M$).
Usually, one chooses : $C(\omega_i / \omega_i) = 1 \quad i = 1, \ldots, M$
and $C(\omega_i / \omega_j) = 0 \quad i,j = 1, \ldots, M \; i \neq j$

and gives a constant value to the reject cost :
$C(\omega_0 / \omega_j) = C_r$
Using the a posteriori density $P(\omega_j / x)$, one can compute the decision cost C(x) :

- $d(x) = 0$, $C(x) = C_0(x) = C_r$
- $d(x) = i$ ($i = 1, \ldots, M$), $C(x) = C_i(x)$,

$$C_i(x) = \sum_{j=1,\, j\neq i}^{M} P(\omega_j / x) = 1 - P(\omega_i / x)$$

$C_i(x)$ is the conditional error probability for classifying x into ω_i. The optimum error probability $e^*(x)$ will be written :

$$e^*(x) = \mathrm{Min}_{i=1,M}\, C_i(x) = 1 - \mathrm{Max}_{i=1,M}\, P(\omega_i / x)$$

Then the decision rule with a reject option follows :

$d(x) = i$ x associated to ω_i if $e^*(x) = e_i(x) \leq C_r$
$d(x) = 0$ x rejected if $Cr < e^*(x)$

This rule can be written using the a posteriori density :

$d(x) = i \quad x \rightarrow \omega_i$
if $P(\omega_i / x) = \mathrm{Max}_{j=1,M} P(\omega_j / x) \geq 1 - C_r$
$d(x) = 0 \quad x \rightarrow \omega_0$
if $1 - C_r > \mathrm{Max}_{j=1,M} P(\omega_j / x)$ (1)

It is easy to establish that a reject option cannot exist if $C_r > \frac{M - 1}{M}$
This decision rule separates the representation space Ω into (M + 1) areas (Fig. 1) :

$$\Omega_i = \{x: \mathrm{Max}_{j=1,M}\, P(\omega_j / x) = P(\omega_i / x) \geq 1 - C_r)\} \quad i = 1, M$$
$$\Omega_0 = \{ x: \mathrm{Max}_{j=1,M} P(\omega_j / x) < 1 - C_r \} \quad (2)$$

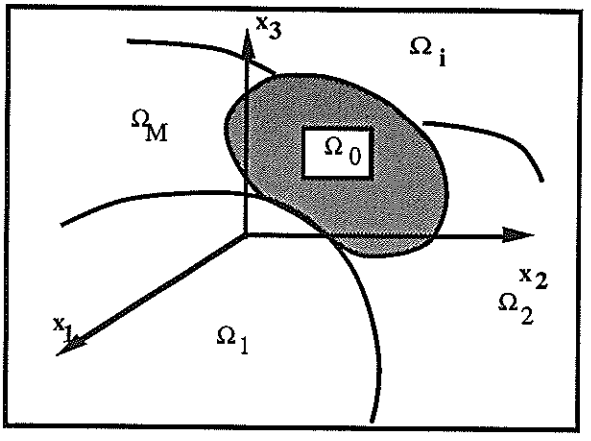


Figure 1

We define the acceptance region Ω_A.
$\Omega_A = \Omega_1 \cup \Omega_2 \cup \ldots\ldots \cup \Omega_M$
Then, one can compute the associated probabilities, which are function of the reject cost :
- global acceptance probability for $d(x) = i$ $P_A(C_r)$:
- reject probability $P_R(C_r)$
- error probability i, $P_E(C_r)$

The following equations are obvious :

$P_A(C_r) = P_E(C_r) + P_C(C_r)$
$P_A(C_r) + P_R(C_r) = 1$

Figure 2 represents the case of two gaussian classes in R (the two gaussian laws have the same variance).

As defined by Chow, this reject option can be defined as an ambiguity reject or an uncertainty reject. In fact, region Ω_0 corresponds always to a region between the different classes ; in this area, different decisions can be chosen. For this reason, we choose the word "ambiguity" for this kind of reject.

In some cases, this notion is not sufficient. Often, a new vector lies in an area of the representation space where no vectors of the learning set were previously observed. It means that this new vector is an element of a new class, for which no information was extracted from the learning set. So, we need a second kind of reject, we have called, a "distance" reject : one must not associate this vector far from usual areas to a known class, if not, he does an error (Fig.3)

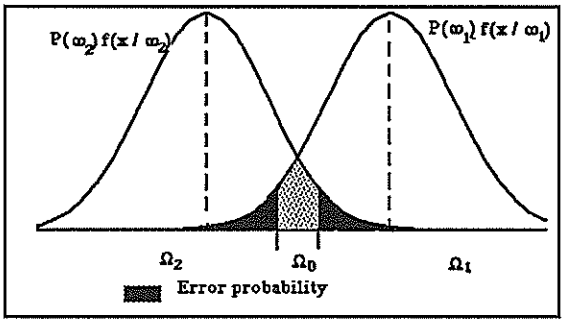


Figure 2

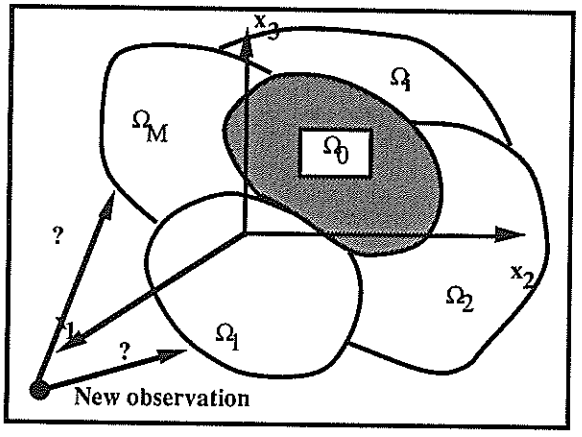


Figure 3

This reject distance is really based on a distance function : the distance to the next classes must be important. Studying the parametric case, we choose to use the probability law to introduce this kind of reject. A new observed vector will be called "far from" known classes if its probability law is small, that is is smaller than a threshold C_d : x "distance" rejected if $f(x) < C_d$ (3)

So, one has to consider two rejects :
- an ambiguity reject given by (1)
- a distance reject given by (3)

We have to introduce in the representation space a reject region Ω_R, defined by : $\Omega_R = \Omega_0 + \Omega_D$

$$\Omega_D = \{ x : f(x) < C_d \}$$

Ω_0 is given by (2).
The reject probability will be written :

$$P_R(C_r, C_d) = \int_{\Omega_R} f(x)\ dx = \int_{\Omega_D} f(x)\ dx + \int_{\Omega_0} f(x)dx$$

In fact, it is easy to verify that using a reject distance reduce the acceptance region.

$$\sum_{j=1}^{M} P(\omega_i)\ f(x / \omega_i) < C_d$$

So,

$$\mathrm{Max}_{j=1,M}\, P(\omega_j)\ f(x / \omega_j) = P(\omega_i)\ f(x / \omega_i) < C_d$$

and

$$P(\omega_j)\ f(x / \omega_j) < C_d \qquad \text{pour } j \neq i$$

Figure 4 gives an example of the different regions, with two gaussian classes in R.

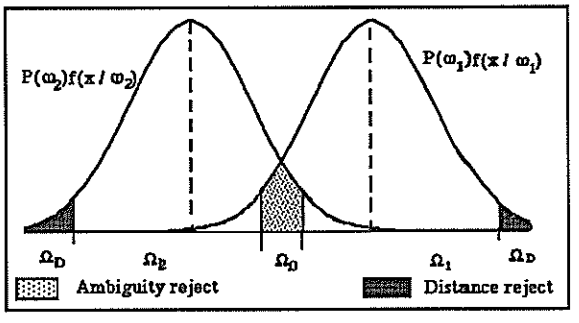


Figure 4

Using this definition of a reject distance, we are not in the case of the optimum Bayes rule, and the usual results are not applicable. We can yet consider this optimum rule by decreasing the acceptance probability with the term P_{RD} :

$$P_{RD} = \int_{\Omega_D} f(x)\ dx$$

This term will increase the reject probability.

3. Non parametric case

There are different non parametric decision rules : we propose to study the most useful one, the k nearest neighbor (k-NN) rule. With this rule, x is assigned to the class which is most heavily represented in the labels of the k-NN.

3.1. Ambiguity reject

It is easy to safeguard ourself against excessive classification error by introducing a reject option, which is an ambiguity reject. x is classified into class ω_i if the number of neighbours from this class is at least equal to a qualifying majority level k' [3] (value k' may be different from one class to another([2]). In [2], the property of this rule has been studied for an infinite learning set. As it is defined, this reject is an ambiguity one.

3.2. Distance reject

K-NN rule does not take in account the distance between x and its neighbors. As it is build up, the k-NN rule works in regions where the density is high. For this reason, most of the results on this rule are asymptotic ones.

The most simple idea in order to consider the distance is to introduce a threshold on the mean distance. x is rejected if this distance mean is higher than T : $\frac{1}{k}\sum_{y^j} d(x, y^j) > T$ (4)

y^j , being a nearest neighbor of x.

T can be dependent on class.

This rule can be combined with the ambiguity reject. Condition (4) must not be verified before applying the ambiguity reject. If the mean distance is smaller than T, x can be assigned to a class or rejected, depending of the number of nearest neighbors of each class. If the mean distance is higher than T, x is rejected. Dasarathy [6] has proposed such a rule, combining distance property and number of neighbors of each class. Error and reject probability cannot be computed in the general case : some results have been got by considering the distance betwween x and a neighbor as a random value with a known probability density function.

4. Learning step

Once some points have been rejected, one has to use them in order to update its knowledge about classes. Points which have been rejected using an ambiguity reject must be assigned to a class. New information must be added : for example, if the points are observed on a system regularly at each sampling period, we propose to consider a sequence of points and analyse them globally using, as an example, a prediction filter. Ambiguity on a point can be, by this method, removed.

Points that have been rejected by a reject distance can be analysed using a clustering algorithm (Dynamic clustering [7]) in order to exhibit new classes. These new classes must, then, be added to the learning set in order to improve the decision rule. The previous decision rule is modified in order to take into account these new classes. So, points, that have been distance rejected because of their positions in the representation space, will be assigned to one of these new classes.

5. Conclusion

In this paper, we have proposed a decision rule based on an incomplete knowledge about the possible classes a point may be assigned to. This decision rule is a reject one : we have distinguished ambiguity reject and distance reject.

These methods have been used to solve diagnosis problems when the knowledge is never complete when beginning the analysis.

References

[1] C. K. CHOW, An optimum character recognition system using decision functions, R.E. Trans. on Electronic computers, (1957), pp 247-254.

[2] P. A. DEVIJVER, J. KITTLER, Pattern recognition, a statistical approach, (Prentice-Hall, Englewood Cliffs, 1982).

[3] M. E. HELLMAN,The nearest neighbor classification rule with a reject option, I.E.E.E. Trans. on System, Man and Cybernetics, (1970), pp 179-185.

[4] B. V. DASARATHY, Nosing around the neighborhood : a new system structure and classification rule for recognition in partially exposed environment, I.E.E.E. Pattern Analysis and Machine Intelligence, (1980).

[5] E. DIDAY, J. C. SIMON, Cluster Analysis, dans Digital Pattern Recognition, in K. S. FU (ed.), (Springer-Verlag, Berlin, 1976).

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SPATIAL REASONING BY KNOWLEDGE-BASED INTEGRATION OF VISUAL AND IR FUZZY CUES

Roberto Feri, Gian Luca Foresti, Vittorio Murino, Carlo S. Regazzoni and Gianni Vernazza

Dep. of Biophysical and Electronic Engineering
University of Genoa
Via all'Opera Pia 11A, I-16145 Genoa, Italy

A distributed framework for Multisensor integration and spatial reasoning is presented, which allow an authonomous system to infer 3D structure of an outdoor environment. To this end, symbolic 2D cues obtained by processing data provided by a visual (b/w) and thermal cameras are used as input representation by a Geometric Reasoner module. 3D models' linguistic description is related to sensors' signals by bounding the possible values assumed by features with costraining functions, and by propagating the information flow between 2D and 3D levels with an opportunistic control strategy. Results above the 3D interpretation of complex road scenes are shown, which confirm the effectiveness of the approach.

1. INTRODUCTION

The identification of 3D objects by integrating data coming from multiple sensors is a fundamental goal for the interpretation of real scenes. In the past few years, new methodologies have been developed, which make it possible to recognize 3D objects located in the surrounding environment starting from a set of 2D visual cues (geometric reasoning) and integrating the information provided by different sources.
Brooks [1] describes a knowledge-based system which can recognize 3D objects from aerial images (ACRONYM) by performing local matches. A prediction tree is created where computed features are organized and matched against a satisfying set of constraints. A SUPINF technique is applied to obtain lower and upper bounds of the geometrical parameters' values.
Barry et al. [2] suggest a hierarchical organization of geometrical knowledge and the use of object models, in order to perform spatial reasoning at different abstraction levels. By following this approach, which is based to algebraic matching operations between data and models, one can utilize a multilevel geometric reasoning scheme to constraint equations describing spatial transformations.
In the present paper, a Knowledge-based module (called Geometric Reasoner) of a distributed recognition system is described, which infers 3D properties of objects contained in a scene, starting from 2D cues. To this end, a Data Fusion process is performed by integrating data coming from multiple sensors (e.g. an IR and a visual camera). A more general description of the recognition system, called DOORS, Distributed Object Oriented Recognition System), can be found in [3,4]. In section 2 attention is focused on the description of the Geometric Reasoner. In section 3 integration between data coming from an infrared and a b/w camera is presented as a case study. Finally, the validity of the proposed approach is assessed, and results are reported.

2. MULTISENSOR INTEGRATION SYSTEM

2.1 Spatial Reasoning

DOORS [4] is made up of three main parts: a set of hierarchical knowledge-sources (KSs), called modules, and an inference engine, which constitute the control structure; a Global Data Base (GDB), where prototypical knowledge is stored inside frame-networks; and a Blackboard [5], where instantiations of the concepts contained in the GDB that have been found to be related to current data are stored during the recognition process. The architecture is shown in Fig. 1.
The role of the Geometric Reasoner module inside the KS hierarchy is to drive the 3D scene interpretation process by requiring and controlling the bottom-up information flow about 2D properties (i.e., projected measures of 3D patches) as observed by a set of lower level logical sensors [6]. Such lower-level sensors are based on data acquired by various physical or simulated devices (e.g., IR and Visual cameras, CAD models, range maps, etc.). These data are processed along the KS hierarchy, according to different strategies [6], by producing frames at different

abstraction levels, such as Regions, Edges, Virtual-Fused-Data (VFD), (i.e., symbolic reports describing results of edge and region fusions obtained over a fixed channel).
Therefore, the GR module has to integrate VFDs obtained over separate channels into a higher-level representation, called 3D-fused data (i.e., 3D descriptions of scene parts), by taking into account the a-priori knowledge regarding both the application domain (i.e., intrinsic and relational descriptions of possible objects contained in a scene) and the multisensor models (i.e.,intrinsic and relational characteristics of the employed sensors). Redundant and complementary characteristics of sensors' data are used in order to obtain better estimates of 3D features and to reduce the related uncertainty. Moreover, 3D structure estimation can also be reached in areas where a single sensor provides observations. Consequently, it is necessary to develop techniques which allow one to infer 3D structures from monocular views, as suggested in [7] and to fuse hypotheses based on such data. Another requirement which must be met is the capability for easily associating linguistic descriptions with object models.

2.2 Knowledge representation

The hypothesis-level fusion type chosen requires to use a common representation format: symbolic models (i.e., frames) are used for logical sensors' characteristics, observed signals, and searched objects.
The sensors' characteristics and relationships (e.g.,transformations between coordinate reference systems, etc.) are described inside each module. In particular, a reference system (Xsi,Ysi,Zsi)(Fig.4) is assigned to each sensor, and its relations with the global reference system (X,Y,Z) is given as a rototranslation matrix. A recognition datum (i.e., an observed sample of the signal to be interpreted, such as a region, a VFD, etc.) is the representation unit corresponding to the prototypical observation of a module. It is composed by a set of attribute-value-pairs; each attribute is a frame which represents the prototype of a measure that a logical sensor can estimate. The computability of an attribute, which depends on the values of other ones, is expressed inside a dependency-network.
A global object model is composed by a set of cues represented inside Hint-frames. Each cue is a linguistic constraint on the space of possible values assumed by an attribute of a recognition datum. An Hint-frame consists of a link (i.e., "judgment-on") to a prototypical attribute and of a fuzzy-membership function. This function returns a value in the range 0-1, depending on the degree of matching between the Hint-frame and the value assumed by the related attribute. Hint-frames are linked together inside a local constraint-network according to the degree of complexity by which they are characterized (see Fig.2). In particular, geometric characteristics of objects' models have been described as a set of 3D Hint-frames which constraint possible relative configurations of planar surfaces' patches in a common coordinate system (Xo,Yo,Zo). Once judgments on all attributes of a certain datum have been expressed, it is possible to obtain the degree of matching between the datum and an object model. This can be achieved by applying a combination rule to the fuzzy values obtained by a separate application of Hint-frames used to describe the object.
In this way, fuzzy sets are used to represent uncertainty, but, when considering geometric parameters, it is also possible to use fuzzy-membership functions to represent the range of possible values that can be assumed by an attribute, as in the case with the SUPINF [1] method. According to the above knowledge representation, each logical sensor can interpret its observations by using models of local constraints' (i.e., Hint-frames). Due to the distributed organization of knowledge and processing, it is also necessary to define correspondences between models and signals at different abstraction levels. Therefore, mapping schemes are required, which allow each KS to transform top-down and bottom-up messages from adjacent logical sensors into local constraints and local attribute values, respectively. In order to meet this requirement, cues and attributes are also organized into two vertical (i.e., inter-KS) networks: a constraint network and dependency one .

3. THERMAL AND VISUAL DATA INTEGRATION

The integration of thermal and visual data acquired by an IR sensor and a visual b/w camera, respectively can be useful thanks to the complementary characteristics of such sensors in certain environmental conditions (e.g., night, mist, fog, etc.). However, even in normal situations, redundant information allows the system to reduce uncertainty. The first step towards the interpretation goal is to transform IR pixel values (i.e., spectral responses in the 8-14 um band, in our case) into an image based on temperature values. A linear transformation between IR intensity and the corresponding temperature is performed, as in [9]. Then, low-level processing is performed

by lower-level modules, in order to obtain a symbolic description of signals in terms of VFD data. The GR module starts its processing, by selecting a model to be searched and an hypothesized viewpoint of the observation. An expectation on the rototranslation between (X,Y,Z), (Xsi,Ysi,Zsi) of each sensor and (Xo,Yo,Zo) of the searched object (Fig.4) is generated. Linguistic descriptions of planar patches have been developed (Fig.2) which are useful, in a road environment, for detecting regular objects (e.g., landmarks, road, cars, etc.). The viewpoint assumption has been modelled by representing views with an high occurency as a link between a Hint-frame at the 3D level and the set of possible 2D cues. A-priori knowledge and observed attributes are used in order to evaluate the chosen mapping. For example, in the case of the object "road" (i.e. a rectangular-flat surface) one among two possible views must be selected, which are taken at a known height above the ground plane (i.e., XZ plane). In the former case, the 3D road model is mapped by the perspective camera model [7] into a trapezoidal 2D patch (central view), while a parallelepiped is suggested in the case of lateral view.

4. RESULTS

An IR and visual image representing an outdoor road scene were considered (Fig.3a). Lower-modules are activated in order to produce 2D regions for each sensor [7]. A central view assumption is made in the case of road searching; the IR and visual regions which have been found (Fig.3b) to satisfy the 2D level description generated by the road model and are passed to the GR module which also computes values of 3D attributes. The 3D description of the model (i.e., 3D Hint-frames) is then used in order to express a final judgment above object's presence. In the case of the road one has FUSED-3D-RECTANGULAR-FLAT-SURFACE 0.84 and FUSED-3D-WIDE-RECTANGULAR-SURFACE 0.95. The belief in the recognized road is therefore equal to 0.84. Finally, the GR module provides, as output data, a 3D reconstruction of the surface examined , as seen from a orthogonal position (Fig.5).
In order to test the GR module, a second set of images was considered, coming from another application. A b/w camera image and a synthetic one provided by considering a 3D CAD model of a complex environment are taken as input data. In this way the CAD image can be used to focus system's attention over interesting areas of the other employed sensors [3]. In Fig.6a original images are shown, while in Fig.6b the results of 3D reconstruction are presented.

5. CONCLUSION

A knowledge-based approach to spatial reasoning has been presented, which allows a multisensor system to interpret the 3D structure of an outdoor environment. Each sensor cooperates to the recognition process by providing an extimation of objects' structure.
The approach has been experimented on different images, provided by physical and simulated devices. Results are encouraging and work is under development in order to extend system's capabilities towards representation and interpretation of more complex scenes.

ACKNOWLEDGMENTS

This work was developed within the framework of the Italian PRO-ART section of PROMETHEUS, a EUREKA project.

REFERENCES

[1] BROOKS R., "Model-based three-dimensional interpretation of two-dimensional images", IEEE Trans. on PAMI, vol.5, No.2, pp.140-150, 1983.
[2] BARRY M., CYRLUK D., KAPUR D., MUNDY J., and NGUYEN V.D., "A multilevel geometric reasoning system for vision", Artificial Intelligence Spec. Issue on Geometric Reasoning, 1988.
[3] MERIALDO P., PECOLLO G.C., REGAZZONI C.S., ZUNINO R., and VERNAZZA G., "Integration of territorial maps in the vision system of an autonomous land vehicle", Proc. 2nd Int. Conf. Intelligent Autonomous Systems, pp. 694-704, Amsterdam, The Netherlands, December 1989.
[4] ARDUINI F., COSOLI P., MURINO V., REGAZZONI C. and VERNAZZA G., "Interpretation of 3D road scenes from thermal and visual images", Proc. 3rd Prometheus Workshop, Torino, Italy, April 1990. (in press).
[5]ENGELMORE R., MORGAN T. eds., "Blackboard Systems", Addison Wesley, 1988.
[6]GIUSTO D.D., POZZI S., REGAZZONI C.S., and VERNAZZA G., and ZELATORE R., "Integration of data-fusion techniques for autonomous vehicle driving", Proc. SPIE Conf. 1198, Philadelphia,1989. (in press)
[7]KANADE T., "Geometrical aspects of interpreting images as a three-dimensional scene" , Proc. of the IEEE, vol. 71, No. 7, pp. 789-802, July 1983,
[8] NANDHAKUMAR N. and AGGARWAL J.K., "Integrated analysis of thermal and visual images for scene interpretation", IEEE Trans. on PAMI vol. 10, pp. 469-480, 1988.

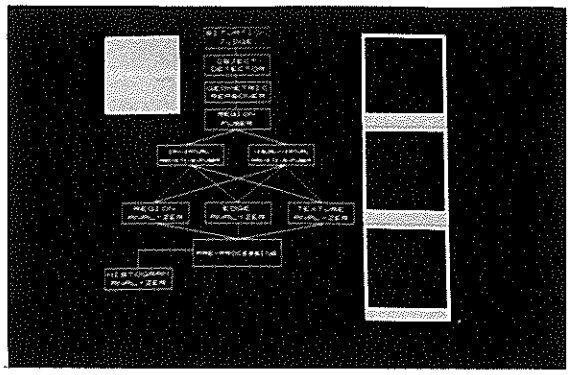


Fig. 1: Systems' Architecture.

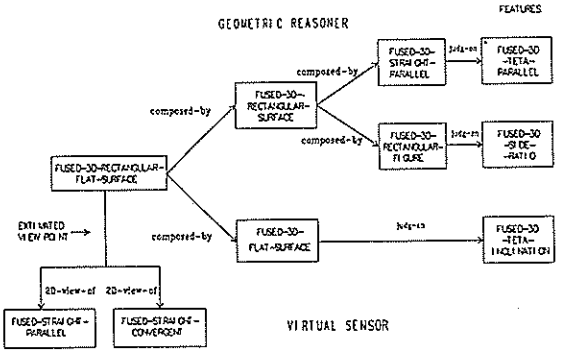


Fig. 2: Structured model of a road (hints and features).

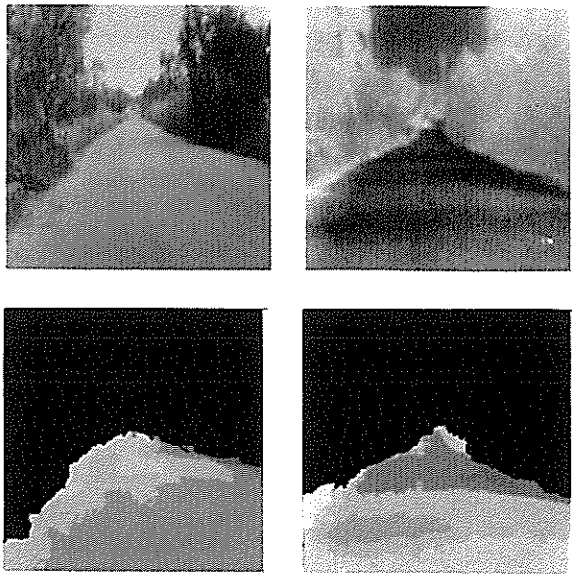


Fig. 3: Visual and IR original images (top) and the related results of closure operation (bottom).

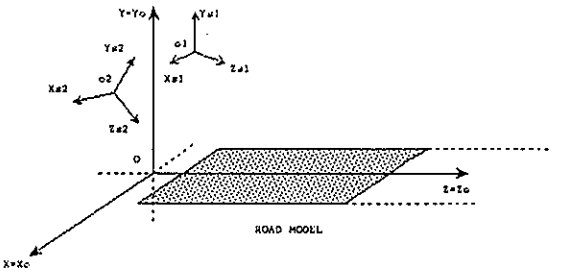


Fig. 4: Roads', IR and visual sensors', and general reference systems.

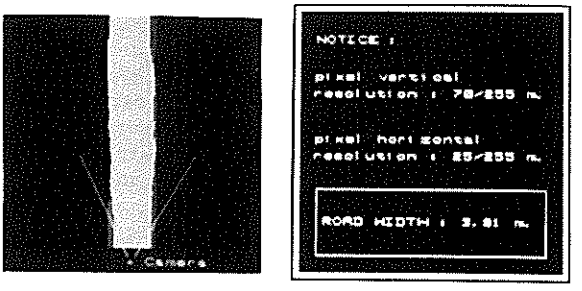


Fig. 5: 3D road interpretation.

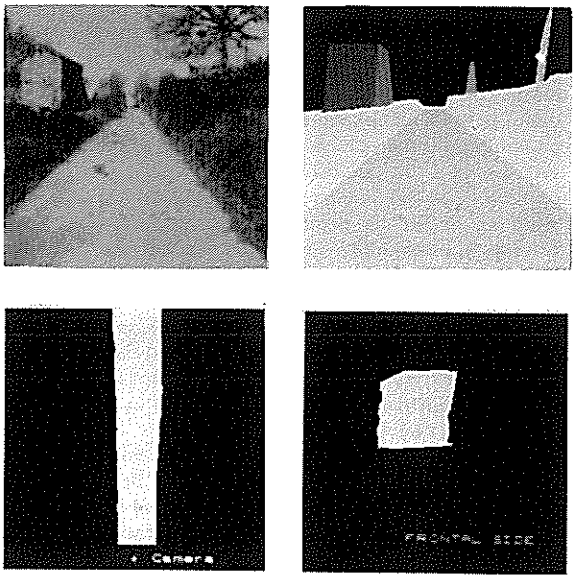


Fig. 6: Original and CAD model images (top) and 3D image interpretation of objects: road and house (bottom).

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A BI-DRIVEN OPTIMAL SEARCH FOR KNOWLEDGE-BASED VISION

Heinrich NIEMANN *)□)
Włodzimierz KASPRZAK □)

*) Universität Erlangen-Nürnberg
Lst. für Informatik 5 (Mustererkennung)
Martensstr. 3, D-8520 Erlangen, F.R.G.

□) Bayerisches Forschungszentrum für Wissensbasierte Systeme, FG Wissensverarbeitung
Am Weichselgarten 7, D-8520 Erlangen, F.R.G.

A domain-independent tree search algorithm for semantic network-based image understanding is presented. The basic transition operators that provide search space expansion, were designed for a (hierarchical) *model-to-image* match. Two operators for *data-dependent* matching are also defined. The first one forces an *iteration* of a model-to-image match, the second one concerns the instantiation of *generic relations*. Minimal *data requirements* of conceptions are applied for search tree pruning.

1. INTRODUCTION

In this paper the image interpretation problem is viewed as an optimal forward search in an implicit space of partial symbolic descriptions [1]. A *semantic net* implementation system *ERNEST* [2, 3] and the *A*-tree search* algorithm constitute the base of presented approach. The basic transition operators that provide search space expansion were designed in *ERNEST* for a (hierarchical) *model-to-image* match.
But one needs *data-driven* search operators in vision systems too because the number of non-competitive instances of given conception may be image-dependent (cannot be predetermined in the model).
Two such operators are proposed here for *data-dependent* matching. The first one causes an *iteration* of the model-to-image match. It is applicable in following cases: an unlimited number of object instances may exist in a scene; iterative volume parts of a solid class may exist; non-merged segments may exist in the image description due to segmentation faults.
The second operator concerns the instantiation of *generic relations* for the verification of hypotheses and for consistency maintenance.
At end minimal *data requirements* may be specified for each conception. They allow a pruning of search space nodes while retaining the admissibility of search.

2. KNOWLEDGE-BASED ANALYSIS IN *ERNEST*

The semantic network in *ERNEST* provides three node types: the *concept*, the *modified concept* and the *instance*, and three link types: *part*, *concrete*, *specialization*. A part is *context dependent* or *not*. Part- and concrete-links of a concept are aggregated into *modality* sets, and each link is marked there by one of the labels: *obligatory*, *optional*, *inherent* or *reference*.
There are three domain-independent rules for the *instantiation* of concepts and three rules for the *modification* of concepts, that describe the use of knowledge.
First a *partial* instance of concept A or modified concept mod(A) is computed by requiring only instances of the *context independent* parts and concretes (RULE 1).
Having the partial instance of A instances of *context dependent* parts M can be computed. Having instances of M the partial instance A may be *completed* (RULE 2). RULE 3 checks whether there are instances of *optional* parts or concretes and generates *extended* instances from a complete instance of A. Constraints can be propagated upwards (RULE 4) or downwards (RULE 5) in the knowledge hierarchy. Initial modifications of concepts are derived by the application of RULE 6 directly to the image data. The rules for instantiation and modification in connection with the *A**-tree search algorithm form the skeleton for different control strategies. The basic *alternating* control consists of a bottom-up selection of (temporary) goal concepts and of matching them to the image data. This matching process is tailored into a top-down *model expansion* (inverse application of the instantiation rules combined with modification of expected conceptions) and bottom-up *instantiation* until the application specified goal is reached.

3. THE BI-DRIVEN CONTROL (Table 1)

3.1 Data-driven goal selection

The primary problem is to find an optimal path in the graph of modified goal concepts. This graph is extended over the concrete-of- and specialization-hierarchies of the model (and the Z axis for multiple modifications of a concept) and a path leads from some initial goal concept from the set C_g to some terminal one (from the set of most abstract and most specialized concepts in the model net).

The search tree is expanded by the application of following operators:
- initilization by the application of RULE 6 to the image data; one successor node is generated for each initialized goal
- superior goal generation (applying RULE 4 to the instance of actual goal); one successor node for each modified superior concept
- more specialized goal generation (applying the inheritance mechanism to the instance of actual goal); one successor node for each partial instance of direct specialization concept

Input: APPLICATION function to provide a list C_g of competing goal concepts					
Initialize: search tree S= (V,E) with V={R}, E= ∅; lists OPEN= ∅, CLOSED= R					
provide APPLICATION function for initial parameters					
FOR all concepts $K \in C_g$ DO:					
apply RULE 6 to K					
FOR all modified concepts $Q_i(K) = o_i$ generated by RULE 6 DO:					
generate one successor node V_i^K of root R in search tree S DATA(V_i^K)= {o_i }; GOAL(V_i^K)= o_i; $h(V_i^K)$= judgement(V_i^K)					
IF	K is a *minimal concept*				
THEN	OBL_PREM(V_i^K,O_i)= T				
ELSE	OBL_PREM(V_i^K,O_i)= F				
refer unlimited objects in DATA(V_i^K) in ITER[V_i^K]					
IF	the segmentation data satisfy MIN_REQUIRE[V_i^K]				
THEN	add V_i^K to OPEN				
WHILE OPEN is not empty DO:					
select the node N with best score from OPEN					
remove node N from OPEN; add it to CLOSED					
IF	the APPLICATION decides that an analysis goal or an end has been reached				
THEN	STOP - successful end of search or end of resource				
activate APPLICATION function to provide a (possibly empty) set S of new goal concepts					
IF	S is not empty				
THEN	FOR all concepts $C_i \in$ S DO:				
	apply RULE 4 to C_i				
	FOR all objects o_l generated this way DO:				
	generate one successor node V_{il} of N in S; add V_{il} to OPEN DATA(V_{il}) = DATA(N) ∪ { o_l }; OBL_PREM(V_{il}, o_l) = F h(V_{il}) = judgement(V_{il}); GOAL(V_{il}) = o_l				
ELSE	IF	one object $o_l \in$ DATA(N) can be instantiated by one of the RULES 1-3			
	THEN	activate ERNEST function *instant* (N) to instantiate the model in node N determine the set Next(N) of successor nodes of N in OPEN activate ERNEST function *consistency_check* (Next(N)) FOR all nodes $N_i \in$ Next(N) DO:			
		refer the unlimited objects in DATA(N_i) in ITER[N_i]			
		FOR all unlimited objects $T_{(i)} \in$ (ITER[N] - ITER[N_i]) DO:			
		generate one successor node N_i^1 of N in S and OPEN copy N_i to N_i^1; DATA(N_i^1) = DATA(N_i) ∪ $T_{(i+1)}$ ITER[N_i^1] = ITER[N_i] ∪ $T_{(i+1)}$ extend the premises of superior objects of $T_{(i)}$ by $T_{(i+1)}$			
	ELSE	IF	there is at least one object $o_l \in$ DATA(N) with OBL_PREM(N,o_l) = F		
		THEN	activate ERNEST function *expand* (N) to expand the model in N		
			determine the set Next(N) of successor nodes of N in OPEN activate ERNEST function *pruning* (Next(N)) to prune the nodes $N_i \in$ Next(N) from OPEN if the available segmentation data does not satisfy MIN_REQUIRE[N_i]		
		ELSE	IF	there is at least one object $o_l \in$ DATA(N) with OPT_PREM(N, o_l) = F	
			THEN	activate ERNEST function *opt_expand*(N) to expand the model in node N	
			ELSE	activate ERNEST function *opt_spec*(N) to consider optional parts and specializations	
STOP – unsuccessful end of analysis					

Table 1: The bi-driven search

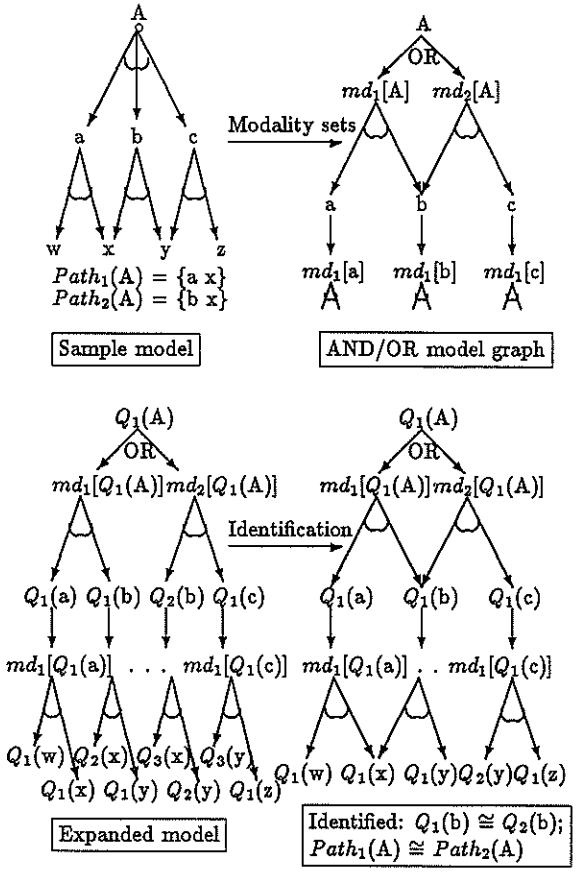


Figure 1: Model expansion with path identification

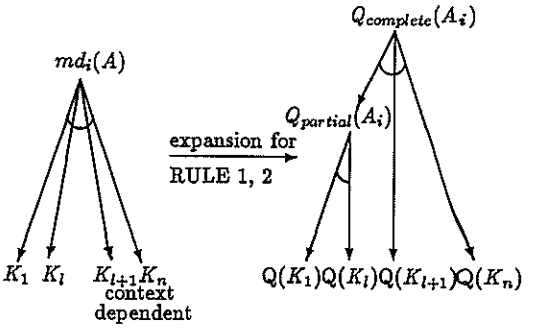


Figure 2: Expanded obligatory model

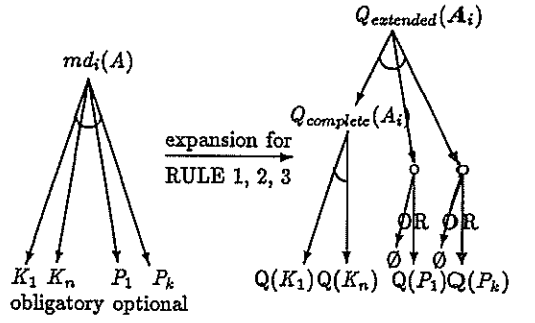


Figure 3: Expanded optional model

3.2 Model–to–image matching

The parts and concrets of a concept are aggregated into a finite set of competitive *modalities* (*md*), i.e. subsets of parts and concrets. The match of selected goal to the image data is a combination of two search problems: a search for a best solution graph in an AND–OR graph (expanded model) M for current goal A (Fig. 1) and the search in the space of competitive instances of entities from M. The entities in M are modified concepts created for model paths starting in actual goal A. These modified concepts are refered by so called *object*–data structures (denoted by Q_i) in a search space node. Due to the *identification* of equivalent paths (as specified in the model) or equivalent objects (from various modality sets of one superior object), one object can represent multiple paths.
Hence there are two search operators applied during the basic matching process. Successors of a search tree node are created either for competitive premises of instantiation (due to different modalities and different modifications made by RULE 5) or for competitive instances of one object (Q_i) from M.
Actually the model expansion mechanism is more complex than presented on Fig. 1 because the elements of one modality are classified into obligatory or optional (among others).

For given goal concept an *obligatory model*–to–image match is performed. In this case only RULE 1 and RULE 2 are considered during the model expansion process (Fig. 2). Due to the optional parts needed in the RULE 3 the *optional model*–to–image match has to be distinguished. Before an *extended* instance can be created by RULE 3 out from the *complete* one, optional part–instances are searched for (empty instances are allowed) (Fig. 3).

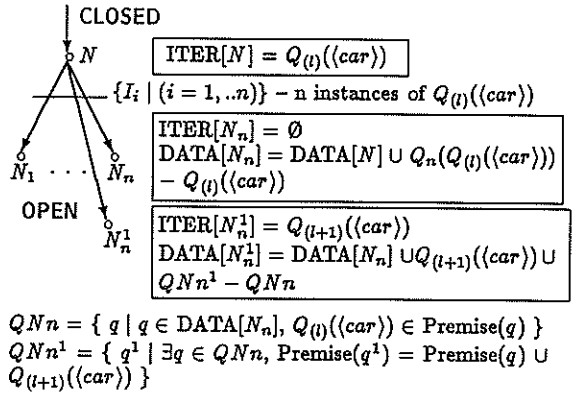
The matching process consists of interlaced expansion– and instantiation–steps. The instantiation step has always the greatest priority. By applying RULES 4, 5 to new generated instances the objects from the data set DATA(N) can be more constrained. In this way the later expansion of each such object can be restricted to these premises only, which satisfy the new obtained constraints.

For the judgement of search space nodes an estimation of the goal object judgement with respect to the set DATA(N) is made. This measure satisfies the admissibility requirements for the A^*–tree search algorithm.

3.3 Iterative match for "unlimited" objects

If the item *dimension* of some (optional) link is filled by the number "unlimited" then it will be searched for an image–dependent number of instances of the apropriate concept. This is represented in the expanded model by an *unlimited* object. The set of such objects in a node N is recognized by the operator ITER[N]. After n instances of an *unlimited*

object (for example object $Q_{(l)}$ of concept $\langle car \rangle$) has been created, the search space node N is firstly expanded by nodes $N_1, ..., N_n$ as usual (Fig. 4).



$QNn = \{ q \mid q \in \text{DATA}[N_n], Q_{(l)}(\langle car \rangle) \in \text{Premise}(q) \}$
$QNn^1 = \{ q^1 \mid \exists q \in QNn, \text{Premise}(q^1) = \text{Premise}(q) \cup Q_{(l+1)}(\langle car \rangle) \}$

Figure 4: Iterative search for object $\langle car \rangle$

After selecting one of the nodes N_i (i=1,...,n) for expansion one additional successor node N_i^1 of node N is created. From this new node the match of those unlimited object will be iterated because a next version $Q_{(l+1)}(\langle car \rangle)$ of the unlimited object is added to the set DATA(N_i^1).
The premises of all superior objects of the unlimited object have to be changed in order to include the next version of this object. The iteration stops because of the shortage of image data – no data can be interpreted twice on one path in the search space. Thus in the subsequent iteration only those image data can be matched which is not interpreted by instances from DATA(N) yet.

3.4 R–objects for generic relations

A specific unlimited object, called a R–object, is given if its part and concrete–links are labeled by the *reference* labels. Such links are not expanded – there are not made distinct objects for concepts reached by them. For each object tuple of concepts from the expanded model that satisfy the premise of the R–concept one apropriate R–object is generated in the expanded model (Fig. 5). This set may be extended by new objects generated during analysis if some link of the R–object refers an "unlimited" object.

One application of generic relations is the representation of relationships for consistency maintenance of the search space. After an inconsistent DATA set was discovered (so called *NOGOOD* search space node) the procedure *inconsistency_check* tries to detect inconsistent subsets. Nodes which contain at least one of the detected inconsistency can be removed from the OPEN set.

3.5 Minimal data requirements for non–expanded objects

Let us finish with a data–driven pruning of search space nodes. A set MIN_REQUIRE is specified for each search node. It contains the minimal image data requirements of all non–expanded objects in given node. In order to pass the pruning test the subset of image data that is available in a node should include the apropriate MIN_REQUIRE set.

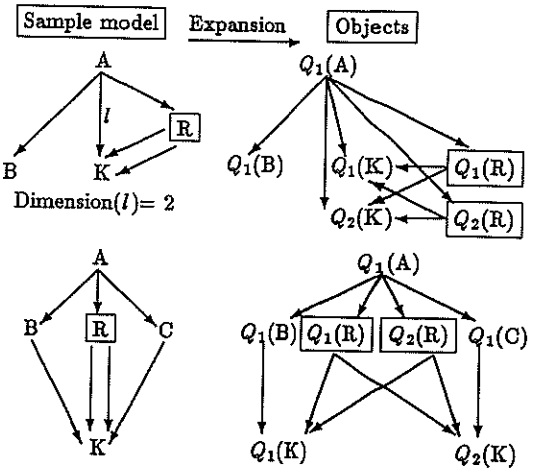


Figure 5: Expansion of the R–concept R

4. CONCLUSIONS

Both model– and data–driven search operators were proposed for knowledge–based image analysis and integrated in an optimal forward tree search.
Contrary to the basic model expansion the number of some objects ("unlimited", R–objects) is not determined by the model, but depends from actual analysis results.

ACKNOWLEDGEMENTS

Dr W. Kasprzak wants gratefully to acknowledge the support from the Alexander von Humboldt–Foundation, Bonn, F.R.Germany.

REFERENCES

[1] Niemann H. et al.: *A Knowledge Based System for Analysis of Gated Blood Pool Studies.* **IEEE Trans Patt Anal Mach Intell, PAMI-7** (1985), 245–259.

[2] Sagerer G., Kummert F.: *Knowledge Based Systems for Speech Understanding.* In:[Niemann H. et al (Ed.): **Recent Advances in Speech Understanding and Dialog Systems.** NATO ASI Series, F46, Springer Vg., Berlin 1988], 421–458.

[3] Niemann H., Sagerer G., Schröder S., Kummert F.: *ERNEST: A Semantic Network System for Pattern Understanding,* **IEEE Trans Patt Anal Mach Intell, PAMI–12**(1990), (*in press*).

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A FIRST STEP IN THE BUILDING OF A SPECTRAL ANALYSIS EXPERT SYSTEM.

C. Adnet N. Martin

CEPHAG, UA346 CNRS, ENSIEG, BP 46, F-38402 ST Martin d'Hères Cedex

In this communication we present the first results of the study of a spectral analysis expert system. It is a user help for choosing methods or algorithms and their associated parameters. Its aim is not to pick out one and only one method, but a set of methods characterizing the signal in the frequency domain. We also address the problem of filtering and undersampling a signal under analysis before using a high resolution algorithm. These processes allow the enhancement of the signal to noise ratio and to improve the resolution.

I) INTRODUCTION

In order to carry out the signal spectral analysis, one has numerous tools or methods in hand. But using or choosing suitable algorithms may be not very easy. In addition, the selection of their parameters is, sometimes, critical. In this communication we are proposing, as user help for picking out methods or tuning parameters, the use of a Knowledge base or Expert sytem. The proposed-system relys on the description of the signal and the algorithms in term of objects. As a matter of fact, the signal under analysis has explicit characteristics (length, sampling period, more or less noisy, stationarity...) and physical properties owing to the experiment it comes from. The methods, in the same way, possess properties and characteristics depending on their application field. Section III brings out the static and dynamical aspect of this system.
In section II the unification principle is discussed. Its aim is to put some spectral analysis methods together, emphasizing their similarities.
In section IV we address the problem of filtering the data as first step to retrieve the frequencies of an harmonic process. It will be shown that this filtering process improves the result of the high resolution methods which is following it.

II) UNIFICATION

The unification principle stems from filter bank analysis [1,2] and states that many methods may be considered as a minimization of a quadratic form, a power of finite impulse response filter with specific constraints. The method is discribed by three steps:

calculus of a "correlation matrix".

calculus of the filter's parameters.

extraction of the relevant information.

Besides, this approach emphasizes 2 families of methods depending on whether the associated filter acts on the data as a band pass filter (familyI) or retains the information (familyII).

familyI

The first family includes FFT-based methods and CAPON-LAGUNAS [1] methods. With the former methods the filter's shape is independent of the data but is used to value resolution, statistical stability and other features of the final spectral estimate which is either computed from the data or from an estimated "correlation" matrix . For the later the filter's shape is deduced from the data, the final spectral estimate is computed from an estimated "correlation" matrix.

familyII

For this family the filter keeps information. It includes autoregressive (AR) or linear prediction (LP) methods, Tufts-Kumaresan's method (TKM) [3]... From the filter one can compute the power spectral density, the filtered signal, the roots of a polynomial whose coefficients are the filter's entries.
Unfortunately all the methods do not fit the unification framework. we shall retain those are useful to retrieve frequencies of an harmonic process. These methods such as MUSIC [3], TAM [4], take advantage of the specific structure of the "correlation-covariance" matrix of this process.

III) KNOWLEDGE BASE.

III.1) STATIC ASPECT.

Our goal is to build an expert system (ES) helping the characterization of the signal under analysis, in the frequency domain. Of course, such a system will have to make use the above mentioned algorithms. Its aim will not be to choose one and only one method, but a set of algorithms relevant to the analysis. In order to carry out this work we are using a expert system generator KOOL. It combines an object representation of simple inheritance, with order 1 production rules (antecedent concequent pairs where an antecedent specifies conditions under which actions decisions, inferences take place). In KOOL the knowledge is structured by three levels : classes, subclasses and instances (fig 1).

-->the classes: they define a generic structure of the manipulated objects. Their properties and characteristics are described by attributes. Moreover, the classes enclose the links instances of a class may have with other object.

-->The subclasses: they inherit the properties of their classes but can have their own attributes. The subclasses enables factorizations and specializations among objects belonging to the same class.

-->The instances: they describe a concrete case. Each instance belongs to a subclass or class defining its structure and properties.

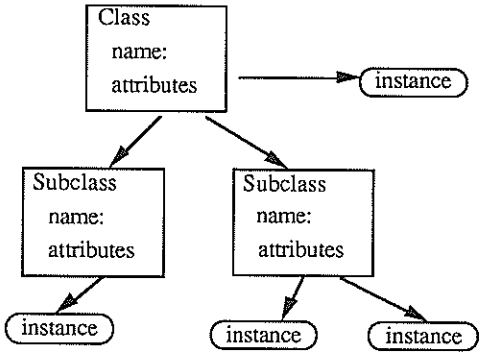


fig 1. : three levels knowledge.

This object representation is well suited to our problem. Signal and algorithms are instances of more general concepts: SIGNAL CLASS and ALGORITHMS CLASSES, specifying their properties, characteristics and links.
The knowledge base we build is organized in seven classes.

-SIGNAL-Class.

-FOURIER-Class.

-Correlation Covariance Matrix-Class: it describes different estimators of the process's correlation matrix: biased or unbiased estimators, forward-backward estimators.

- HR-Class: containing high resolution methods
  1) output structure: Lagunas
  2) filter structure: AR, TKM, PISARENKO
  3) correlation structure : TAM, MUSIC,..

-ISM-Class: (Information on Spectrum or Model.). This is the information obtained during the analysis.
  1) filter output.
     spectrum.
     frequencies: peaks of the spectrum
  2) frequencies: from internal representation of the filter (its poles)
  3) frequencies from HR-C.3

-POWER-Class: power representation of the signal
  1) CAPON
  2) phases amplitudes associated to frequencies (second step of Prony's method)

-MUSTIG-class: This class allows the ES to communicate with a signal processing software [5]. Its attributes find their values among the above classes.

III.2) DYNAMICAL ASPECT

This section is focused on the problem of finding a dynamical strategy providing a frequency characterization of the signal. In the above section the classes depicted the static state of the system, the modelization of a spectral analysis workshop. The dynamical aspect, that is to say the way the analysis is carried out, is depicted by the rule base. The rule base encloses procedural knowledge which results from the experience and knowledge of the experts in signal processing. Rules are writen down as an antecedent-concequent pair where an antecedent specifies conditions under which actions decisions, inferences take place. There are 2 sets of rules.

The first set of rules allows to produce qualitative information from quantitative one, at the beginning of the analysis or during its course. For example the length of the signal (short, intermediate, long) or the signal to noise ratio (low,intermediate,high) take their qualitative values by means of rules. These informations fill the base of facts.
The second set of rules defines actions or decisions such as picking out parameters or methods.
The general strategy is divided into 2 steps: primary and secondary analysis.

-The primary analysis is computed with 2 Fourier-based methods: 'Correlogram', 'Wosa' [6]. Their parameters are deduced from the signal's attributes or from prior knowledges on the signal. They set up the resolution and the stastistical stability of the spectral estimates. This step provides spectral density estimates computed by Mustig and interpreted by a layer of rules. The extraction of the caracteristics of the spectral pictures (broad-band, narrow-band..) involves a discussion with the user, increasing the base of facts.

-The secondary analysis consists in improving the former results. Three directions are then allowed. The analysis may be procedeed with the previous Fourier-based methods, with new Fourier-based methods such as STUSE, LAG-RESHAPE [6] or with HR algorithms. The choice of the new analysis direction is guided by the results of the primary analysis, with the aim of improving the resolution or confirming the previous spectral shapes. When HR methods must be brought into operation, the choice of an algorithm is guided by the sisnal's type. If it is a broad-band signal the ES will activate Lagunas's method. If it is a narrow-band signal , an AR method will

be used. And, from the primary analysis, if the signal is thought to be a sum of sinusoids, specific algorithms such as TKM, TAM, MUSIC... will be chosen. The parameters of the method which is held by the ES are chosen by means of theoritical or empirical criteria.

IV) HIGH RESOLUTION ON A FILTERED SIGNAL

A primary analysis of the signal may point out that the spectral information is located in a certain region of the frequency domain. As a matter of fact, when the signal consists of closely spaced sinusoids in additive noise, an ordinary Fourier's method as a periodogram (square of the Discrete Fourier Transform of the signal) exhibits more energy at some frequency locations. This information can be used to filter the signal on a frequency band. This way, the signal to noise ratio can be enhanced. Moreover in order to achieve a better resolution the signal is undersampled in a non usual way.

-Filter design technic:

So as to build a FIR (finite impulse response) filter, a frequency band (2F) and its central frequency fo is first selected. The filter's impulse response r(n) is constructed from the weighted impulse response of the ideal bandpass filter [7].

$r(n) = h(n)\, w(n)$ $\quad -M \le n \le +M$
w(n) weighting window

$h(n) = \sin(2\pi Fn) / \pi n \; 2\cos(2\pi fon)$ $\quad -\infty < n < +\infty$
ideal impulse response

The type of windows sets, the transition bandwidth between the passband and the stopband and the rejection.The parameter M sets the impulse response duration. As M is increased the overshoot is confined in a smaller frequency range. Fig 2 shows the frequency response of a bandpass filter centered on fo=0.234 hz and F=0.708 hz, the ideal impulse was weighted by a Blackman Harris 4-terms window of length 2M+1=135 producing a rejection of -74 db in the stopband.

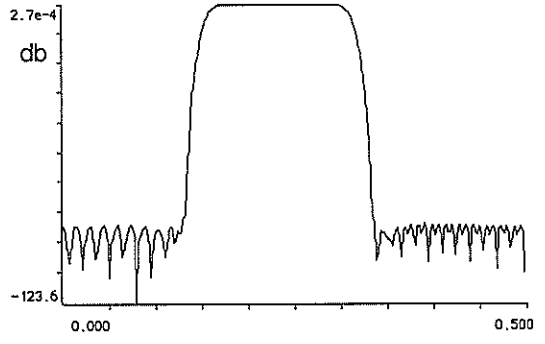


fig2: frequency response

-Improving the resolution:

The filtering process is carried out by means of the Fast Fourier Transform algorithm:

$x_{filtred}(n) = TF_N^{-1}\,(\,TF_N(x(n)) \times TF_N(g(n))$
TF_N^{-1} N-point DFT.
TF_N N-point IDFT.
$N \ge 2M+L$, L = length of the signal.

Instead of working on these filtered data, one builds a bandpass filter with the same bandwidth 2F but centered on $f1 = F+\delta$ where δ depends on the length of the transitionband. Furthermore it is assumed that $2\,f1 \le Fe/4$ where Fe is the sampling period of the signal. Then the frequency domain of the signal that one wants to study is shifted (positive and negative frequencies) in the filter's passband. Next the IDFT is performed on the frequencies ranging from [-Fe/4, Fe/4]. This second step may enable a better resolution, as the high resolution methods which are following this filtering process are sensitive to closely spaced sinusoids: before filtering if 2 frequencies are k Fe/N away, after filtering they are 2k Fe/N away.

simulations and discussion:

This section provides some simulation results. The considered problem is the retrieval of 2 closely spaced sinusoids in additive white noise from 256 data samples. More precisely assuming the sampling period Fe = 1 hz, the frequencies are 0.260 hz and 0.270 hz, the signal to noise ratio are -3db and +7db respectively. The filter is computed as mentioned earlier, with a Blackman-Harris weighting window of length 135.

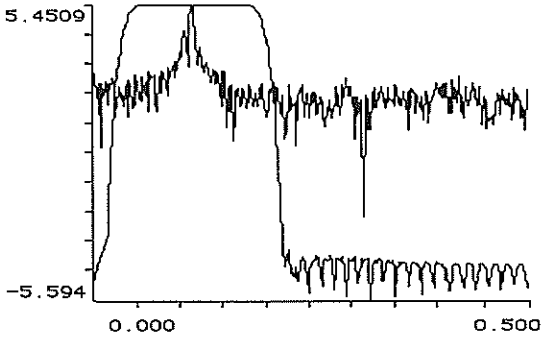


fig3 response filter and shiftedsignal's PSD.

Fig 3 displays the superposition of the frequency response of the filter with the shifted signal's power spectral density. The filtering provides 60 samples, when transient is not taken into account. Next to retrieve the frequencies of the signal one uses the TUFTS-KUMARESAN 's method (TKM). They are phases of the roots of a polynomial whose coefficients represent a vector of minimal norm; this vector belongs to the noise subspace of the filtered signal's covariance matrix.

In this study, the TKM, with polynomial order 32 is performed (it is assumed that the number of sinusoids is known). Figures 4.1 and 4.2 display the zeros of the polynomials for 60 independant trials computed from the filtered and non filtered data respectively. It is shown that the filtering and 'undersampling' processes allow a better resolution and detection of the 2 sinusoids. This filtering process is achieved by a FIR filter, a futher study will investigate the effects of this process, for example the influence of the zeros's filter (false detection), the emergence of a colored noise after filtering. Other type of filter such as infinite impulse response filter (IIF) shall be studied too.

V) CONCLUSION

As it was mention earlier, the goal of this expert system is to characterize the signal in the frequency domain. The Fourier-based methods yield spectral density estimates which can be useful as references for deeper analysis. Currently we are seeking the links or relations beetwen methods.The underlying problem, in fact, is to value the spectral density estimates, and the most suitable methods for improving estimates. In addition, this system will include schemes for pre-processing the data (filtering, trend removal..) or linking together some methods (AR method + Fourier-based method).

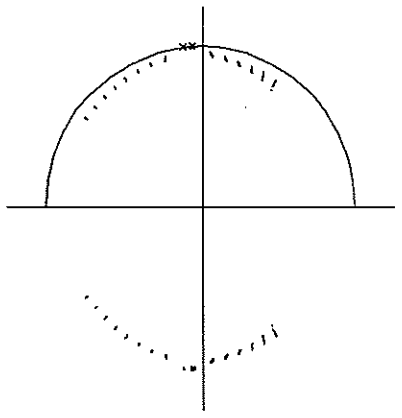


fig 4.1 : zeros of polynomials computed from filtered data.
x : true poles

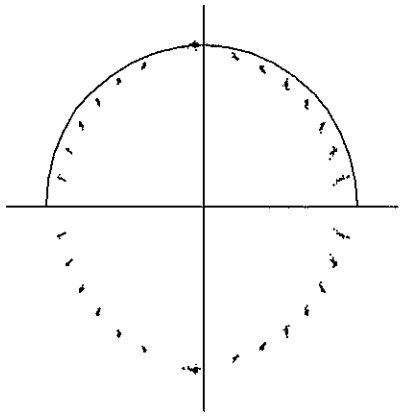


fig 4.2 : zeros of polynomials computed from the unfiltered data.

References

[1] M.A. Lagunas M.E. Santamaria A. Gasull
maximum likelihood filters in spectrum estimation.
Signal Processing n°10 1986
[2] N. Martin J.L. Lacoume M.E. Santamaria
Unified spectral analysis. EUSIPCO 88 Grenoble (F)
[3] S.L. Marple
Digital spectral analysis with applications. Prentice Hall
[4] S.Y. Kung D.V.B. Rao
State space and SVD-based approximation methods for the harmonic retrieval problem.
J.O.S.A Vol 73 n°12 87
[5] G. Lejeune J. Lienard
MUSTIG. GRETSI Juan les Pins France.
[6] V.J. Mathews D.A. Youn N. Ahmed
A Unified approach to non parametric spectrum estimation
IEEE ASSP Vol 35 n° 3 March 87
[7] L.R. Rabiner B. Gold
Theory and application of digital signal procesing
Prentice Hall 1975

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A KNOWLEDGE-BASED INTERFACE TO ASSIST IN SIGNAL ANALYSIS.

Roberto Barbò (*), Claudio Ferri (**) and Paolo Salvaneschi(**)

(*) Department of Computer Science, University of Milano
(**) ISMES S.p.A.
Viale G. Cesare, 29
24100 Bergamo, Italy

A knowledge-based interface able to assist engineers with the use of a signal analysis software package is described. A conceptual model expressing different types of knowledge and reasoning has been developed and implemented, creating a specialized shell for the generation of knowledge-based aids in this specific field.

1. INTRODUCTION

In order to use signal analysis software packages effectively, engineers require not only data bases, a full range of computational tools, and an efficient user interface, but also a *knowledge-based system* to advice on which tools should be employed and how to use them.

In the field of signal analysis there is a growing interest in methods and technology related to Artificial Intelligence [1,2]. Emphasis has generally been placed on new programming technologies and less attention has been paid to the identification and representation of various types of knowledge and a clear separation between conceptual modelling and implementation phases.

Our work [3] fits into this context with two objectives. First of all, to model the problem solving processes and the knowledge used by engineers in carrying out signal analysis by means of complex software. Secondly, to develop a specialised shell for generating knowledge-based aids for signal analysis.

2. THE PROBLEM

Our work arose from the need to give support to engineers in the use of ISA (Ismes Signal Analysis) [4]. This is a software package developed by ISMES and now widely used in the processing of signals, related to the dynamic behaviour of civil and mechanical structures, and vibratory phenomena.

The problems in operating this system are concerned not with ISA "user friendliness" but rather with the *level of the available domain knowledge*. In fact an engineer using ISA must possess a background of specialised knowledge, both theoretical and experiential. Such knowledge is not always easily accessible bearing in mind the variety of computational tools that are available and needed in solving a specific engineering problem.

The approach adopted has been that of augmenting ISA by developing a knowledge-based interface able to help with the choice and the use of signal analysis tools. To develop the system we need firstly to model the knowledge and problem solving of the engineers. Of particular importance are the identification and adequate representation of the various types of knowledge used during the reasoning process and the clear separation of the problem modelling phase from its implementation in a specific environment. Secondly, bearing in mind the need for supplying expert advice for different signal analysis activities, a specialised shell must be developed for generating corresponding knowledge-based aids.

3. THE SYSTEM MODEL

The ISA assistant is made up of: (1) a *domain knowledge model,* (2) *reasoning mechanisms,* and (3) a *man/machine interface.*
For a specific signal analysis activity, it is possible, using the system, to build a model of the knowledge related to the activity itself (fig.1). The reasoning mechanisms use the model enabling the system to help the user in the execution of the relevant activity by means of a man/machine interface.

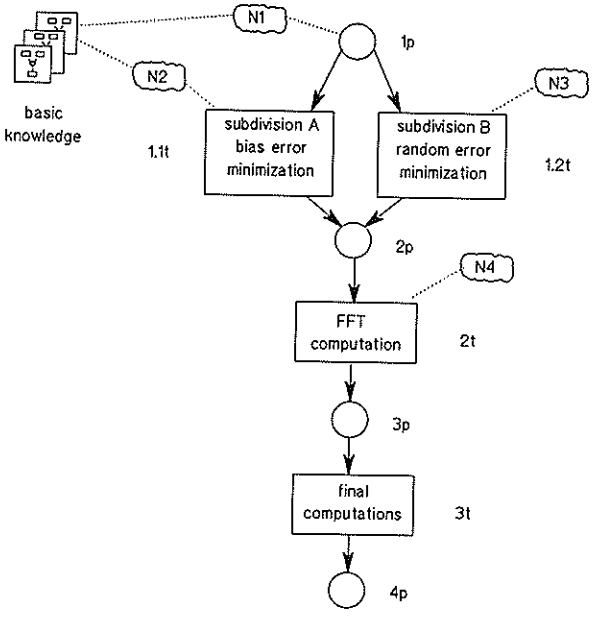


Figure 1

The net for a spectral density computation, using Bartlett method. The node 1p is a decision point and the nodes 1.1t, 1.2t and 2t are transitions with expert support.
The modules N1 and N2 utilise basic knowledge.

3.1. The knowledge model

We have identified three types of knowledge employed whenever any kind of signal analysis is undertaken: *procedural knowledge, heuristic knowledge* and *basic knowledge.*

3.1.1. Procedural knowledge

The result of the problem solving activity of an engineer is the identification of possible procedures applicable to sets of signals for a specific goal. The engineer uses *schemes of possible procedures* for a given class of known problems. The formal approach used in modelling these procedural schemes is that of Place-Transition Petri nets (P/T nets) [5]. This type of scheme is composed of well-structured procedural knowledge linked to decision points which have to be resolved in order to create an executable procedure.

The P/T nets employed make use of a particular interpretation based on a particular meaning given to their components in our application. A *place* is interpreted as containing a signal. Specific types of places are decision and evaluation points. A decision point (conflict point in Petri nets language) represent situations in which the engineer must decide on the path to be followed within a procedural scheme. An evaluation point represents a state in which the user, dissatisfied with the (partial or final) results can order a re-run of the procedural scheme. In this case, using the execution history, the system is able to offer advice bearing in mind the unsatisfactory nature of the earlier choices. The execution history is a data structure in which information related to execution of the procedural scheme under consideration is recorded.
A *transition* contains a signal analysis activity. There are particular types of transitions, such as compound transitions and transitions with expert help. Within our system it is possible for a transition to be exploded into further P/T net. Transitions with these characteristics are called compound. This hierarchical structuring of P/T net requires an appropriate hierarchical control mechanism: the net at a lower level of abstraction inherits information on initial conditions from net at the higher level; it returns information to the higher level at the end of the execution. Transitions with expert help are those requiring help in defining the processing parameters for signal analysis. Finally, a *token* [5] models a signal.

3.1.2. Heuristic knowledge

In our system, heuristics are used to aid the execution of a procedural scheme. Heuristic knowledge is made up of various *rule-based modules,* each associated with a specific node

(place or transition) in the procedural scheme. The rule-based modules have different functions according to the type of node with which they are connected. If the node is a transition (with expert help), the rule-based module helps the engineer to determine correctly the processing parameters for the related computational activity. If the node is a place (either a decision or evaluation point), the rule-based module aids the user in the related decisional processes. Each rule-based module may call modules of basic knowledge.

3.1.3. Basic knowledge

During the problem-solving process, the engineer makes use not only of heuristic knowledge, but also of theoretical. The latter is comprised of *fundamental mathematical relationships* existing between the various parameters involved in signal analysis. Such relationships have been represented by means of concept-actor nets (C/A nets) [6]. In our particular case, each actor contains a set of mathematical formulae while a concept models a parameter.
C/A nets are clustered in modules, each related to a specific key concept in signal analysis (i.e. sampling or time/frequency transformation).

3.2 Reasoning mechanisms

A specific reasoning mechanism is activated for each type of knowledge. Initially, the *procedural knowledge reasoning mechanism* is activated. Beginning with an initial set of tokens, it executes the Petri net. A rule-based module is activated as soon as execution of the net reaches the node to which it is linked. The *heuristic knowledge reasoner* is then activated, using a mixture of backward and forward chaining. The *basic knowledge reasoning mechanism* may be activated by a rule-based module, in a goal-directed execution of the C/A net with the aim of calculating the value of a particular variable.

3.3. The man/machine interface

At the moment, our system interacts with the user on the basis of a Question/Answer interface, although we are presently working on an animated and interactive graphic presentation. The concept of an interface with a graphic presentation is consistent with the representation of knowledge in net form and enables the execution of the graphs to be animated on the screen. Hypertext features allow the user to interact with the graphic objects displayed on the screen.

4. INTEGRATION WITH ISA

Although the present system supplies only "off-line" help, it will in the future be integrated with ISA. Communication will take place by means of three modules, whose respective functions are the generation of partial ISA procedures, the running of ISA tools, and the data acquisition from the ISA data base. Furthermore the support system interface will become part of the ISA interface in a workstation environment.

5. SYSTEM IMPLEMENTATION

The system has been developed on a SUN/UNIX workstation, using Nexpert Object, a hybrid environment for expert systems development, and the C programming language. Nexpert Object has been used as an object-oriented language for the implementation of P/T and C/A nets and for the coding of rule-based modules (using also its rule-based language). The reasoning mechanisms related to both types of nets have been developed using C language.

The system consists of a *specialised shell* which permits the creation and running of knowledge-based signal analysis aids. Using the shell, the prototypes of two specific aids have been developed in order to validate our approach. One of these gives advice on the computation of the spectral power density, using the Bartlett method, while the other gives advice on choosing a digital filter from a set of possible options.

6. CONCLUSIONS

Our work has led to two main results. Firstly, the definition of a conceptual model suited to the modelling of knowledge-based aids in the field of signal analysis. This kind of model integrates and separately represents the various types of knowledge. Secondly, the implementation of a

specialised shell which has been used for the development of two specific assistants.

REFERENCES

[1] Li, X., Morizet-Mahoudeaux, P., Trigano, P., Gaillard, P., "A Spectral Analysis Expert System", Dept. Génie Informatique, Compiègne Cedex (FR), 1985.

[2] Gui Lin Nie, Morizet-Mahoudeaux, P., Gaillard, P., "A Knowledge-Based System for Digital Filters Synthesis", EURASIP, 1988.

[3] Barbò, R., "Un Sistema Intelligente di Supporto all'Analisi dei Segnali", Degree Thesis, University of Milano, February 1990.

[4] Salvaneschi, P., Ferri, C., Gambirasi, P., "ISA: a Package for Signal Analysis", Decus Europe Symposium, 1985.

[5] Reisig, W., "Le Reti di Petri", Arnaldo Mondadori Editore, 1984.

[6] Sowa, J.F., "Conceptual Structures", Addison-Wesley, 1984.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

HIERARCHICAL IMAGE SEGMENTATION: A K-B SYSTEM USING FUZZY FUNCTIONS*

Marco Ronco, Roberto Vio, Silvana Dellepiane and Gianni Vernazza

Dep. of Biophysical and Electronic Engeneering, University of Genoa Via All'opera Pia 11A, I-16145 Genoa, Italy

This paper deals with the problem of segmenting tomographic Magnetic Resonance images. The system presented in this paper a knowledge-based one and utilizes different segmentation criteria to reach a hierarchical decomposition of an image; the hierarchy corresponds to different levels of detail on segmentation. Fuzzy functions to represent uncertain situations are also used togheter with different levels of detail of resolution. The segmentation process performed by the system is based on the integration of two kinds of primitives: edges and regions.

1. INTRODUCTION

Segmentation of an image is the process that splits it into primitives. For our purposes, a successful segmentation yields regions which can serve as a useful input to a higher-level vision process, i.e. the interpretation of groups of regions such as anatomical slices. This means that segmentation results can be considered satisfactory if they provide a sufficiently rich and accurate description with minimal redundancy to be utilized for future processes.
Algorithms are usually based on one kind of primitives (regions or edges), each able to characterize different image details. This reason has led us, but also researchers e.g. Nazif and Levine [5], to integrate regions and edges to make them co-operate during the segmentation process. This process starts working on a large set of small regions in to which a scene has beeen divided by applaying a fast pre-segmentation algorithm. The idea is to group these regions by successive merging steps: the possibility of merging neighbouring regions depends on the similarity of their features and on the positions with respect to edges. The whole process is accomplished using a "blackboard system".
In the following, a detailed description of the system is given, together with some results on MR slices.

2. SYSTEM DESCRIPTION

The whole segmentation process can be subdivided into two separated phases: in the first one, the numerical processing and the transformation from the numerical into the symbolic level are performed; in the second one the partial results of the first phase are managed to produce the final segmentation results.

2. 1. Numerical level

This is the first part of the system where we operate at the level of input images. In order to exploit the information derived from various input images, a numerical data fusion is performed: this yields a very detailed pre-segmentation image made up of regions built according to the gray-level values of the pixels in the two input images describing the same anatomical slice. This allows us to obtain a final image in which organs still consist of small sets of regions, without loos in important details. The following recognition process will merge these sets, using domain- knowledge to identify the anatomical entities in the scene.
To reduce noise images, a filtering process is used: first, a linear filter cuts noise spikes; second, a nonlinear edge- preserving smoothing algorithmenhancees real edges. The latter applies two different sets of masks (see Fig. 1) to take into account and preserve the presence of small structures. The first set is made up of masks in three pixels thickness: they are the

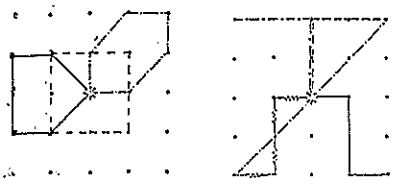


Fig. 1

original Nagao- Matsuyama masks [1]. The second set consist of masks in one pixel thickness as proposed by Vernazza et al. [2]. For each pixel, the selection of the most suitable set of masks is based on the gradient evaluation.
The following step consists in the primitives extraction: by the term "primitives" we mean groups of pixels that have similar features. Two kinds of primitives are used: regions and edges.
Regions are produced by a fast pre-segmentation

*This work has been supported by C.E.E. A.I.M. (Advanced Informatics in Medicine) under a contract granted by CO.VI.RA. (COmputer VIsion in RAdiology)

process (a region-growing method) that groups pixels on the basis of their gray-level values in the T1 and T2 images. The idea is to group only very similar pixels in order to maintain details and, at the same time, to reduce the complexity of the problem.
To extract edges from input images, an edge-detecting algorithm based on the Marr-Hildred approach is used[3].
A feature-extraction process is performed on these sets of primitives. For each region, geometrical and densitometric features are computed, while edges are defined, by their positions and length. This allows the generation of a graph to describe edge-region adjacencies. Regions are the primitives on which the following merging process is based; therefore their characterization is more detailed. Gray level values and variances are computed from the T1 and T2 channels to evaluate the optical similarity, while the main axes are necessary to find similar shape characteristics usefull to the merging process.

2. 2. Symbolic Level

This is the real merging process: it yields the final segmentation results starting from the pre-segmentation image and working on the region and edge primitives. To better understand how it behaves, it can be useful, to divide the description of the system in three sections. The first analyzes the strategy used for the merging process; the second describes the "blackboard approach"; the third treats the merging process in more detail, including implementation characteristics.

2. 2. 1. Strategy

The main idea is to merge adjacent regions that have very close characteristics and that are not separated by edges. This process has to be performed by successive steps, until no more regions can be merged. Thus a complete description of a step can clarify the behaviour of the system.

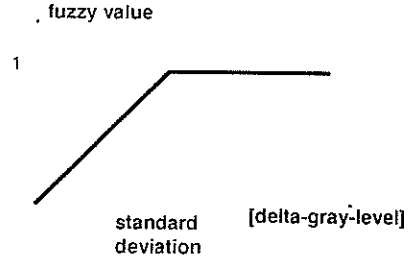


Fig. 2

First, a set of "focus-regions" are selected, on which we must work to find the most promising regions from which to start the merging process, thus allowing for a reduction in computational time. To choose such regions from the whole image, a dynamic "focus- cyclic-strategy" is provided to guarantee the selection of different regions.
Candidate regions are selected is made by comparing each "focus-region" with each neighbouring region to compute the dissimilarity of each feature. By use of fuzzy functions a value in the range [0.0- 1.0] is supplied for each feature: two regions are thus completely characterized by a "dissimilarity-vector" on which to test the feature distances. An example of fuzzy function for gray level dissimilarity is given in Fig.2.
The standard deviation, computed on the whole image, makes the function adaptive to the scene under test.
Only the regions that satisfy a threshold condition on the "dissimilarity vector" and that are not separated by any edge can be merged. This condition must be satisfied for all the vector values but only for a vector sub-part. Thus the merging condition is satisfied when at least n elements of the "dissimilarity-vector" are less than ft and the merged regions have a "reliability" less than rt; (where n is the minimum number of candidate features, tf is the feature threshold, and rt is the reliability threshold). This yields two candidate regions when such regions have similar optical features or similar shapes and are on the same side of an adjacent edge. Reliability has a value (100 for the regions of the starting Data Base) that decreases after each merging according to the dissimilarity (distances of the features) of the regions merged.
The reduced set is further investigated to search for possible conflict situations (a conflict situation occurs when a region can be merged into two different "focus-regions") Such situations are solved allowing only the "best" merging in the sense of the pairs with closer features that produce the minimum reliability worsening.
The feature threshold is gradually increased from 0 to 1, in order to allow the best merging. Only regions with still "good" reliability value are considered for possible subsequent merging. The process stops only when no more regions can be merged.

2. 2. 2. The Blackboard Approach

The blackboard approach, as proposed by B. Hayes-Roth [4], defines a conceptual structure of the system. In this context, it is possible to identify two important conceptual parts: a memory space, called blackboard, in which data must be stored and updated during processing. It also contains information at different levels of detail: primitives with their features and all the intermediate results after each step of the segmentation process. This part can be defined as the passive module of the system. The elements that operate on data represent the active part: they are organized hierarchically, in the present system [6] and are called Tasks. Each of them performs a subpart of the whole process by operating on data. Since contemporary manipulations of data by several tasks are not allowed, a monitor is necessary. This "engine" of the system has to schedule the Tasks according to their needs for operating on data. Each Task consists of a condition-part and an action-part: the scheduler has to test the

condition part to verify the possibility of "firing" the Task, and has also to establish which Task sequence will be

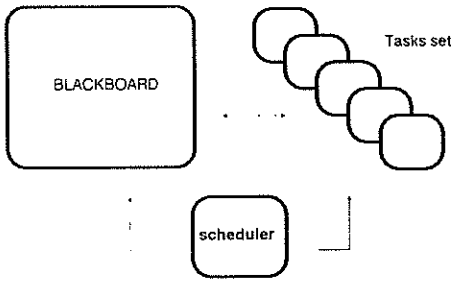


Fig. 3

performed (Fig. 3). A simple way of deciding is to order the Tasks according to their different importance .
A frame net has been chosen to represent regions and edges: each primitive is a frame whose slots are the attributes (features) of the primitive itself and the links to other adjacent primitives. Each Task is physically represented by using a frame. Here is an example:

```
(def-task 'r-focus-strategy
      :level-state      'YES
      :precondition     '(NULL *FR-list*)
      :condition-part   '(focus-strategy)
      :has-subtask      '(r-gl1-L r-gl1-M r-gl1-H)
      :priority         1
      :task-level       1
      :type             'strategy)
```

The level-state allows the turning off of the Task if necessary; the condition-part is the condition to be satisfied to "fire" the Task. The action-part represents a call to a Lisp function operating on the problem data. Priority defines the importance of the Task. Other two slots, subtask-of and has-subtasks, describe the position of the Task in the hierarchical tree. This structure allows the "scheduler" to work in a particular way: when a Task can be "fired", after its action-part (if any), has been executed, the monitor continues its test on the "sons" of the Task, using the "depth-first" technique. This hierarchical tree makes the system highly flexible: any Task can be inserted by simply adding its name to the slot "has-subtask" belonging to the father, and any Task can be turned off, by simply setting to NO the slot "level-state"of the Task itself. This fact allows a very easy integration of this system into a larger framework, for the final purpose of a scene interpretation.

2. 2. 3. The Merging Process

According to the hierarchical organization of the Tasks, the whole segmentation system can be devided into five principal Tasks:
- CREATE DATA-BASE
- FOCUS OF ATTENTION
- CANDIDATE
- PREPARE
- MERGE

In addition, an INTERFACE Task has been created to better interact with the system.
These Tasks and their sub-Tasks make up the tree structure.
The merging process starts with the activation of the CREATE DATA-BASE Task which, after performing the numeric level, builds region and edge frames; on this Data-Base, the other Tasks perform the processing steps to produce the final result.
The FOCUS OF ATTENTION Task extracts from all the regions a set of them on which to work during the merging process. The current strategy can be, for example, to select regions with very high gray level on channel one (T1) and regions adjacent to long edges. This means that the scheduler, after the execution of the action part of the FOCUS Task (strategy), will fire the GL1-L and EDGE-H Tasks. The first will look for regions with gray-level values belonging tothe lower sub-range of the possible values of this feature in the T1 image; The EDGE-H Task acts in the same way, working on the length of the edges. To simplify the following step, we assume that two "focus-regions" cannot be adjacent.
The CANDIDATE Task analyzes one "focus-region" at a time, using three subtasks:
- the EDGE-DISCRIMINATOR subtask eliminates adjacent regions, separated by edges from the focus-region;
- the FEATURE-COMPUTING subtask produces, for each "focus-region"/adjacent region pair a "dissimilarity-vector"; the elements of this vector are computed using a set of fuzzy functions, as shown in Fig. 2;
- the CANDIDATION subtask has to threshold this vector and the reliability of the region to be merged to decide if merging is possible. For the final regions to be merged, conflicts are solved as described before.

The PREPARE Task works on each "focus-region" that has passed the CANDIDATE test. Depending on the desired maximum dimension of the merged groups, it prepares suitable data for the merging step.
Finally, the MERGE Task performs the physicalmergingof the groups, and updates the features of the new regions. This task recomputes all the features; to update the optical features these formulas are used:

$$var(M_a + M_b) = S(M_a + M_b) - m^2(M_a + M_b)$$

$$S(M_a + M_b) = \frac{1}{M_a + M_b}\left(M_a\, S(M_a) + M_b\, S(M_b)\right)$$

where m(Ma) is the mean value of the current feature computed for the region A, Ma is the area (number of pixels) of the region A, S(Ma) is the summation of the square values of the current feature for the region A, and var(Ma) is the variance of the current feature for the region A.

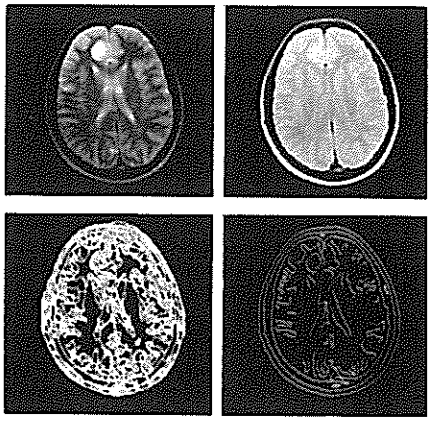


Fig. 4

3. FINAL RESULTS

We started with the two images shown in the upper portion of Fig.4; they refer to the same slice. The lower-left portion of the same figure shows the result of the pre-segmentation process, while in the lower-right portion, the result of the edge-extraction process is presented.
The pre-segmented image is made up of 854 very small regions, while in the image on the right we have only very sharp edges. Our segmentation is mainly based on regions: the edges can aid in segmenting very thin structures, like skin and bone.

Fig. 5

On this initial data-base, the system performs iterative merging steps. The figure shows four successive steps (from top to bottom and from left to right respectively we have: 599, 352, 142 and 65 regions). The last image (lower, right) represents the final segmentation result. The previous image shows that very thin details like those of "fissura longitudinalis", skin and bone, are well preserved.

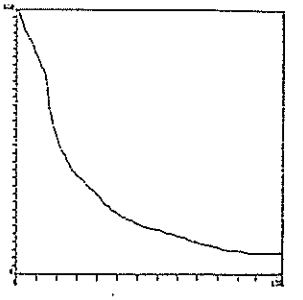


Fig. 6

The graph (Fig. 6) gives the number of regions for each iteration step. One can notice that after 70% of steps, the number of merged regions is almost constant.
Finally we give some processing information: the system has been implemented on a SUN4 computer, and the CPU time has been about 2100 sec. for the complete segmentation process.

4. CONCLUSIONS

Results have been obtained by integrating edge and region primitives, and by using specific knowledge to decide, on the region merging. The system flexibility allows one, to insert other heuristics based on additional knowledge. For example, a function that computes similarity values for regions that, when merged, fit pre-defined geometric shapes (circle, rectangle, etc.) could be useful in segmenting geometrical structures (e.g. in industrial images).

ACKNOWLEDGEMENT

Autors wish to thank Philips for its support for providing images used in this paper.

REFERENCES

[1] Nagao, M. and Matsuyama, T., A Structural Analysis of Complex Aerial Photographs, Plenum Press, N.Y., 1980.
[2] Vernazza, G., Serpico, S.B., and Dellepiane, S., Edge-preserving smoothing of digital images by means of thin masks, Proc. of the IASTED International Symposium on Measurement, Signal Processing and Control. MECO '86, Taormina (Italy), Sept. 1986.
[3] Marr, D. and Hildreth, E., Theory of edge detection, Proc.Roy.Soc.London, Vol. B207, 1980, pp.128-217.
[4] Hayes-Roth, B., A blackboard architecture for control, Journal of Artificial Intelligence, 26, 1985.
[5] Nazif, M. and Levine, D., Low level image segmentation: an expert system, I.E.E.E. Trans. Vol. P.A.M.I. 6, No. 5, pp. 555-557, Sept. 1985.
[6] Dellepiane, S., Regazzoni, C., Serpico, S., and Vernazza, G., An application-independent knowledge-base framework for complex image recognition, Proc. I.A.P.R. 5th International Conference on Image Analysis and Processing, Positano, Sept. 89 (in press).

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

GEOPHYSICAL SIGNAL INTERPRETATION: A KNOWLEDGE-BASED SYSTEM

Vito ROBERTO, Adriano PERON
Dipartimento di Matematica e Informatica, Università di Udine, via Zanon 6, I-33100 Udine, Italy

Claudio CHIARUTTINI
Istituto di Geodesia e Geofisica, Università di Trieste, Italy

Giuliano BRANCOLINI
Osservatorio Geofisico Sperimentale, Trieste, Italy

The problem of seismic stratigraphic interpretation is addressed. The analysis of the domain expert knowledge and interpretive behavior shows that flexible programming tools as those provided by Artificial Intelligence techniques are necessary, and that they have to be combined with image processing tools. The prototype of a Knowledge-Based System, named *Horizons*, was designed according to the Blackboard problem-solving paradigm. We present its architecture and the result of tests with real data.

1. Introduction

The aim of exploration geophysics is investigating the earth subsurface to detect underground structures and resources. In the last years, considerable progress has been obtained in data acquisition and processing; however, in the domain of data and signal *interpretation* most of the work is still accomplished by trained experts, which results in a bottle-neck for the whole exploration work. This happens mainly because knowledge has to be collected from different sources (e.g. general geological models, local geological models, seismic reflection images, well-logs and other geophysical data), and arranged in a unique, consistent framework. There follows the need of automation in such domain, with the aim of the reduction of time and cost of the interpretation; a more effective use of the informative content of the data; a standardization of the procedures of analysis. Artificial Intelligence techniques, combined with image processing and pattern recognition tools, provide a new and promising framework for such purposes.
Seismic stratigraphic analysis [1] aims at understanding the processes of sediment deposition in a basin. This paper reports a brief summary on a research project named *Horizons*, for the study, design and application of a Knowledge-Based System (KBS) to support the stratigraphic interpretation of seismic signals and data.

2. The Problem

Two main techniques are employed for signal and data acquisition: *well logging*, where a set of physical and lithological parameters is measured from the cores or in the drilled hole, providing precise information, however restricted to a small area; *reflection seismics*, where elastic waves are generated by explosions, and the waves reflected back by the subsurface layers are recorded and put in form of images, named *seismic sections*: they provide information over vaste areas. The aim of the interpretation is integrating well-log and seismic-section data, to identify the main features of the area and their time evolution. This is done using the basic knowledge models: a *geophysical model*, i.e. a collection of objects and processing methods related to the data currently being examined, and a *geological model*, embedding all information about the basin under investigation. The latter model involves three kinds of objects: *structural elements*, mainly geometric, related to tectonic deformations; *stratigraphic elements*, other geometric features due to the sedimentary evolution of the basin; *lithologic elements*, i.e. physical/chemical patterns of the rocks. Our project concerns the automation of the stratigraphic interpretation. We adopt an Artificial Intelligence approach [2], since it provides several tools to effectively model and code the expert knowledge.

3. Modelling the Expert Knowledge

Objects, concepts and relations in the expert reasoning are individuated by means of *semantic networks*, through the relations *part_of*, *qualified_by* and *is_a*. In this way several stratigraphic concepts (for example, *seismic sequence, internal sequence, prograding clinoform*) are suitably represented: it should be observed that this is the simplest formal description of such objects, since no geometrical parametrization can be attempted for them. In addition, hierarchies among objects are individuated (e.g. the *taxonomic hierarchy*): this allows a more efficient coding of knowledge, by

exploiting all the logical/algebraic properties of the relational links. Moreover, the basic building blocks of the domain knowledge are outlined (the *semantic primitives*), as the *less structured* objects in the hierarchy: this is the case of the *reflective events* in our context.

The *resolution strategies* of the analyst—i.e. the dynamics of the expert knowledge—have been studied and distinctive patterns have been evidentiated. A *selective focus-of-attention* attitude is present: large attention areas in the image data are examined first, and more specific subareas, at a finer level of detail, are subsequently scanned. Second, we observe the lack of a definite direction in the analysis (e.g. top-down, bottom-up), where, instead, decisions and reasoning steps proceed *opportunistically*, according to the current data and hypothesis contexts.

In view of the above-quoted observations, we propose the *Blackboard Model* [3,4] as a suitable problem-solving scheme. In particular, the interpretation (called *hypothesis* in the following) evolves via successive activations of tasks, with no a priori defined order; such tasks are basically *generation* of elements of the hypothesis, *test* of pieces of the hypothesis and *debugging* of all items (new data or assertions) recognised to be inconsistent with the current system database [5].

The architectural scheme we propose is outlined in Figure 1, four basic units are present:

- The *Permanent Database*, encoding the knowledge permanently present in the system, partitioned into *static* parts: *objects* and *facts*, and *dynamic* parts: the *Knowledge Sources* (KS), modules capable to perform specific interpretation tasks.
- The *Blackboard*, a volatile database that records the current status of the analysis, including the *current best hypothesis* (CBH), the *control database* and the *agenda*: both used to direct the flow of computation, and the *reasoning chain*: the explanation in natural language of the inferences made by the system.
- The *Control Unit*, needed to *schedule* the KS execution according to a *strategy* that may be selected at run time.
- The *User Interface* and the *Justifier*, that allow the user to inspect the data and examine the CBH and the reasoning chain.

4. The Prototype: Implementation, Results

A *demonstration prototype* of *Horizons* has been constructed, with the aim of obtaining a concrete insight on the complexity of the interpretive problem, and to plan further steps in the project. The system has been implemented on a SUN 3/110 workstation.

Low-level processing modules, coded in C-language for

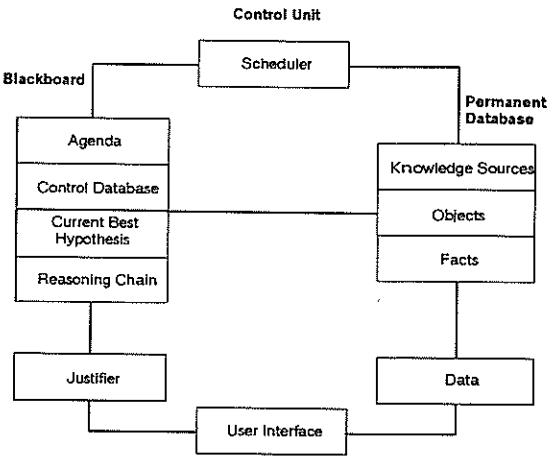


Figure 1. The architectural scheme of *Horizons*.

a total of 3000 lines of code, perform several operations on seismic sections such as *enhancement*, to increase the signal-to-noise in noisy patterns; *texture analysis*, through run-length detection, to single-out candidate linear patterns and define the appropriate data structures; *region segmentation*, to individuate the attention areas as those with coherent dominant geometrical patterns [6].

At the symbolic processing level, all modules (i.e. objects, knowledge sources and rules) are coded in interpreted Prolog for a total of 2000 lines of code.

The current capabilities of the prototype include:

- Construction of the knowledge base, by a sequence of the numerical and statistical analysis steps reported above.
- Initialization of the inferential processes, via KSs instantiating objects in the working memory. Suitable communications with graphic tools is also activated to support the dialogue with the user.
- Communication between numerical and logical processing steps (signal-to-symbol transformation), by suitably interfacing the C- and Prolog-coded procedures.
- Generation of a rough interpretation (*hypothesis*). Figure 2 reports a few results from *Horizons*, including some of the low-level processing steps and a rough hypothesis about the detected stratigraphic profile.
- Test of its consistency against data, analysed at a finer level of detail. Figure 3 reports another example of *Horizons* at work; in this case, part of the hypothesis was found to be inconsistent with data.
- Justification of the reasoning chain in natural language; an appropriate data structure (the *reasoning chain*) is defined accordingly.

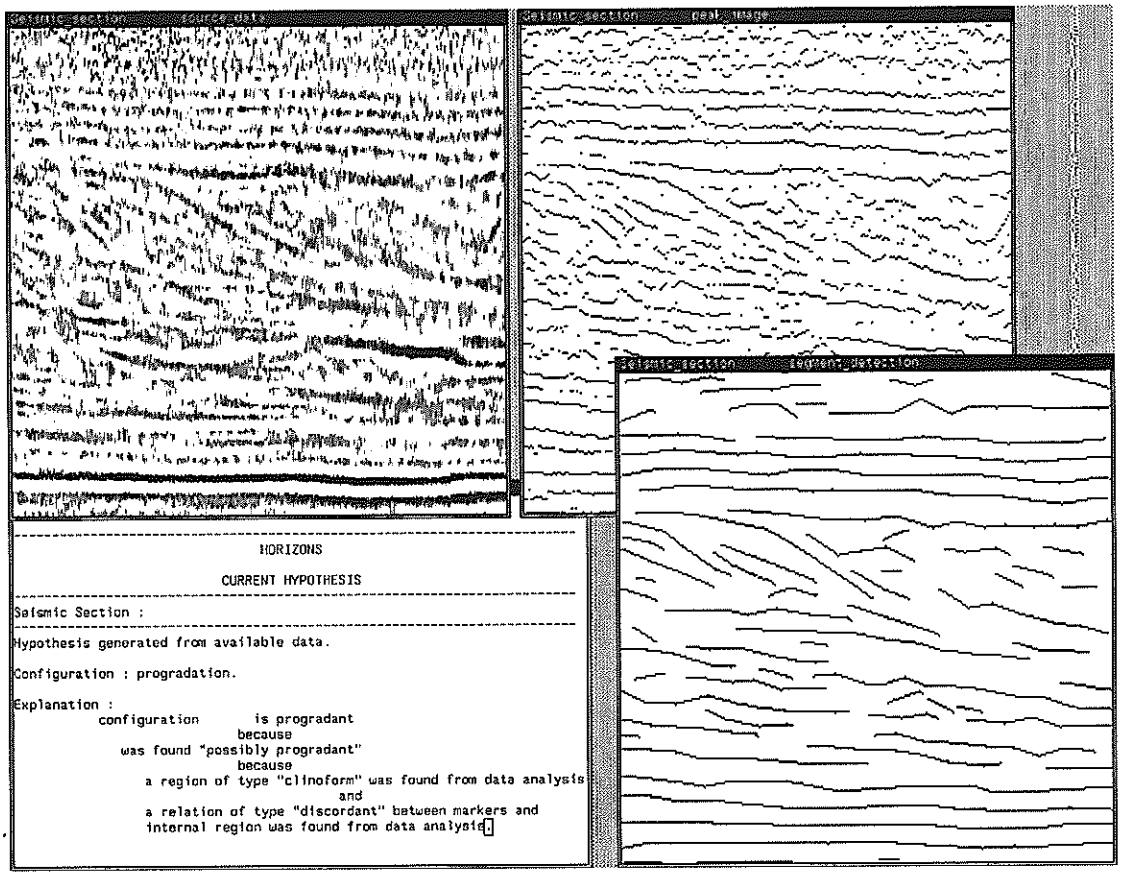


Figure 2. Display of a few outputs from the prototype of *Horizons*: input image data (seismic section), upper-left corner; image of the positive peaks, after enhancement (upper-right); image of the segments (reflective events) extracted after texture analysis and line-following (lower-right); hypothesis generated and justified by the reasoning modules (lower-left).

Preliminary results concern the stratigraphic analysis carried out on real data (seismic sections) in a simple context and with limited goals; the relevant stratigraphic elements are correctly identified in all cases. The performances of the prototype are encouraging, for what concerns both the effectiveness of the problem-solving scheme and the efficiency in the numeric and symbolic processing tasks. An interpretation such as those reported in Figures 2 and 3 is generated in roughly 20–30 CPU seconds, most of the time being spent in the numerical/statistical steps.

Further developments of the system capabilities are under way, especially for what concerns the extension of interpretation to three-dimensional data sets, and the definition of suitable control strategies.

References

[1] Berg, O. R. and Woolverton, D. G. (eds.), *Seismic Stratigraphy II: an Integrated Approach* (AAPG Memoir 39, Tulsa, Oklahoma).

[2] Frost, R. A., *Introduction to Knowledge Base Systems* (Collins, London, 1986).

[3] Engelmore, R. S. and Morgan A. J. (eds.), *Blackboard Systems* (Addison-Wesley, Wokingham, England, 1988).

[4] Roberto, V., Gargiulo, L., Peron, A. and Chiaruttini, C., A Knowledge-Based System for Geophysical Interpretation, in *Proc. IEEE Intern. Conf. on Acoustics, Speech and Signal Processing* (Albuquerque, New Mexico, April 1990).

[5] Simmons, R. and Davis, R., Generate, Test

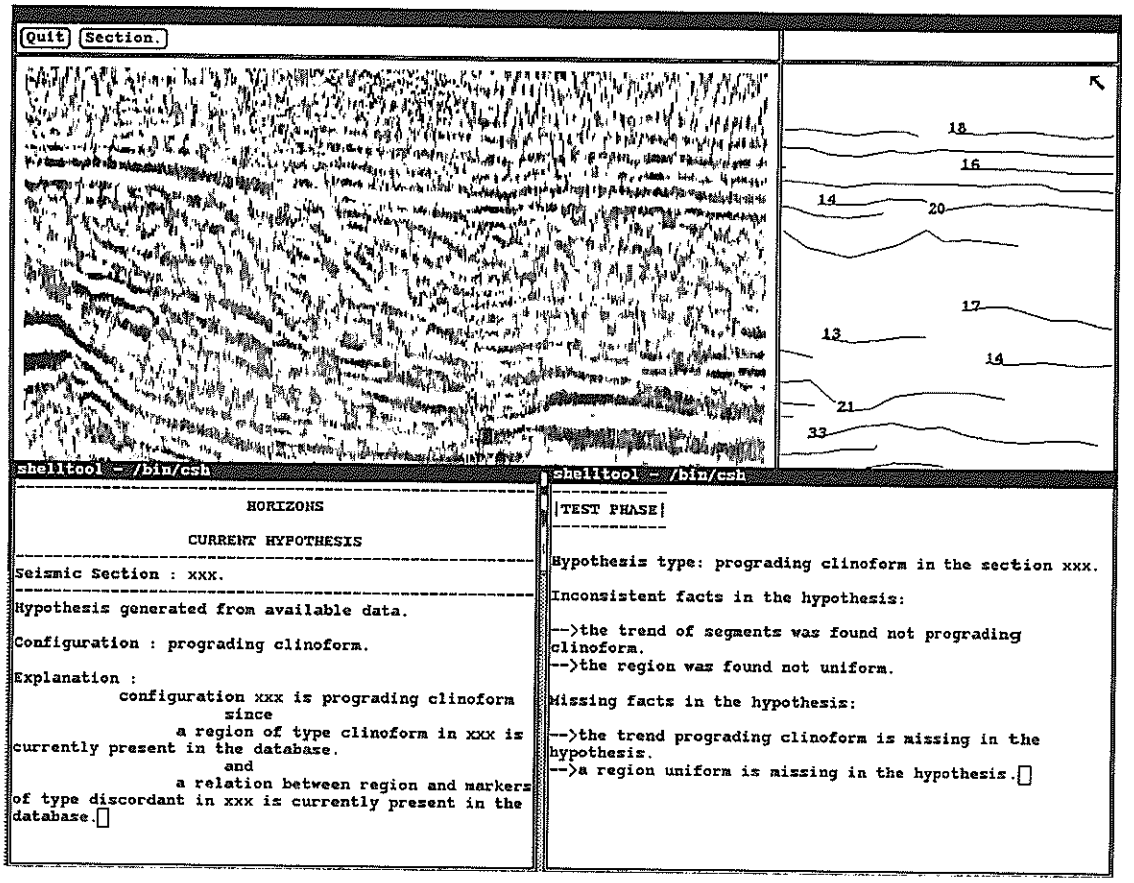


Figure 3. Another display from the prototype: input seismic section (upper-left corner); the segments (reflective events) extracted and labelled (upper-right); the hypothesis generated (lower-left) and tested (lower-right).

and Debug: Combining Associational Rules and Causal Models, in *Proc. Intern. Joint Conf. on Artificial Intelligence* (Milan, Italy, August 1987) pp. 1071–1078.

[6] Roberto V., Peron A. and Fumis P. L., *Pattern Recognition Letters* 10 (1989) 111–122.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Configuration of Systems for Recognition of Raised Characters using Knowledge-Based Techniques

M. Dehesa, K. Hörger, E.v. Hinüber, C.-E. Liedtke

Institut für Theoretische Nachrichtentechnik und Informationsverarbeitung, Universität Hannover
Appelstr. 9 A, 3000 Hannover 1, Fed. Rep. Germany

The problem of automatically recognizing embossed or molded characters which are raised from the surface on industrial materials (like rubber and glass) is discussed. Some differences between this task and that of recognizing printed characters are emphasized. A knowledge-based approach is proposed to configure automatically a suitable image processing and classification procedure to accomplish a predefined task. Experimental results with raised characters on rubber tires and glass bottles are presented.

1. Introduction

In industrial automation an increased request for the recognition of embossed or molded characters in relief (denoted hereafter as *raised characters*) can be observed. There are important differences between this task and that of recognizing printed characters. An example of this can be found in the classification of rubber tires based on their alphanumeric inscriptions. The characters themselves have approximately the same grey-value as the background, and, because of the tires geometry, illumination of the surface is nonhomogeneous. Therefore, it is not possible to get proper blobs for the classification by using a simple threshold operation as in the preprocessing of images containing printed characters [1], [2]. Raised characters are not only found on opaque materials but also on transparent ones as, for example, on glass bottles. If the bottle is defective, the batch number (on the bottle bottom) must be read to determine its origin. Using adequate illumination the characters on the bottle may be relatively easy to distinguish from the background. Nevertheless, reflections due to the transparency of the material may be unavoidable. Further problems arise from uncertainty in the position, size and orientation of the characters and from the fact that the font is not standardized.

Due to the diversity of objects and materials, the recognition system should be able to adapt itself at least to some different situations. This means, given a recognition task, the system must be configured as a sequence of general purpose processing modules and the parameters of these modules need to be adapted in order to solve the task. In this paper we report the advance in the development of a knowledge-based system for this purpose. We will consider only the above mentioned cases of characters on rubber tires and glass bottles.

2. Configuration method

The recognition system used is shown in Fig. 1. For each of the *processing steps* (with exception of the *digitization*) appropriate processing modules and their parameters must be chosen.

The configuration of a processing procedure has some analogy with the developement of a skill by a human being. In both cases the task to be performed must be analysed and an appropriate sequence of steps must be chosen. For describing this process of skill acquisition three phases have been proposed [3]: 1) *declarative*, 2) *associative* and 3) *procedural*. The first and third phases take their name from the type of knowledge which is mainly used throughout the duration of the phase. The name of the second one is due to the associations done between the input image and a sequence of processing modules, to determine appropriate values for the parameters of these modules.

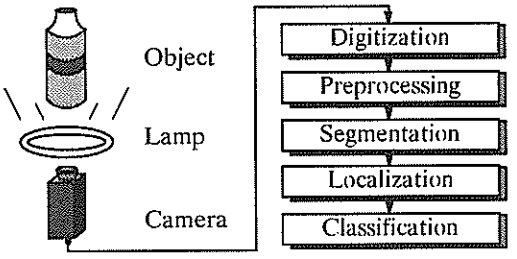


Fig. 1. The recognition system.

1. *Declarative Phase.* In this phase the task to be done is analyzed, and one or several modules for each *processing step* are proposed by the system. The first step is to use a *context model* to ask the user about properties of the characters to recognize. This model is shown as a semantic net in Fig. 2. The context model contains implicitly the conditions under which the system is able to recognize characters. The net can also be viewed as an AND/OR graph [4]. AND-nodes are those where the incoming rays are connected by an arc. All other nodes are OR-nodes. AND-nodes represent conditions all of which must be asked. In the case of OR-nodes only one of the conditions must hold. For processing purposes the context model is represented with schemata.

Beginning at the node "Character" the net is inspected using a first-depth search algorithm which is independent of the number of nodes and rays. Each time a ray is inspected, the system queries the user to tell if the condition represented by the ray and the two corresponding nodes does hold or not, or if the answer is unknown. In the case of a ray with the name "has value" the user is requested to enter a value for the corresponding variable. An example of such an inquiry is:

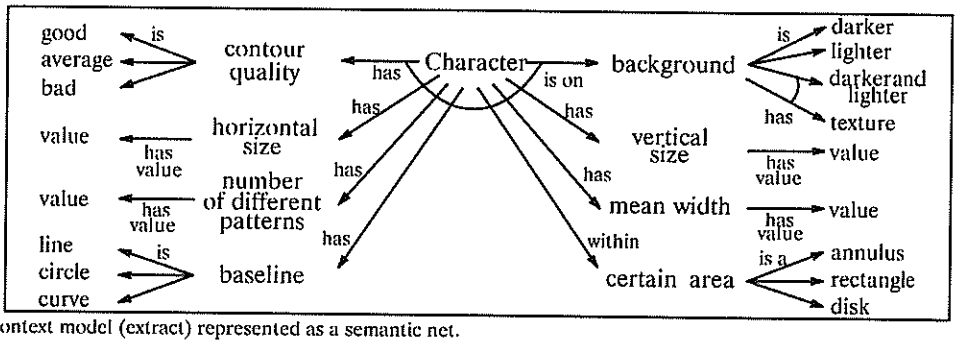


Fig. 2. Context model (extract) represented as a semantic net.

Character has contour quality ?	[y/n/u]	yes
Contour quality is good ?	[y/n/u]	no
Contour quality is average ?	[y/n/u]	yes
Character has horizontal size ?	[y/n/u]	yes
Horizontal size has value:		20

etc. As a result of net inspection a set of *directives* for the configuration is achieved. Each directive consists simply of a question which was asked to the user and of the corresponding answer (Fig. 3). The directives are represented in a convenient format for later processing. A set of rules is used to test for inconsistencies in the directives and to assign default-values to important parameters (e.g. the characters mean width) which the user didn't specify. Using the complemented directives list a second set of rules decides about which module or modules could be used for each processing step. The results are passed to the second phase. The speed during this first phase is slow due to intensive use of declarative knowledge.

2. *Associative Phase.* For each processing step, one or several modules were proposed in the declarative phase. In this second phase, heuristic rules and repeated application of the different modules are used to select only one module for the step. Initially, default values for the parameters of the modules are set. These values are adapted to data material according to some specified quality criterion. The best module in the sense of this criterion is selected for the step. Note that the quality criterion may not be the same for the different steps. The evaluation of the results is based on heuristic criteria which are represented as rules.

CHARACTER-IS-ON-A-BACKGROUND	T
BACKGROUND-IS-DARKER	T
BACKGROUND-HAS-TEXTURE	T
CHARACTER-HAS-HORIZONTAL-SIZE	T
HORIZONTAL-SIZE-VALUE	20
CHARACTER-HAS-MEAN-WIDTH	T
MEAN-WIDTH-VALUE	5
CHARACTER-WITHIN-A-ANNULUS	T
⋮	

Fig. 3. Example of list of directives.

3. *Procedural Phase.* This phase is reached when the configured procedure has been sufficiently tested. There is no need to evaluate repeatedly the results of this procedure. Nevertheless, because viewing conditions may change with time, the results delivered by the running procedure may be tested to see if they are still in accordance to the goal stated initially by the user. If necessary, some parts or the whole configuration process may be repeated. Processing is very fast because the knowledge involved is mainly of procedural type.

3. Procedural modules

These are algorithms for image processing or other tasks like object localization or classification. For the sake of brevity, let us suppose that only the modules shown in Fig. 4 are available. The modules are identified by a short name and are grouped according to their function. *Filtering* and *pre-search* are part of the *preprocessing*. Note that *preprocessing*, *segmentation*, *localization* and *classification* correspond to the steps shown in Fig. 1. The function and parameters of the modules shown in Fig. 4 are explained in [5]. Here only a brief explanation of some of them will be given. References to figures corresponding to results will be made while explaining the modules.

Filtering	
local mean	il_MEAN
local ranking	il_RANK
local mean weighted	il_MEANW
Pre-search	
on a line	cs_LINE
on a circle	cs_CIRC
Segmentation	
global threshold	i_THRE
local segmentation	il_SEG
Localization	
using size and area	cp_SA
by building words	cp_WB
Classification	
statistical	cc_NUM
structural	cc_STR

Fig. 4. List of modules for the configuration.

3.1 Preprocessing and segmentation

In Figs. 5(a) and 6(a) original images corresponding to a rubber tire and the bottom of a glass bottle, respectively, are shown. After *digitization*, *preprocessing* takes place to *filter* the image and to perform a *pre-search* of the approximate location of the characters. This search is made by using the gradient magnitude of the original image (cs_CIRC). The gradient is shown for the tire case in Fig. 5(b); in Fig. 5(c) the areas of interest found are shown in black. Afterwards, *segmentation* is done to attain areas of only three grey-values (il_SEG). The method followed for the segmentation is described in [6]. The result of this step is shown in Fig. 6(b) for the bottle case.

From the segmented images it is easy to find the contours of the objects of interest (in the case of Fig. 6(b) the white ones). In the method used for segmentation, object bound-

aries are found in a similar way as in zero-crossing algorithms [7], [8]. As is well known, such methods produce spurious or phantom contours [9]. Therefore not all the contour points are relevant. This may be seen, for example, by comparing Figs. 6(a) and 6(b). It is evident that not all contour points of the white objects in the segmented image correspond to relevant grey-value transitions in the original one. Thus, a validation procedure is necessary to discard the non-interesting contour points (see [5]). The validated contours are shown in Fig. 6(c) for the bottle case.

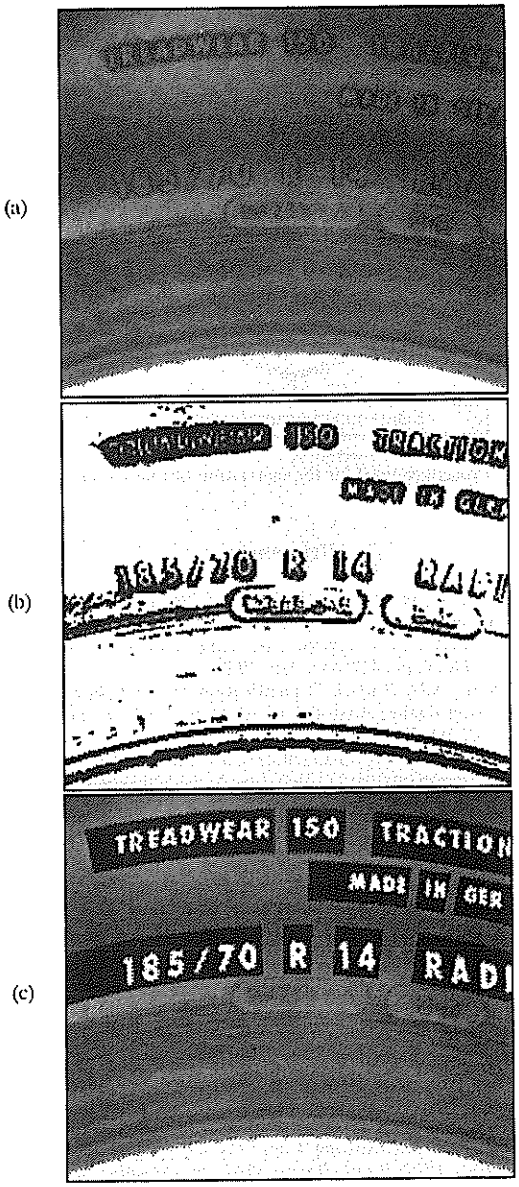


Fig. 5. Rubber tire. (a) Original, 512x512x8 bits. (b) Binarized image of the gradient magnitude. This image is used for the pre-search. (c) Original image with the areas of interest in black. The characters found in these areas were rotated to vertical orientation and displayed again at their original position.

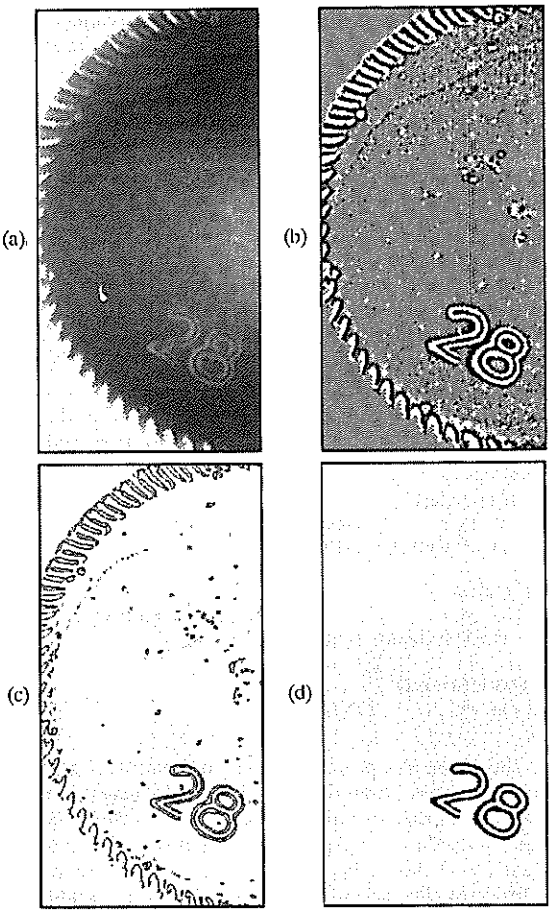


Fig. 6. Glass bottle. (a) Original, 512x256x8 bits. (b) Segmented image. (c) Relevant contours. (d) Extracted characters.

3.2 Localization and classification

Using the validated contours a further selection of the objects in the segmented image is performed. This is done by simply discarding the objects which do not have at least 50% of validated contour points on their contours. From the remaining objects only those which have appropriate size and area (cp_SA) are considered for the recognition step.

The relevant blobs found are finally classified. If we know the position of the center of the circle on which the characters are, then the direction of the characters may be estimated. The position of the wheel center is automatically calculated using the wheel border (cs_CIRC) which is easy to find in the gradient image (Fig. 5(b)). The position of the glass bottle center can be mechanically positioned and therefore it is used as a-priori information to rotate the characters found to vertical rotation. In Fig. 5(c) the result of this step for the wheel case can be seen. The blobs thus obtained are normed in size and finally classified by statistical means (cc_NUM) [10].

4. Results

As previously described, the result of the declarative phase is a list which consists of processing steps, and one or several possible modules for each step. Such a list is shown for the case of the tire in Fig. 7. During the first phase, a

unique module for the *filtering* could not be determined. Two modules are proposed for this step. This is also the case with the *segmentation*. For all other steps only one module was proposed.

Filtering	il_MEAN il_RANK
Pre-search	cs_CIRC
Segmentation	i_THRE il_SEG
Localization	cp_GA
Classification	cc_NUM

Fig. 7. List of steps and possible modules for each step.

If there are several modules for a step, a particular set of rules and a quality criterion are used during the associative phase to select only one of the modules. To explain how this is done let us take for example the *segmentation*. In the system used, processing steps and modules are represented as schemata:

```
{{ FILTERING
     IS_A:          PROCESSING_STEP
     HAS_RULES:     FILTERING_RULES
}}
{{ i_THRE
     IS_A:                  PROCESSING_MODULE
     HAS_PARAMETER:         THRESHOLD
}}
{{ THRESHOLD
     IS_A:          INTEGER
}}
```

etc. The name of a processing step and that of its possible modules are read from the list in Fig. 7. The general procedure used for the evaluation of each module is shown in Fig. 8. This procedure is independent of the processing step, but for each step the quality criterion used for evaluation may be different. The schema corresponding to the module to evaluate is instantiated. The instantiated schema contains as meta-information initial values for the parameters of the module and eventually lower and upper bounds for these values. Using the initial values, the module is applied to a test image. The result is evaluated using the rule set with the name FILTERING_RULES (see schema FILTERING). If necessary, the parameter values are changed and the modul is applied again. Especially in the case of i_THRE and il_SEG, the quality criterion used is the segmentation error described in [6]. After a good result (in the sense of the quality criterion) was attained, the procedure in Fig. 8 is executed for the next possible module. Finally, an evaluation of all module results is performed. The module which delivers the minimal segmentation error is selected.

The system proposed was simulated on a digital computer. The context model was represented using schemata. This model and the control mechanism were implemeted in LISP using the software tool Knowledge Craft®. The rule sets were implemented in OPS5 [11] and the processing modules in the language C.

5. Summary

A knowledge-based approach for the configuration of recognition procedures for characters on industrial materials was described. Different types of knowledge representation like schemata, rules and algorithms were used to gain flexibility. Procedures for the recognition of characters on rubber tires and glass bottles where configured and used. The system capabilities are still very restricted, but the approach used enables easy extension of the context model and the rule sets to add new cases.

Acknowledgements

The authors are indebted to the following companies for their kind supply of work-pieces for the experiments: Nienburger Glas (D-3070 Nienburg) and Continental Gummi-Werke AG (D-3000 Hannover).
The authors would also like to thank A. Blömer for his aid in the implementation of the control mechanism and S. Martucci for his assistance in the writing of this manuscript.

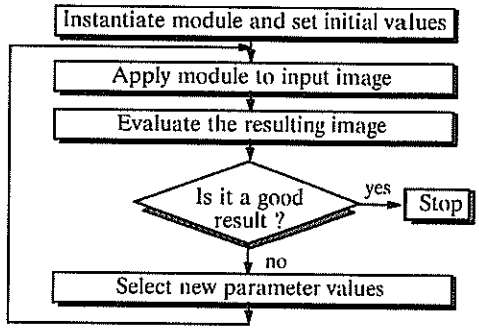


Fig. 8. Procedure used for the evaluation of one module.

References

1. F.-H. Cheng, W.-H. Hsu, M.-Y. Chen, "Recognition of Handwritten Chinese Characters by Modified Hough Transform Techniques", *IEEE Trans. Pattern Anal. and Machine Intell., vol. PAMI-11*, No. 4, pp. 429-439, Apr. 1989.
2. D. Wang, S.N. Srihari, "Classification of Newspaper Image Blocks Using Texture Analysis", *Comp. Vision Graphics Image Process*, vol. 47, pp. 327-352, 1989.
3. P.L. Ackerman, "Individual Differences and Skill Aquisition", in P.L. Ackerman et. al (Eds.) *Learning and Individual Differences*. New York: W.H. Freeman and Co., 1989.
4. N.J. Nilsson, *Principles of Artificial Intelligence*. Berlin, Heidelberg, New York, Tokyo: Springer-Verlag, 1982.
5. M. Dehesa, *Wissensbasierte Schriftzeichenerkennung auf industriellen Materialien*. Dissertation, Fakultät für Maschinenwesen, Universität Hannover, 1990. [German]
6. M. Dehesa, C.E.-Liedtke, "Image Segmentation for the Recognition of Characters on Different Materials", *11. DAGM-Symposium Proceedings*, Hamburg, Fed. Rep. Germany, 2-4 Oct. 1989.
7. R.M. Haralick, "Digital Step Edges from Zero Crossing of Second Directional Derivatives", *IEEE Trans. Pattern Anal. and Machine Intell., vol. PAMI-6*, No. 1, Jan. 1984.
8. J.L.C. Sanz, T.T. Huang, "Image Representation by Sign Information", *IEEE Trans. Pattern Anal. and Machine Intell., vol. PAMI-11*, No. 7, pp. 729-738, Jul. 1989.
9. J.J. Clark, "Authenticating Edges Produced by Zero-Crossing Algorithms", *IEEE Trans. Pattern Anal. and Machine Intell., vol. PAMI-11*, No. 1, pp. 43-57, Jan. 1989.
10. E.v. Hinüber, *Erkennung erhabener Schriftzeichen auf Kraftfahrzeug-Reifen*. Diplomarbeit, Inst. f. Theor. Nachrichtentechnik u. Informationsver., Universität Hannover, 1988. [German]
11. L. Brownston et.al, *Programming Expert Systems in OPS5: An Introduction to Rule-Based Programming*. Reading, Massachusetts: Addison-Wesley, 1986.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

Global Positioning System Integrated Navigation and Attitude Determination System (GINAS) [1]

R.Lucas graduated in Telecommunications Engineering, with ESA since 1986.

M.A.Martínez graduated in Aeronautics Engineering, with GMV since 1988.

M.Martin-Neira graduated in Telecommunications Engineering, with GMV since 1989.

Abstract

An integrated system for Navigation and Attitude Determination of a mobile platform using the Global Positioning System is presented. The method is based in the processing of triple-difference phase measurements taken with a very-short baseline. The system has been evaluated experimentally using two commercial off-the-shelf GPS C/A code receivers and results are given. An accuracy of 0.1° for a baseline of 2 meters is demonstrated. In addition an advanced system based on the same concept and that reduces drastically the required initialization time is described. Finally areas of further research in the Signal Processing field for GINAS applications are proposed.

1 Introduction

The Global Positioning System (GPS) is a USA DoD's satellite based radionavigation system which allows to users equipped with a GPS receiver to determine autonomously and in real-time their current position, velocity and time (see ref.1).

The GPS will be the main navigation system of the 90's and beyond superseding Loran-C, Transit and Omega. An integrated system that provides Attitude Information in addition to Navigation out of the GPS measurements has a high potential interest for the users. The GINAS will provide to the user all the performances of a multisensor system but with less mass and power consumption. Platform integration of GINAS will be much simpler too.

In the next section, the system for Attitude Determination is presented. In section 3, the experimental results achieved so far are given and in section 4 and advanced GINA sensor is introduced. This advanced sensor has the characteristic of requiring almost negligible initialization time. Finally in section 5, areas for further R&D and improvements on the Signal Processing part of GINAS are proposed. This paper concludes with the prospects for this kind of sensors.

2 GINAS System Description

The method (ref.2) is based in the processing of the phase of arrival difference of a GPS signal as measured at the ends of a short rigid baseline of known distance -d- ($d < 2$ meters). The **single difference** -SD_1- (figure 1) formed in that way carries the basic attitude information and besides has the good characteristics of cancelling some bias error of the GPS system itself (e.g. signal atmospheric propagation perturbations) which are common to both antennas.

In order to eliminate the bias errors of the receivers (clock biases) we form the so-called **double difference** -DD_{12}-. At this point we have a good quasi bias-free phase measurement with a low random noise added. However and since the baseline length is several times that of the received signal wavelength, this measurement contains an ambiguity of an integer number of carrier cycles. We remove this ambiguity by forming the **Triple Difference** -TD_{12}- which is basically a frequency doppler measurement.

The baseline attitude information can be obtained either from the -DD- or from the -TD- measurements. However the TD, being a time difference measurement is only suited for measuring changes in attitude and therefore the absolute attitude of the baseline (respect to a local frame) has to be obtained processing the -DD- measurement. In section 4 we will show in which case is also possible to measure attitude with the -TD-.

[1]This work has been supported by the European Space Agency under ESTEC/Contract No. 7617/88/NL

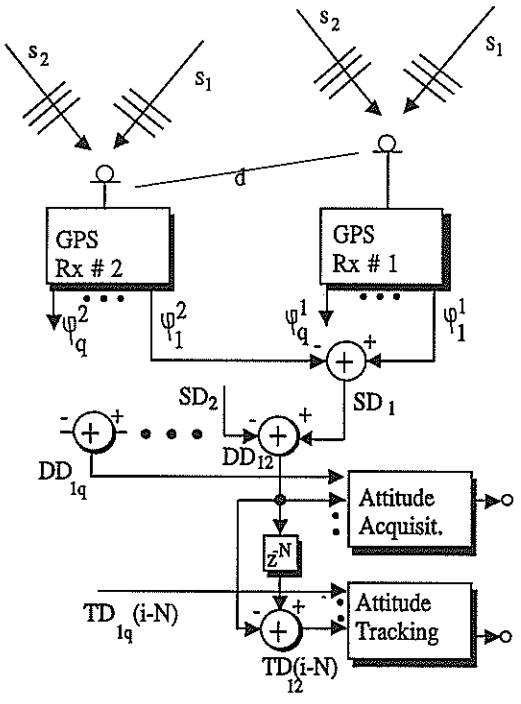


fig.1,'GINAS System View'

3 Results

Two GPS antennas were mounted at the end of a 2 meters long baseline (fig.2, ref.3). Each antenna was connected to an independent commercial GPS receiver. As there was no synchronization signal controlling the measurement time of both receivers, the measurements were contaminated by residual clock bias and clock drift which had to be estimated .

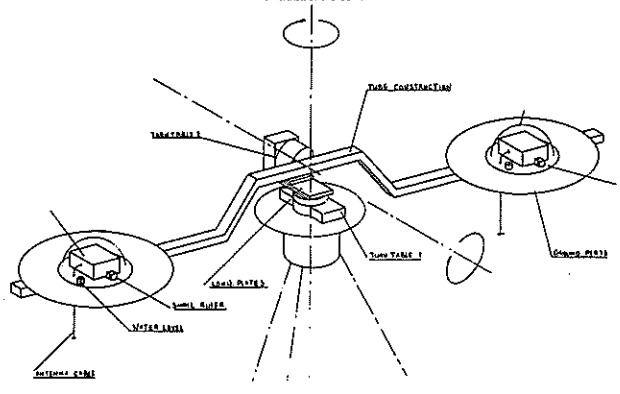


fig.2,'Antenna Baseline'

The baseline was kept horizontally with a fixed azimuth angle during the first 30 minutes of calibration. Then the azimuth was changed by -10°- and after 15 minutes the baseline was placed again at the initial position (figure 3). The GPS computed azimuth and elevation are shown in figures 4 and 5 for two algorithms: Least Squares and Kalman Filtering. Both filters operated on the TD's inputs. The result for elevation seems to be noisier but it just reflects the real-world: the baseline was vibrating on the vertical direction due to the wind. An accuracy of -0.1°- or -2 mrads- was achieved and even better performances can be expected using the advanced sensor described in the next section.

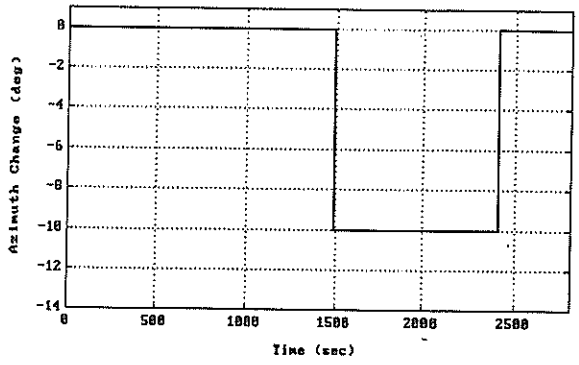


fig.3,'Baseline Azimuth Angle Movement'

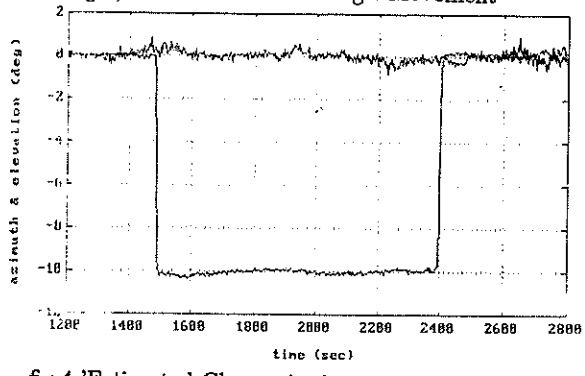


fig.4,'Estimated Change in Azimuth with a LS filter'

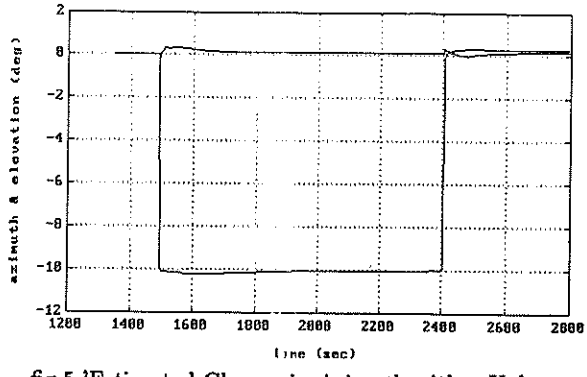


fig.5,'Estimated Change in Azimuth with a Kalman filter'

4 Advanced Attitude Sensor

In the previous case the baseline had to be kept with an arbitrary fixed orientation during 10 to 30 minutes to ensure a good initialization of the algorithm. The initialization problem is that of measuring the initial attitude of the baseline (absolute attitude) respect to a local frame. If we have a good estimate of this attitude we can later on follow and track changes of the baseline orientation (relative attitude) with very good accuracy. With the sensor configuration depicted in figure 6 we manage to eliminate the need for a long initialization time. The idea behind is to measure the absolute attitude of the platform by turning voluntarily the baseline. On this case, the antenna baseline can rotate around a vertical axis passing through its center. If we rotate the baseline exactly -180°- the absolute attitude and the relative attitude are easily related (see fig.7):

$$\Omega = \beta - (-\beta) = 2\beta$$

Ω = change in attitude (relative att.),
β = absolute attitude.

Therefore rotating the baseline on that way back and forth we can measure the absolute attitude at the time of the rotations. After this initialization (in total would take less than 5 seconds), the algorithm for attitude tracking can be employed.

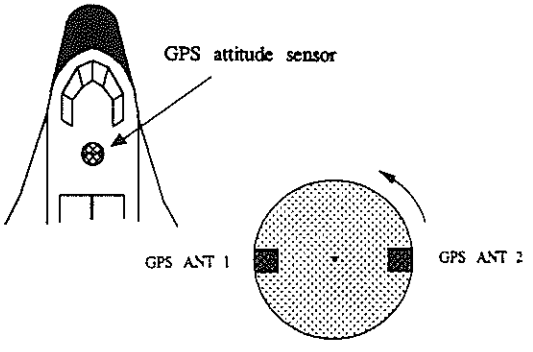


fig.6,'Rotary GPS Attitude Sensor'

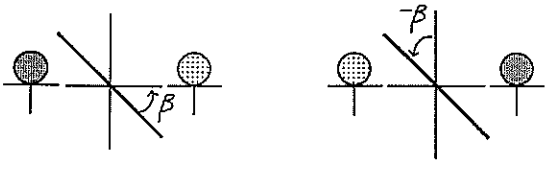


fig.7,'..a 180° rotation'

Finally in table 1, the characteristics of this advanced sensor are given.

SPECIFICATIONS	
Absolute Elevation Bias Error <	2 mrad
Absolute Azimuth Bias Error <	2 mrad
Absolute Elevation Random Error <	2 mrad (1 σ)
Absolute Azimuth Bias Error <	2 mrad (1 σ)
Time to first solution	1 second
Measurement rate	1 Hz
Baseline Length	0.5 m

Table 1. 'Advanced GINAS Specifications'

5 Areas for further Signal Processing R&D on GINAS

Going trough all the work done so far, there are still some points that require additional R&D on the area of Signal Processing.

System Integration: Currently the work is based on a GPS receiver configuration as shown in figure 8.a. however a better architecture would be that of figure 8.b with two big improvements: more integration and elimination of receiver's interchannel bias error since a common SP for both receivers is used.

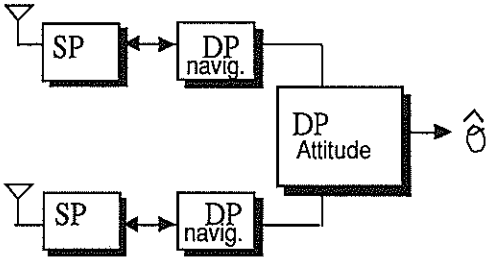


fig.8.a,'Current Architecture'

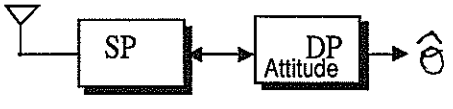


fig.8.b,'Integrated Architecture'

Observable Measurement: A SP which instead of tracking the phase of each satellite would track the differential phase between satellites would provide better results than a conventional receiver.

Electrically Steerable Antennas: The sensor of section 4, requires a mechanical turning of the antennas. A major requirement is that the phase of the signals has to be track continuously during all the rotation. Could this movement be substituted by an electrically switchable antenna scheme?. A careful system analysis is required.

Integer Ambiguity Resolution: The problem of initialization solved with the rotation of the baselines -180°- is to estimate the initial integer number of cycles contained in the ambiguous GPS measurement. Algorithms for fast solution of this ambiguity could be developed avoiding the need of moving the baseline.

GPS Attitude Sensor Antenna: The antenna array involved in the attitude sensor must have special characteristics as perfect symmetry. Studies on array configuration would be of great interest to measure GPS attitude determination accuracy.

6 Conclusions

The prospects for GINAS are very optimistic. Important potential users of this sensor will be aircrafts and Low Earth Orbit Satellites. A new generation of small satellites is emerging which due to the limitation in mass (< 500 Kg) cannot afford to have a sensor for position and a different one for attitude (star trackers, sun sensors, etc..) and therefore GINAS will fulfill optimally the mission requirements.

On the other hand GINAS offer advantages respect to conventional inertial gyro sensors as it is that GINAS does not require on-board realignment or at least it is easier. This will make GINAS also attractive to users already relying on other technologies for getting the attitude information.

References

1. R.J.Milliken et al. 'Principle of operation of NAVSTAR and System Characteristics', ION 'red book', Vol.1. 1980.

2. R.Lucas et al. 'Attitude Determination with GPS', Proceedings of the ION International Conf., Colorado Springs, September 1988.

3. M.Martin-Neira et al. 'Attitude Determination with GPS: Experimental Results'. NAECON'90, Dayton, Ohio.

4. M.Martin-Neira. 'GPS Absolute Attitude Determination Sensor'. GMV Internal Technical note at ESA/ESTEC, Noordwijk February 1990.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A ROBUST METHOD FOR SUBMERSIBLE TRAJECTORY ESTIMATION BY VIDEO SEQUENCE ANALYSIS

J.J. Jacq, F. Aguirre and J.M. Boucher

Groupe Traitement d'Images, ENST de Bretagne, B.P. 832, 29285 BREST CEDEX, FRANCE

This paper describes a system which computes the trajectory of a submersible vehicle from a video sequence. The camera is carried out by the vehicle and the movement is estimated by applying the Generalized Hough Transform (GHT) on successive images where areas, called markers, have been selected. In order to make the estimation more robust, many different markers are chosen on the same image and processed in parallel. A confidence factor, which takes the GHT shape into account, is used for each marker to weight the estimation. A Kalman filter smooths theses estimations and can be used to predict the position of the markers in the following image to decrease the error. The performance of the method was evaluated on two undersea sequences.

1. INTRODUCTION

The problem of position estimation in an undersea vehicle can be solved by using hydrophones, measurement of phase differences, or inertial guidance. However these systems are difficult to use for three essential reasons: voluminous equipment compared to vehicle size, cost and poor accuracy. Video image sequences have been used recently for aircraft trajectory estimation, mobile robots and for other mobile guidance applications. So it can also be used for a submersible trajectory estimation by taking into account the difficult aspects of the undersea video images: lack of contrast, absence of information in a large part of the image due to a quasi uniform sea bed intensity, halo due to the light response of the camera. Thus gradient-based approaches for motion estimation are not satisfactory in real seabed images [2].

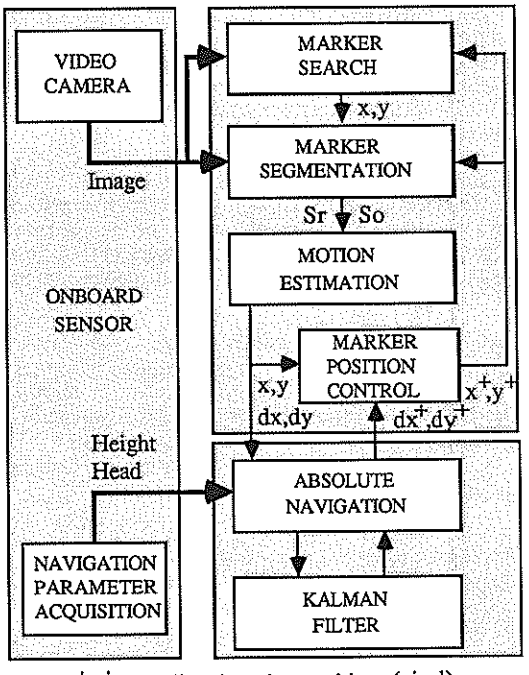
Our new approach to identify the movement in the sequence is to establish correspondences between a set of points, called markers, in two successive images by a generalized Hough transform.

2. SYSTEM DESCRIPTION

The Algorithm operates on a data flow which consists of two sets of five markers and their associated structures. One set - the reference one - belongs to the previous image; the other - the observation one - belongs to actual image. All the kinematic parameters in small characters are given in the shot coordinates system (pixels). The navigation system proposed in this paper is sketched in figure 1. There are eight main blocks.

- The CCD camera is mounted on the underwater vehicle; its optical axis is automatically controlled to stay vertical.

- The navigation parameters block provides informations such as the height above the seabed, the head, the pitch and roll which are used to keep the optical axis verticality.



x^+y^+ Predicted marker position (pixel)
x, y Reference marker position (pixel)
dx ,dy Prediction displacement error (pixel)
dx^+,dy^+ Predicted displacement (pixel)
So Observation structure
Sr Reference structure

Fig. 1 Trajectory estimation system (schematic)

- A marker is defined as an area of NxM pixels; five markers, which must not overlay, are selected on each

image. The marker search procedure uses the amplitude gradient computed on the overall image; its sum is computed on all possible areas of NxM pixels and the five greatest values are kept, which provides a new reference marker location (x,y) (upper left corner of the window)

• Marker segmentation is used to extract the major edge elements (**structure**) from the marker area. This is done by a Sobel operator which gives the edges in the window; major edge elements are then extracted by thresholding edge amplitude. The threshold is computed so that remain a fixed percentage (Q) of the major edge elements belonging to the NxM initial edge elements [3,4].

• Motion estimation is based on the structural matching method. This computational approach allows to establish the correspondence between the points belonging to the reference state structure and the respective observation state structure in the next image [1]. Our approach is able to handle limited occlusion and disocclusion also.

The structural matching method is based on the **Generalized Hough Transform** with gradient direction information [5,6] applied on displacement vectors between edge elements in reference state and observation state of the structure. Conceptually this consists of individual cross correlations between the two states of each substructure; the eight correlation functions (one for each direction) are then added to constitute the final correlation function; the peak eccentricity of the correlation function gives the associated displacement ($\Delta x, \Delta y$).

This correspondence process is successively applied on the five structures. The mean prediction displacement error (dx,dy) is computed by the way of a confidence measurement related to each ($\Delta x, \Delta y$) displacement. Section 3 explains the computation of the confidence measurement.

• The marker position control process assumes the coordination and the tracking of the markers. It detects the exit of a marker from the shot, and asks for another if required. Also it ensures the transition of the parameters for each new marker. This gives the new locations (x^+, y^+) of the active markers in the next image and the exact location of valid markers in actual image.

• Absolute navigation is computed by adding the navigation parameter acquired by the onboard sensors and the period of frame acquisition. This allows the computation of the vehicle's velocity and position (which is obtained by velocity integration) in the navigation coordinates system. This block also interfaces the navigation coordinates system and the shot coordinates system.

• The input data to the Kalman filter block is the vehicle's velocity.

The vehicle's dynamic model [8] proposed is composed of the position, velocity and acceleration over the X and Y components. The measurement variable is the vehicle's velocity. This process produces a smoothed trajectory, which can be used by the absolute navigation block, when for some reasons there is no accurate marker in the sequence. The Kalman filter can track the trajectory as long as the vehicle does not manoeuvre during the markers's absence. Nevertheless, the estimation error becomes greater when the absence of a marker is long in the sequence.

3. CONFIDENCE FACTOR

The estimation for each marker of the same image may vary in a great proportion, because of all the cumulative errors in the different step of the algorithm. A procedure to make the global estimation more robust is needed.

Two parameters, derived from the GHT can help us in the estimation quality evaluation: the height of the GHT accumulation matrix peak value and its secondary peak location and height [7]. This leads to two confidence factors.

- The first factor that has been thought, is the ratio of the maximum peak value (Hmax) to the total number of marker point N. It indicates the proportion of well-matched points in both structures. But, in many cases, it happens that only a part of the marker is used for the GHT computation . A new criteria has been defined: it is the ratio of the number of matched points and the total number of points in the overlapping area. If Sr (resp. So) denotes the set of points of this area in the reference (resp. in the observation) window, we have:

$$F = \frac{|Sr \cap So|}{|Sr \cup So|}$$

Figure 2 illustrates in a geometrical way the computation of this confidence measurement.

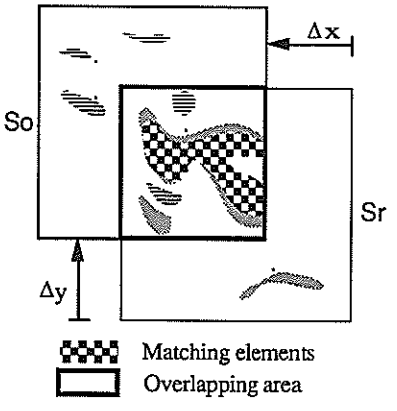


Fig. 2 Definition of the overlapping area

- The second confidence factor acts like a signal to noise ratio. The noise, in the GHT, can be evaluated by the mean square of the secondary peaks in a little neighborhood (v) of the maximum peak.

$$SNR = \frac{H\,max}{\sqrt{E}}$$

with:

$$E = \frac{1}{|v|} \sum_{(i,j) \in v} H(i,j)^2$$

H: Accumulation matrix

Three steps are used to select the marker:

- If the first factor is less than an experimental given threshold (F0), the marker is not reliable enough and therefore is eliminated.
- The SNR ratio is then computed for each remaining marker and the best is retained.
- Then a comparison is then made between this marker displacement estimation and the others. If it is further than four pixels it is not taken into account.

The first factor allows to compute a weighted estimation on such markers. The mean error of the predicted displacement (dx, dy) is then obtained by:

$$(dx, dy) = \frac{\sum_{F > F0} F.(\Delta x, \Delta y)}{\sum_{F > F0} F}$$

(dx,dy) gives the displacement between the reference markers (x,y) in the previous image and the observation markers (in their predicted positions (x^+,y^+) in the actual image. (dx,dy) gives the prediction error on the displacement.

4. SIMULATION RESULTS

Algorithm simulations are supported by data from realistic sequences. Up to now, two different sequences have been tested; each one is obtained from video interlacing raster scanning at a rate of 25 images per second with 512 by 512 pixels resolution.

In order to limit computer requirements we must consider images taken at longer interval than the sampling acquisition interval (40 msec). On the other hand, the sampling interval must be small enough to deal with the maximum displacement that could be encountered so that remains a significance intersection between two consecutive areas of interest. At present images are considered at a sampling interval of 0.2 sec.

The live recording is made in daylight. For testing, we use sequence without preprocessing, such as halo correction, ...; so we have to deal with low contrast images containing a central halo and fluctuation in height and heading.

The instantaneous height measurement and the image interval time (0.2 sec) will enable the computation of absolute navigation based on pixel location knowledge. Thus all displacements are computed and related in pixel intervals.

The first video sequence of 250 images (generic name SF) shows a wreck in Brest bay (the "Swansea"). The second one of 100 images (generic name SG) shows a natural seabed with "concretions". These two sequences are retained because they represent opposing seabed textures.

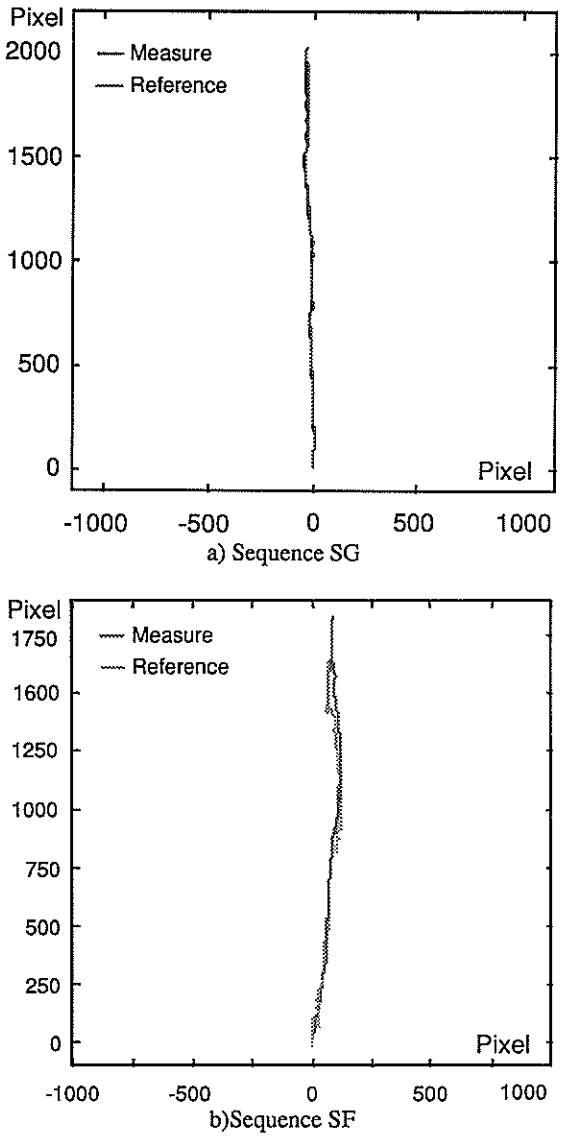


Fig. 3 Computed trajectories

In order to validate the computed trajectory and the theoretical hypothesis, we did a manual sequence analysis. The reference trajectory is obtained by simultaneous tracking of four markers.

Results given here are obtained with major algorithm parameters fixed as below:

Sampling interval:	T= 0.2sec
Image size :	512x512 pixels
Marker size :	M= 75 and N= 75 pixels
Number of markers :	5
Marker's confidence threshold:	F0=10%
Percentage of edge pixels retained in one marker :	Q = 12%

Figure 3 shows simulation results for the trajectories on the two sequences. The reference and computed motion components are shown together. The computed component which is taken into account is the measured one.

The computed trajectory quality is evaluated using comparisons of statistical error parameters (mean and variance) on each computed displacement versus the reference one. Figure 4 shows the error analysis on x and y components of sequences SG and SF.

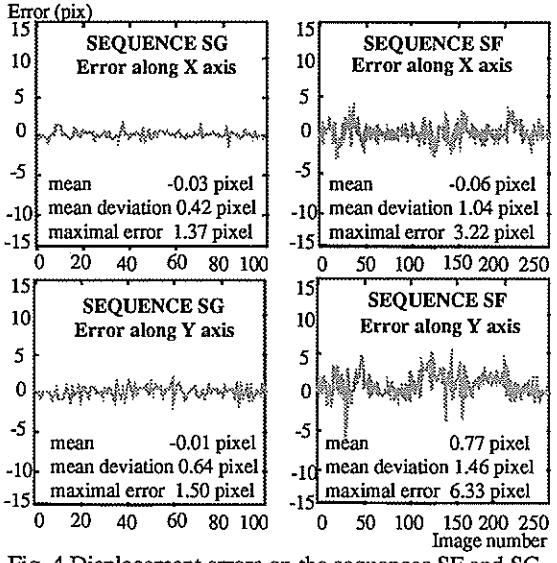


Fig. 4 Displacement errors on the sequences SF and SG

We can see that for the two sequences, the computed trajectory is fairly similar to the reference one. Nevertheless, in the case of the sequence SF, a bias appears on the Y axis between the computed trajectory displacement and the reference one. Meanwhile in the two cases, mean deviations are within limitations of manual analysis; this consequently demonstrates good noise immunity.

Sequence SF presents some problems. Due to the content of the sequence (a wreck), in a same image, two markers can be at very different heights (for example: one marker on the masthead, the other on the seabed); this produces optical aberration resulting in a difference between the computed displacements of each marker. The marker's locations retained for the reference trajectory analysis are not necessarily the same as the locations computed by the algorithm. So this may introduce a bias between the two trajectories when the algorithm chooses markers concentrated at a location whose height is different from those retained in the reference. However this appears to be a very particular case for undersea observation.

On the other hand, the sequence SG trajectory shows the algorithm's accuracy ; the final y axis component error is about 7 pixels out of a total displacement of 2000 pixels with a mean deviation of less than one pixel.The "concretions" are distinct from the seabed, so this gives accurate markers. In the shot boundaries, the seabed is nearly flat so that the computed displacement is independent of the marker location in the image.

5. CONCLUSIONS

The use of a video camera to obtain the trajectory of a submersible vehicle seems to be an efficient accurate way for some applications. An algorithm, based on structural matching process between points of windows in two successive images gives an estimate of the movement; many windows are simultaneously used with a confidence measurement connected to the accumulator matrix parameters in order to have more reliable values.
Simulations have been done with two real image sequences, a wreck and a natural seabed with "concretions", obtained from an experiment of the IFREMER (French Ocean Research Center) .
Some unreliable estimations are found on the sequence of the wreck, because all the markers are not at the same height. Despite this fact, the results show that the trajectory can be estimated by this method with an accuracy less than five pixels on hundred images.

REFERENCES

[1] Aggarwal, J.K, L.S. Davis and W.N Martin, Correspondence Processes in Dynamic Scene Analysis, Proc. of the IEEE, Vol.69, N°5, 1981, pp.562-572.

[2] Aguirre, F., J.M. Boucher and J.P. Hue, Passive navigation of a submersible vehicle by Image Processing, Proc. 4th European signal processing conference, Grenoble, France, 1988, pp. 963-966.

[3] Aguirre, F., J.M. Boucher and J.J. Jacq, Video trajectory estimation of an undersea vehicle, Proc. 6th Symposium on transportation systems IFAC-CCCT'89, 19-21 sept. 1989,Paris, France.

[4] Hue, J.P., F. Aguirre, J.M. Boucher and J.J. Jacq, Underwater optical navigation system - a new concept, Proc. Marine instrumentation'90, 28 Feb 1990, San Diego, USA.

[5] Ballard, D.H. and Brown C.M, Computer Vision, Prentice-Hall INC., Englewood Cliffs, 1982, pp. 128-131.

[6] Ballard, D.H., Generalizing the Hough transform to detect arbitrary shapes, Pattern recognition, Vol. 13, No. 2, pp. 111-122, 1981.

[7] Brown, C.M., Inherent bias and noise in the Hough transform, IEEE trans on PAMI, Vol. PAMI-5, No. 5, pp. 493-505, 1983.

[8] Singer, R.A., Estimating optimal tracking filter performance for manned maneuvering targets, Trans. on AES, Vol. AES-6, N°4, 1970, pp. 473-483.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

ADAPTIVE RECOGNITION OF HEAD BIOSIGNALS FOR BIOSIGNAL CONTROL IN ROBOTICS

Stevo Bozinovski, Georgi Stojanov, Mihail Sestakov

Electrical Engineering Faculty, Karpos II, 91000 Skopje, Yugoslavia

Two trainable algorithms for bioelectric mobile robot control using the biosignals from the human head are described. The signals used are EEG and EOG. The control of the robot is performed by eye movement. In the described experimental design, experiments with EEG start/stop control and experiments with five command EOG control are carried out.

1. INTRODUCTION

There are several communication channels between humans and robots, including tactile, speech, and visual, most commonly used in human communication. Direct bioelectric communication and control has not been widely reported in the robot control literature, although some effort has been done with EMG signals [1], which are used in biomedical rehabilitation and functional stimulation.

Our effort concerns on bioelectric signals which could be obtained from the human head. Objective is to explore possibilities of communication between the human and artificial intelligence with "head to head communication". Parts of this task are contemporary widely explored as separate disciplines, like machine vision and spech communication, and are not of primary interest for our research, although we have made some experiments in those directions.

In the sequel we describe our first results using head biosignals for a robot control. Our results with other types of mobile robot control, are described elsewhere [2,3].

2. CONTROL BIOSIGNALS USED: EEG AND EOG

The biosignals we used in our investigation are EEG signal taken from the occipital region where usualy alpha rhythm is recorded, and EOG signals taken from the eyes movement [5]. The signals are obtained using biomedical amplifier, 14-bit A/D converter, PC/XT computer, 100 Hz sampling rate, and our signal processing software.

Figures 1 and 2 give examples of these signals (lower part) with segment enlargement (central part of the figure).

Figure 1 gives a pattern of the EOG activity during the eyes movement. up/down, left/right, and winking.

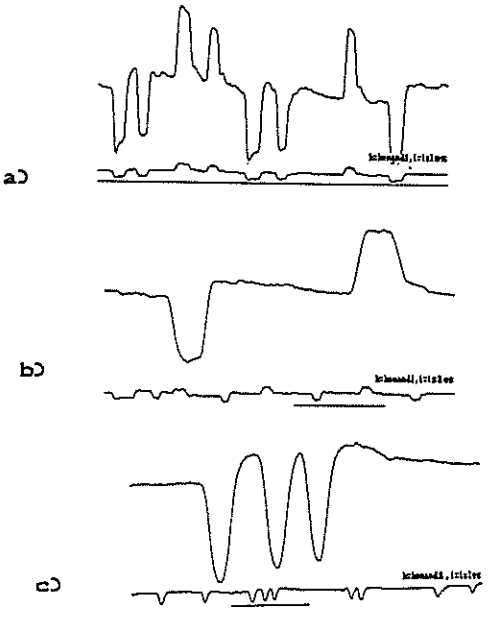


Figure 1. EOG activity due to the eyes movement: Fig 1.a) up/down Fig 1.b) left/right Fig 1.c) winking

Figure 2 shows change of the EEG activity when the operator opens and closes his eyes.

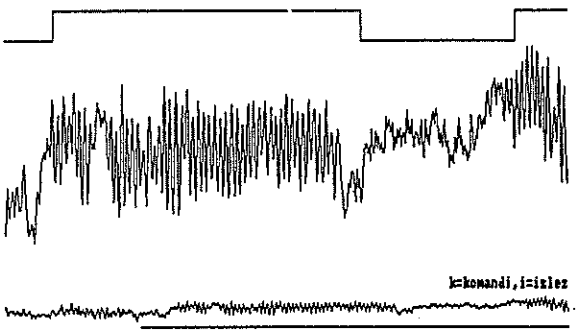


Figure 2. Example of an EEG activity during open and closed eyes (lower and central part) and performance of the closed-eyes recognition algorithm (upper part)

3. THE CONTROL ALGORITHMS

To perform control using the mentioned signals, we designed a control algorithms which are able to recognize the appropriate patterns of the bioelectric activity. The patterns are sensitive to the human operator who generates them, so it was neccessary to design them with ability to adapt their parameters to the considered human operator. The adaptive control algorithms we designed have three working regimes: learning regime, examination regime, and exploatation regime. The exploatation regime is actualy the real-time control of the robot, with the decisions described in the examination regime.

Figure 3 gives the EEG control algorithm an a pseudo-Cobol language. In our experiments we used required_votes=3. Figure 2 in it upper part shows the performance of this algorithm for the shown EEG control pattern.

```
Procedure Learning:
     Perform 10 sec Acquisition
       during which
         the operator has eyes open;
     Compute distributions for
       the time intervals betweeen two extreme points, and
       the amplitudes of those points.
 Perform Procedure Learning
     replacing "eyes open" with "eyes closed".
 Compute decision border points
     for the pairs of distributions
     for "open" and "closed" case.

Procedure examination:
While the operator opens and closes his eyes in real time do:
     Perform Acquisition until a next extreme point is found,
         compute its time interval
         compute its amplitude;
     Compare with the respective "open"/"close" distributions
         if they fall in "open" region
            vote "open"
            increment open-counter
            reset close-counter
            if open-counter=required_votes
               then decision="OPEN";
         if they fall in "close" region
            vote "close"
            increment close-counter
            reset open-counter
            if close counter=required_votes
                then decision="CLOSE";
```

Figure 3: The EEG control algorithm

```
Procedure Learning:
     Perform Acquisition during which
       the operator moves the eyes "up",
       store values for "up".
     Perform Procedure Learning
       replacing "up" with "down","left","right","wink".

Procedure Examination:
     While the operator moves his eyes in real time do:
         measure the amplifiers values L and D
           from the left and the right eye respectively,
           and the frequency F of the signal between L and D;
         if L="PLUS" and D="PLUS" decision="UP"
         if L="PLUS" and D="MINUS" decision="RIGHT"
         if L="MINUS" and D="PLUS" decision ="LEFT"
         if L="MINUS" and D="MINUS"
                 if F="LOW" decision="DOWN"
                 if F="HIGH" decision="WINK".
```

Figure 4. The EOG control algorithm

The EOG control algorithm is given on Figure 4. Its examination procedure can be understand considering the Figure 5.

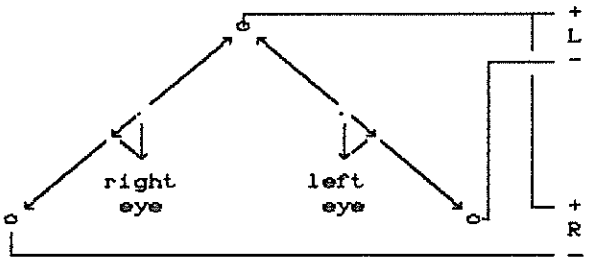


Figure 5. The EOG amplifiers measure the eye dipols projection. It is showh a case when the sight points down

The recognition algorithm during the examination procedure is based on the observation that the EOG amplifiers measure actualy the projection of the eye dipole vector to the line defined by the electrodes. That gives the same polarity for "down" and "wink", winking being significantly faster (fastest human movement).

Figure 6 shows the animation software during the exploatation regime. During the real time control the screen shows its animated representation: both the eyes movement and the robot movement.

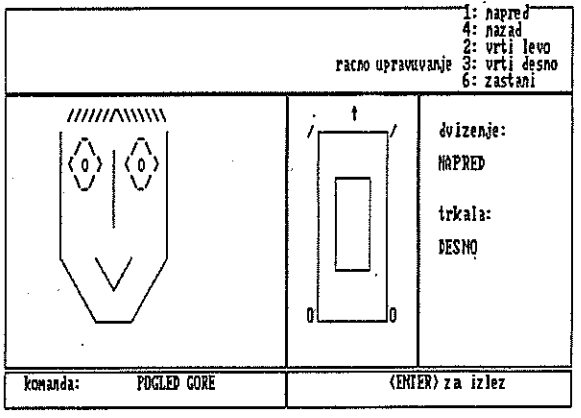


Figure 6. Animatied representation of the mobile robot control during the exploatation regime of the control algorithm

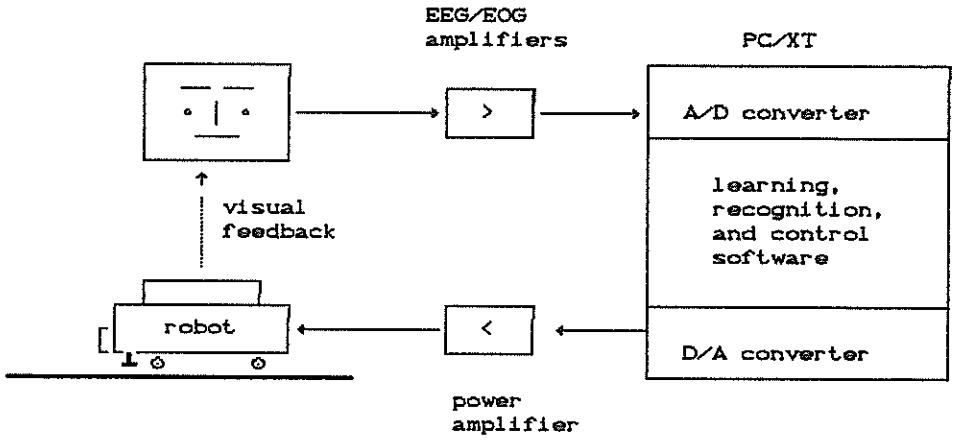


Figure 7. The experimental design

4. THE EXPERIMENTAL DESIGN AND THE EXPERIMENTS

Our experimental investigation is based on the experimental design given on Figure 7.

The EEG control experiment includes an Elehoby Line Tracer mobile robot which has its own intelligence to follow a black trajectory drawn on a white floor. The operator performs start/stop control during the robot movement along the trajectory, by stopping the robot when he opens his eyes, and continuing robot movement as long as the eyes are closed.

The EOG control experiment includes our mobile robot which we designed for our investigation with various modes of robot control [2-6]. On a terrain with obstacles, the operator has a task to lead the robot from a starting to a goal point using five command (Forward, Bacward, Left, Right, Stop), given by his eyes movement (Up, Down, Left, Right, Blink).

The exploatation modules of the control algorithms perform the appropriate mapping between the recognized eyes movement and robot action.

The experiments are carried out with various subjects, students of the subject Robotics and biocybernetics, ninth semester of the Computer Science Division of the Electrical Engineering Faculty. The average time a student to be trained to successfully perform a control task is 30 min.

5. CONCLUSION

Our experimental result have confirmed our expectation on possibility of robot guidance using only a head biosignals. Until now we experimented with simple EEG and EOG signals. Our further effort will be in design a helmet which will enable other head signals also to be used in robot control.

REFERENCES

[1] A. Kobrinskij, A. Kobrinskij "Manipulation Robot Systems" (In Russian) "Nauka", Moskow, 1985

[2] S. Bozinovski, "Adaptive inteligent robots: Learning, goal seeking, and experience gathering," (In Serbocroat) Proc. Symp. Applied Robotics, Vrnjacka Banja, 1985

[3] S. Bozinovski. "Mobile robots in flexible manufacturing systems" Proc. Symp. on Applied Robotics and Flexible Automation," (In Serbocroat) Bled, 1987

[4] S. Bozinovski, M. Sestakov, L. Bozinovska. "Mobile robot control using alpha rhythm from the human brain" (In Serbocroat) Proc. Symp. JUREMA, Zagreb, 1988

[5] S. Bozinovski, M. Sestakov, G. Stojanov, L. Bozinovska, "Bioelectric mobile robot control" (In Macedonian) Proc. Symp. on Robotics and Flexible Automation, Novi Sad, 1989

[6] S. Bozinovski, M. Sestakov, G. Stojanov, "Learning control of mobile robots using head biosignals" USSR-Yugoslav Symposium on Robotics, Moskow, 1989

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Real-Time Movement Detection

S.Mathis, A.Gunzinger, W.Guggenbühl

Electronics Laboratory, Swiss Federal Institute of Technology
CH-8092 Zürich, Switzerland

Movement detection is an essential step in many image processing applications for robotics, automation and surveillance. To meet speed requirements of these applications, real-time execution of movement detection algorithms is needed. In this paper the real-time implementation of such an algorithm on the synchronous dataflow machine (SYDAMA [1]) is described. As an application, the use of movement detection for a cooperative robot is shown.

1 Introduction

Movement detection is an essential step in many image processing applications for robotics, automation and surveillance. Different approaches for detection of moving objects in a sequence of images have been published. Some solutions to this problem will be mentioned below, before our own algorithm and its real-time implementation is described.

In [2] an algorithm is presented using cross correlation for evaluating displacement vector fields between succesive frames in a sequence. Statistical methods to detect moving targets are also proposed. In [3] a statistical clustering algorithm to detect moving targets is described. The statistical segmentation algorithm in [4] takes information from motion prediction to detect moving targets. Measurement of 'change energy' is used in [5]. The algorithm is combined with pyramid transforms to minimize the required computing power. Yet another approach is presented in [6]. At each pixel candidate, trajectories are hypothesized. These trajectories are mapped onto nodes in a tree. Using 'multistage hypothesis testing' each canditate is tested, the most probable being taken.

Several differenceing algorithms are presented in [7] for subpixel target detection and tracking. Interframe difference between two adjacent images in a timesequence is another often used technique ([8], [9], [10]). Frame to frame changes are interpreted as motion related phenomena. With this method, differences occur where a moving object is covering or uncovering the background, making it difficult to extract the exact shape of the moving object.

An improvement to this method is not to compare two adjacent images in a sequence but to compare an incomming image with a stored reference image of the background ([11], [12]). To improve this method further, it is possible to update the reference image periodically, to take into account changes occuring in the background. With this update, slow changes in the background due to changes in illumination are suppressed.

Other techniques must be used if significant changes to the background are possible, for example,when the position of the camera is not fixed; e.g. extraction of moving objects from a scene taken by a moving camera ([2], [7]).

In this paper, an algorithm is described based on comparing each image in a sequence with a stored reference image. The reference image is updated recursively to eliminate changes in the environment caused by changeing light. The implementation of the algorithm on the synchronous dataflow machine (SYDAMA) is shown. Some results are presented to show the capabilities of the implemented algorithm. The results are taken from a viedeo tape showing a natural scene with moving objects. Finally, an application out of the field of robotics is presented in which this algorithm could be used: surveillance of a cooperative robot operating in the same workspace as a human.

2 Movement Detection Algorithms

A given scene is observed by a video camera; any movement within this scene must be detected. First the background and the moving object must be modelled.

As a first approach, the background is modelled by a constant pattern. Any pixel within the image, differing by more than a given threshold from the corresponding background value, is considered to belong to a moving object.

This may be done by storing an image of the observed scene with no moving object in it. This stored image is now used as a reference image. Differences between the reference image and a newly taken image from the same scene, which are greater than the threshold value, are classified as the moving object:

$$r(x,y) = in(x,y,t_0)$$

$$obj(x,y,t_k) = \begin{cases} 1 & \text{if } abs(in(x,y,t_k) - r(x,y)) > thr \\ 0 & \text{otherwise} \end{cases}$$

$in(x,y,t_k)$ is the two dimensional input image taken from the image sequence at time t_k, x and y denote the space coordinates. See figure 3 for an example of an image with moving objects. $r(x,y)$ is the reference image taken at time t_0. $obj(x,y,t_k)$ is the binarized representation of the recognized moving object at time t_k (1 stands for a moving object and

0 stands for a non-moving image part (background)). See figure 5 for an example of a segmented image. thr is the threshold value.

Due to changes in the environment (changeing light etc.) this algorithm leads to false results; moving objects are detected even when there is no movement in the scene.

By using another model for background pixels, changes in environment may be eliminated. Instead of using a fixed reference, the mean value of the last n images is calculated and used as a reference. Because of the huge amount of memory needed to store n images, this approach may not be used directly. Therefore the mean value is approximated by a recursive calculation, such that only one image must be stored:

$$r(x,y,t_{k+1}) = \alpha \times in(x,y,t_k) + (1-\alpha) \times r(x,y,t_k)$$

$$obj(x,y,t_k) = \begin{cases} 1 & \text{if } abs(in(x,y,t_k) - r(x,y,t_k)) > thr \\ 0 & \text{otherwise} \end{cases}$$

$r(x,y,t_k)$ is the recursive updated reference image at time t_k. See figure 2 for an example of a reference image. α is the refreshing factor for reference image update ($\alpha < 1$).

Although the sensitivity to environmental changes is eliminated, another disadvantage has been introduced: objects moving into the scene and stopping there will be recognized as moving objects only as long as they move. Once they stopp, the recursive update algorithm will filter these objects out and they are lost. Once they begin to move again, both the object and the underlying background will be recognized as moving objects, since the area left by the objects is temporarly interpreted as moving. To eliminate this undesirable behavior, the reference image update can be controlled by the detected movements:

$$d(x,y,t_k) = in(x,y,t_k) - r(x,y,t_k)$$

$$againobj(x,y,t_k) = \begin{cases} 1 & \text{if } abs(d(x,y,t_k)) > thr \\ 0 & \text{otherwise} \end{cases}$$

$$r(x,y,t_{k+1}) = r(x,y,t_k) + \alpha \times d(x,y,t_k) \times (1 - obj(x,y,t_k))$$

$d(x,y,t_k)$ is the difference image between the input image and the reference image. See figure 4 for an example of a difference image.

Due to non-idealities in the algorithm, single pixels may be recognized as moving objects. Since real objects are, in general, greater than any one single pixel, these misidentified pixels can be easily eliminated. This is done by using a median filter on the binary image $obj(x,y,t_k)$ in the local neighbourhood. The kernel size of the median filter depends upon the size of the moving objects to be detected.

3 Implementation on the Synchronous Dataflow Machine

The algorithm described above has been implemented on the synchronous dataflow machine in real time (50 pictures per second, 256×256 pixels).

This machine has been developed at the Electronics Laboratory, Swiss Federal Institute of Technology. The *sy*nchronous *da*taflow *ma*chine (SYDAMA) is a fast special purpose parallel computer for real-time image processing. The central concept of this computer is the direct hardware mapping of a static dataflow graph. Initially, a vision algorithm is described in a functional programming language especially designed for low level, real-time image processing. This algorithm is then translated by the compiler into a static dataflow graph. This graph is subsequently partitioned into subgraphs, each representing one processing element. These subgraphs are automatically mapped onto processing elements, whereby the goal is to minimize communication paths.

Example: The implementation of the above-described algorithm will be shown. The algorithm is expressed in our functional programming language which is close to the mathematical description of the algorithm.

```
d(x,y,t)     <- in(x,y,t)-r(x,y,t);
obj(x,y,t)   <- CASE | 1 IF ABS(d(x,y,t)) > thr(t)
                     | 0 OTHERWISE;
r(x,y,t+1)   <- r(x,y,t)+a(t)*d(x,y,t)*(1-fobj(x,y,t));
fobj(x,y,t)  <- rank(obj(x,y,t),5,5,med);
feat(x,y,t)  <- fobj(x,y,t);
```

`feat` denotes a specialized processing element for calculation of up to second order moments. `rank` is a specialized processing element for applying the median filter to the binarized image. A kernel size of 5×5 pixels is used here, but kernel sizes of up to 8×8 pixels would be possible.

This program is translated into a static dataflow graph, which is automatically mapped onto the existing hardware. Now the algorithm may be executed on the synchronous data flow machine in real time. In this case, real time means that 50 pictures per second are processed, each having a resolution of 256×256 pixels.

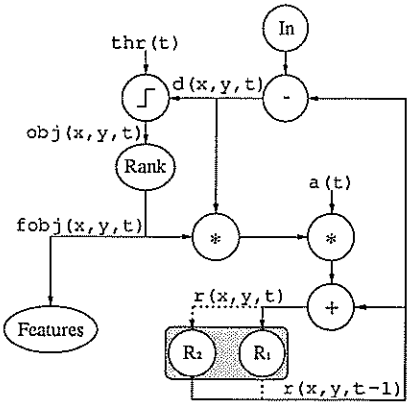


Figure 1: Dataflow graph for the described movement detection algorithm

The filtered and binarized difference image *fobj(x,y,t)* is used for further processing. In the next processing part, within the dataflow machine, up to second order moments of the binarized image are calculated. These moments are sent to the host computer in order that the center of gravity and the dimensions of the moving objects may be calculated. With this information, a post-processor on the host may take some application-dependent actions.

Figure 2: Reference image of a natural scene

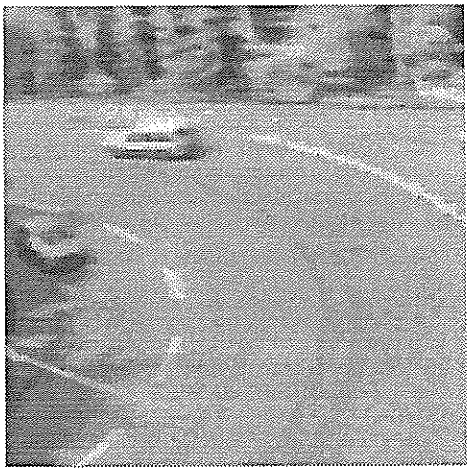


Figure 3: Image with moving objects

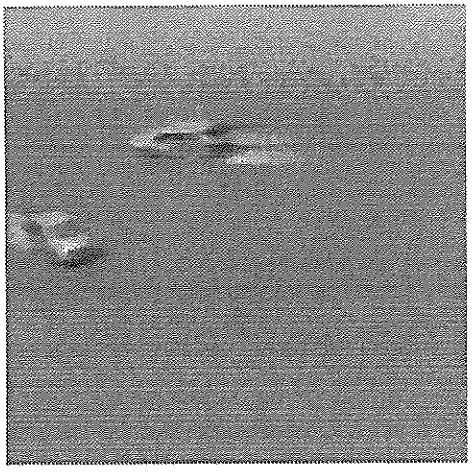


Figure 4: Grayvalued difference between the two images in figure 2 and figure 3

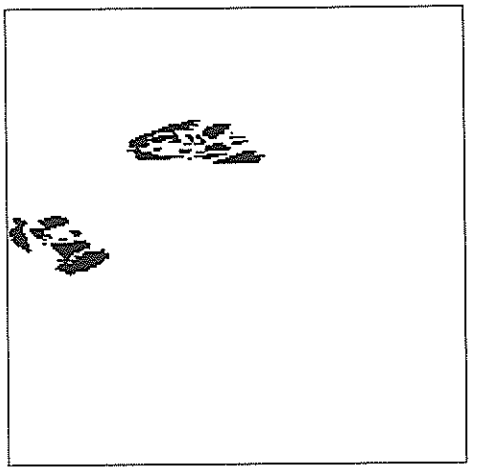


Figure 5: Segmented image

In figures 2 ... 5 some results using the described algorithm on the synchronous dataflow machine are presented. The images are taken from a video tape and processed in real time (50 frames per second). Figure 2 shows the stored reference image. In figure 3, an image with two moving objects is shown. The difference between these two images is shown in figure 4. Figure 5 shows the segmented image from which the features can be extracted.

4 Application

One application of this algorithm is in the field of a cooperative robot system. This means, a robot working with a man in the same working area. To avoid accidents, some precautions have to be taken to protect the man from being touched by the robot. A security system performing this task may be implemented using a vision system.

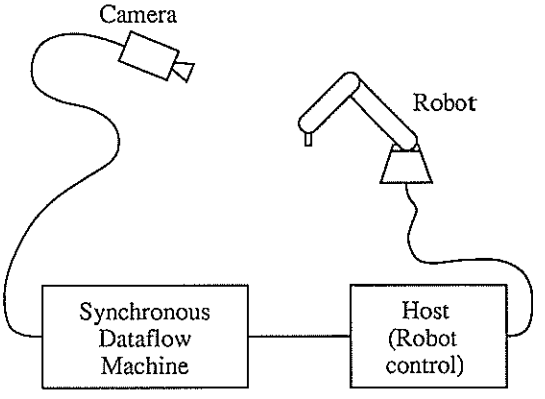


Figure 6: System overview for a cooperative robot: The image from the camera is processed by the synchronous dataflow machine. Results from the movement detection algorithm are fed to the host computer which controls the robot

Meeting the speed requirements for such a system is

very difficult. The robot system must stop if something comes within reach of the robot arm. However reaction time may be easily determined. Suppose the robot arm moves with a tangential speed of 5m/sec, and a moving object is not permitted to come closer than 0.5m to the robot arm, a reaction time of less than 100msec is required. In these 100msec, the moving object must be recognized, the command to stop the robot must be executed and the robot must stop, without touching the object. With the described system, this speed is possible.

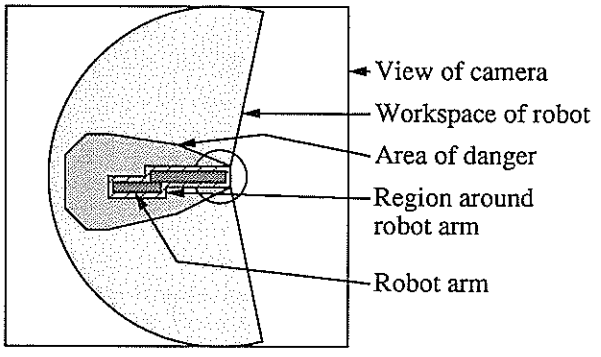


Figure 7: View of the camera with different regions

For the implementation of this algorithm, the image is divided into several regions. Depending upon where the moving object is located, the behavior of the robot can be controlled accordingly. The first region is outside the working area of the robot. If an object is recognized as being inside this region, no action is required, because it is impossible for the robot to reach the object. If an object is in the second region which is within the possible workspace of the robot, the speed of robot movement must be reduced, as it would be possible for the robot to touch the object. An object located in the third region causes an emergency stop of the robot arm. The fourth region is the robot arm itself. Because the robot is itself a moving object, it must be discounted by the movement detection algorithm. A view of the camera with the different regions is shown in figure 7.

The first and fourth of the above-mentioned regions are implemented using a mask, so that only movements occuring in regions two and three are detected by the movement detection algorithm. The required mask depends upon the exact position of the robot arm. It is therefore updated under control of the host computer which knows the exact position of the robot arm. The decision if an object is in the second or third region is also calculated by the host computer.

5 Conclusion

A movement detection algorithm which works in a wide range of environments has been presented. Changes in background illumination do not effect the detection of moving objects. A real-time implementation on the synchronous dataflow machine meets the difficult speed requirements for use in robotics.

References

[1] S. Mathis, A. Gunzinger, W. Guggenbühl, *A Synchronous Dataflow Machine for Real-Time Image Processing*, Proceedings of EUSIPCO-88, September 5–8, 1988, Grenoble, France, pp 911–914.

[2] Peter Spoer, *Displacement Estimation for Objects on Moving Background*, NATO ASI Series, Vol. F2, Image Sequence Processing and Dynamic Scene Analysis, Edited by T.S. Huang, Springer Verlag Berlin Heidelberg 1983, pp 424–436.

[3] Minami Yamada, Shinji Ozawa, *Estimation of Detecting Reliability and Tracking Condition for a Picture Tracking System of Moving Targets*, Computer Vision and Pattern Recognition, IEEE 1983, pp 394–396.

[4] D.S. Kalivas, A.A. Sawchuk, R. Chellappa, *Estimation and Segmentation of Image Sequences*, Proc. SPIE Vol. 829, Applications of Digital Image Processing X (1987), pp 15–23.

[5] C.H. Anderson, P.J. Burt, G.S. van der Wal, *Change Detection and Tracking Using Pyramid Transform Techniques*, Proc SPIE Vol. 579, Intelligent Robots and Computer Vision (1985), pp 72–78.

[6] Steven D. Blostein, Thomas S. Huang, *Detection of Small Moving Objects Using Multistage Hypothesis Testing*, Proc. of International Conference on Acoustics, Speech and Signal Processing, Vol. II, Multidimensional Signal Processing, pp 1068–1071.

[7] D. Casasent, B.V.K. Vijaya Kumar, Y.L. Lin, *Subpixel Target Detection and Tracking*, Proc. SPIE Vol. 726, Intelligent Robots and Computer Vision: Fifth in a Series (1986), pp 206–220.

[8] Hans-Helmut Nagel, *Overview on Image Sequence Analysis* NATO ASI Series, Vol. F2, Image Sequence Processing and Dynamic Scene Analysis, Edited by T.S. Huang, Springer Verlag Berlin Heidelberg 1983, pp 2–39.

[9] Ramesh Jain, W.N. Martin, J.K. Aggarwal, *Segmentation through the Detection of Changes Due to Motion*, Computer Graphics and Image Processing, Vol. 11 (1979), pp 13–34.

[10] Ramesh Jain, *Extraction of Motion Information from Peripheral Processes*, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. PAMI-3, Number 5, September 1981, pp 489–503.

[11] Gregory W. Donohoe, Don R. Hush, Nasir Ahmed, *Change Detection for Target Detection and Classification in Video-Sequences* Proc. of International Conference on Acoustics, Speech and Signal Processing, Vol. II, Multidimensional Signal Processing, pp 1084–1087.

[12] Klaus-Peter Karmann, Achim von Brandt, *Moving Object Recognition Using an Adaptive Background Memory*, Proc. of the 3rd International Workshop on Time-Varying Image Processing and Moving Object Recognition, 2, Edited by V. Cappellini, Elsevier Science Publishers B.V., 1990, pp 289–296.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

FREQUENCY DOMAIN ANALYSIS OF NONUNIFORMLY SAMPLED SIGNALS BY DIRICHLET TRANSFORM

Andrzej Wojtkiewicz, Michal Tuszyński

Institute of Electronic Fundamentals, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, POLAND*

In this paper we propose to use Dirichlet transform for spectral analysis of deterministic and stochastic nonuniformly sampled signals processed by digital filters. The results presented will be limited to basic properties of a nonuniformly sampled analog signal in Dirichlet frequency domain. This theory has been developed and used mainly to analyse signal processing in radars; however it may be applied much wider.

1. INTRODUCTION

Let us assume that an analog signal x(t) is sampled at time instants $t_{np}=nT_0+t_p$, $n=0,\pm1,\pm2,\ldots$, $p=0,1,2,\ldots,M-1$, $t_{p+1}>t_p>0$, $t_0=0$, $t_{rM}=rT_0$, $r=0,\pm1,\pm2,\ldots$. Sampling instants belonging to one T_0 period may also be written as $t_p=p\tau+\delta_p$, where δ_p is a displacement from the average sampling period $\tau=T_0/M$ ($\delta_0=\delta_{rM}=0$). The $\{x(t_{np})\}$ sample train is then processed by a stationary digital filter with transfer function H(z). This periodically nonuniform (staggered) sampling and subsequent digital filtering is often used in coherent Doppler radars [1].

As the *Z*-transform does not preserve information about sampling instants, we propose Dirichlet transform [2,3], defined as:

$$X(\omega)=D[x(t_{np})]=\sum_{p=0}^{M-1}\sum_{n=-\infty}^{\infty}x(nT_0+t_p)e^{-j\omega(nT_0+t_p)} \tag{1}$$

which is a natural generalization of the Z-transform on the unit circle and is related to Dirichlet series, discrete Laplace transform and Fourier series of almost-periodic functions.

The inverse Dirichlet transform $D^{-1}[X(\omega)]$ may be found from expanding Dirichlet spectrum X(ω) into Fourier series:

$$x(t_{np})=D^{-1}[X(\omega)]=\lim_{\omega_1\to\infty}\frac{1}{2\omega_1}\int_{-\omega_1}^{\omega_1}X(\omega)e^{j\omega t_{np}}d\omega \tag{2}$$

2. DIRICHLET SPECTRUM OF DETERMINISTIC SIGNALS

Let x(t) be a finite energy analog signal. For further analysis, the Dirichlet spectrum (1) of a nonuniformly (but periodically) sampled signal $x(t_{np})$ may be written as:

$$X(\omega)=\frac{1}{M}\sum_{m=0}^{M-1}F_m^*(\omega)X^I(\omega+m\omega_0) \tag{3}$$

where ($\omega_0=2\pi/T_0$, * denotes conjugation)

$$F_m(\omega)=\sum_{p=0}^{M-1}e^{j\omega\delta_p}e^{-j\frac{2\pi mp}{M}} \tag{4}$$

is a DFT of a function of a M-periodic displacements' sequence $\{\delta_p\}$ and

$$X^I(\omega)=\sum_{p=0}^{M-1}\sum_{n=-\infty}^{\infty}x(nM\tau+p\tau+\delta_p)e^{-j\omega(nM+p)\tau} \tag{5}$$

is the *Z*-transform of the $\{x(t_{np})\}$ sequence on the unit circle (Fourier spectrum). The main difference between spectra (3) and (5) lays in the fact that the spectrum $X^I(\omega)$ is periodic with period $2\pi/\tau$, while the Dirichlet spectrum X(ω) is, generally, an almost-periodic function.

An interesting property of Dirichlet spectrum may be found from expressing it in terms of analog signal Fourier spectrum. Let us assume

* This work was sponsored by CPBP 02.14

that complex harmonic signal $e^{j\omega_d t}$ (for instance Doppler radar echo signal) is modulated by envelope function g(t) (for instance antenna directional characteristic). Now we can prove that the Dirichlet spectrum of the nonuniformly sampled input signal

$$x(t_{np}) = g(t_{np}) e^{j\omega_d t_{np}} \tag{6}$$

is

$$X(\omega) = \frac{1}{T_0} \sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} F^*_{-k}[(mM+k)\omega_0] G^F[\omega - \omega_d - (mM+k)\omega_0] \tag{7}$$

where $G^F(\omega)$ is the Fourier spectrum of the envelope g(t), while the sample sequence $\{x(t_{np})\}$ spectrum is:

$$X^I(\omega) = \frac{1}{T_0} \sum_{k=0}^{M-1} \sum_{m=-\infty}^{\infty} F_k[\omega - (mM+k)\omega_0] G^F[\omega - \omega_d - (mM+k)\omega_0] \tag{8}$$

If g(t) is narrow-band then

$$F^*_{-k}[(mM+k)\omega_0] \approx F^*_{-k}(\omega - \omega_d) = F^*_{M-k}(\omega - \omega_d) \tag{9}$$

so the train of shifted replicas of $G^F(\omega)$ in (7) is modulated by functions $F^*_{-k}(\omega-\omega_d)$ while in (8) this train is modulated by

$$F_k[\omega - (mM+k)\omega_0] \approx F_k(\omega_d) \tag{10}$$

As $F_k(0) = M$ for $k = 0, \pm M, \ldots$ and $F_k(0) = 0$ for $k \neq 0, \pm M, \ldots$ so property (9) may be used for detection and Doppler frequency estimation of nonuniformly sampled signals by MTD-like systems working in the Dirichlet frequency domain [4].

A Dirichlet spectrum X(ω) of narrow-band, nonuniformly sampled signal (6) modulated by Gaussian shaped function g(t) is presented on Fig. 1 for M = 7, $1/\tau = 400$ Hz, $f_d = \omega_d/2\pi = 0$ (solid line), $f_d = 200$ Hz (dashed line) and displacements δ_p [ms]: $\delta_0=0$, $\delta_1=0.2$, $\delta_2=-0.2$, $\delta_3=-0.7$, $\delta_4=-1.0$, $\delta_5=-0.6$, $\delta_6=0$

The sample sequence spectrum $X^I(\omega)$ of the signal (6) is shown on Fig. 2. This spectrum is, as expected, a periodic function with 400 Hz period, so it is not possible to recover unambiguous Doppler frequency from it. However, it may be possible to recover unambiguous Doppler frequency f_d, in range depending on nonuniform sampling pattern $\{\delta_k\}$, from the Dirichlet spectrum by use of (9) and knowledge of functions $F_k(\omega)$.

In many practical applications (for instance, in radar data processing) signal $x(t_{np})$ is processed by stationary digital filter with transfer

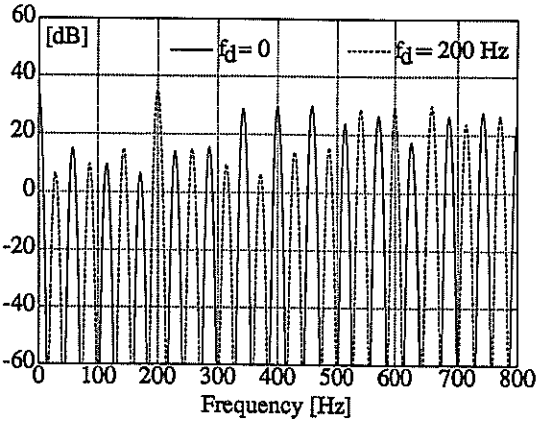


Fig. 1. Dirichlet spectrum (7) of a signal (6)

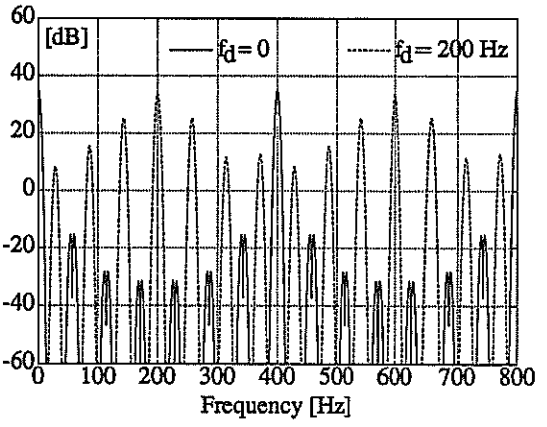


Fig. 2. Fourier spectrum (8) of a sequence (6)

function H(z). It can be proved that Dirichlet spectrum Y(ω) of a digital filter output signal $y(t_{np})$ is:

$$Y(\omega) = \frac{1}{M} \sum_{m=0}^{M-1} F^*_m(\omega) H(\omega + m\omega_0) X^I(\omega + m\omega_0) \tag{11}$$

where $H(\omega)=H(z)$, $z = e^{j\omega\tau}$ and $X^I(\omega)$ is a Fourier spectrum (5) of the input sample sequence $\{x(t_{np})\}$. The spectrum of the output sequence $\{y(t_{np})\}$ is $Y^I(\omega) = H(\omega) X^I(\omega)$.

Fig. 3 shows Dirichlet spectra (11) of a signal (6) for $f_d=0$ (solid line) and $f_d=200$ Hz (dashed line) processed by a highpass, second order digital FIR filter with coefficients $a_0=a_2=1$, $a_1=-1.969$ One could see from Fig. 1 and 3 that the signal with $f_d=0$ is heavily attenuated, in contrast to signal with $f_d=200$ Hz. In radar data processing, we may assume that signal (6) with $f_d=0$ represents ground clutter and signal with $f_d=200$ Hz represents target signal and that those signals are narrow-band.

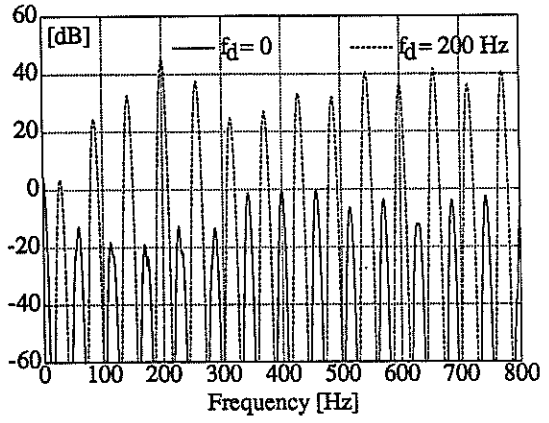


Fig. 3. Dirichlet spectrum (11) of a digital filter output signal

An important case is analysis of a steady-state digital filter (for instance, MTI filter) output for almost-periodic, finite power input:

$$x(t_n) = e^{j\omega(n\tau+\delta_n)} \tag{12}$$

If we expand periodic displacements' function $e^{j\omega\delta_n}$ into Fourier series:

$$e^{j\omega\delta_n} = \frac{1}{M}\sum_{m=0}^{M-1} F_m(\omega)\, e^{j\frac{2\pi mn}{M}} \tag{13}$$

where coefficients $F_m(\omega)$ are defined by (4), then the input signal (12) may be decomposed into a sum of M uniformly sampled (with period $\tau = T_0/M$) harmonic signals located at frequencies $\omega+m\omega_0$ and having amplitudes $F_m(\omega)/M$;

$$x(t_n) = \frac{1}{M}\sum_{m=0}^{M-1} F_m(\omega)\, e^{j(\omega+m\omega_0)n\tau} \tag{14}$$

Now, it is easily seen that the steady-state digital filter output signal is:

$$y(t_n) = \frac{1}{M}\sum_{m=0}^{M-1} F_m(\omega)\, H(\omega+m\omega_0)\, e^{j(\omega+m\omega_0)n\tau} \tag{15}$$

The MTI filter frequency response is defined as a ratio of output (15) and input (12) signal powers. The output signal power is a sum of powers of each output signal harmonic component:

$$A^2(\omega) = \frac{1}{M^2}\sum_{m=0}^{M-1} |F_m(\omega)|^2\, |H(\omega+m\omega_0)|^2 \tag{16}$$

Eq. (16) was used in a design of adaptive MTI filter for ground, weather and bimodal clutter suppression [5].

3. DIRICHLET SPECTRUM OF STOCHASTIC SIGNALS

Formula presented above may be generalized for nonuniformly sampled stochastic signals [6].

Let x(t) be a finite variance stationary stochastic signal of zero mean and known autocorrelation function $R_x(t') = \mathcal{E}[x^*(t)x(t+t')]$. The Dirichlet power spectrum of the nonuniformly sampled signal $x(t_{np})$ is defined as a Dirichlet transform of this signal autocorrelation function:

$$S_x(\omega) = \sum_{s=0}^{M-1}\sum_{p=0}^{M-1}\sum_{m=-\infty}^{\infty} R_x(mT_0+t_p-t_s)e^{-j\omega(mT_0+t_p-t_s)} \tag{17}$$

where $R_x(mT_0+t_p-t_s) = \mathcal{E}[x^*(t_{ns})x(t_{n+m,p})]$, while standard (Fourier) power spectrum of a stochastic sample sequence $\{x(t_{np})\}$ is defined as:

$$S_x^I(\omega) = \sum_{s=0}^{M-1}\sum_{p=0}^{M-1}\sum_{m=-\infty}^{\infty} R_x(mT_0+t_p-t_s)e^{-j(mM+p-s)\omega\tau} \tag{18}$$

The relation between (17) and (18) is much more complicated than in previous case, because here we have to deal with two-dimensional Fourier transforms. However, one can prove that:

$$S_x(\omega) = \frac{1}{M^2}\sum_{k=0}^{M-1}\sum_{r=0}^{M-1} F_r^*(\omega)F_k(\omega)S_x^I(\omega_r,-\omega_k) \tag{19}$$

where

$$S_x^I(\omega_r,-\omega_k) = \sum_{s=0}^{M-1}\sum_{p=0}^{M-1}\sum_{m=-\infty}^{\infty} R_x(mT_0+t_p-t_s)\cdot e^{-j[\omega_r(mM+p)\tau-\omega_k s\tau]} \tag{20}$$

for $\omega_r = \omega+r\omega_0$, $\omega_k = \omega+k\omega_0$.

If an stochastic sequence $\{x(t_{np})\}$ is processed by stationary digital filter H(z) then the two-dimensional power spectrum of the output sequence $\{y(t_{np})\}$ may be written (20) as:

$$S_y^I(\omega_r,-\omega_k) = H(\omega_r)H^*(\omega_k)S_x^I(\omega_r,-\omega_k) \tag{21}$$

The Dirichlet power spectrum of the output signal $y(t_{np})$ can be computed from (19).

Dirichlet power spectrum can also be expressed by a power spectrum $S_x^F(\omega)$, which is a Fourier transform of autocorrelation function $R_x(t')$ of analog input signal. It may be proved that:

$$S_x(\omega) = \frac{1}{T_0}\sum_{k=0}^{M-1}\sum_{m=-\infty}^{\infty} |F_{-k}[(mM+k)\omega_0]|^2 S_x^F[\omega-(mM+k)\omega_0] \tag{22}$$

This expression is a generalization of (7) for stochastic signals.

In practice, the autocorrelation function and Dirichlet power spectrum is usually not known and should be estimated from a finite number of samples taken from one realization of a stochastic process. A convenient, asymptotically unbiased estimator of this power spectrum - Dirichlet periodogram - is:

$$\hat{S}_N(\omega) = \frac{1}{2N+1} |X_N(\omega)|^2 \qquad (23)$$

where

$$X_N(\omega) = \sum_{p=0}^{M-1} \sum_{n=-N}^{N} x(nT_0+t_p) e^{-j\omega(nT_0+t_p)} \qquad (24)$$

is Dirichlet spectrum of truncated signal $x_N(t_{np})$.

The estimation of Dirichlet power spectrum is relatively simple; it may be done by estimating (for instance, by FFT) power spectrum $X_N^1(\omega)$ of sample sequence $\{x_N(t_{np})\}$ and computing Dirichlet power spectrum $S_x(\omega)$ from (3) and (23).

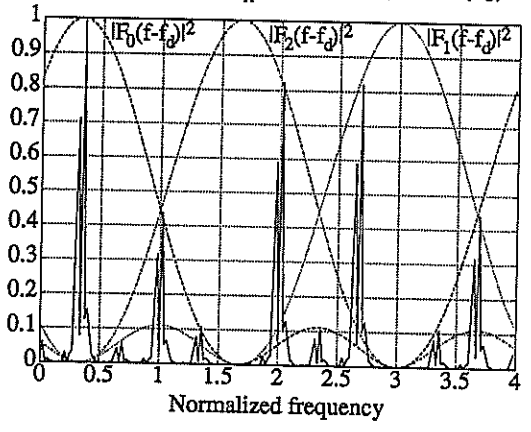


Fig. 4. Dirichlet periodogram

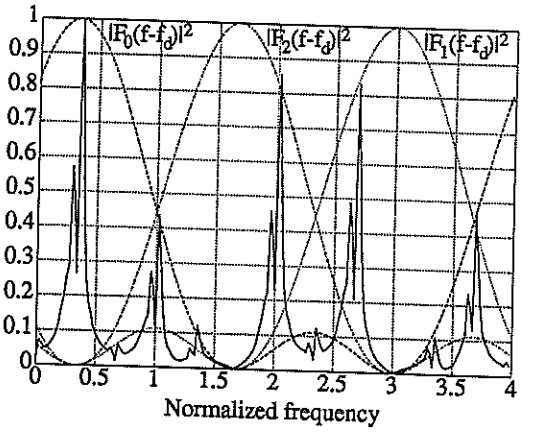


Fig. 5. Dirichlet power spectrum

Fig. 4 shows Dirichlet periodogram (23) of a nonuniformly sampled complex stochastic signal with Gaussian power spectrum for M = 3, $\delta_0=0$, $\delta_1=0.25$, $\delta_2=-0.25$, $\tau=1$ and center frequency $f_d\tau=0.33$. Fig. 5 shows the Dirichlet power spectrum estimated from autocorrelation function samples.

4. CONCLUSIONS

Spectral analysis of nonuniformly sampled signals processed by digital filters has been developed and used mainly to analyse digital signal processing in radars. This results have practical relevance, because they facilitate simple spectral analysis of nonuniformly sampled signals processed by digital filters, simpler MTI filter analysis and synthesis and development of new clutter suppression schemes working in the Dirichlet frequency domain, similar to MTD systems.

It seems that results of this analysis may be used in search for new, non-stationary variable delay filters. For instance, filter impulse response $h(t_{np})$ may be a nonuniformly sampled analog filter impulse response h(t).

REFERENCES

[1] Skolnik M.I. (ed.): "Radar Handbook", McGraw-Hill, New York, 1970.

[2] Wojtkiewicz A., Tuszynski M.: "The Z-domain matrix analysis of nonuniformly sampled data processing by the digital filters", Proc. ECCTD-83, Stuttgart 1983.

[3] Wojtkiewicz A., Tuszyński M., Klimkiewicz W.: "Analysis and design of MTI digital filters processing nonuniformly sampled signals", Proc. ECCTD-85, Prague 1985.

[4] Chambon P., Domette G.: "Radar application of staggered sampling signal processing and of spectral pattern recognition techniques", Int. Conf. on Radar, Paris 1984.

[5] Tuszyński M., Wojtkiewicz A.: "Bimodal clutter MTI filter for staggered PRF radars", Proc. IEEE Int. Radar Conf., Washington 1990, to be published.

[6] Wojtkiewicz A., Tuszynski M.: "Frequency domain analysis of staggered sampling MTI filters by Dirichlet transform", Proc. Int. Symp. on Noise and Clutter Rejection in Radars and Imaging Sensors, Kyoto 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

APPLICATION OF DAP BASED DFTS TO FAST SAR PROCESSING

JOHN J. SORAGHAN Signal Processing Div, Dept. of Elec.Eng, Univ. of Strathclyde, Glasgow, Scotland.
DEREK G. APPLEBY Dept. of Electronics and Comp. Science, Univ. of Southampton, Southampton, UK.
RICHARD G. GREEN First Base, Beacon Tree Plaza, Gillett Way, Reading, Berks., UK.

ABSTRACT: Mapping DFT problems onto the family of DAPs and DAP based SAR processing are examined.Algorithm timing results are presented. A hybrid DAP based PFA/FFT/WFTA is proposed which has both computational and operational advantages over the conventional power-of-two based range-Doppler SAR processing algorithms.

1. Introduction

This paper discusses the implementation of the DFT on a distributed array of processors such as the AMT family of DAPs. The DAP is described in Section 2 together with programming considerations. Section 3 describes the mapping choices for a range of discrete Fourier transform (DFT) problems and presents the corresponding parallel algorithms. The performance for these are given for 32-bit floating point and 16-bit fixed bit arithmetic using the DAP-510. The hybrid PFA/FFT/WFTA DAP based DFT algorithm is developed in section 4. Mapping the SAR processing problem onto the DAP is described in section 5. Some simulation results are given in Section 6. Both SEASAT-SAR and ERS-1 simulated data are used in the experiments on the DAP-510. The paper concludes with a discussion of the consequences of using the hybrid DFT algorithm for SAR processing.

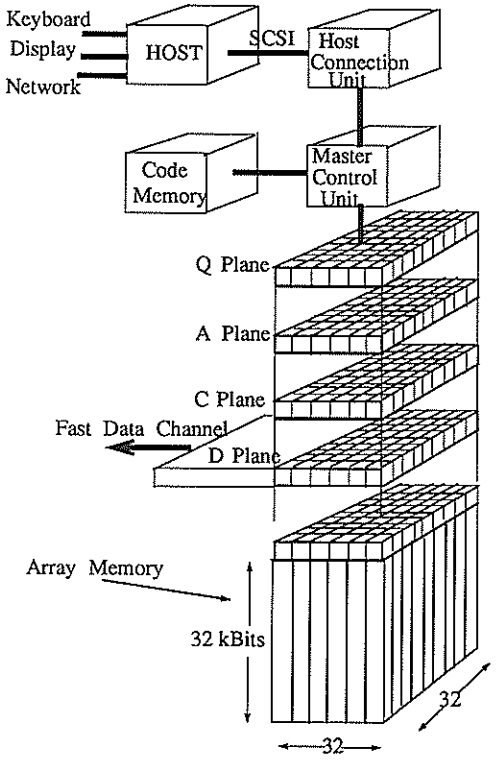


Figure 1 The DAP-510-4 Schematic

2. The Distributed Array Processor (DAP)

Figure 1 shows a schematic of the massively parallel DAP (Distributed Array Processor). The heart of the machine is a 2-D array of simple 1-bit processing elements (PEs). Each PE within the array has connections to its 4 nearest neighbours, the other PEs in its row and column, and to its own local memory. Memory addressing is under the control of the MCU (Master Control Unit) so a memory access by the PEs can be considered as the processing of a bit plane. Within an individual PE there are essentially four registers: the Q, C A, and D registers. The Q (accumulator) and the C (carry) registers are used to develop the arithmetic whereas the A and D registers have more specialised purposes. The A or activity register controls the writing of results to memory, so while all PEs execute the same instruction not all results need be written back to memory. The D register provides the array with a route to the real world. Planes of memory can be loaded into the D plane of registers then, asynchronously to the subsequent operations can be fed to the Input/Output devices at a sustained rate of 50 MBytes per second. At this I/O rate there is only a 4% overhead on processing for the 500 series machines and a 1% overhead for the 600 series. The DAP can perform high-speed arithmetic, input/output, and data movement; in short all the functions needed for high-speed real-time signal processing.

Floating point, fixed point, Boolean and user-defined number formats are supported by DAP computers. Access to these number formats is essentially by two programming languages: Fortran-plus and APAL. Fortran-plus is an extended Fortran, having array and vector data types, with a comprehensive set of functions and operators to process arrays and vectors. Most notably there are extensions that allow the user to utilise the nearest neighbour and broadcast features of the DAP. Alternatively the user may choose to program the DAP using APAL; the DAP's assembly language.

While floating point and fixed point arithmetic and their use in signal processing algorithms is well understood, the potential benefits of user defined-arithmetic are less well understood. As the DAP builds up its arithmetic functions from bit serial operations, arithmetic of arbitrary precision can be performed. In particular the arithmetic precisions can be tailored to the precise requirements of the algorithm. For example, high performance FFTs can be programmed by tapering the arithmetic

precision to the word growth through the algorithm.

3. DFT on the DAP

The degree of parallelism offered by a DAP containing P PEs is variable and ranges from 1 (Scalar Mode) to P (Maximum Machine Parallelism). The design of efficient parallel DFT algorithms begins with a proper choice of data mapping.

Given N DFTs each of length M and assuming that $N > P$ then clearly $[N/P]_F$ transforms may be simultaneously performed using a maximum parallelism of P. For greatest efficiency it is advantageous for N to be an exact multiple of P. (The $[.]_F$ is the 'floor' function.) This type of problem is referred to as the ideal parallel problem wherein an optimum sequential algorithm is simultaneously performed within each PE. In the case of the DFT set it is defined as the Vertical Transform (VT).

When $N < P$ then the N transforms may be simultaneously performed using a parallelism of N. For optimum efficiency it is advantageous that N divides P exactly. This type of algorithm (where the data is spread across/down PEs) is defined as the Horizontal/Vertical Transform (HVT). Depending on N and the data length M the algorithm may consist of a pure Horizontal Transform, HT (NM=P) or a HVT (NM > P).

When N and P are powers of 2 and $N < P$ then for N,M-point DFTs the DAP is firstly partitioned into N sections each containing P/N PEs. Each data set of length M is mapped under the P/N PEs down a depth of (NM/P) into the DAP store. The 1-d data sets are thus mapped into a natural 2-d structure and a parallel version of the familiar ROW/TWIDDLE/COLUMN DFT algorithm [1] may be used to implement the sets of 1-d transforms. In the general case the resulting DAP algorithm would consist of a VT, twiddling and finally a series of HTs.

The difference between a VT and a HT is that in the latter data communication is required between the PEs. Hockney [2] has shown that where possible, it is more economical to perform the complete transform using VTs and rotation of data if the data sets extend into memory by more than two samples i.e. if $(NM/P) > 2$. In this paper it is assumed that the DFTs may be performed using only VTs.

In [3] a general formula for the family of DFTs described above is developed. A performance table for various sized DFT sets implemented on the DAP-510 is shown in Table 1.

The performance index used is the execution time per transform. The throughput for both 32-bit floating point and 16-bit fixed point arithmetic is shown in the table. As the parallelism of the problem increases the throughput increases as expected.

4. Enhancing the Throughput : FFT Alternatives

In many problems the DAP will be used with a medium degree of parallelism. This may be due to the nature of the problem (non-stationary) or due to a limited DAP memory. In [4] it was shown that in such cases, the relative cost of the Twiddling component of the DFT algorithms accounted for between 11% and 35% of the overall transform costs. The figure depends on the precision used and the size of the problem.

One of the advantages of FFT alternatives such as the Prime Factor Algorithm (PFA) or the Winograd DFT [5] is that the complex 'Twiddle' factors described above are replaced by noiseless permutations of data elements. When the DAP is operating with a medium degree of parallelism then unlike sequential algorithms, the permutations involved in the PFA/WDFT may not be performed using simple indexing.

The two data permutations encountered in the PFA and WDFTA are:

(A) The simple permutation (SP).

(B) The Chinese Remainder Theorem permutation (CRTP) each of which is defined as follows.

Given a 1-D set of numbers x_n, n=0,1,...$M-1$ and that $M = Lr$ where $(L,r)=1$ [i.e. L and r are relatively prime] then the SP is obtained using the following index substitution:

$$n = [\, L.i \;+\; r.j \,] \; modulo \; M \qquad (8)$$

$$\text{for } i = 0,1,2...r-1 \text{ and } j = 0,1,2...L-1.$$

The CRTP is obtained using the following substitution for the index n:

$$n \;=\; [\, L.i.g \;+\; r.j.p \,] \; modulo \; M \qquad (9)$$

$$\text{for } i = 0,1,2...r-1 \text{ and } j = 0,1,2...L-1$$

The integers g and p are obtained as solutions to the following Diophantine equations:

$$L.g \;=\; 1 \; modulo \; r \;\text{ and }\; r.p = 1 \; modulo \; L \qquad (10)$$

The above permutations may be extended to any number of dimensions, once all the dimensions are relatively prime.

An M-point PFA with $M = p_1,p_2,\ p_3...p_D$ and $(p_i,p_j) = 1$, $i \neq j$, involves the following tasks:

(a) An SP to permute the 1-D sequence into a D-dimensional sequence

(b) $\left(\frac{M}{p_i}\right)$ short length DFTs for each dimension 'i', i=1,D.

(c) A CRTP to permute the output sequence to its natural order.

In an M-point WFTA tasks (a) and (c) are similar to those of the PFA. For task (b) Winograd devised an efficient nesting scheme which minimises the total number of multiplications required in the resulting algorithm .

To construct efficient DAP SP and CRTP algorithms the following constraints are imposed given N sets of M-point DFTs :

(a) L must be a power of two and $r = L-1$

(b) If either the Natural to CRTP or its inverse map is required then the data must be mapped under L PEs.

(c) If either the Natural to SP or its inverse is required then the data must be mapped under r PEs.

DAP permutation algorithms for the PFA/WFTA are presented in [6] and their complexity analised.

The hybrid DAP PFA/FFT/WFTA algorithm simultaneously performs N, M-point DFTs with M=Lr, under conditions (a) - (c) given above. A block diagram of the algorithm is shown in figure 2.

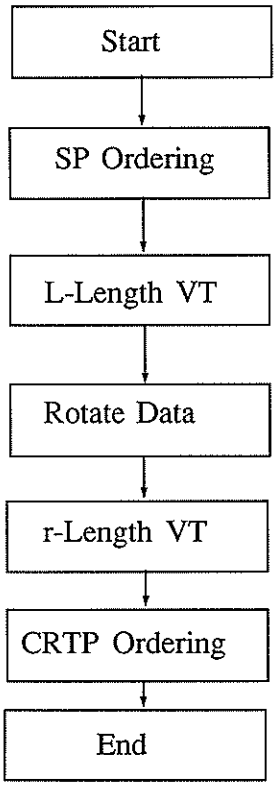


Figure 2 M Length DFT with M=Lr and (L,r)=1

Following the SP permutation an L-point VT is carried out using an efficient DAP-FFT VT algorithm. Having rotated the data, either a WFTA or a PFA based VT of length 'r' is performed. The choice depends on the actual value of 'r'. A final CRTP permutes the data to normal order.

5. SAR Processing

In general the SAR azimuth compression [7] problem is a 2-dimensional correlation problem. However if the point target azimuth reference function is stored as a 1-dimensional vector and a 'straightening' procedure is applied across the synthetic aperture then efficient 1-dimensional time domain correlation may be utilised. However for each new azimuth position a new straightened data set is required which results in low throughput. Wu's frequency domain range migration correction (RMC) scheme is based on the fact that the azimuth response to point targets located at the same slant range but at different azimuth positions map to the same curve in the Doppler frequency space. Thus by straightening a block of range compressed data in the Doppler space the efficient 1-dimensional FFT may be used for fast correlation over the block.

Speckle exists in all coherent systems but due to the wavelengths used in SAR it is particularly annoying. To reduce speckle the complete aperture can be subdivided into a number of 'looks' and a separate image produced for each look. All the 'looks' are incoherently combined. Of course a separate azimuth reference is required for each look. Further details of these SAR algorithms may be found in [7].

Mapping the Problem (DAP-600)

The 2-dimensional array of complex range compressed data is subdivided into data blocks of length 64 range vectors by length 4096 azimuth vectors. This dimension has been chosen so as to set up a parallelism within the problem that may be exactly matched by the proposed array processor. The degree of data block overlap requirement is dependent on the synthetic aperture length in the azimuth direction and on the amount of range migration measured in range samples in the range direction.

The following memory mapping technique was chosen. A complete azimuth vector is folded under a row of 64 PEs. This scheme accommodates 64 azimuth vectors of maximum length that depends on the wordlength selected. Three points are noted:

(a) This mapping allows us to directly use the parallel DFT algorithms described earlier.

(b) The azimuth vectors are located side by side and hence with the nearest neighbour connectivity of the DAP very efficient RMC is possible.

(c) Selecting contiguous Doppler bands corresponding to the separate 'looks' may be accomplished by simply segmenting the DAP store in equal sections thus avoiding complex data manipulation routines. (Note: In the case of the DAP-510 the value of 64 would be replaced by 32 in the above discussion)

6. Simulation Results (DAP-510)

The SAR system specifications [7] define the geometry of the system, the antenna gain pattern and the synthetic aperture length. For simulation purposes a grid within the actual swath is user defined. This sets up the minimum slant range the number of range samples and the desired azimuth extent. Point targets of variable strengths are located on this grid. The range compressed return from these point targets is obtained by coherently summing the response of each point target in turn across the image grid .

Algorithm run time is measured using the DAP-510 timer. Table 2 shows the times for the single LOOK algorithm (SLA) and Multi-LOOK Algorithm (MLA) for a block of 32 azimuth vectors each of length 4096 complex samples.

Note that the cost of processing 4 looks of a 4 look system is comparable to the single look cost. The MLA offers significant increase in throughput when only a subset of the total number of looks are processed.

7. Hybrid Algorithm and SAR Processing on the DAP-600

The value of L and 'r' for the DAP-600 are 64 and 63 respectively. Thus 64 azimuth vectors each of length 4032 complex samples map under 63 PEs. Using the hybrid algorithm described in section 4 to transform the data into the Doppler domain sets up the array for Wu's RMC. An efficient power-of-two algorithm is used for the 64 length VT while a WDFT is used for the 63 length VT. Notice that during the 64-point VT one PE per azimuth vector is idle. The multilook data extraction process involves segmenting the transformed data into the desired number of LOOKs. As the factor 63 has exact divisors 3,7,9,21 and 63 these can be combined with the factors 2,4,8,16,32 and 64 to offer a wide range of natural divisions within the problem size. Choosing a 7-LOOK system, for example, would result in the division of each 64x63 block into 9 sub-blocks each of dimension 64x7. The dimensions remain relatively prime and hence a suitable hybrid algorithm may be used for the inverse transformation. The multilook processor requirements are thus catered for using the alternative DAP based alternative algorithms, and in fact in the case of the 64x63 azimuth vector the range of naturally occurring unique divisors is increased compared to the power-of-two based algorithm.

Currently a complete (raw data to SAR image) DAP based SAR processor in under development. This will incorporate both the FFT based and the hybrid DFT based range-Doppler SAR processing algorithms.

References

[1] H.J. Nussbaumer, "Fast Fourier Transforms and Convolution Algorithms", Springer Verlag, New York, (1982).

[2] R.W. Hockney and C.R. Jesshope, "Parallel Computers", Adam Hilger Ltd., (1981) U.K.

[3] Soraghan J.J. and Green R.G. "Parallel DFT Algorithms for Radar Signal Processing", Int. Radar Conf. Versailles April 1989.

[4] Green R.G. and Soraghan J.J. "Parallel DFT Algorithms on a distributed array of processors", IMA Conf. on Math in Signal Proc., Warwick, Dec 1988.

[5] Blahut R. "Fast Algorithms for Digital Signal Processing", Addison-Wesley, 1985.

[6] Soraghan J.J. "New Data Movement Algorithms for Processor Arrays", ICASSP-88,pp. 1930-1933, New York 1988.

[7] K. Tomiyasu, "Tutorial Review of synthetic aperture radar (SAR) with applications to imaging of the ocean surface", Proc. IEEE, May 1978, Vol. 66, 563-583.

Table 1
Performance Table showing FFT throughput for the DAP-510 using both 32-bit Floating Point (FlP) and 16-bit Fixed Point (FxP) arithmetic. The figures represent the cost per transform in msecs. The table shows N transforms each of length M

DAP-510 FFT Throughput Table				
N M	32 (FlP,FxP)	64 (FlP,FxP)	256 (FlP,FxP)	1024 (FlP,FxP)
32		(.13,0.047)	(0.06,0.013)	(0.043,0.011)
128	(0.61,0.21)	(0.56,0.13)	(0.34,0.085)	(0.276,0.074)
512	(2.54,0.81)	(2.0,0.6)	(1.76,0.46)	(1.46,0.376)
1024	(4.13,1.2)	(4.02,1.1)	(3.84,0.99)	
2048	(9.03,2.6)	(8.9,2.54)	(8.48,2.17)	
4096	(19.8,5.8)	(17.59,4.94)		

Table 2
DAP-510 Block Processing times for both the SLA and MLA. The blocksize is 32x4096 complex 16-bit samples and a 4-LOOK MLA is assumed.

SLA		MLA			
		Cost per Block		Cost per Look	
Oper	cost msecs	Oper	cost msecs	Oper	cost msecs
Data Format	60	Data Format	38	Look Select	2
FFT	186	FFT	186	Mult.	11
RMC	33	RMC	33	IFFT	39
Mult.	42			Store	5
IFFT	186				
Total	507	Total	257	Total	57

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SYNTHESIS OF FREQUENCY HOP CODES WITH IDEAL RANGE-DOPPLER AUTO-AMBIGUITY PROPERTIES FOR RADAR AND SONAR SYSTEMS

JEROME R. BELLEGARDA
IBM RESEARCH, T. J. WATSON RESEARCH CENTER, YORKTOWN HEIGHTS, NEW YORK 10598, USA

SVETISLAV V. MARIC, EDWARD L. TITLEBAUM
DEPT. OF ELECTRICAL ENGINEERING, UNIV. OF ROCHESTER, ROCHESTER, NEW YORK 14627, USA

IVAN SESKAR
COMPUTER, CONTROL, AND MEASUREMENTS INST., UNIV. OF NOVI SAD, NOVI SAD, YUGOSLAVIA

The synthesis of frequency hop pulse train signals from specified auto-ambiguity characteristics is discussed using the hit array formalism, originally developed for frequency hop waveform analysis in the context of coherent active radar and sonar echolocation systems. Under certain conditions, the hit array can be viewed as a discrete version of the ambiguity function, which makes it a suitable input to the synthesis procedure. A general backtracking-type algorithm is presented to reconstruct the frequency hop pattern(s) associated with any given (realizable) hit array. For limited, *a priori* range and Doppler shifts, this is the first efficient algorithm for direct synthesis from ideal ambiguity function specifications. A reconstruction example is shown for illustration purposes.

I. INTRODUCTION

The ambiguity function has proven a useful analysis tool in evaluating the performance of signals used in coherent active radar and sonar systems. Since an ideal "thumb tack" shape is necessary to perform reliable target and/or scattering function measurements [1], performance can be predicted from the departure of an (auto-)ambiguity function from this (non-realizable) ideal shape. In general, a narrow main lobe and adequately small sidelobes are pre-requisites to achieve high resolution in both range and velocity. It is therefore of interest to study the *synthesis* problem, i.e., how to retrieve a signal from such "close to ideal" ambiguity characteristics. Early efforts, however, have failed to produce a systematic direct synthesis algorithm (cf., e.g., [2]).

Because of that, the preferred approach has been "synthesis-by-analysis," based on the properties of a selected family of realizable waveforms. Since high-efficiency multi-components signals must have disjoint components in *both* time and frequency [3], a natural choice is the family of (multiple access) frequency hop pulse train signals. In this context, Costas proposed a class of signals with nearly ideal ambiguity properties [4]. Although his original code arrays were found through an exhaustive computer search, algebraic constructions are also possible for Costas arrays [5]. Such synthesis-by-analysis methods, however, quickly become unpractical as the size of the time-frequency space increases.

To summarize, neither starting directly from ambiguity domain specifications as in [2] nor enumerating potential code arrays from a known suitable class as in [4] leads to a satisfactory solution to the synthesis problem. Recently, an intermediate entity was developed, which, under certain conditions, has the same information content as the ambiguity function [7]. In this paper we show that this concept, called hit array formalism, is also useful as a synthesis tool, and derive an algorithm for generating frequency hop signals from specified hit array/ambiguity function characteristics.

The paper is organized in the following way. The next section briefly reviews the concept of *hit array* for multiple access frequency hop signals. In Section III, we develop a backtracking algorithm for synthesizing frequency hopping pattern(s) corresponding to a given realizable hit array. Finally, Section IV discusses synthesis from ideal ambiguity properties for limited, *a priori* range and Doppler shifts.

II. SIGNAL SETS AND HIT ARRAY

Frequency hop signals used in echolocation systems can be defined as follows. Consider a rectangular pulse of length T seconds, divided into N equal segments. Let B be the (radian) bandwidth available, so that each signal considered occupies a time-bandwidth product of approximately $2BT$. In each segment of the pulse (time slot) we place one, and only one, subpulse written as:

$$s_k(t) = p\Big(t - \frac{T}{N}k\Big)\, e^{j(\omega_k t + \theta_k)}, \qquad \text{for } 1 \le k \le N, \quad (1)$$

where $p(t) = 1/\sqrt{T}$ if $-T/N \le t \le 0$ and 0 otherwise, and:

$$\omega_k = \omega_0 + y_k \frac{B}{N}, \qquad \text{for } 1 \le k \le N, \quad (2)$$

with some suitable ω_0. The (ordered) set of integers $\{y_k\}_{k=1}^N$ is obtained through a *placement operator*, as defined through Definition 1 of [7] (see also [6] in these Proceedings), which can represented through a two-dimensional frequency hopping pattern (grid) $\{(k, y_k) : 1 \le k \le N\}$. (An example is provided in Fig. 1.a for $N = 6$: it corresponds to a Welch–Costas placement operator using 3 as a primitive root for 7 [5].) Finally, the *placement difference* $v_{i,k}$ is given by:

$$v_{i,k} = y_{i+k} - y_i, \quad (3)$$

for all $1 \le i \le N - k$ and $0 \le k \le N - 1$.

This work was sponsored in part by SDIO/IST and managed by ONR under contract N00014-86-K-0511.

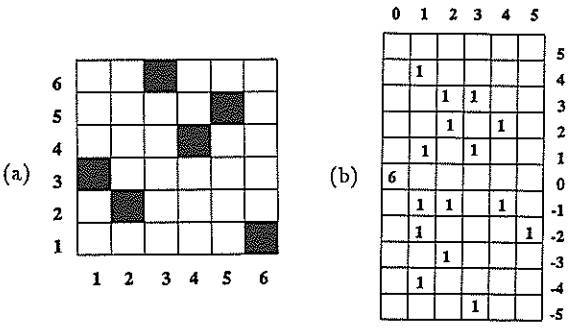


Fig. 1. (a) Example of Code Array $\{(k, y_k) : 1 \leq k \leq N\}$, for $N = 6$; (b) Corresponding Hit Array $\{(h_{k,\ell})\}$.

A coincidence, or "hit," is defined as a time-frequency shift for which one element of each of the two sequences $\{y_{i+k}\}$ and $\{y_i\}$ occupy the same time-frequency position. In other words, a hit corresponds to a placement difference of zero. The collection of all hits, together with their location, forms the hit array, as defined in Definition 2 of [7]. The hit array is addressed by the placement $(k; v_{i,k})$, for $0 \leq k \leq N-1$ and all $1 \leq i \leq N-k$, and its value at position (k, ℓ) is obtained as:

$$h_{k,\ell} = \sum_{i=1}^{N-k} \delta(v_{i,k} - \ell), \tag{4}$$

where $\delta(m) = 1$ if $m = 0$ and 0 otherwise. Among other properties (cf. [7]), the hit array is odd symmetric, so we need only display half of it. For illustration purposes, the right hand side of the hit array corresponding to the placement operator of Fig. 1.a is shown in Fig. 1.b.

The usefulness of the above formalism lies in three facts: (i) if $N^2 \ll BT$ the hit array can be viewed as a discrete version of the ambiguity function [7]; (ii) in contrast with the ambiguity function itself, necessary and sufficient conditions for the realizability of a hit array can be easily derived [8]; and (iii) a frequency hop signal can be synthesized from a given realizable hit array without having to resort to an exhaustive search. While for space reasons (i) and (ii) cannot be addressed here, we elaborate on (iii) below.

III. BACKTRACKING ALGORITHM

In this section we construct a general backtracking algorithm which admits as input any potential hit array (realizable or not). For simplicity, we assume that the elements of this input hit array are all zero or one except of course for the center value (N); also, for convenience we take N to be an odd number. These assumptions are without loss of generality. From this given hit array, it is straightforward to collect, for each $1 \leq k \leq N-1$, all the values ℓ_i such that $h_{k,\ell_i} = 1$, where $1 \leq i \leq N-k$. This leads to a $N-1$ row triangle whose k^{th} row is written as:

$$\ell_1 \;\; \ell_2 \;\; \cdots \;\; \ell_{N-k} \; . \tag{5}$$

The issue lies in the *ordering* of the row (5), information which is not available in the hit array. What we need to find is *some* permutation(s) of (5) which could be realized by a signal with placement difference $v_{i,k}$, so that (5) could be identified with:

$$v_{1,k} \;\; v_{2,k} \;\; \cdots \;\; v_{N-k,k} \; , \tag{6}$$

for some appropriate values $v_{i,k}$. As it turns out, to solve the problem it is sufficient to order the first row only: for a realizable triangle all subsequent rows can be uniquely determined from the first row, and from the triangle the code array follows immediately by solving a system of difference equations [8]. Given the following identities for all $1 \leq k \leq N-1$, easily obtained from the definition (3):

$$v_{1,k} = \sum_{j=1}^{k} v_{j,1} \; , \tag{7}$$

$$v_{N-k,k} = \sum_{j=1}^{k} v_{N-k+j-1,1} = \sum_{j=N-k}^{N-1} v_{j,1} \; , \tag{8}$$

it is clear that if (5) is to be identified with (6), and if $k \neq N-1$, then the left most and right most elements of this row are distinct and *must* satisfy (7) and (8), respectively. Starting with $k = N-2$ and proceeding to $k = (N-1)/2$, the backtracking algorithm will work by relating this pair of elements to a pair of elements in the first row symmetrically placed with respect to the center of the first row.

More specifically, let us re-arrange (7) and (8) as:

$$v_{1,k} - \sum_{j=1}^{N-k-2} v_{j,1} = v_{N-k-1,1} + \sum_{j=N-k}^{k} v_{j,1} \; , \tag{9}$$

$$v_{N-k,k} - \sum_{j=k+2}^{N-1} v_{j,1} = \sum_{j=N-k}^{k} v_{j,1} + v_{k+1,1} \; , \tag{10}$$

for all $(N-1)/2 \leq k \leq N-2$, and suppose for a moment that the left hand sides of both (9) and (10) are known. Under this condition, the problem of ordering the pair of first row elements which occur in positions $v_{N-k-1,1}$ and $v_{k+1,1}$ is reduced to determining two possible combinations of the first row elements which satisfy (9) and (10). Simple analysis of element occurrence in those combinations gives the desired result: any element occuring once in each combination must be eliminated, leaving only one element in the combination satisfying (9) (which therefore must occur in position $v_{N-k-1,1}$) and only one element in the combination satisfying (10) (which therefore must occur in position $v_{k+1,1}$). Should more than two combinations of elements satisfy (9) and (10), it is necessary to stack these combinations into two lists and keep track of the results. The algorithm proceeds for the next row in the range mentioned above, and, in case no combination can be found, backtracking occurs and the next potential solution is popped from the stack and further extended. If the hit array is realizable, eventually one finds at least one ordering which satisfies both (9) and (10) for all $(N-1)/2 \leq k \leq N-2$.

Given these basic principles, let us look at the practical implementation of the algorithm. At the beginning, we set $k = N-2$, by virtue of which (9) and (10) become:

$$v_{1,N-2} = v_{1,1} + v_{2,1} + \cdots + v_{N-2,1} \; , \tag{11}$$

$$v_{2,N-2} = v_{2,1} + \cdots + v_{N-2,1} + v_{N-1,1} \; . \tag{12}$$

Note that the values in the set $\{v_{1,N-2}, v_{2,N-2}\}$ are known from the hit array. Furthermore, for $k = N-2$ the order of the elements is not essential, since, from Property 4 of [7],

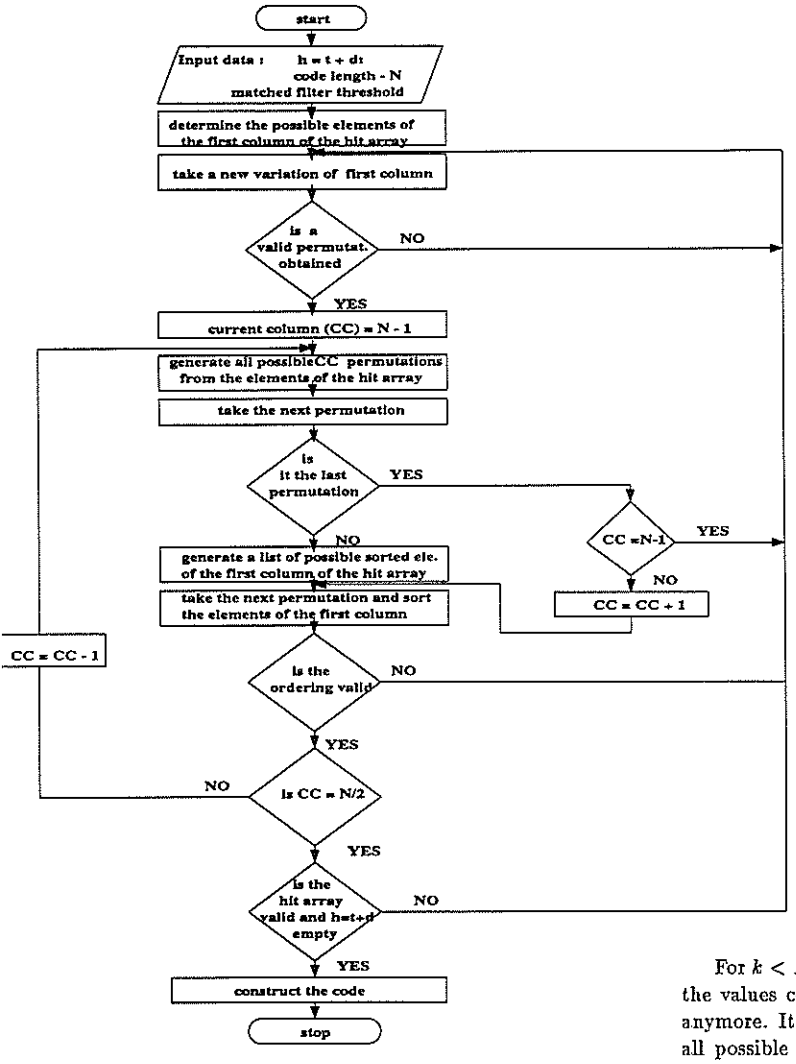


Fig. 2. Backtracking Algorithm Overview.

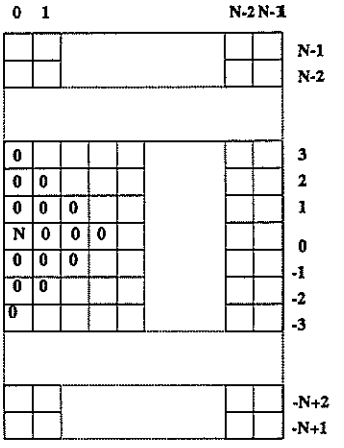


Fig. 3. Generic Hit Array with Forbidden Hit Positions.

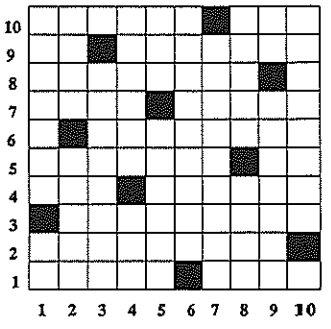


Fig. 4. Code Array Obtained Using the Backtracking Algorithm.

and provided the hit array is realizable, the two orderings lead to two related code words; we can therefore take the order of $v_{1,N-2}$ and $v_{2,N-2}$ arbitrarily. As a result, the left hand sides of (11) and (12) are known. Thus, applying the procedure outlined above, one has to generate all the possible combinations of length $N-2$ out of the $N-1$ elements composing the first row of the hit array, and check whether or not they satisfy (11) and (12). Any combination which satisfies (11) (respectively, (12)) is stacked into the list of "left" (respectively, "right") combinations. Analysis of elements occurrence is performed on the top combination of each list, resulting in potential candidates for $v_{1,1}$ and $v_{N-1,1}$. These two elements will be considered determined in the sequel, unless backtracking proves necessary at some further step, in which case new candidates for $v_{1,1}$ and $v_{N-1,1}$ would be computed. If, on the other hand, no combination can be found which satisfies (11) and (12), the hit array is declared non-realizable and the algorithm is terminated.

For $k < N-2$ things get a little more complicated because the values corresponding to $v_{1,k}$ and $v_{N-k,k}$ are not known anymore. It is therefore necessary to try out in (9) and (10) all possible left sides which can be generated from all the elements of the k^{th} row. For a given choice of $v_{1,k}$ and $v_{N-k,k}$, the above algorithm is applied. If it fails, another choice of $v_{1,k}$ and $v_{N-k,k}$ is considered; if it succeeds in finding potential candidates for $v_{N-k-1,1}$ and $v_{k+1,1}$, the indices of the elements of the k^{th} row that were used for $v_{1,k}$ and $v_{N-k,k}$ are stored in parallel lists for later reference. In this way the left most and right most elements of any row in the range considered are always known when backtracking occurs.

The above step is repeated until all of the first row elements are ordered. However, since the algorithm essentially implements a suboptimal search, this does not guarantee that the ordered first row is a valid one. It remains to construct the complete triangle from this first row and check that all the values obtained match the original specifications from the hit array. If the solution is inconsistent, backtracking must be resorted to in order to find another potential candidate for the ordered first row. Finally, if all the candidates turn out to be inconsistent, the hit array is declared non-realizable and the algorithm is terminated. To illustrate the procedure described in this section, an overview of the backtracking algorithm is presented in Fig. 2.

IV. SIGNAL SYNTHESIS

The above (hit array-based) procedure implicitly assumes that an adequate hit array has already been selected from the ambiguity funtion originally specified. Suppose that the latter, supposedly close to the ideal "thumb tack" ambiguity function, is defined in general terms as having a clear region around the origin. Using the usual Hamming measure (see, e.g., [5]) on the hit array, the distance from the origin to a hit in position (k, ℓ) is given by:

$$d_{k,\ell} = |k| + |\ell| . \tag{13}$$

Thus, a clear region of size D around the origin *in the hit array* is obtained for:

$$d_{k,\ell} \geq D , \qquad \text{for all } 1 \leq k, \ell \leq N-1. \tag{14}$$

Note that the value $D = 2(N-1)$, corresponding to the ideal "thumb tack" ambiguity function, is precluded from the hit array definition. Consequently, the desired, realizable ambiguity function may only have ideal characteristics for limited time and frequency shifts. However, in practice range and Doppler shifts are bounded by physical conditions [9], so this is not a severe restriction. Fig. 3 illustrates the requirement (14) in the case $D = 3$ by isolating the forbidden area in the hit array. Empty squares comprise all permissible positions for the $N^2 - N$ hits not yet placed. Even then, not all possible placements of the remaining hits lead to valid hit arrays, as properties mentioned earlier need to be satisfied. One solution may be in terms of a hit array generator, which would complete partially specified input hit arrays by automatically enforcing the relevant constraints.

In the simple example $N = 10$, the solution determined through the backtracking algorithm is the code array given in Fig. 4; in the terminology of [5], this code array has "non-attacking kings." The associated ambiguity function is shown in Fig. 5. Although the resolution characteristics are limited, one can definitely see a "clear" (more appropriately, a low-pedestal) region around the origin, and accordingly a distribution of higher peaks further away from the origin, each of these peaks corresponding to one hit in the hit array. We emphasize that the code array obtained in this example does not belong to any known class of algebraically constructed frequency hop patterns. The algorithm, however, can be used to construct any other patterns, such as Costas arrays of any desired length.

V. CONCLUSION

The hit array formalism, originally developed for frequency hop waveform analysis, can also be used for synthesis purposes. In essence, it is a suitable intermediate step for both the specification of a realizable ambiguity function and the retrieval of the associated signal itself.

A general backtracking-type algorithm has been derived to reconstruct patterns associated with a hit array. This represents the first efficient procedure for synthesizing a set of frequency hop waveforms directly from any given (realizable) hit array, and thus, because of the correspondence between hit array and ambiguity function, from any physically meaningful ambiguity function.

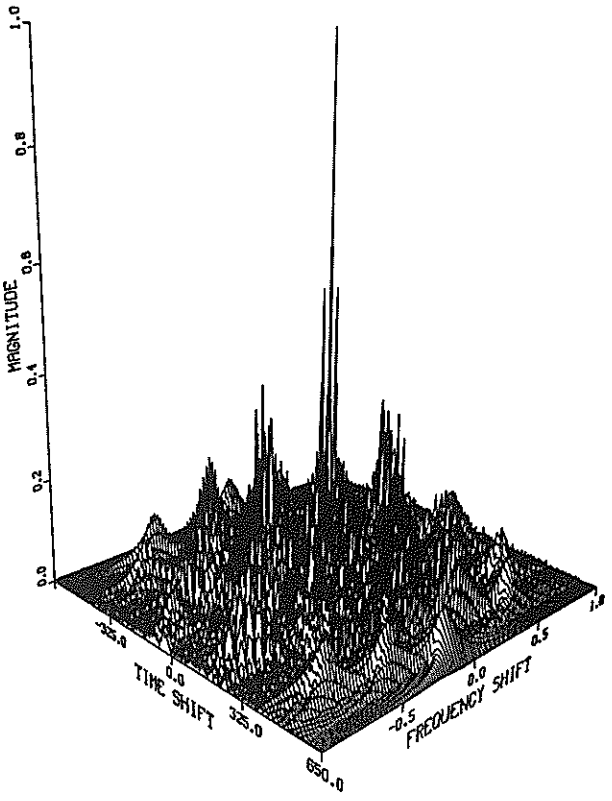


Fig. 5. Auto-Ambiguity Function Associated with the Code Array of Fig. 4.

REFERENCES

[1] H.L. Van Trees, *Detection, Estimation, and Modulation Theory — Part III*, New York: Wiley, 1971.

[2] W. Blau, "Synthesis of Ambiguity Functions for Prescribed Responses," *IEEE Trans. AES*, Vol. AES-3, pp. 656-663, 1967.

[3] E.L. Titlebaum and L.H. Sibul, "Time-Frequency Hop Signals Part II: Coding Based Upon Quadratic Congruences," *IEEE Trans.AES*, Vol. 17, pp. 494-499, July 1981.

[4] J.P. Costas, "A Study of a Class of Detection Waveforms Having Nearly Ideal Range-Doppler Ambiguity Properties," *Proc. IEEE*, Vol. 72, pp. 996-1009, August 1984.

[5] S.W. Golomb and H. Taylor, "Constructions and Properties of Costas Arrays," *Proc. IEEE*, Vol. 72, No. 9, pp. 1143-1163, September 1984.

[6] J.R. Bellegarda, "Time-Frequency Properties of Six Classes of Congruential Frequency Hop Signals," in *Proc. V EUSIPCO*, September 1990.

[7] J.R. Bellegarda and E.L. Titlebaum, "The Hit Array: An Analysis Formalism for Multiple Access Frequency Hop Signals," *IEEE Trans. AES*, to appear, January 1991.

[8] J.R. Bellegarda, S.V. Marić, and E.L. Titlebaum, "The Hit Array: A Synthesis Tool for Multiple Access Frequency Hop Signals," *IEEE Trans. AES*, submitted.

[9] W. Chang and K. Scarbrough, "Costas Arrays with Small Number of Cross-Coincidences," *IEEE Trans. AES*, Vol. AES-25, No. 1, January 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A PARALLEL AND PROGRAMMABLE ARCHITECTURE FOR RADAR SIGNAL PROCESSING

S. Bottalico*, L. Gabbani*, M. La Manna**

Selenia, Italy.

Computing structures oriented towards Radar Signal Processing mainly includes filtering algorithms (DFTs, FFTs, convolutions, etc.) involving hundreds of arithmetic operations per sample, sampling interval of input signal being of the order of 200 nsec. - 1 microsec. These throughput figures correspond to computing power requirements of up to some GIPS (Billion Fixed-point Instructions per Second)/GFLOPS (Billion Floating-point Operations per Second). These structures have to be based on multiprocessor machines with hundreds of processors, with relative problems in communication, sizing, power, etc. As a consequence, the use of special-purpose digital video processors has been generally preferred, where the filtering algorithms are executed by hardwired logic. In an attempt to overcome the drawbacks exhibited by hardwired processors, a feasibility study on a flexible Radar Signal Processor led to the definition of a Parallel and Programmable Architecture, based on a programmable and configurable array of processing elements (PEs), with hundreds of PEs connected together through fast communication links. In this paper this Parallel and Programmable Architecture is presented and a measure of its performance is given, dimensioning the architecture in a typical application. The results show the applicability of the architecture to the chosen case-study and the possibility to configurate the architecture depending on the specific application with the use of a limited set of boards based both on commercial components and on ASICs (Application Specific Integrated Circuits). Industrial advantages of this approach are also considered, as the reduction of costs in the design, development, production, storage, logistics and the possibility to introduce technological advances through the easy upgrading of the basic modules, without changing the basic architecture.

1. INTRODUCTION

Radar Signal Processing consists of a set of computations on the radar echoes to detect presence of a target and to estimate the target parameters (range, azimuth, velocity, etc.). The radar covered area (Fig. 1) is subdivided in M azimuthal sectors, each of which including N range-bins (a range-bin corresponds to half radar resolution). Each computation, relative to a range-bin, involves the echoes returning from the considered range-bin and from the neighbouring ones (each echo is represented as a complex sample).

This computation, which does not depend on the particular range-bin, mainly consists of linear non-recursive filtering. More precisely, for a given range-bin a filtering algorithm can involve quantities relative to range-bins belonging to the same sector or quantities relative to range-bins belonging to the same ring. Observe that the number of bits employed to represent the quantities to be processed is not constant during the overall computation. The dynamic range of input samples is usually covered by 12 bits, while the intermediate results require a decreasing number of bits to be represented, and tend, in the most simplified context, to become binary (presence or absence of target).

* Selenia s.p.a., Via Tiburtina Km. 12.4 - I00131 Roma, Italy

**Selenia s.p.a., Via S. Maria 83 - I56126 Pisa, Italy.

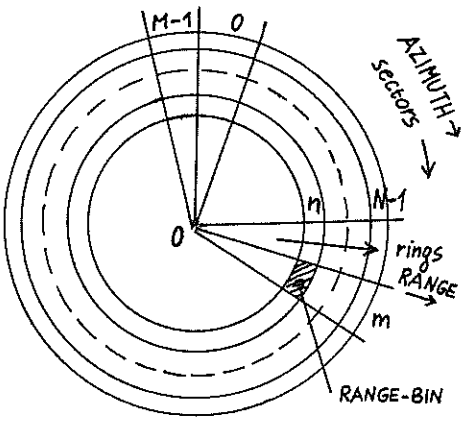


Fig. 1 Schema of the radar covered area

The echoes returning from the N range-bins of one sector are thought of as arranged in a vector of N elements, so that the signals to be processed consist of a series of vectors spaced by a time interval PRT (Pulse Repetition Time).

In order to perform radar signal processing, a Parallel and Programmable Architecture is proposed, whose main part is a two-dimensional array, of Q rows and P columns, of Processing Elements (PEs) (Fig. 2), each of which is connected to four adjacent PEs. At every time interval T, the PEs of a row perform the computations relative to S=N/Q elements of an input vector. The number P of columns of the array depends on the computational load. If this load is not too high, only one column may be used, otherwise a pipeline structure of P PEs per each row is used. The number Q of rows, the number P of columns and the

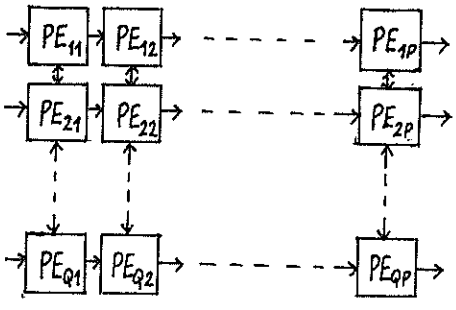


Fig. 2 Structure of the two-dimensional array

number S of vector elements that are processed by each PE are parameters to be properly defined for each specific radar. They depend on the number N of the vector elements, on the computational load to be done for each vector element and on the computational load which can be performed by each PE.

2. DESCRIPTION OF THE PARALLEL AND PROGRAMMABLE ARCHITECTURE

The Parallel and Programmable Architecture is based on a two-dimensional array of PEs. The PEs are connected as a matrix through high rate communication channels. The overall structure of the architecture is given in Fig. 3. It is composed of a two-dimensional array of PEs, an Input Formatter, an Output Formatter and an Array Control Unit.

The two-dimensional array of PEs constitutes the bulk of the architecture, where the fundamental signal processing operations (e.g. convolution, filtering, thresholding, etc.) are performed. Each PE is responsible for a part of the total computing task. In particular, the array is structured in Q rows and P columns. The computations relative to a set of S rings, corresponding to range-bins of different sectors, are performed by the PEs of the same row.

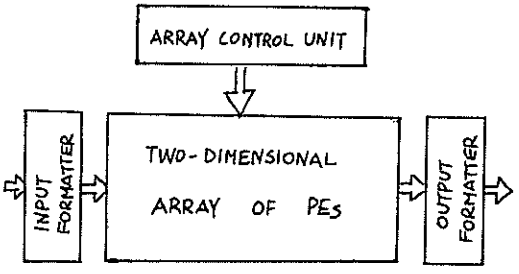


Fig. 3 The Parallel and Programmable Architecture

The Input Formatter receives the input data (raw video signals) from external equipment (e.g. A/D converter) and distributes the data into the array. In particular, if N is the total number of range-bins and Q is the number of rows of the array, the input data are subdivided into Q sets of S=N/Q elements. Each set of S elements, or range-bins, is sent to a row of the array, to be processed.

The Output Formatter receives the output data (processed video signals) from the PEs of the last column of the array and collects them into a single stream of data, to be processed by the Data Processing Computer and presented on a display.

The Input and Output Formatter are connected to the two-dimensional array through high rate communication channels. The communication protocol uses a Time Division Multiplexing and Addressing mode.

The Array Control Unit is a module, based on a conventional microprocessor, whose main functions are the array initialization and set-up, debugging and optional interface to a host computer. The Array Control Unit is connected to the PEs of the array through a dedicated channel at low rate.

2.1. The Two-Dimensional Array of PEs

Each Processing Element (PE) of the array is a computing unit with intensive arithmetic capability and with an access capability to a communication network.

Each PE (Fig. 4) is composed of:

- a DSP processor, TMS320C30 at 33 MHz, by Texas Instruments (3), featuring 16.6 MIPS and 33 MFLOPS;
- a Communication Controller based on a 1 micron 100K gates ASIC (Application Specific Integrated Circuit), which manages five bidirectional communication channels at 20 Mbytes/sec. each;
- a memory, shared between TMS320C30 and the Communication Controller, with two possible formats: 16Kx32 bits and 128Kx32 bits;
- a boot channel at 5 Mbytes/sec.

The five bidirectional channels (North, South, East, West, Extra) provided by the Communication Controller support the communications with neighbouring PEs and the Input and Output Formatters. In particular, the North and South channels support the communications between the PEs of the same columns and adjacent rows, while the East and West channels support the communications between the PEs of the same row and adjacent columns. The West channel of the PEs of the first column is connected to the Input Formatter, and the East channel of the PEs of the last column is connected to the Output Formatter.

The Extra channel may be used either to increase the throughput of a channel (e.g. to increase the input data rate of the West channel), or to interface specialized external devices (e.g. FFT processor).

2.2. The Input and Output Formatters

The Input and Output Formatters consist of special modules, whose main function is to interface the two-dimensional array with the radar equipment.

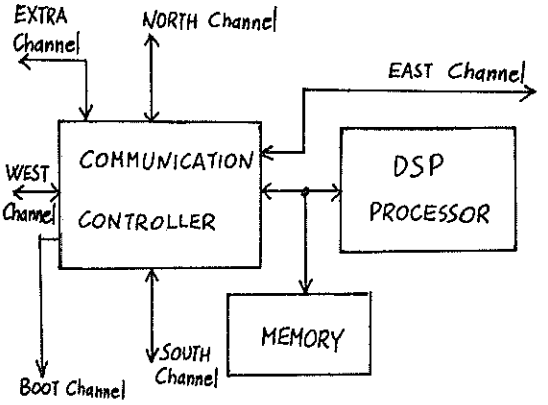


Fig. 4 Structure of the PE

Each module is constituted of a standard part, which interfaces the array, and a specialized part which interfaces the radar equipment.

The standard part of the Input and Output Formatters is constituted of:

- A programmable controller, which initializes the input/output parameters, i.e. defines the input/output data format to/from PEs, the number of bytes per transfer cycle, etc.;
- Two memory banks, alternatively connected to the Array Processor and to the external equipment;
- A programmable address generator, which defines, at every transfer cycle, the correct address of the data to be transferred from/to the memory banks to/from the array.

The specialized part of the Input Formatter is typically constituted of the A/D converters and associated input equipment, while the specialized part of the Output Formatter contains the interfaces to the Radar Data Processor.

2.3. The Array Control Unit

The Array Control Unit supervises the initialization and configuration of the Array Processor (e.g. program loading through the boot channel) and performs tests and diagnostics. It can be used either in stand-alone mode or as a slave unit of a general purpose host computer (e.g. VAX).

3. IMPLEMENTATIVE DETAILS

The feasibility study on the Parallel and Programmable Architecture has shown that a possible implementation includes the following boards:

- Processor Board, in two formats: (a) 4 PEs each with 128Kx32 bits; (b) 8 PEs each with 16Kx32 bits. The first format is used in applications with high memory requirements, while the second format is used in applications with high computing power requirements.
- Input Formatter Board. This board consists of a standard interface to the array and a non-standard interface to the system. It allows the transfer of data from the system to the array.
- Output Formatter Board. This board consists of a standard interface to the array and a non-standard interface to the system. It allows the transfer of data from the array to the system.
- Array Control Unit Board. This board, based on a conventional microprocessor, allows to use the architecture both in a stand-alone configuration and slaved to a host computer, such as VAX.

The boards are of Double Europe format, and are supposed to use the SMD (Surface Mounting Devices) technology. The maximum weight allowed per board is 1 kg., and the maximum power consumption per board is 60 Watts. The maximum number of boards per rack is 18.

4. MAPPING THE APPLICATION ON THE ARCHITECTURE

For sake of clearness, we will consider machines with a single column of PEs (P=1), but the present methodology could be easily extended to the two-dimensional case. Observe that a single column Array Processor can be used when the total time to perform all the computations on a single range-bin is less than the PRT, as it is in the majority of cases.

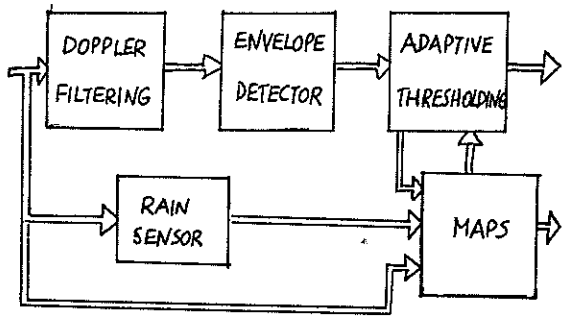


Fig. 5 Signal Processing for Air Traffic Control Radar

A radar signal processor generally includes one or more processing channels with low mutual interactions. Each channel is dimensioned separately and generally one channel includes an Input Formatter Board, a number of Processor Boards, an Output Formatter Board and an Array Control Unit Board.

For each channel, the dimensioning process includes 3 steps:

- Input-output balance: the input/output capacity of each machine has to be large enough to support the input/output data rate. As an example, using 2 input channels, the West and the Extra channels (40 Mbytes/sec. total), up to 8 bytes every 200 nsec. can be taken as input data;
- Computing power balance: each PE works on Nrb=PRT/Trb range-bins, where PRT is the Pulse Repetition Time and Trb is the time taken to perform all the computations on a single range-bin. For algorithms working on groups og range-bins, the time Trb is calculated dividing the total time by the number of range-bins. As N=PRT/ST (ST is the sampling time) is the total number of range-bins, the total number of PEs is NPE=N/Nrb=Trb/ST;
- Memory balance: the total amount of memory required per PE is Nrb times the memory required per range-bin, plus a small amount needed as working memory and for the code. If this total exceeds the maximum (128Kx32 bits), the number of range-bins per PE has to be reduced, in order to have less memory

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

LOW COMPLEXITY A/D–CONVERSION AND PREPROCESSING FOR DIGITAL PHASED ARRAYS

W. STAMMLER, A. ELTERICH

TELEFUNKEN SYSTEMTECHNIK GmbH
Sedanstrasse 10, 7900 Ulm / West Germany

In this paper a concept for a low complexity digital phased array receiver is presented. Basic features are low resolution A/D–conversion in the IF–range, significant oversampling, efficient quadrature demodulation and beamforming in the baseband. Theoretical as well as simulation results are given to prove the tolerability of distortions due to 1 to 3 Bit Quantization.

1. INTRODUCTION

Digital phased array (DPA)–systems have been in the state of euphoric discussion for a long time [2]. Recently commercial interest has increased again e.g. for satellite navigation with DPA–receivers, installed on vehicles. So far, however, practical implementations still suffer from extremely high complexity and ultimately unattractive high cost [1]. To overcome part of these difficulties a concept is presented and evaluated for low complexity A/D–conversion and digital preprocessing of a DPA–receiver. Here the term "digital phased array" is not used in the sense of analog beamforming with digital control. Instead DPA stands for a fully digital array with digital preprocessing, digital beamforming and digital control.

The array is designed

- to perform beamforming in the baseband (Figure 1). This offers low cost multibeam extension and use of adaptive digital beamforming techniques.
- to employ digital demodulation and filtering of the IF–signal in order to minimize channel mismatching and signal distortions due to analogue components.
- to use monolithic integration for the digital components and as far as possible for the analog frontend as well (e.g. Monolithic Millimeter Wave Integrated Circuits) in order to end up with compact and low cost modules for each channel.

Figure 1 shows the structure of the DPA receiver with N antenna elements and baseband beamforming. It is a "narrowband" array in the sense, that the beamforming is performed via phase shift instead of time delay. This imposes the restriction

(1) $f_0/B \gg N$

on the ratio of carrier frequency fo and signal bandwidth B. The above condition is derived from the requirement, that the plane wave traveltime across an array of N elements with half wavelength spacing ought to be small compared to the transient times resulting from signal modulation.

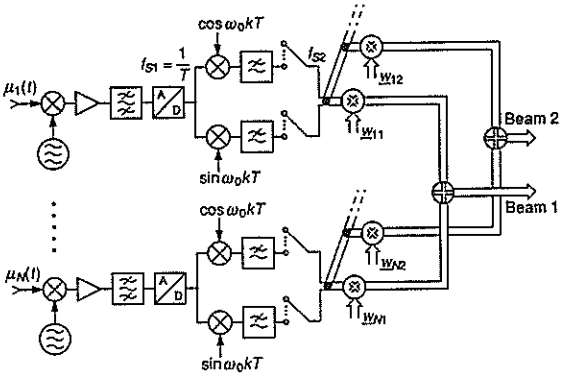


Figure 1: Structure of the digital phased array

The main point to be considered here is how to employ A/D–conversion and digital preprocessing at the earliest stage without causing unattractively high processing loads or tremendous signal distortions.

2. THE CONCEPT

Our solution may be characterized as follows:

- A/D–conversion of the IF–signal with 1 to 3 Bits, favourably 2 or 3 Bits, based on the fact, that typically SNR–values at the A/D–converter are below zero dB. As a consequence of low resolution, the hardware effort for subsequent signal processing can be kept at an extremely low level. The signal distortions obtained are small enough to live with (see next section).
- Significant oversampling (sampling frequency f_{S1} = 10 · B) to achieve both low degree of the antialiasing filter (and hence low differences for the various channels) and high processing–gain by subsequent digital filtering. Especially the antialiasing filter as the analog component with the highest selectivity demands encountered, turns

out to be a critical component. Nonideal filter characteristics are tolerable as long as they are identical for all channels – a demand, which can be fulfilled easily for large f_m/B ratios (f_m–intermediate frequency), not however for our case of low IF frequencies and rather high bandwith B ($f_m/B = 1...3$). Calculations for a first order lowpass filter (3 dB–corner frequency at $f_m + B/2$, $f_m/B = 2.5$) indicate, that ± 10 % tolerance in corner frequency accounts for 1.5 degrees of phase tolerance within the band of interest. Thus oversampling is an absolute necessity.

– Appropriate selection of the decimation rate V2 and of the rate V1 of sampling frequency and intermediate frequency in order to simplify quadrature demodulation as well as the combination with FIR–filtering and subsampling. This leads to

(2) $V1 = f_{S1}/f_m = |\, n / (1 - i \cdot n) \,|$
$V2 = f_{S1}/f_{S2} = m \cdot n$
with m = 1, 2, 3 ...; i = 0, 1, 2, 3, ...;
n–rational number > 2.

With V1 = 4 (i = 0, n = 4) and m = 2 no multiplications in demodulation are required and the filtering process can be simplified.

The gain in signal to noise ratio due to the whole processing chain from ADC to beamformer is given by the following relation:

$$(3)\quad G[dB] = \frac{SNR\ after\ beamforming}{SNR\ at\ ADC\ input}$$

$$= -LQU + GFIL - LWIN + 10 \cdot logN$$

with LQU–Quantization loss (see next section)
GFIL – Gain by Lowpass–Filtering, (approximately $10 \cdot \log V2$)
LWIN – Loss due to windowing during beamforming.

3. QUANTIZATION EFFECTS

3.1 Theoretical Model

First of all a statistical model is designed to describe quantization effects not only for 1 Bit (see [3]). The model according to Figure 2 assumes identical signals $s(k) = s(k \cdot T)$ on all channels, i.e. a plan wave, which strikes the array perpendicularly. For the noise (primarily thermal noise) a Gaussian probability density function (pdf) $p_\eta(u)$ is chosen. As a result, the noise–corrupted input signals $x_i(k \cdot T)$ may be considered representations of a normally distributed stochastic process $\eta(k)$ with expected value s(k) and variance 1.0. Uniform quantization is described as a transformation with a symmetric nonlinear function $f(\eta) = \xi$. The resulting discrete process ξ is characterized by its probability distribution

$$(4)\quad P\{m\} = \frac{1}{\sqrt{2\pi}} \int_{L_e}^{L_u} e^{-(u-s)^2/2} du.$$

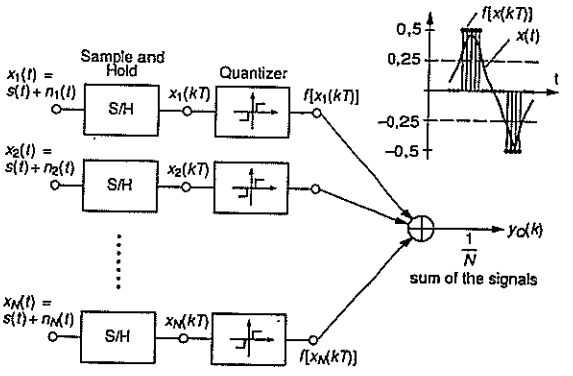


Figure 2: Statistical model to describe the effect of quantization

For further calculation we restrict ourselves to M being an odd number of quantization states ("representatives"). Then with L = (M–1)/2

$$(5)\quad L_e = \begin{cases} (m-0.5)a & \text{for } -L < m < L \\ -\infty & \text{for } -L = m \end{cases}$$

$$L_u = \begin{cases} +\infty & \text{for } L = m \\ (m+0.5)a & \text{for } -L < m < L \end{cases}$$

are the boundaries of the quantization intervals, where a stands for the quantizer stepsize. Now the expected value $<f(\eta)>$ is given by

$$(6)\quad y = <f(\eta)> = a \cdot \sum_{m=-L}^{L} P\{m\} \cdot m.$$

Using the integral notation of P{m} from equ. (4) and expanding the integrand in a power series of s, we obtain after integration

$$(7)\quad y = 2 \cdot a \cdot \sum_{n=0}^{\infty} \frac{s^{2n+1}}{(2n+1)!} \cdot \sum_{m=0}^{L-1} p_\eta^{(2m)}(a \cdot m + a/2).$$

Rewriting it into a power series of s

$$(8)\quad y = c_1 \cdot s + c_3 \cdot s^3 + c_5 \cdot s^5 + \ldots$$

shows, that the nonlinear distortion due to quantization can evaluated rather simply for any type of signal s(k) or signal combination by applying equation (8).

Now an appropriate value for the quantizer stepsize a needs to be chosen. With optimum uniform quantization (MSE–criterion) for a Gaussian process, the parameter a results as a solution of the following nonlinear equation

$$(9)\quad <f^2(\eta)> = c_1 \qquad or$$

$$2a^2 \sum_{m=1}^{L} P\{m\} \cdot m^2 = 2 \cdot a \sum_{m=0}^{L-1} p_\eta(a \cdot m + a/2).$$

With the optimum values of parameter a for given M the coefficients c_i are explicitely known (equ. 7). Now the SNR–loss LQU due to quantization may be specified. It turns out, that

$$LQU = 10 \cdot \log c_1 \tag{10}$$

holds, provided that a small SNR (< 0 dB) is assumed. The above equations (4) to (10), which were derived for odd values of M are applicable to even values of M with slight modifications.

3.2 Numerical Evaluations

First of all the SNR–loss LQU as a function of the number of representatives M is depicted in Figure 3. Based on equ. (10) and the underlying assumption of rather small SNR, it becomes evident, that sign quantization implies a loss of 1.96 dB, whereas an additional representative will reduce the loss to 0.9 dB. With optimum uniform 3 Bit quantization (M = 8) the loss is below 0.2 dB.

To check for the signal distortions according to equation (8), the following table contains an excerpt of the coefficients c_i, assuming optimum quantization for a zero mean Gaussian process

M	a	c_1	c_3	c_5
2	1.596	0.637	–0.1061	0.01592
3	1.224	0.810	–0.0842	0.00603
4	0.9957	0.881	–0.0669	0.00200
8	0.5860	0.963	–0.0327	–0.00224

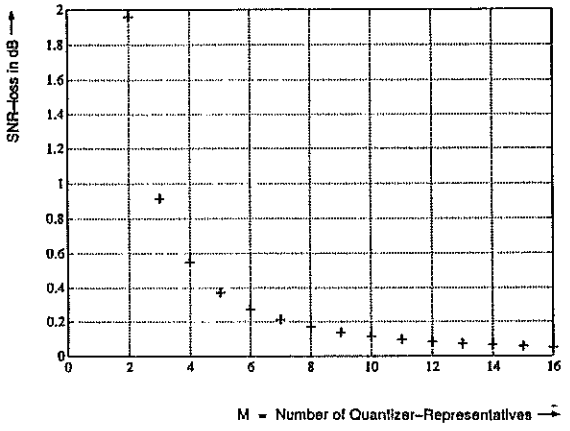


Figure 3: SNR–loss due to quantization with M representatives for the case of low input SNR (< 0 dB)

Signal distortion curves y(s) for M = 3, 8 and 256 are illustrated in Figure 4, showing the trend towards y(s) = s with increased resolution.

In phased array configurations the nonlinear terms in y(s) are responsible for "ghost"–signals, appearing at different azimut (or elevation) position, thus pretending e.g. additional targets in Radar applications. The exact knowledge of their amplitude is crucial for system design and analysis. Figure 5 shows the ratio of the mainlobe (desired signal) and the maximum sidelobe (ghost signal) resulting from quantization with 3 representatives as a function of input SNR. From equ. (8) a third–, fifth–and fifteenth order expansion was used for these computations. It becomes evident, that the ghost signals are less significant, the smaller an input SNR is given.

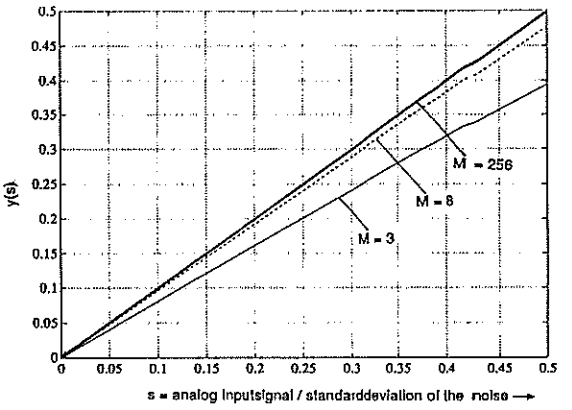


Figure 4: Signal distortion $y = c_1 \cdot s + c_3 \cdot s^3 + c_5 \cdot s^5 + \ldots$ due to quantization for M = 3, 8 and 256

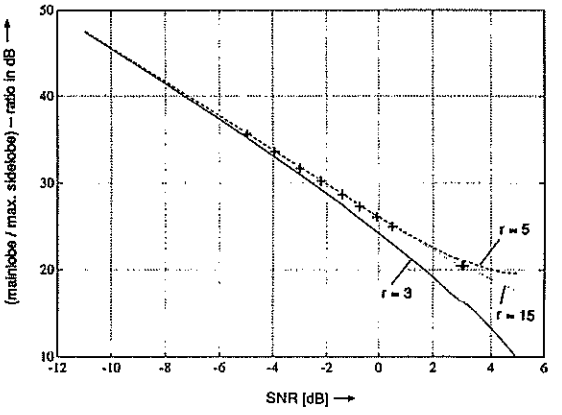


Figure 5: Ratio of mainlobe and maximal sidelobe as a function of SNR assuming sinusoidal input signal, M = 3 and third–, fifth–, and fifteenth–order power series expansion according to equation (8). Points marked by + result from numerical simulations

To check the theoretical results from section 3.1 discrete numerical simulations were performed. Figure 6 illustrates results for M = 3 and SNR of –15 dB at the input of each quantizer. Summing up N = 200 channels according to Figure 2, the SNR observed for quantization with 3 representatives corresponds to the theoretical results (Figure 3) qualitatively and quantitatively. Finally the amplitude of the "ghost"–signal is determined by simulation (Figure 7).

This implies integration for each beam in order to reduce the noise to a sufficiently low level. Figure 7 shows the result, conciding well with the theoretical results from Figure 5.

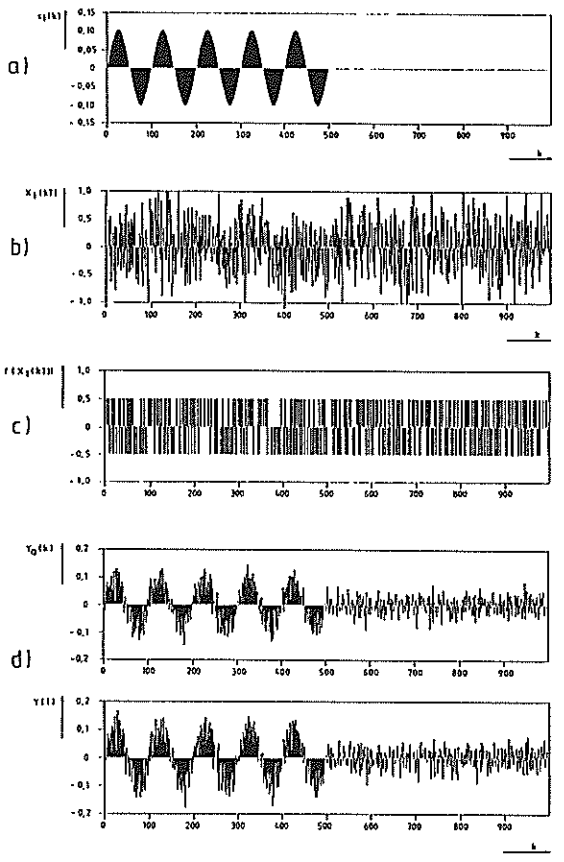


Figure 6: Numerical simulation of the structure from Figure 2
- a. Sinusoidal signal with 100 taps per period (identical for each channel)
- b. Noise corrupted signal (SNR = –15 dB, normal distribution of noise, stat. independent noise from channel to channel)
- c. Noise corrupted signal after quantization (M = 3)
- d. Signals after summation over 200 channels
  bottom: without quantization
  top: with quantization (see c)

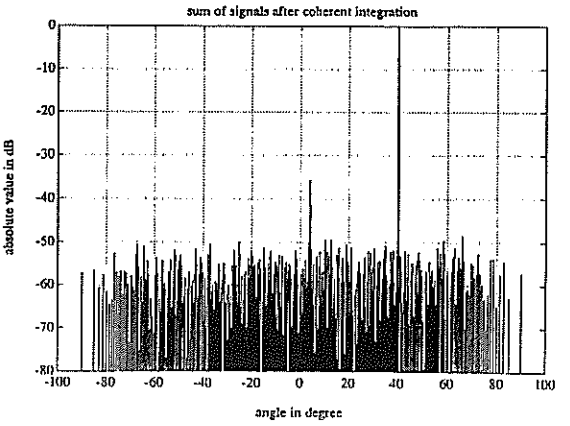


Figure 7: Result of simulation for quantization and beamforming, showing the amplitude of the ghost signal in relation of the desired signal (0 dB)
(512 antenna elements; FFT for beamforming; Target from 40.13 degrees; SNR befor quantization = –6 dB)

ACKNOWLEDGEMENT

This work was supported by the Federal German Government. This is gratefully acknowledged here.

REFERENCES

[1] Steyskal H.: Digital beamforming antennas – An introduction,
Microwave Journal, january 1987,
pp. 107–124.

[2] Barton P.: Digital beamforming for Radar,
IEE Proceedings, Vol. 127, Part F, no.4, August 1980,
pp. 266–277.

[3] Wong A. C. C.: Radar digital beamforming,
Military Microwaves Conf. 1982,
pp. 287–294.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SIGNAL PROCESSING FOR RADAR TARGET ANALYSIS

F. CHRISTOPHE, A. BERGES , P. BORDERIES, A. SARREMEJEAN

O.N.E.R.A.-C.E.R.T. - Microwaves Department
B.P. 4025 - 31055 Toulouse Cedex - France

I - INTRODUCTION

Radar cross section analysis is of increased importance either for tailoring of modern radars to their targets and environment (i.e. clutter), or for designing stealth vehicles.

Equipment is currently operated for such purpose in the field, in large static facilities, or in smaller anechoïc chambers with scale targets.

The signal processing algorithms and hardware associated to the radar sensor have to fulfill the following requirements :

- high spatial resolution in either 1, 2 or 3 dimensions resulting from real, synthetic or inverse synthetic aperture processing, and pulse compression ;

- high accuracy and sensitivity, related to specific calibration procedures ;

- reasonable computing time and adequate display of the results.

This presentation will mainly focus on the methodology and signal processing used for indoor target analysis : some specific requirements for accurate results will be first reviewed ; the now classical inverse synthetic aperture radar (ISAR) imaging through rotating the target on a turntable will be then shortly described, and we will finally emphasize a light experimental set up for fix target imaging through synthetic aperture radar (SAR).

II - REQUIREMENTS FOR ACCURATE BACKSCATTERING MEASUREMENTS

II.1. - Reduction of the effects of the environment

The target is supposed to be located in the quiet zone of an anechoïc chamber, illuminated by a plane wave.

At frequency f, a continuous wave (CW) coherent radar receiver measures the complex signal $M(f) = G(f)\left[X(f) + C(f)\right]$ (1)

where : . X(f) is the backscattering coefficient of the target to be determined,

. C(f) represents the coupling beetween the transmitter and receiver ports due to the antenna(s) and the whole environment, including the backscattering by the anechoïc chamber itself,

. G(f) is a gain factor due to the amplifiers and antennas.

Subsequent measurements $L(f) = G(f)\ C(f)$ of the signal produced by an empty chamber and $N(f) = G(f)\left[A(f) + C(f)\right]$ produced by a calibrating target of known backscattering coefficient A(f) allows to eliminate unknown G(f) and C(f) :

$$X(f) = A(f)\ \frac{M(f) - L(f)}{N(f) - L(f)} \qquad (2)$$

under the assumptions of stability of G and C during the whole sequence, and of validity of equation (1), i.e. excluding multiple scattering beetween the target and its environment.

Stability of G is mainly achieved through thermal control of the amplifiers, and of C through mechanical means. Since the physical locations of the main contributors to the coupling C are the antenna(s) and the back wall of the anechoïc chamber, a substantial reduction of C may be reached by a range gating, either by generating and receiving after the proper delay an impulse signal, or by synthetic pulse generation and digital signal processins. Successive CW signals stepped df from each other are then transmitted with overall frequency shift Df ; the Fourier transform gives an unambiguous time response over an interval of 1/df which has to be larger than the double transit time from the antenna to the backwall of the anechoïc chamber. After multiplying this time response by a window centered on the target, an inverse Fourier transform results in a frequency response of the target cleared from most of the spurious echoes.

The interest of the very high time resolution 1/Df which can be obtained by such signal generation and processing will be later discussed.

II.2. - Polarimetric calibration

Calibration problems are more difficult when the backscattering matrix is to be measured. When working in linear polarisations, 3-bounce corner reflectors and spheres are adequate calibrators for copolarized backscattering. The crosspolarized terms may be calibrated with a 45 degree tilted dihedral corner reflector, but this latter needs a careful orientation. We have successfully used such dihedral reflector as single calibrator for the 4 complex terms of the backscattering matrix, thanks to high precision positioners allowing for the search of a maximum of the backscattered signal.

III - ISAR Imaging

2D ISAR imaging is obtained by coherently summing synthetic impulse responses at azimuth angles stepped $d\theta$ over the angular sector $D\theta$. The measurements result in a complex 2 dimensional array $X(f,\theta)$. Fourier transform of X versus f gives the depth response (axis x) and versus θ the transverse response (axis y). Theoretical developments of such processing are given in ref(1) and (2).

Practically, we must keep in mind that the resolution is :

$$dy = \frac{c}{2\,Df} \qquad dx = \frac{\lambda}{2\,D\theta}$$

while the non-ambiguous zone in characterized by

$$Dy = \frac{c}{2\,df} \qquad Dx = \frac{\lambda}{2\,d\theta}$$

Weighting may be used in both dimensions, with poorer resolution. We must note that in transverse dimension exists a natural weighting due to the narrow directivity of most diffracting points. Besides, without improving physical resolution, we interpolate by using zero-padding in range FFT and increasing the number of calculation points in the azimuth integration.

This method is set up at CERT for several years, and used in anechoïc chamber to characterize small targets or reduced scale models. Working frequency bands are 8-18 GHz, 26-40 GHz and 90-100 GHz, so that resolution is few centimeters. Targets are located on a continuously rotating platform which activates analog to digital conversion of the received signal every $d\theta$ degrees and turns over 360° for each frequency. Calibration is as previously described.

As examples of such measurements fig. (1) shows the image of a model of a Vautour jet fighter measured at 35 GHz, reconstructed by incoherently summing over 360°, the 12 sectorial coherent images over 30°.

IV - SAR imaging

IV.1. - Principles

Synthetic aperture radars are currently used for long range imaging of the ground from aircrafts or satellites. At short range, they may be adapted to the needs of RCS analysis of targets for which ISAR is unrealistic.

It may be the case for analysing the coupling of real targets with their environment, or for economic reasons since light antennas are easier to move with millimetric accuracy than heavy targets.

The transverse resolution for a system of moving antennas on a line of given length L imaging at range R, at wavelength λ, is given by $dy = \lambda R/2L$, and the sampling increment for avoiding ambiguities with omnidirectional antennas is $dL = \lambda/2$. These relations may be obviously connected with those established for ISAR, by replacing the azimuthal sector $D\theta$ by L/R, and the azimuthal increment $d\theta$ by dL/R.

A light experiment has been set up at CERT for demonstrating the feasibility of such applications.

IV.2. - Experimental set up

The step frequency generator and the coherent receiver available from a commercial network analyser were connected to 2 small pyramidal horn antennas, supported by a translating trolley under computer control. The desktop computer controlled also the signal generator, and stored the received signal in phase and quadrature form. Off line processing was, as usual for SAR, a combination of pulse compression through FFT and of focussed azimuth compression through correlation ; a Hanning window was used in both dimensions.

The system was operated at X band, with a bandwidth of 2 GHz and the trolley moved over 1 m, with resulting theoretical resolutions of 15 cm (range) by 10cm (cross-range) at range 5 m.

An example of data obtained when imaging a scene of 3 trihedral reflectors in an anechoïc chamber is given fig. 2. The location of each scatterer and its level are correct ; resolution and side lobes are close to the anticipated values.

V - CONCLUSIONS

Since SAR and ISAR imaging principles are similar, performances of both may be closely related and systems built according to them for RCS analysis make use of same signal processing schemes. But such systems may widely separate from the mechanical point of view, and their complementarity should then be kept in mind.

REFERENCES

(1) C. POUIT "Imagerie radar à grande bande passante" - Colloque international sur le Radar - Paris, déc. 78.
(2) D. MENSA "High resolution radar imaging" - Artech House.

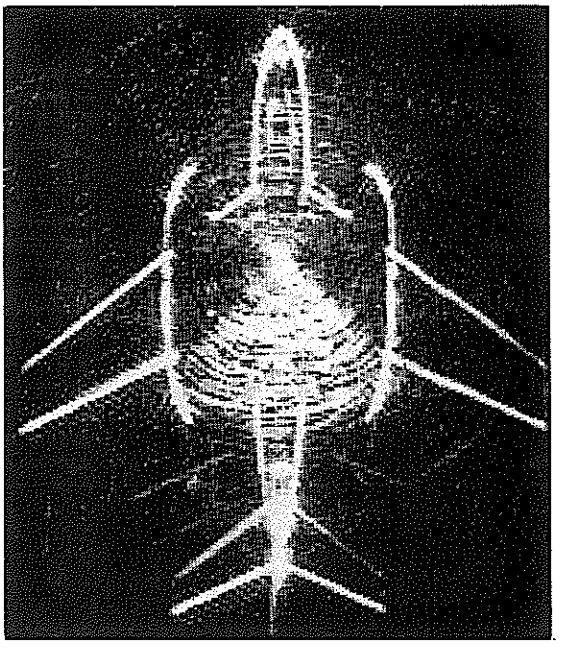


Figure 1 : Image reconstructed from ISAR processing of a 1/10 scale model of Vautour around 35 GHz with 10 GHz bandwidth.

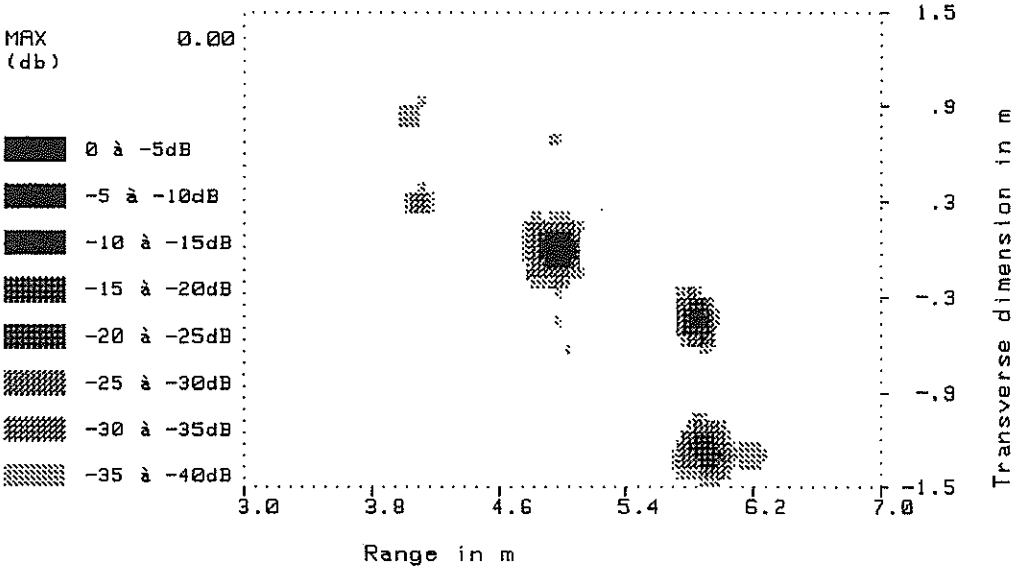


Figure 2 : Image obtained from SAR measurement at 10 GHz of 3 corner reflectors in anechoïc chamber - Bandwith 1.5 GHz, antenna displacement 1 m.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

AN AIRBORNE PULSE DOPPLER RADAR MODEL

J. MARTIN AND B. MULGREW

Department of Electrical Engineering, The King's Buildings, University of Edinburgh, Mayfield Road, Edinburgh, EH9 3JL, Scotland, U.K.

This paper will describe an airborne pulse Doppler radar model which simulates the radar return signal from airborne targets, with particular attention paid to the frequency spectrum of the return signal. The model simulates the return signal from the airframe and rotor or propeller blades of the target. The return signal also includes noise and ground clutter.

1. INTRODUCTION

Over the years various airborne pulse Doppler radar models have appeared in the literature, e.g. [1]-[4]. Most of these have attempted to model the return signal from ground clutter, not that from airborne targets.

Pulse Doppler radars are used mainly in airborne applications which require the detection of moving targets in a ground clutter environment. In pulse Doppler radar, the radar signal is transmitted as a series of coherent pulses. From the return signal the radar is able to calculate the range and Doppler frequency shift of the target.

Although the model is described as an airborne pulse Doppler radar model, in practice it can also be used as a ground-based pulse Doppler radar model, in which case there will be no ground clutter.

The velocity of the airframe will tend to cause a Doppler frequency shift of the return signal. The rotation of the rotor or propeller blades will tend to cause a modulation of the return signal. The modulation is a form of frequency modulation and results in a number of sidebands about the airframe line of the target.

The model is useful for two main reasons. First, target acquisition and tracking is sometimes achieved using the Doppler frequency shift of the airframe of the target [5]. At the development stage of acquisition and tracking algorithms it would obviously be much less expensive to test the algorithms on simulated data, than to pay for flight trials in order to test them on real data. Second, target classification and identification is sometimes achieved using the sidebands of the target [6]. As with acquisition and tracking algorithms, it would obviously be much less expensive to test any classification and identification algorithms on simulated data, than on real data.

2. MODEL STRUCTURE

Figure 1 shows the basic structure of the model. The radar signal is transmitted by the radar and travels towards the target. The signal is then reflected by the airframe and rotor or propeller blades of the target and travels back towards the radar. At some antenna angles the signal is also reflected by the ground and this also travels back towards the radar. Zero-mean white Gaussian noise is then added to the signal. The return signal is then received by the radar, and after pre-processing is down-converted into real and imaginary components. The fast Fourier transform (FFT) of the signal is then taken. Finally, the time series and frequency spectrum of the signal are

3. AIRFRAME

A model which describes the theoretical return signal from an airframe is given by

$$v_a(t) = A_a e^{j(\omega_c t - \frac{4\pi}{\lambda}(R_a + v_a t))} \quad (1)$$

where A_a = a scale factor,
R_a = range of the airframe,
t = time,
v_a = radial velocity of the airframe with respect to the radar,
λ = wavelength of the transmitted signal,
ω_c = radian frequency of the transmitted signal.

The Fourier transform of Equation (1) is given by

$$V_a(f) = A'_a \delta(f - f_c - f_{d_a}) \quad (2)$$

where $A'_a = A_a e^{-j\frac{4\pi R_a}{\lambda}}$,
f = cyclic frequency,
f_c = cyclic frequency of the transmitted signal,
$f_{d_a} = -2v_a/\lambda$ = cyclic Doppler frequency shift of the airframe,
$\delta(f)$ = unit impulse function.

In the theoretical case, the airframe will consist of a single scatterer, having a single cross section, range and velocity. In the practical case, however, the airframe will consist of many scatterers, each having random variations of its cross section, range and velocity.

The cross section will have a Rayleigh or chi-square probability density function (pdf), corresponding to Swerling's four fluctuation models, and the range and velocity will have a Gaussian pdf. In practice these pdfs will not always correspond to the pdfs of practical targets, however in many cases they will be a reasonable approximation.

The random variations of the cross section, range, and velocity are mainly caused by random variations of the position and orientation of the airframe. The position and orientation of the airframe can be resolved into six degrees of freedom: roll, pitch and yaw, i.e. rotation about three orthogonal axes; and surge, sway and heave, i.e. translation along the three axes, respectively.

In the theoretical case, the airframe line will consist of a single spectral line. In the practical case, the random variations of the velocity will cause the airframe line to have a Doppler frequency spread. The variance of the spread depends mainly on the variance of the velocity and the wavelength of the transmitted signal.

4. ENGINES

A model which describes the theoretical return signal from the blades of an aircraft propeller is given by

$$v_b(t)=\sum_{n=0}^{N-1}A_b(L_2-L_1) \tag{3}$$
$$\cdot e^{j(\omega_c t-\frac{4\pi}{\lambda}(R_b+v_b t+\frac{L_1+L_2}{2}\cos(0)\sin(\omega_r t+\frac{2\pi n}{N})))}$$
$$\cdot\mathrm{sinc}(\frac{4\pi}{\lambda}\frac{(L_2-L_1)}{2}\cos(\theta)\sin(\omega_r t+\frac{2\pi n}{N}))$$

where A_b = a scale factor,
L_1 = distance of the blade roots from the centre of rotation,
L_2 = distance of the blade tips from the centre of rotation,
N = number of blades,
R_b = range of the centre of rotation,
v_b = radial velocity of the centre of rotation with respect to the radar,
θ = angle between the plane of rotation and the line of sight from the radar to the centre of rotation,
ω_r = radian frequency of rotation.

A number of assumptions are made in the derivation of Equation (1):

1. Each blade acts as a homogeneous, linear, rigid antenna.
2. Each blade is always visible to the radar, i.e. there is no shielding of the blades.
3. The rotor or propeller is in the far field of the antenna.

The Fourier transform of Equation (3) is given by

$$V_b(f)=\sum_{k=-N_1}^{N_1}c_{Nk}\delta(f-f_c-f_{d_b}-Nkf_r) \tag{4}$$

where $c_{Nk}=\sum_{l=0}^{\infty}\frac{2(-1)^{Nku(0)}A_bN}{\frac{4\pi}{\lambda}\cos(\theta)}e^{-j\frac{4\pi R_b}{\lambda}}$
$$\cdot(J_{|Nk|+2l+1}(\frac{4\pi}{\lambda}L_2\cos(\theta))-J_{|Nk|+2l+1}(\frac{4\pi}{\lambda}L_1\cos(\theta))),$$

$f_{d_b}=-2v_b/\lambda$ = cyclic Doppler frequency shift of the centre of rotation,
f_r = cyclic frequency of rotation,
$J_k(\cdot)$ = Bessel function of the 1st kind and kth order,
N_1 = highest significant sideband,
$u(k)$ = unit step function.

When $L_1=0$, Equation (3) describes the return signal from a helicopter rotor and Equation (4) describes a frequency modulated signal with N_1 pairs of sidebands about the airframe line, each separated by Nf_r. When $L_1\neq 0$, Equation (3) describes the return signal from an aircraft propeller and Equation (4) still describes a frequency modulated signal, however, depending on L_1, the sidebands nearest to the airframe line will be approximately zero, resulting in a frequency notch about the airframe line. The greater the value of L_1, the greater the notch will be. In general:
When $L_1=0$

$$N'=\frac{8\pi L_2\cos(\theta)}{N\lambda} \tag{5}$$
$$\Delta f=Nf_r \tag{6}$$
$$B=\frac{8\pi f_r L_2\cos(\theta)}{\lambda} \tag{7}$$

When $L_1\neq 0$

$$N'=\frac{8\pi(L_2-L_1)\cos(\theta)}{N\lambda} \tag{8}$$
$$B=\frac{8\pi f_r(L_2-L_1)\cos(\theta)}{\lambda} \tag{9}$$

where N' = number of significant sidebands,
Δf = frequency separation of each sideband,
B = bandwidth of the sidebands.

As with the airframe, in the theoretical case, each incremental chord-wise section of blade will consist of a single scatterer, having a single cross section, range and velocity. In the practical case, each section of blade will consist of many scatterers, each having random variations of its cross section, range and velocity.

In the theoretical case, each sideband will consist of a single spectral line. In the practical case, the random variations of the velocity will cause each sideband to have a Doppler frequency spread.

It should be noted that some rotating objects will not tend to cause a modulation of the return signal. In general, a rotating object will not tend to cause a modulation of the signal if it is a body of revolution about its axis of rotation, e.g. a flat disc, a sphere, etc. This is why there is usually a frequency notch about the airframe line in the case of propeller aircraft - because the propeller spinner (a streamlined faring which fits over the propeller hub) is a body of revolution about its axis of rotation.

In the case of propeller aircraft, the propeller blades can have significant blade pitch. This will tend to cause a periodic variation of the cross section of each incremental chord-wise section of blade, which will tend to cause a second type of modulation of the return signal. The modulation is a form of amplitude modulation and results in the attenuation of the upper or lower sidebands, depending on the aspect angle.

5. CLUTTER

At some antenna angles the return signal will include ground clutter. Various clutter types are included in the model, using various pdfs (e.g. exponential, Gaussian, log-normal, Rayleigh and Weibull). In each case the clutter is uncorrelated from pulse to pulse, although in practice the clutter is sometimes correlated.

6. SIMULATION RESULTS

Figure 2 shows the frequency spectrum of a helicopter aircraft in which $N=4$, $L_2=5$ m, $f_r=6$ Hz, $\lambda=2$ m, and $\theta=0$ rad., and in which the airframe and rotor blades each consist of a single scatterer (i.e. $N_a=1$ and $L_b=1$), each having a single cross section, range and velocity.

It can be seen that $N'=16$, $\Delta f=24$ Hz, and $B=384$ Hz. Note that the airframe line and sidebands each consist of a single spectral line, and that the frequency spectrum is approximately symmetrical about the airframe line.

Figures 3-5 show the frequency spectrum of the same target as the number of scatterers in the airframe and rotor blades are each increased to 8, 64, 512 and 4096, respectively. In each case, the cross section has a Rayleigh pdf, corresponding to a Swerling Case 2 model, and the range and velocity have a Gaussian pdf.

It can be seen that, for a small number of scatterers, the frequency spectrum is unrealistically noisy, and that as the number of scatterers is increased, the frequency spectrum becomes similar to that for practical targets.

Figure 6 shows the frequency spectrum of a propeller aircraft in which $N=4$, $L_1=0.25$ m, $L_2=1$ m, $f_r=40$ Hz, $\lambda=0.20$ m, and $\theta=\pi/4$ rad., and in which the airframe and propeller blades each consist of 4096 scatterers.

It can be seen that $N'=9$, $\Delta f=160$ Hz, and $B=1.28$ kHz. Note that there is a frequency notch of 480 Hz about the airframe line, caused by the propeller spinner, and that the upper sidebands are approximately zero, caused by amplitude modulation of the return signal from the propeller blades.

7. FURTHER WORK

There are a number of ways in which the model could be improved. First, the model could be adapted to simulate cross section, range, and velocity pdfs other than those described above. Second, the model could be developed to simulate the return signal from the compressor and turbine blades of jet engines. (The model would then have to simulate multiple reflections from the airframe and engine cowling.) Third, the model could be extended to simulate volume clutter.

8. CONCLUSIONS

This paper has described an airborne pulse Doppler radar model which simulates the return signal from airborne targets. The return signal also includes noise and ground clutter.

The model is useful at the development stages of target acquisition and tracking algorithms, and target classification and identification algorithms.

It has been shown that, for a small number of scatterers, each having random variations of its cross section, range and velocity, the frequency spectrum is unrealistically noisy, and that as the number of scatterers is increased, the frequency spectrum becomes similar to that for practical targets.

ACKNOWLEDGEMENTS

The authors gratefully acknowledge the support of the Science and Engineering Research Council and Ferranti International, Radar Systems Division in the execution of the work reported here.

REFERENCES

[1] Mooney, D., and Ralston, G., "Performance in Clutter of Airborne Pulse MTI, CW Doppler, and Pulse Doppler Radar", *IRE Convention Record*, pp. 55-62, 1961.

[2] Farrell, J. L., and Taylor, R. L., "Doppler Radar Clutter", *IEEE Transactions on Aerospace and Navigational Electronics*, vol. ANE-11, no. 5, pp. 162-172, September 1964.

[3] Friedlander, A. L., and Greenstein, L. J., "A Generalized Clutter Computation Procedure for Airborne Pulse Doppler Radars", *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-6, no. 1, pp. 51-61, January 1970.

[4] Ringel, M. B., "An Advanced Computer Calculation for Ground Clutter in an Airborne Pulse Doppler Radar", *IEEE NAECON Record*, pp. 921-928, 1977.

[5] Skolnik, M. I., *Introduction to Radar Systems*, ch. 5, pp. 190-191, McGraw-Hill, New York, 1962.

[6] Skolnik, M. I., *Radar Handbook*, ch. 15, pp. 31-36, McGraw-Hill, New York, 1970.

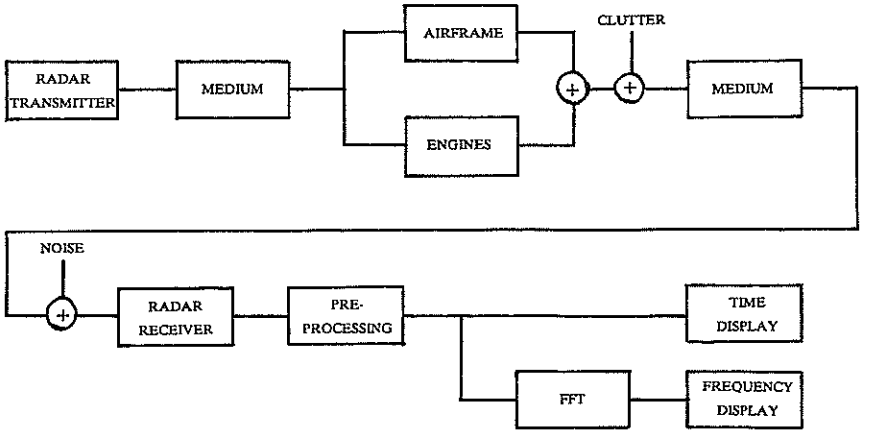


Figure 1.
Basic structure of the airborne pulse doppler radar model.

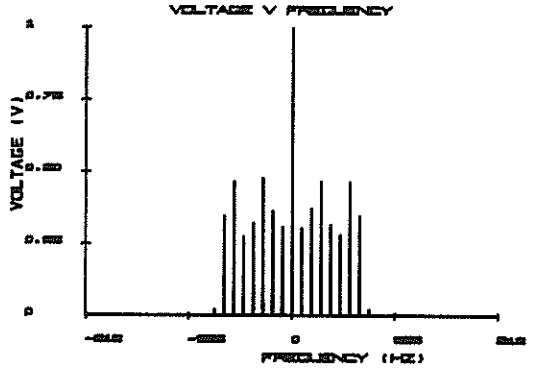


Figure 2.
Frequency spectrum of a helicopter aircraft in which $N_a = 1$ and $N_b = 1$.

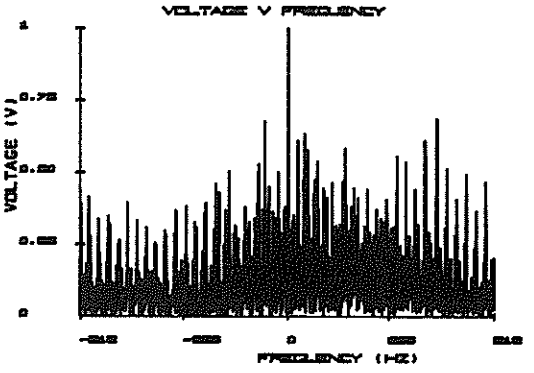


Figure 3.
Frequency spectrum of target with $N_a = 8$ and $N_b = 8$.

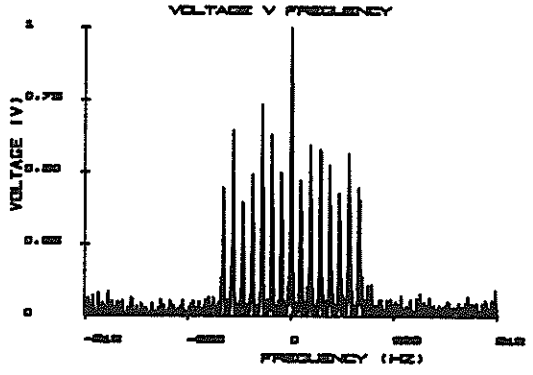


Figure 4.
Frequency spectrum of target with $N_a = 64$ and $N_b = 64$.

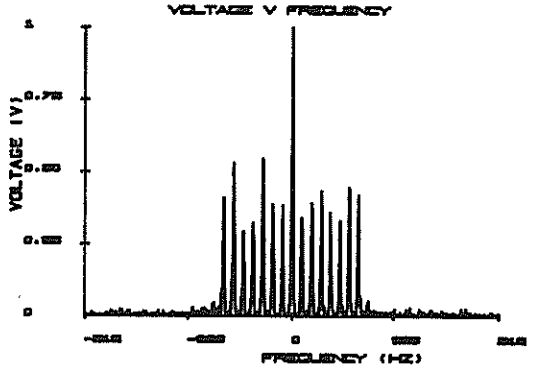


Figure 5.
Frequency spectrum of target with $N_a = 512$ and $N_b = 512$.

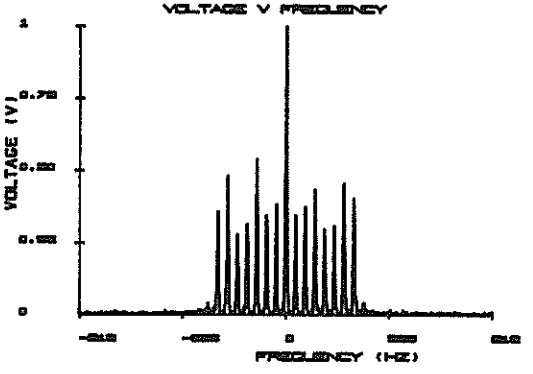


Figure 6.
Frequency spectrum of target with $N_a = 4096$ and $N_b = 4096$.

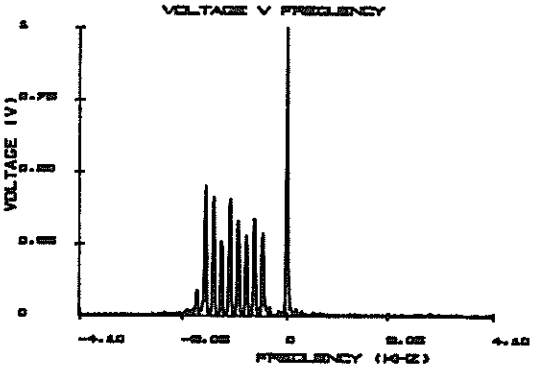


Figure 7.
Frequency spectrum of a propeller aircraft in which $N_a = 4096$ and $N_b = 4096$.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

Estimation of the Height of Swerling Fluctuating Targets Using the Maximum Likelihood Method

E. Bossé, R.M. Turner and M. Lecours*

Defence Research Establishment Ottawa
3701 Carling Avenue, Ottawa, Ontario, Canada, K1A 0Z4
(* Université Laval, Ste–Foy, Québec, J1K 7P4)

Accurate tracking of targets flying at low altitude above a smooth surface is difficult because of the surface reflection. We propose a solution to this problem based on deterministic physical modelling of the specular multipath used with the maximum likelihood method. We incorporate the effects of target fluctuation statistics in our model to develop a family of estimators for different target fluctuation models. We derive the Cramer–Rao bounds corresponding to the various target fluctuation models and the refined propagation model. Monte–Carlo simulations compare performance with the Cramer–Rao bound and demonstrate the threshold effects for both fluctuating and non–fluctuating Swerling target models.

1.0 INTRODUCTION

This paper presents a new method for low–angle tracking in presence of specular multipath. We define new estimators based on the maximum likelihood criterion for estimating the height of fluctuating and non–fluctuating targets flying at low altitude over the sea. The problem investigated is direction finding for a single target in the presence of specular reflection from a relatively smooth flat surface such as the sea. In recent years a considerable amount of research has been devoted to this problem. Much work has been done in the area of signal processing, using superresolution algorithms to separate a target from its image and to provide an accurate estimate of the target's elevation. Our approach is essentially different; we assume that interfering multipath signals are present and that identifying and modelling the nature of the multipath interference can have a beneficial effect on the estimation of desired parameters.

Barton [1] summarizes the low–angle tracking work up to 1974. His paper presents a number of methods to combat multipath errors. Most of these techniques use monopulse and they all fail for target separations which are less than 1/4 of a beamwidth. It seems accepted [2] that the limit of 1/4 of a beamwidth applies to most techniques including the high resolution techniques. For low–angle tracking, the requirements for resolution can be as high than 1/20 of a beamwidth.

We propose a technique [3,4] which incorporates a highly descriptive model of the physics of the low–angle problem. Litva [5], first used a similar model in an algorithm called CHA (Correlation Height Analysis) to obtain resolution up to 1/28 of a beamwidth [5]. In this paper we use the idea of a detailed propagation model but in the context of the estimation theory based on maximum likelihood (ML). The model we use represents multipath with one signal having three unknowns parameters: an amplitude, a phase and the target height. To obtain the information required by the model, the radar is operated in an acquisition mode to determine target range and doppler velocity. Prior knowledge about the geometry and the specular reflection coefficient is also assumed. The accuracy of the height estimate is directly related to the accuracy of the model.

In the previous works [3,4,6], we have considered a signal model where the complex amplitude of the signal is regarded as unknown but deterministic. However, in radar, the appropriate signal model corresponds to that of fluctuating or random signals. The likelihood function (defined in section 3) is considerably modified when the distributions of the complex amplitudes are taken into account. This is particularly true when the distribution is not gaussian. In this paper, we treat target fluctuations according to the Swerling models.

We use the following notation: matrices are represented by bold upper–case letters, vectors by bold lower–case letter, scalars by both upper and lower italic letters. The superscripts, $\hat{}$, $*$, ^{T}, ^{H}, $\| \ \|$ denote estimate, conjugation, transposition, conjugate transposition, and vector norm respectively.

2.0 The Propagation model

We consider a two–way transmission model (radar) and a one–way transmission model (beacon). The noise–free observation model for a signal received at the k^{th} element of an array of K sensors is $v_k = b\, f_k(h)$ where $b = Q\, exp(-j\xi R)$ for the beacon model and $b = Q\, exp(-j2\xi R)\, exp\{-j\xi(z_e^2 + h_e^2)/2R\} \times$

$$[exp(j\xi h_e z_e/R) + A_1\, exp(-j\xi h_e z_e/R)]$$

for the radar model.

The propagation model [7] is

$$f_k(h) = exp\{-j\xi(z_k^2 + h_k^2)/2R\} \times [exp(j\xi h_k z_k/R) + A_2\, exp(-j\xi h_k z_k/R)] \quad (1)$$

with $\xi = 2\pi/\lambda$ (λ, the wavelength), R, the target range, and Q, an unknown complex amplitude due to target characteristics. In the radar model, we consider four separate propagation paths for each element in the array: direct and reflected signals going from the transmitter to the target and direct and reflected signals returning from the target to the receiving array. In the beacon model, we consider only the last two paths. The z_e and h_e are the transmitter and target heights with respect to the tangent plane at the reflection point, O_1, for the transmitted signal; z_k and h_k are the antenna element and target heights measured with respect to the tangent plane at the reflection point, O_2, for the signal returning from the target; and A_1, A_2 are the complex multipath reflection coefficients at point O_1 and O_2 respectively. A_1, A_2 are determined from the reflection and specular scattering coefficients and divergence factors. Polarization enters the model via the reflection coefficients. All quantities are corrected for the earth curvature and these corrections are different for each element height. Therefore, we represent the target height by h_k.

Assume now that the observation vector or snapshot s coming from the output of an array having K sensors is given by $\mathbf{s} = b\,\mathbf{f}(h) + \mathbf{n}$ where b can be deterministic (non–fluctuating case) or random (fluctuating case) and the noise vector n is assumed to be stationnary, additive, spatially white and independent of the target signals. $(\mathbf{s}, \mathbf{f}, \mathbf{n}) \in \mathbb{C}^{K\times 1}$ and $b \in \mathbb{C}^{1\times 1}$.

3.0 ML using the physics of the low–angle problem

The ML estimate of an unknown parameter vector θ is that value of θ which maximizes the conditional density $p(s|\theta)$ of the observations, or likelihood function, L. The optimal receiver takes into account the propagation model and depending on the assumptions on b, different estimators are derived. In the previous works [3,4,6], we have considered b as an unknown deterministic constant; this is the Marcum 0 or Swerling 0 non–fluctuating model [8]. We present briefly the estimator for this case. We then treat fluctuating target cross–sections with b a random variable having probabilities densities described by Swerling [8].

3.1 The non–fluctuating case: Marcum 0 (Swerling 0)

We assume that the observation vectors are statistically independent from snapshot to snapshot. We also assume that they are non–coherent because we use frequency agility. The noise levels may differ on each frequency as a result of the frequency dependence of the receiver noise figure.The observation model is given by

$$\mathbf{s}_m = b_m \mathbf{f}_m(h) + \mathbf{n}_m \quad (2)$$

where the index m indicates the m^{th} frequency.

In [3,4], we obtain the ML estimator of the height of a non–fluctuating target (Marcum 0) as the value of h that maximize the following function:

$$C^n_{M0}(h) = \frac{1}{P} \sum_{m=1}^{M} \frac{\|\mathbf{s}_m^H \mathbf{f}_m(h)\|^2}{\sigma_m^2\, \|\mathbf{f}_m(h)\|^2} \quad (3)$$

where $P = \sum_{m=1}^{M} \|s_m\|^2/\sigma_m^2$ and ${}^{n}_{M0}$ means Marcum 0 non–coherent pulse train.

Because the function $C^n_{M0}(h)$ has multiple peaks [4] , we are forced to make an exhaustive search over a range of values. The estimate of h corresponds to the largest peak of (3). We solve the ambiguity problem of the multiple peaks by combining many data vectors taken at different frequencies [4].

3.2 ML height estimates for Swerling I & II

The Swerling I target model treats the amplitude of the entire pulse train as a single random variable with a Rayleigh probability density function. In addition, the initial phase of each pulse is assumed to have uniform distribution on $(0,2\pi)$. Swerling II differs from Swerling I in that the amplitude of each pulse is a statistically independent random variable with the same Rayleigh density function.

The observation vector for the Swerling II model is given by (2) where we have replaced b_m by $r_m\, exp(j\psi_m)$ with r_m having Rayleigh distribution and ψ_m having a uniform distribution on the interval $(0,2\pi)$. Under these assumptions, the observation vectors $\mathbf{s}_m$ are zero–mean independent complex gaussian vectors with a covariance matrix given by

$$\mathbf{R}_m = \mathrm{E}\{\mathbf{s}_m \mathbf{s}_m^H\} = \sigma_{tm}^2 \mathbf{f}_m \mathbf{f}_m^H + \sigma_m^2 \mathbf{I} \quad (4)$$

and where $\sigma_{tm}^2 = \mathrm{E}\{\|b_m\|^2\}$. Using $\mathbf{S}_m = \mathbf{s}_m \mathbf{s}_m^H$, the log–likelihood function is reduced to

$$L(\theta) = -\sum_{m=1}^{M} (log|\mathbf{R}_m| + \mathrm{tr}\{\mathbf{R}_m^{-1}\mathbf{S}_m\}) \quad (5)$$

where , $\theta = [h, \sigma_{t1}^2, \sigma_{t2}^2, \ldots, \sigma_{tM}^2]$.

We eliminate σ_{tm}^2 by setting to zero the derivative of (5) with respect to σ_{tm}^2 [7]. Then, we maximize (5) by determining the value of h that maximizes the following function:

$$C^n_{SWII}(h) = \frac{1}{P} \sum_{m=1}^{M} \left[\frac{\|\mathbf{s}_m^H \mathbf{f}_m(h)\|^2}{\sigma_m^2 \|\mathbf{f}_m(h)\|^2} - log\left\{ \frac{\|\mathbf{s}_m^H \mathbf{f}_m(h)\|^2}{\sigma_m^2 \|\mathbf{f}_m(h)\|^2} \right\} \right] \quad (6)$$

where ${}^{n}_{SWII}$ means Swerling II non–coherent pulse train.

The estimator for the Swerling I case is simply

obtained by assuming that the model and the noise characteristics do not change during the pulse train ($\mathbf{f}_m$ is replaced by $\mathbf{f}$ and σ_m^2 by σ^2). The ML estimators of the target height for the models Swerling I and II have the same form as for the Marcum 0 case. The peaks appear at the same height values. Equation (6) indicates that the target fluctuations increase the width of the $C_{M0}^n(h)$ peaks since we can show [7] that the logarithm in (6) is always negative. Target fluctuations increases the variance of the ML estimate of the target height since the variance is related to the curvature of the estimator peaks.

3.3 ML height estimates for Swerling III & IV

The Swerling III target model is similar to Swerling I in that each pulse in the train has the same amplitude. In the Swerling III model, the amplitude of the received pulse train is assumed to be a random variable with a one–dominant–plus–Rayleigh probability–density function. The Swerling IV is similar to Swerling II but the amplitude of each pulse in the train is assumed to be an independent random variable characterized by a one–dominant–plus–Rayleigh probability density.

The observation vector is given by the relation (2) but now r_m has probability density given by [7]

$$p(r_m) = \frac{8r_m^3}{\alpha_m^2} exp(-2r_m/\alpha_m) \qquad r_m \geq 0 \tag{7}$$

The likelihood function for a single snapshot, $p(\mathbf{s}_m|h)$, is then

$$p(\mathbf{s}_m|h) = \iint p(\mathbf{s}_m|r_m,\psi_m)\, p(r_m,\psi_m)\, dr_m\, d\psi_m \tag{8}$$

The extension to multiple snapshots (m = 1,2,...,M) is given in [7]. We first eliminate α_m by setting to zero the derivative of the log–likelihood with respect to α_m. We then obtain a second order polynomial with two roots. However, provided the signal–to–noise ratio is high and the true target height is close to the model height, the correlation between $\mathbf{s}_m^H$ and the model vector $\mathbf{f}_m(h)$ is high and the root with a + sign gives a valid estimate of α_m. We have

$$\hat{\alpha}_m = \frac{-v_{2m} + \sqrt{v_{2m}^2 - 4v_{1m}v_{3m}}}{2v_{1m}} \tag{8a}$$

where $v_{1m} = a_{1m}^2 + a_{1m}^2 b_{1m}$, $v_{2m} = 4a_{1m}^2 - b_{1m}^2$,
$v_{3m} = 4a_{1m} - 4b_{1m}$, $a_{1_m} = \|\mathbf{f}_m\|^2/\sigma_m^2$, $b_{1_m} = \|\mathbf{s}_m^H\mathbf{f}_m\|^2/\sigma_m^4$.

Using $\hat{\alpha}_m$, the ML estimator of the height of a low flying target fluctuation according to Swerling IV model is given by the height corresponding to the largest peak of the following function ($C_{SWIV}^n(h)$: means Swerling IV non–coherent pulse train):

$$C_{SWIV}^n(h) = \frac{1}{P}\sum_{m=1}^{M}\left(\frac{\hat{\alpha}_m\|\mathbf{s}_m^H\mathbf{f}_m(h)\|^2}{\sigma_m^4\, A(\hat{\alpha}_m)} - 2\,log\{A(\hat{\alpha}_m)\} + log\left[1 + \frac{\hat{\alpha}_m\|\mathbf{s}_m^H\mathbf{f}_m(h)\|^2}{\sigma_m^4\, A(\hat{\alpha}_m)}\right]\right) \tag{9a}$$

with
$$A(\hat{\alpha}_m) = \frac{\hat{\alpha}_m\|\mathbf{f}_m(h)\|^2}{\sigma_m^2} + 2 \tag{9b}$$

The ML estimator for a Swerling III model is given by the relations (9a, 9b) where $\mathbf{f}_m$ is replaced by $\mathbf{f}$ and σ_m^2 by σ^2.

4.0 Formulation of the Cramer–Rao bounds

The Cramer–Rao bound on the variance of the height estimate of a non–fluctuating target has been derived in [3,6] and it is given by

$$CR_{M0}(\hat{h}) \geq \left(2\sum_{m=1}^{M} SNR_m \left[\left\| \frac{\partial \mathbf{f}_m}{\partial h} \right\|^2 - \frac{1}{\|\mathbf{f}_m\|^2} \left\| \frac{\partial \mathbf{f}_m^H}{\partial h}\, \mathbf{f}_m \right\|^2 \right] \right)^{-1} \quad (\text{with } SNR_m = r_m^2/\sigma_m^2) \tag{10}$$

We illustrate the Cramer–Rao bound for data vectors that are coherently integrated into a single data vector by setting M = 1. Figure 1a shows an example of that bound versus the height for a target at 5 km and smooth sea conditions ($A \approx .9$) where A is the ratio of the amplitude of the reflected ray to the direct ray. The SNR is fixed at –6 dB. The horizontal line represents the case where $A = 0$. This case corresponds to the best that can be done with a model using a single plane wave (monopulse, Fourier beamforming). The results of fig.1a indicate that, over a very wide range of target heights of interest, modelling the multipath by using two plane waves is a very significant aid in measuring target height provided we know the correct relation between these two waves.

The Cramer–Rao limit when the target fluctuates according to Swerling models III & IV leads to almost intractable equations. We limit our derivations to zero–mean complex gaussian observation vectors as in Swerling I & II models. We use an extension [7] of the Bangs relation [9] to include the effects of frequency agility. The elements of the Fisher information matrix are given by:

$$J_{ij} = \sum_{m=1}^{M} tr\left(\mathbf{R}_m^{-1} \frac{\partial \mathbf{R}_m}{\partial \theta_i} \mathbf{R}_m^{-1} \frac{\partial \mathbf{R}_m}{\partial \theta_j} \right) \tag{11}$$

By using the inversion matrix lemma, we obtain the Cramer–Rao bound on the variance of the estimate of the height of a target fluctuating according to Swerling model I & II as $CR_{SWII}(\hat{h}) \geq$

$$\left[\sum_{m=1}^{M} \frac{2\ \mathrm{SNR}_m}{(\mathrm{SNR}_m\ \|\mathbf{f}_m\|^2 + 1)}\left[\|\mathbf{f}_m\|^2 \left\|\frac{\partial \mathbf{f}_m}{\partial h}\right\|^2 - \left\|\frac{\partial \mathbf{f}_m^H}{\partial h}\mathbf{f}_m\right\|^2\right]\right]^{-1} \quad (12)$$

$$(\text{with } \mathrm{SNR}_m = \sigma_{tm}^2/\sigma_m^2)$$

With high value of SNR the limit becomes the same as obtained for the non–fluctuating case (Marcum 0) suggesting that the estimator $C_{M0}(h)$ is sufficiently robust to use with fluctuating targets. Equation (12) indicates that target fluctuations increase the variance on the estimate of target height when the SNR decreases.

5.0 Results of Monte–Carlo simulations

We have used the root–mean–square errors, RMSE, as the measure of performance. The results shown in fig.1b use a linear array of K=8 equally spaced omnidirectional elements, with an antenna aperture of one–metre length. The height of the target was arbitrarily fixed at 10 metres and two X–band frequencies were used (9 and 10 GHz) operating over a smooth sea. We carry out Monte–Carlo simulations to determine the effects of target fluctuations. Figure 1b compares the results obtained with the Swerling II target model (6) and the non–fluctuating Marcum 0 (3). We have also plotted the Cramer–Rao bound for the non fluctuating case. Our estimators lie very close to the Cramer–Rao limit for large SNR but large deviations occur as the SNR decreases below a threshold value. Results show that target fluctuations do not have a big effect on the estimation variance when the SNR is large. However, target fluctuations increase the threshold value of SNR required to obtain good estimates.

6.0 CONCLUSION

There are strong theoretical reasons to believe that the ML estimation using a detailed propagation model provides a very significant improvement in radar tracking performance. We have derived a family of ML estimators for low angle tracking. We have evaluated the performance using both Monte–Carlo simulations and the Cramer–Rao bounds. Experimental verification is still ongoing but results obtained until now are extremely encouraging.

7.0 REFERENCES:

[1] Barton, D. K., " Low–Angle Radar Tracking ", Proceedings IEEE, vol.62, no. 6, pp.687–704, Jun. 1974.

[2] Various authors, " Antenna Array Signal Processing in Phased Array Radar ", NATO Report, AC/243 (Panel 3 /RSG.15) D/8, p.105, May 1988.

[3] Turner, R.M, Bossé, E., " Maximum Likelihood Tracking Using a Highly Refined Multipath Model ", 21st Asilomar conference on Signals, Systems and Computers, Pacific Grove, CA, Nov.2–4, 1987.

[4] Bossé, E., Turner, R.M, " Height Ambiguities in Maximum Likelihood Estimation with a Multipath Propagation Model ", 22nd Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, Oct. 31–Nov. 2, 1988.

[5] Litva, J., " Superresolution Based on the Use of Deterministic Physical Modelling ",Proc. of ICASSP–89, pp.2148–2151, Scottland, 1989.

[6] Turner, R.M., Bossé, E., " The Use of Highly Refined Propagation Models in Maximum Likelihood Estimation of Target Elevation for Radar Tracking of Low–Altitude Targets over the Sea ", EUSIPCO–88, Grenoble, Sept.5–8, 1988.

[7] Bossé, E., " Localisation radar de cibles rasantes utilisant le maximum de vraisemblance et la physique des phénomènes de propagation", Thèse de Doctorat, Université Laval , Québec, 1990.

[8] Meyer, D.P., Mayer, H.A., " Radar Target Detection: Handbook of Theory and Practice ", Academic Press, Londres et New–York, 1973.

[9] Bangs, W.J., " Array Processing with Generalized Beamformers ", Ph.D Dissertation, Yale University, New–Haven, CT, 1971.

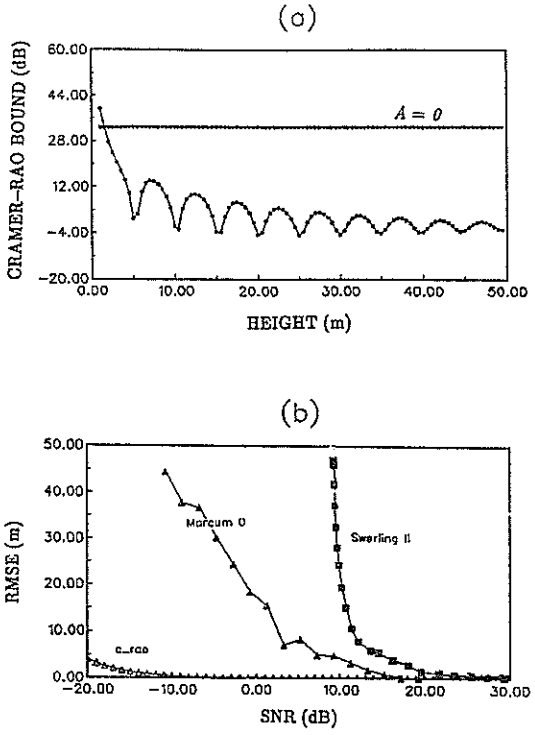


Figure 1
(a) The Cramer–Rao bound (Marcum 0)
(b) Threshold effects.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Quantization Effects in Multiple Frequency IFM Receivers

Jim Lansford, David Zurn
Georgia Tech Research Institute
Atlanta, Georgia, USA

William McCormick
Wright State University
Dayton, Ohio, USA

BACKGROUND

The Instantaneous Frequency Measurement (IFM) receiver has been extensively studied for use in Electronic Support Measures (ESM) systems [1][2][3]. The IFM offers many advantages in the ESM environment, such as multi-octave instantaneous bandwidth, near 100% probability of intercept, and ease of interface with other pulse parameter measurement subsystems in the overall suite. The IFM is ideal for tracking frequency agile signals as well as for sorting and identifying more conventional signals. A simplified block diagram of the IFM is given in Figure 1.

Despite these advantages, the IFM has yet to find widespread usage in operational systems. The most serious limitation of the IFM is its inability to resolve multiple simultaneous signals within its passband, which is a serious drawback given the likelihood of overlap in a wide bandwidth. The reason for this limitation is twofold: 1) A limiter is typically used to increase dynamic range, causing a capture effect, and 2) The theoretical basis for the IFM limits it to a single frequency. To understand why the latter is true, consider Figure 2. If the video bandwidth is small relative to the RF bandwidth, then the output of the lowpass filter will be very close to the autocorrelation function $R(t,\tau)$, where τ is the delay line time; note that the autocorrelation is also a function of the time t chosen to evaluate it [5]. Classical theory then dictates that the frequency ω_1 of the input sinusoid is then [1, p.184] :

$$\omega_1 = \frac{1}{\tau}\tan^{-1}\left(\frac{\mathrm{Im}[R(\tau)]}{\mathrm{Re}[R(\tau)]}\right) \qquad (1)$$

Upon closer inspection, this is actually an AR(1) model based on the complex autocorrelation function at lag τ. Since this is only a first order predictor, the all-pole model can only have a single pole in the z-plane, corresponding to a single input frequency. A logical extension to this concept would be to increase the order of the model to accommodate more signals, as suggested by McCormick and Tsui [5]. This approach has proven less than satisfactory for several reasons; The analog circuitry employed to date has not been capable of giving the desired dynamic range (>30 dB) but use of a limiter to extend dynamic range has caused the AR(N) model to perform poorly because of intermodulation products due to the non-linearity. Digital approaches could potentially eliminate these limitations but the required bandwidth requires processing speeds that are quite high.

APPROACH

The best solution method for the multifrequency IFM problem is digital calculation of the correlation estimates. The multifrequency IFM uses an autoregressive estimator of the form:

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(p) \\ R(-1) & R(0) & \cdots & R(p-1) \\ \cdots & \cdots & \cdots & \cdots \\ R(-p) & R(-p+1) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_p \\ a_{p-1} \\ \cdots \\ a_1 \end{bmatrix} = \begin{bmatrix} R(p+1) \\ R(p) \\ \cdots \\ R(1) \end{bmatrix} \qquad (2)$$

which is Hermitian symmetric but not Toeplitz in the general (complex) case. The entries in the vector $\underline{a}$ are the coefficients of the all pole predictor polynomial:

$$H(z) = \frac{G}{1-\sum_{i=1}^{p}(a)_i z^{-i}} \qquad (3)$$

where G is an arbitrary (in this application) gain term and z is the unit delay operator. The poles of H(z) (or the roots of the denominator) form the frequency estimates in the z-plane. In general, the entries in (2) are complex, thus allowing a p^{th} order predictor to represent p sinusoids (in theory) [6]. If the entries in (2) are real, then a p^{th} order model can only represent $\lfloor p/2 \rfloor$ sinusoids since the roots must occur in conjugate pairs.

This study is an investigation of the architecture given in Figure 3, which solves the two frequency case. The form taken by equation (2) then becomes:

$$\begin{bmatrix} R(0) & R(1) \\ R(-1) & R(0) \end{bmatrix} \begin{bmatrix} a_2 \\ a_1 \end{bmatrix} = \begin{bmatrix} R(2) \\ R(1) \end{bmatrix} \quad (4)$$

where it can be shown that $R(-1)=R^{*}(1)$ [8]. In this simple case, a solution by Cramer's rule yields an architecture that has much inherent parallelism. Using this architecture, the algorithm has been evaluated for a variety of word sizes (number of bits in the A/D converter) to study the effect of quantization on frequency accuracy.

Prior work by Blythe [9] has shown that the spurious frequencies introduced by quantizaton have a magnitude that can be related to Bessel functions. These spurious tones can perturb the estimates of the "true" signals but will, in general, cause peaks in the model spectrum as computed by Equation 3.

The effects of these spurious signals can most effectively be negated through the use of an overdetermined AR model, where the excess order will allow the spurious frequencies to be modeled. The image frequencies can then be pruned by evaluating the magnitude of the corresponding sinusoid in the Prony model; if the magnitude is small, the tone is considered spurious while a large magnitude indicates a valid signal. Figure 4 shows the effect of number of bits on the magnitude and frequency of the Prony model terms; this clearly shows the increase in amplitude of the model term corresponding to a spurious harmonic and an error in the estimate of the true frequencies.

SUMMARY

This paper has briefly shown a technique for extending the IFM to multiple frequencies and some important issues in quantization of these wide bandwidth signals. The ultimate tradeoff is that for fewer bits of quantization, a larger model must be used to accomodate spurious frequencies.

REFERENCES

[1] Tsui, J.B.Y., *Digital Microwave Receivers: Theory and Concepts*, Artech House, 1989.

[2] Heaton, Dean, "The Systems Engineer's Primer on IFM Receivers, *Microwave Journal*, Feb. 1980, pp. 71-85.

[3] East, P.W., "Design Techniques and Performance of Digital IFM", *IEE Proceedings*, Vol. 129, Part F, No. 3, June 1982, pp. 154-163.

[4] Lee, J.P.Y., "Detection of Complex and Simultaneous Signals Using an Instantaneous Frequency Measurement Receiver", *IEE Proceedings*, Vol. 132, Part F, No. 4, July 1985, pp. 267-274.

[5] McCormick, William S., and J.B.Y. Tsui, "A Real-time, Wide Bandwidth, Multiple Signal Frequency Estimator", Internal Report, Wright-Patterson AFB, Wright Research Development Center, Avionics Laboratory, 1988.

[6] Marple, S. Lawrence, *Digital Spectral Analysis with Applications*, Prentice-Hall, 1987.

[7] Kay, S.M., *Modern Spectral Estimation: Theory & Application*, Prentice-Hall, 1988.

[8] Tsui, J.B.Y., "IFM Receiver with Capability of Detecting Simultaneous Signals", Internal Report, Wright-Patterson AFB, Wright Research Development Center, Avionics Laboratory, 1988.

[9] Blythe, J.H., "The Spectrum of a Quantized Sinusoid", *GEC Journal of Research*, Vol. 3, No. 4, 1985, pp.229-242.

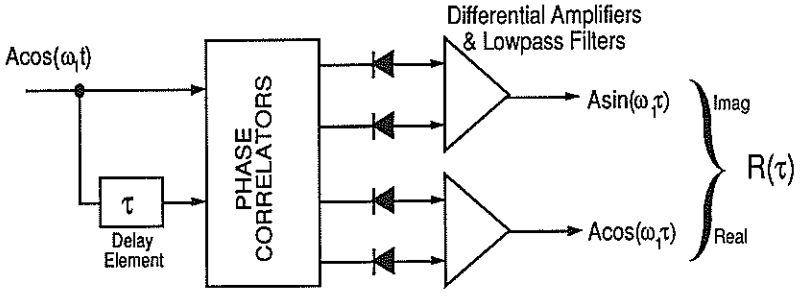


Figure 1: Simplified block diagram of IFM Receiver

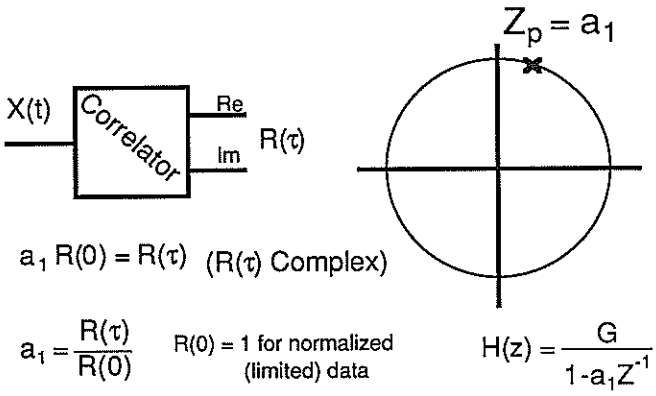


Figure 2: Single frequency IFM as an AR model

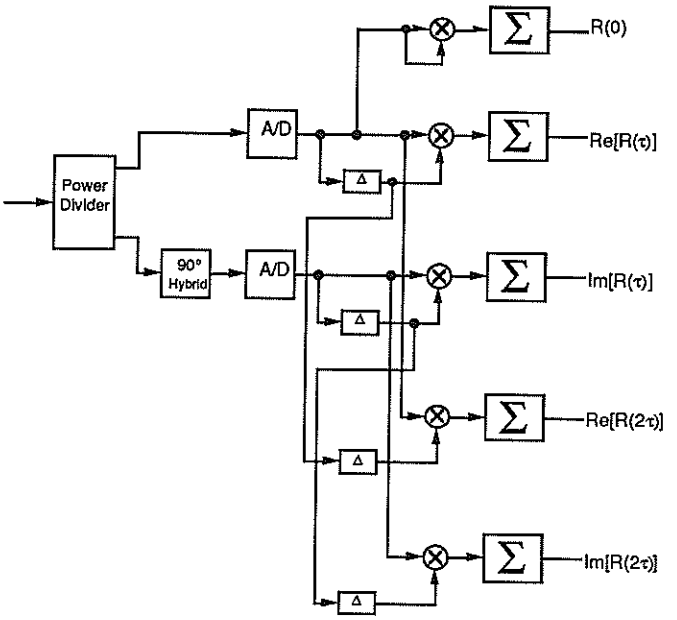
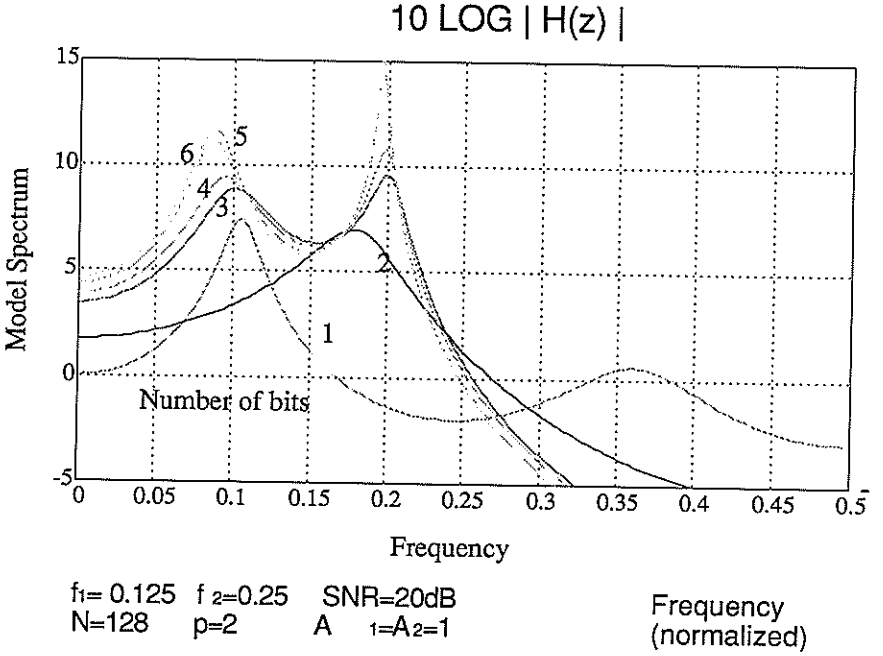


Figure 3: Block diagram of correlator for two-frequency architecture



f_1= 0.125 f_2=0.25 SNR=20dB
N=128 p=2 $A_1=A_2=1$

Frequency (normalized)

Figure 4: Spectral estimate for various quantization levels

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

DEROTATION TECHNIQUES IN RECEIVERS FOR MSK-TYPE CPM SIGNALS

Alfred Baier

Philips Kommunikations Industrie AG
D-8500 Nürnberg 10, Fed. Rep. of Germany

1. INTRODUCTION

Binary continuous phase modulations (CPM) with a modulation index of h=0.5 represent an important class of constant-envelope modulation schemes with fast spectral roll-off [1]. Starting from the classical full response Minimum Shift Keying (MSK) [2], a multitude of MSK-type partial response CPM schemes has been derived in the past. Amongst these are well-known modulations such as Raised-Cosine MSK, Gaussian MSK (GMSK), Tamed Frequency Modulation (TFM), and Generalized TFM (GTFM) [1].

In this paper, a signal processing technique called '*derotation*' will be presented which simplifies the structure and reduces the complexity of receivers for MSK-type CPM signals.

Section 2 leads off with a brief review of MSK-type CPM signals. By applying a linear approximation introduced in [3], it will be shown in Section 3 that these signals can be modelled as quadrature amplitude modulated (QAM) signals with a rotational signal structure. In Section 4, a generalized derotation technique is derived which abolishes this rotational signal structure and, thus, results in a simplified linear transmission model. Easy-to-implement versions of the derotation technique are presented in Section 5.

In Section 6, we will demonstrate how the derotation technique can be applied to different types of receivers for MSK-type CPM signals. In particular, a simple coherent threshold receiver, a noncoherent matched filter receiver for MSK-type spread spectrum signals, and an adaptive maximum likelihood Viterbi receiver for signals transmitted via intersymbol interference (ISI) channels will be considered.

2. MSK-TYPE CPM SIGNALS

A CPM signal has the general form

$$\tilde{s}(t) = A \cos [2\pi f_0 t + \Phi(t,\underline{b}) + \Phi_0] \qquad (1)$$

where f_0 is the carrier frequency, A is an amplitude factor assumed to be 1, and Φ_0 is a constant phase offset assumed to be 0 in the following. With the data sequence to be transmitted denoted by $\underline{b} = (\ldots, b_i, \ldots)$, the modulating phase function is given by [1]

$$\Phi(t,\underline{b}) = 2\pi h \sum_{i=-\infty}^{\infty} b_i \int_{-\infty}^{t-iT} g(\tau)d\tau \; . \qquad (2)$$

h is the modulation index, T is the symbol period, and $g(t)$ is a smooth pulse shaping function over a finite time interval $0 \le t \le LT$ and zero outside. $g(t)$ determines the smoothness of the continuous phase frequency modulation and, hence, has a great influence on the spectral properties of the CPM signal [1].

Following common conventions, we use '*MSK-type CPM signals*' as synonym for CPM signals characterized by a modulation index of $h = 0.5$ and by binary data symbols b_i taking on the values +1 and -1 [1], [3]. Furthermore, we will present all signals in complex baseband notation. With $j = \sqrt{-1}$ the CPM signal in baseband notation is

$$s(t) = \exp [j\Phi(t,\underline{b})] \qquad (3)$$

3. LINEAR QAM SIGNAL MODEL

According to [3] a h=0.5 CPM signal $s(t)$ can be closely approximated by the linear QAM signal

$$v(t) = \sum_{i=-\infty}^{\infty} \exp \left(j\frac{\pi}{2} \sum_{k=-\infty}^{i} b_k\right) p(t-iT) \; . \qquad (4)$$

$p(t)$ is a real-valued pulse shaping function spanning the time interval $0 \le t \le (L+1)T$. If $p(t)$ is properly matched to the frequency pulse shape $g(t)$ of equ. (2), the approximation error is negligible [3]. As an example, $p(t)$ is plotted in Fig. 1 for a GMSK signal with BT=0.3 and with $g(t)$ truncated to a length of $5T$ [1].

For MSK-type CPM signals with $b_i = \pm 1$, $v(t)$ can be further simplified to

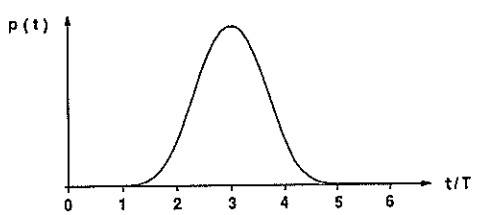


Fig. 1 Pulse shape $p(t)$ for GMSK (BT=0.3, L=5)

$$v(t) = \sum_{i=-\infty}^{\infty} c_i \, j^i \, p(t-iT) \qquad (5)$$

where the modified data symbols $c_i = \pm 1$ are determined by the recursion

$$c_i = c_{i-1} \, b_i \, . \qquad (6)$$

Obviously, the data sequence $\underline{c} = (\ldots, c_i, \ldots)$ is obtained from $\underline{b}$ by differential encoding assuming that the symbols +1 and -1 represent the logical 0 and 1 states, respectively. It is worth noting that an equivalent version of (5) exists in which j is substituted by -j and b_i in (6) is substituted by $-b_i$.

According to (5) and (6), the MSK-type CPM signal may be regarded as a linear partial response QAM signal with differentially encoded data symbols $c_i j^i$ which are phase-rotated in the complex plane by consecutive multiples of $\pi/2$. Fig. 2 illustrates this QAM signal model.

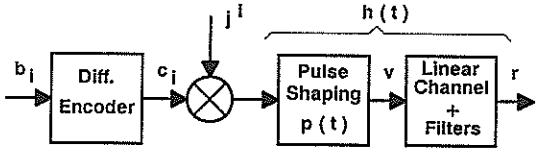


Fig. 2 Linear QAM model of MSK-type CPM signal transmission

Note that alternative MSK-type CPM schemes may be defined in which the data bits are not directly mapped onto the instantaneous frequency as in (2) but are mapped as in the Offset QPSK case [2]. For these modulations, the differential encoder in Fig. 2 has to be omitted and the symbol sequence $\underline{c}$ directly represents the data stream to be transmitted.

In Fig. 2, the QAM model has been extended to cover the effect of linear transmission channels and receiver filters. The received signal $r(t)$ of Fig. 2 is given by

$$r(t) = \sum_{i=-\infty}^{\infty} c_i \, j^i \, h(t-iT) \qquad (7)$$

where $h(t)$ is the overall complex impulse response of the linear transmission system including the pulse shaping filter $p(t)$.

4. GENERALIZED DEROTATION TECHNIQUE

In the following, we will formulate the derotation technique for continuous-time signals. However, a discrete-time formulation may easily be derived using an equivalent approach.

Let $q(t)$ be a complex function of time which fulfills the condition

$$q(t+iT) = q(t)(-j)^i \quad \text{for every } i. \qquad (8)$$

Due to the rotation factor $(-j)^i$ which exhibits the opposite sense of rotation with respect to the rotation factor in (7), $q(t)$ will be called '*derotation function*' henceforth.

Applying $q(t)$ of (8), we now define the derotated received signal

$$r'(t) = r(t)q(t) = \sum_{i=-\infty}^{\infty} c_i \, j^i \, h(t-iT)q(t)$$

$$= \sum_{i=-\infty}^{\infty} c_i \, j^i (-j)^i \, h(t-iT)q(t-iT) \, . \qquad (9)$$

By introducing the derotated impulse response

$$h'(t) = h(t)q(t) \, , \qquad (10)$$

$r'(t)$ can finally be expressed as

$$r'(t) = \sum_{i=-\infty}^{\infty} c_i \, h'(t-iT) \, . \qquad (11)$$

In (11), the derotated received signal $r'(t)$ is represented as a binary pulse amplitude modulated (PAM) signal, i.e. the rotational signal structure of $r(t)$ has been abolished. As a consequence, applying the derotation technique will result in simplified receiver structures.

5. PRACTICAL DEROTATION FUNCTIONS

The two most practical derotation functions $q(t)$ are

$$q_1(t) = \exp\left[-j\frac{2\pi}{4T}(t-t_0)\right] \, , \qquad (12)$$

$$q_2(t) = (-j)^i \quad \text{for } iT \le (t-t_0) < (i+1)T \qquad (13)$$

where t_0 is an optional time offset. Obviously, both $q_1(t)$ and $q_2(t)$ obey the condition (8).

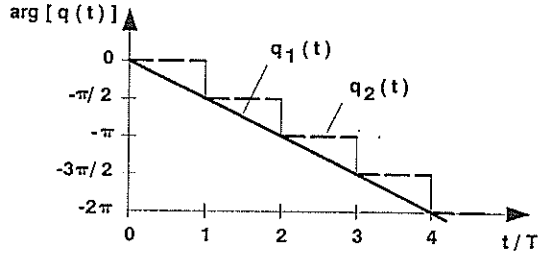


Fig. 3 Phase of derotation functions $q_1(t)$ and $q_2(t)$ for t_0=0

The derotation function $q_1(t)$ simply corresponds to a negative frequency shift by a quarter of the symbol rate $1/T$. $q_2(t)$, on the other hand, represents a stepwise derotation function which takes on the values ±1 and ±j only, cf. Fig. 3. This property makes the stepwise derotation function well suited for a low-complexity implementation in digital signal processing.

6. MSK-TYPE CPM RECEIVERS WITH DEROTATION

6.1. Coherent Threshold Receiver

In the following sections, we will demonstrate how the presented derotation technique can be utilized to derive simplified receiver structures. In the first example, we will consider a coherent threshold receiver with perfect carrier phase and symbol synchronization [1].

After having passed an appropriate receiver filter [1], the received signal $r(t)$ is sampled at optimum sampling instants $iT+t_1$ and derotated using the discrete-time derotation function $q_i = (-j)^i$, cf. Fig. 4. The derotation function must be synchronized with $r(t)$ to enable a coherent demodulation. The real part (inphase component) of the derotated signal r_i' is fed to a sign detector which recovers the data symbols c_i. Finally, the symbols c_i may be differentially decoded to obtain the data symbols b_i.

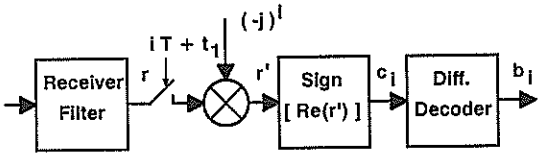


Fig. 4 Coherent threshold receiver with derotation

Alternatively, the derotation could be performed prior to sampling by means of the frequency shifting derotation function $q_2(t)$ according to (12). This leads us to the serial MSK receiver proposed by Amoroso [4].

6.2 Noncoherent Matched Filter Receiver

Next we consider a noncoherent matched filter (MF) receiver for finite-length MSK-type spread spectrum signals which is realized as quadrature receiver in the baseband [5].

Let $\underline{b} = (b_0,\ldots,b_{N-1})$ be a binary pseudo-noise (PN) sequence of lenght N which is used to generate an MSK-type CPM modulated spread spectrum signal $s(t)$ according to (2) and (3). T now represents the '*chip period*'. $\underline{b}$ is assumed to be chosen such that the differentially encoded sequence $\underline{c} = (c_0,\ldots,c_{N-1})$ defined by (6) has good autocorrelation properties, i.e.

$$\sum_{i=0}^{N-1} c_i c_{i+k} \cong \begin{cases} N & \text{for } k=0 \\ 0 & \text{otherwise.} \end{cases} \qquad (14)$$

The function of the noncoherent MF receiver is to decide whether the spread spectrum signal $s(t)$ is present in the received signal or not [5]. Note that the PN chips b_i and c_i are not explicitly recovered by the MF receiver.

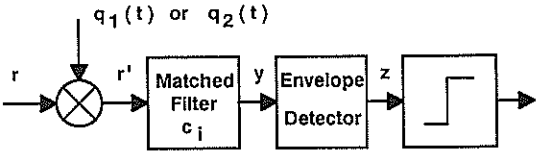


Fig. 5 Noncoherent MF receiver with derotation

In the noncoherent MF receiver of Fig. 5, the received signal $r(t)$ is derotated using the derotation function $q_1(t)$ or $q_2(t)$. $r'(t)$ is fed to a MF followed by an envelope detector and a threshold decider. The MF has a tapped-delay-line structure with a tap spacing of T and with the binary tap weights $c_i = \pm 1$. This MF produces the output signal

$$y(t) = \sum_{i=1}^{N} c_{N-i}\, r'(t-iT) \; . \qquad (15)$$

Using (11) and (14), it can be proven that

$$y(t) \cong N\, h'(t-NT) \; . \qquad (16)$$

As $|q(t)| = 1$ for the considered derotation functions, the output signal $z(t)$ of the envelope detector in Fig. 5 becomes

$$z(t) = |y(t)| \cong N\, |h(t-NT)| \; . \qquad (17)$$

Hence, the output signal of the noncoherent MF directly corresponds to the magnitude of the impulse response $h(t)$ or, if the effects of the transmission channel and receiver filter can be neglected, to the pulse shape $p(t)$ of the MSK-type CPM.

It should be pointed out that due to the envelope detection according to (17), the derotation function $q(t)$ does not need to be phase or time synchronized with the received signal, i.e. the time offset t_0 introduced in (12) and (13) may be chosen arbitrarily in this kind of application.

An examination of the MF structure presented in [5] for MSK-type signals makes apparent that the MF structure is substantially simplified by introducing the derotation technique since now only real-valued binary MF tap weights are required.

6.3. Adaptive MLSE Viterbi Receiver

In the third example, an adaptive maximum likelihood sequence estimation (MLSE) Viterbi receiver with derotation will be presented. This receiver takes into account ISI caused by a time-dispersive transmission channel and, thus, acts as an adaptive equalizer [6], [7].

Let us again start from an MSK-type CPM signal $s(t)$ according to (2) and (3). The effect of the time-dispersive transmission channel is assumed to be included in the impulse response $h(t)$ introduced in (7). Based on the linear QAM model (7), an MLSE receiver without derotation may be designed using the matched filter approach presented in [6]. In order to recover the data sequence $\underline{b}$ or $\underline{c}$ hidden in the received signal $r(t)$, the non-derotation MLSE receiver calculates the metrics

$$M(\underline{c}) = \sum_{i=-\infty}^{\infty} \mathrm{Re}\,[c_i \mathrm{j}^{-i}(x_i - \sum_{k\geq 1} R_k c_{i-k} \mathrm{j}^{(i-k)})] \quad (18)$$

with

$$x_i = \int_{-\infty}^{\infty} r(\tau+iT)h^*(\tau)\mathrm{d}\tau \quad (19)$$

$$R_i = \int_{-\infty}^{\infty} h(\tau+iT)h^*(\tau)\mathrm{d}\tau \quad (20)$$

for any possible sequence $\underline{c}$ and searches for that particular sequence which maximizes $M(\underline{c})$. The asterisk in (19) and (20) denotes complex conjugation. The search for the optimum sequence $\underline{c}$ is efficiently carried out by means of the Viterbi algorithm [6]. As to techniques for estimating the impulse response $h(t)$ required for the metric calculation refer to [7].

The metric calculation in the described MLSE Viterbi receiver is rather complicated due to the rotation factors occuring in (18). This can be avoided and complexity can be reduced if the MLSE receiver operates on the derotated signal $r'(t)$. Starting from the binary PAM model (11), the metric

$$M'(\underline{c}) = \sum_{i=-\infty}^{\infty} \mathrm{Re}\,[c_i(x_i' - \sum_{k\geq 1} R_k' c_{i-k})] \quad (21)$$

with

$$x_i' = \int_{-\infty}^{\infty} r'(\tau+iT)h'^*(\tau)\mathrm{d}\tau \quad (22)$$

$$R_i' = \int_{-\infty}^{\infty} h'(\tau+iT)h'^*(\tau)\mathrm{d}\tau \quad (23)$$

is obtained. Using eqs. (8), (9), and (10), it can be verified that $M(\underline{c})$ of (18) and $M'(\underline{c})$ of (21) are identical provided $|q(t)| = 1$.

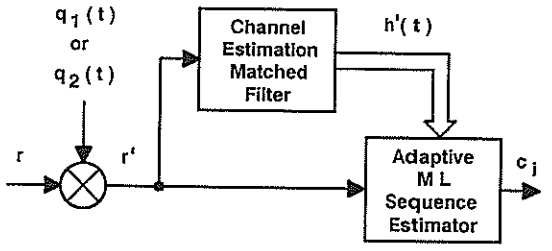


Fig. 6 Adaptive MLSE Viterbi receiver with derotation

The derotated impulse response $h'(t)$ required in (22) and (23) can be directly estimated in the receiver by observing the derotated signal $r'(t)$. For example, consider a MF which is matched to a finite-length pseudo-random sequence of training symbols c_i which are included in the transmitted signal and are known in the receiver [7]. From Section 6.2 it is evident that such a MF operating on $r'(t)$ directly produces an estimate of $h'(t)$, cf. (16).

Fig. 6 illustrates the structure of an adaptive MLSE Viterbi receiver with derotation comprising a channel estimation MF. Owing to the estimation of $h'(t)$ and the receiver adaptation, the derotation function $q(t)$ does not need to be synchronized with the received signal $r(t)$.

REFERENCES

[1] Sundberg, C.-E., "Continuous phase modulation", *IEEE Commun. Magazine*, Vol. 24 (1986), No. 4, pp. 25-38.

[2] Pasupathy, S., "Minimum shift keying: A spectrally efficient modulation", *IEEE Commun. Magazine*, Vol. 17 (1979), No. 4, pp. 14-22.

[3] Laurent, P.A., "Exact and approximate construction of digital phase modulations by superposition of amplitude modulated pulses (AMP)", *IEEE Trans. Commun.*, Vol. COM-34 (1986), No. 2, pp. 150-160.

[4] Amoroso, F., and Kivett, J.A., "Simplified MSK signaling technique", *IEEE Trans. Commun.*, Vol. COM-25 (1977), No. 4, pp. 433-441.

[5] Baier, A., "A low-cost digital matched filter for arbitrary constant-envelope spread-spectrum waveforms", *IEEE Trans. Commun.*, Vol. COM-32 (1984), No. 4, pp. 354-361.

[6] Ungerboeck, G., "Adaptive maximum-likelihood receiver for carrier-modulated data-transmission systems", *IEEE Trans. Commun.*, Vol. COM-22 (1974), No. 5, pp. 624-636.

[7] Baier, A., "Correlative und iterative channel estimation in adaptive Viterbi equalizers for TDMA mobile radio systems", in: *ITG Report 107 'Stochastic Models and Methods in Information Theory'* (VDE-Verlag, Berlin, 1989), pp. 363-368.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

NEW FAST TRANSFORM BASED COMPLEX TRANSMULTIPLEXER IMPLEMENTATION.

I. R. Corden, Dr. R. A. Carrasco.

Dept. of Electrical & Electronic Engineering, Staffordshire Polytechnic, Beaconside, Staffs., ST18 OAD, England, U.K..

Abstract.
This paper presents results concerning the functional assessment of an asynchronous fast transform based quadrature transmultiplexer algorithm designed to facilitate multi-band quadrature PSK data transmission and reception, such as are to be employed in the forthcoming third generation of 'intelligent' satellites to utilize on-board multicarrier demodulation (MCD) techniques. The algorithm which has been implemented using a TMS320C25 signal processor and tested in conjunction with a TMS320C30 processor uses a new technique for the fulfilment of the band multiplexing process. The method relies upon a splitting of the transform processor into two conjugate transforms.

1. INTRODUCTION.

During recent years, interest has arisen with regard to the use of digital signal processing (DSP) methods for the implementation of quadrature phase shift keying (QPSK) modulation digital data modems. In the case of single channel modem systems, the strive for improved reconfigurational flexibility and increased data transmission rates has led to the use of general purpose DSP devices such as those from the TMS320 family, rather than implementation-specific expedients in order to construe greater economic viability. Further, such methods have been implemented with specific regard to computational load [1]. Though the concepts utilized in single channel DSP based modems may be extended toward the implementation of a single group-string in a multi-band modem, the computational effort that ensues is so overwhelming that it is not fruitful to pursue them. Instead, digital transmultiplexers (TMUXs) in conjunction with per-channel modems may be used to achieve the modulation and multicarrier demodulation (MCD) functions.

This contribution addresses the problem of the implementation of the TMUX equipment for the modulating function. The algorithm used herein was introduced previously [2], and represents a new more efficient approach for the band multiplexing process which will be referred to here as a 'conjugate band processing' strategy. A further novel feature to be discussed is that of a direct interfacing of an un-filtered complex baud signal to the TMUX modulator.

2. CHOICE OF TMUX STRUCTURE.

Various classes of TMUX algorithms are being researched for use in MCD functions. These have been discussed in detail in [2]. Following [3], it is postulated here that since the cost to sample conversion time relationship is not a linear one that an all-complex processing TMUX structure is to be preferred. The argument for this stems from the fact that by using this approach, two continuous time - discrete time system interfaces are required, but these may have double the conversion time of that allowable when a single conversion interface is utilized in an equivalent real frequency division multiplexing (FDM) process. Further, it is desirable to deal with a complex FDM signal, as the single sideband (SSB) spectral translations to and from the IF band required in typical satellite systems are achieved with greater simplicity in the analogue time domain in that case. It may also be noted that although there is no absolute requirement for either even or odd channel stacking, those algorithms using the odd structure do not require one of the spectral packets to fall upon the transition band of the analogue filter; further, as will be shown, the odd technique allows the formulation of the conjugate band multiplexing algorithm.

3. COMPLEX TRANSMULTIPLEXER ALGORITHM.

3.1 Modulator.

This section presents a Z-domain derivation of the TMUX modulator algorithm used. Using contiguous band channel stacking, the modulation function may be expressed in the form [3]:

$$Y(z) = \sum_{i=0}^{N-1} z^{-i} B_i(-z^N) \sum_{k=0}^{N-1} S_k(-z^N)\, e^{+j\pi(2k+1)i/N}. \quad (1)$$

The parameters: $B_i(-z^N)$ denote the band inverted polyphase subfilters, $S_k(-z^N)$ refers to the k^{th} band inverted complex baud signal, and Y(z) indicates the complex FDM assemblage of QPSK data signals. (N is assumed throughout to be an even integer). The form of equation (1) indicates that 2N real polyphase subfilter operations must be implemented along with an N-point complex inverse odd discrete Fourier transform (IODFT) which may be realized through a complex inverse fast Fourier transform (IFFT) and a set of N complex phase offsets. Utilizing a conjugate pair of exponential modulation schemes, where an odd channel stacking multiplex is desired equivalent to that attained in equation (1), then one may derive:

$$Y_E(z) = \sum_{k=0}^{M-1} S_{2k}(z'^N)\, H(z'), \quad (2a)$$

$$z' = z \exp\{-j\pi(4k+1)/N\},$$

where M=N/2, H(z) denotes a prototype filter, and $Y_E(z)$ indicates the FDM multiplex consisting of the even parts of Y(z) only. Similarly for the odd components: $Y_O(z)$ of Y(z), one obtains:

$$Y_O(z) = \sum_{k=0}^{M-1} S_{N-2k-1}(z''^N)\, H(z''), \quad (2b)$$

$$z'' = z \exp\{+j\pi(4k+1)/N\}.$$

Decomposing the prototype filter into a set of digital polyphase subfilters [4], re-ordering summations and utilizing:

$$Y(z) = Y_E(z) + Y_O(z), \text{ yields:} \quad (3)$$

$$Y(z) = \sum_{i=0}^{N-1} z^{-i} B_i(-z^N) \Big\{ \sum_{k=0}^{M-1} S_{2k}(-z^N)\, e^{+j\pi(4k+1)i/N} + \sum_{k=0}^{M-1} S_{N-2k-1}(-z^N)\, e^{-j\pi(4k+1)i/N} \Big\}, \quad (4)$$

which can be reformulated as shown in equation (5) where, two M-point complex conjugate DFT processors are required along with 2N complex phase offsets.

$$Y(z) = \sum_{i=0}^{N-1} z^{-i} B_i(-z^N) \Big\{ e^{+j\pi i/N} \sum_{k=0}^{M-1} S_{2k}(-z^N)\, e^{+j2\pi ki/M} + e^{-j\pi i/N} \sum_{k=0}^{M-1} S_{N-2k-1}(-z^N)\, e^{-j2\pi ki/M} \Big\}. \quad (5)$$

Then,

$$Y(z) = \sum_{i=0}^{M-1} z^{-i} B_i(-z^N) \Big\{ e^{+j\pi i/N} \sum_{k=0}^{M-1} S_{2k}(-z^N)\, e^{+j2\pi ki/M} + e^{-j\pi i/N} \sum_{k=0}^{M-1} S_{N-2k-1}(-z^N)\, e^{-j2\pi ki/M} \Big\} + j \sum_{i=0}^{M-1} z^{-i-M} B_{i+M}(-z^N) \Big\{ e^{+j\pi i/N} \sum_{k=0}^{M-1} S_{2k}(-z^N)\, e^{+j2\pi ki/M} - e^{-j\pi i/N} \sum_{k=0}^{M-1} S_{N-2k-1}(-z^N)\, e^{-j2\pi ki/M} \Big\}, \quad (6)$$

which requires two M-point discrete complex transforms. It may also be noted here that the phase offsets can be computed in general by way of a 4/6 multiply/add strategy rather than a dual 4/2 process, though further improvements are possible as a result of trivial operations. Alternatively, dual 3/3 processes [5] may be replaced by single 6/5 schemes. Traditionally, the computational complexity of DSP structures is evaluated in terms of the required number of multiplies. However, when modern DSP devices are used for the implementation, the total number of operations required is of more concern. This then leads to a predilection for the 4/6 process.

3.2 Demodulator.

The demodulation process may be formulated in the Z-domain thus:

$$S_{2k}(z^N) = 1/N \sum_{i=0}^{N-1} B_i(z^N) \Big\{ z^{-i} \sum_{m=0}^{N-1} Y(z\, e^{-j2\pi m/N}\, e^{+j\pi(4k+1)/N})\, e^{+j2\pi mi/N} \Big\}.$$

$$;k=0,1,\ldots,(M-1). \quad (7a)$$

$$S_{N-2k-1}(z^N)=1/N \sum_{i=0}^{N-1} B_i(z^N) \{ z^{-i} \sum_{m=0}^{N-1} Y(z\, e^{-j2\pi m/N}\, e^{-j\pi(4k+1)/N})\, e^{+j2\pi mi/N} \}. \quad ;k=0,1,...,(M-1). \qquad (7b)$$

Substituting:

$$Y'_i(z^N) = 1/N \sum_{m=0}^{N-1} Y(z\, e^{-j2\pi m/N})\, e^{+j2\pi mi/N}\, z^{-i}, \qquad (8)$$

into equations (7) yields the forms:

$$S_{2k}(z^N) = \sum_{i=0}^{M-1} \{B_i(z^N)\, Y'_i(-z^N) + j\, B_{i+M}(z^N)\, Y'_{i+M}(-z^N)\}\, e^{+j2\pi ki/M}\, e^{+j\pi i/N} \quad ;k=0,1,...(M-1). \qquad (9a)$$

$$S_{N-2k-1}(z^N) = \sum_{i=0}^{M-1} \{B_i(z^N)\, Y'_i(-z^N) - j\, B_{i+M}(z^N)\, Y'_{i+M}(-z^N)\}\, e^{-j2\pi ki/M}\, e^{-j\pi i/N} \quad ;k=0,1,...(M-1). \qquad (9b)$$

The phase offsets can be separated as:

$$G(i,N) = e^{+j\pi i/N}, \text{ and } G^*(i,N) = e^{-j\pi i/N}. \qquad (10)$$

Further,

$$V'_n(z^N) = \sum_{i=0}^{M-1} \{B_i(z^N)\, Y'_i(-z^N) + j\, B_{i+M}(z^N)\, Y'_{i+M}(-z^N)\},$$

and,

$$V_n(z^N) = \sum_{i=0}^{M-1} \{B_i(z^N)\, Y'_i(-z^N) - j\, B_{i+M}(z^N)\, Y'_{i+M}(-z^N)\}; \qquad (11)$$

for: n=0,1,...(M-1).

The demodulated output signals are obtained by way of the transform output mappings:

$$S_{2k}(z^N) = S'_k(z^N) \quad ;k=0,1,...(M-1),$$
$$S_{N-2k-1}(z^N) = S''_k(z^N) \quad ;k=0,1,...(M-1). \qquad (12)$$

From an implementational standpoint, it is more appropriate to situate the band inversion processes in the demodulator immediately subsequent to the commutator, and in the modulator, immediately prior. In this manner, the subfilter coefficients need not be band inverted. This simplifies the downloading of coefficients to the DSP processor from host machine filter design programs.

4. IMPLEMENTATION.

4.1 Asynchronous data interface.

The modulator algorithm of equation (6) has been implemented using a TMS320C25 system with an asynchronous baud interface. It has also been implemented using a TMS320C30 device. Pseudo-aperiodic baud signals were utilized in the TMS320C25 system, these being derived from a pair of shift register integrated circuits (ICs). The digital data streams were then sampled by means of a direct digital-to-digital interface to the TMS320C25 data bus. A sampling baud density of 4 samples per symbol was chosen to minimize jitter in the sampled signals. For hardware simplification, no pre-filtering was undertaken.

4.2 Prototype filter design.

Since the sampled signal characterizes digital step functions, then an equalization process is required consisting of the frequency domain correction factor: $1/\mathrm{sinc}(\pi f/R)$, (where R denotes the baud rate). In practice, this is coupled to the frequency response of the prototype filter which in this case consisted of a 480 taps FIR linear phase square root raised cosine design with an excess bandwidth of 40%. This filter for an eight channel system was designed using a type-1 frequency sampling method, and facilitates a peak passband ripple of 0.303dB and a stopband attenuation in excess of 38dB when a critically sampled solution is sought.

4.3 DSP program structure.

The polyphase subfilter TMUX structure calls for N different filters to act upon as many complex signals which in the case of the modulator follow the transforms and phase offsets. In order to implement these, use was made of the TMS320C25 'CALA' and TMS320C30 'CALLU' instructions. A restriction concerning the TMUX implementation when using the TMS320C25 DSP is its limited data page length of 128 words. In order to overcome this difficulty, a program structure was elaborated consisting of multiple data pages to hold the data buffers for each of the 2N real filters. This restriction does not occur when using the TMS320C30 device due to its data section page length of 64Kwords. Further, the TMS320C30 cyclic addressing mode allows each of the subfilters to be programmed easily by storing pointers to each coefficient and data buffer array as indicated in table 1.

5. RESULTS.

In order to test the quality of the TMUX algorithm in an implementational sense the algorithm was programmed using both 16 bit fixed point integer (TMS320C25) and fully floating point (TMS320C30) arithmetic. The modulator output was then saved directly to RAM memory in addition to being output to DAC devices.

5.1 Frequency domain results.

In order to establish the nature of the magnitudes of the Fourier transforms of the multiplexer output signals in both cases sampled at the rate: 9.6KHz, a block of length 2047 samples was processed with a Kaiser window with a β-parameter equal to 13.2216, corresponding to a spectral leakage value throughout the transformed data of no more than -99.65dB. The block length used realizes a spectral resolution of 40.66Hz, or one part in 236. The spectragram of figure 1 was obtained with both algorithms operating under identical conditions (identical prototype filters and impulse inputs), except for the difference in fixed and floating point arithmetic. Figure 1(a) indicates for the critically sampled N=8 case the spectragram obtained when only channel 0 is loaded in the TMS320C25 system, and figure 1(b) indicates that obtained when the TMS320C30 is used.

6. CONCLUSIONS.

A new more efficient complex input - complex output block processing TMUX structure has been derived in the Z-domain and advocated as a suitable choice for an efficient implementation of the band multiplexing structure for use in a QPSK group modem. The algorithm has been implemented for N=8 channels using 16 bit integer arithmetic using a TMS320C25 device and floating point arithmetic using a TMS320C30 DSP. The resulting spectragrams have been obtained for both implementations, indicating the performance of the algorithm.

REFERENCES.

[1] Corden I. R. and Carrasco R. A.,'A high speed DSP PSK modem for real time digital speech transmission.', Proc. 2nd IEE National Conference on Telecommunications, York U.K., April 1989, pp.333-339.

[2] Corden I. R. and Carrasco R. A.,'New fast transform based complex transmultiplexer algorithm for QPSK digital multicarrier demodulator application.',IEE Colloquium, 'Modems and codecs for satellite communications', London U.K., Oct. 1989, pp.12/1-5.

[3] Takahata F. et al, 'A PSK group modem based on digital signal processing: algorithm, hardware design, implementation and performance.', International Journal of Satellite Communications, Vol.6, 1988, pp.253-266.

[4] Bellanger M. G. and Daguet J. L., 'TDM-FDM transmultiplexer: digital polyphase and FFT.', IEEE Transactions on Communications, Vol.COM-22 No.9, Sept. 1974, pp.1199-1205.

[5] Vetterli M. and Nussbaumer H. J., 'Simple FFT and DCT algorithms with reduced number of operations.', Signal Processing, (North Holland), No.6, 1984, pp.267-278.

Table 1 - TMS320C30 subfilter algorithm.

```
    LDI     @XSTAR,AR7
    LDI     @HSTAR,AR6
    LDF     @E0RN0,R1
    STF     R1,*AR7++%
    LDF     0.0,R0
    LDF     0.0,R2
    RPTS    NTAPS-1
    MPYF3   *AR6++%,*AR7++%,R0
||  ADDF3   R0,R2,R2
    ADDF    R0,R2
    STF     R2,@F0RN0
    STI     AR7,@XSTAR
    STI     AR6,@HSTAR
```

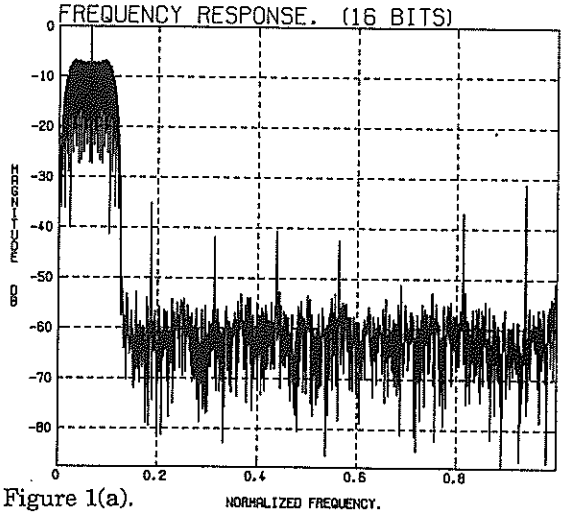


Figure 1(a).

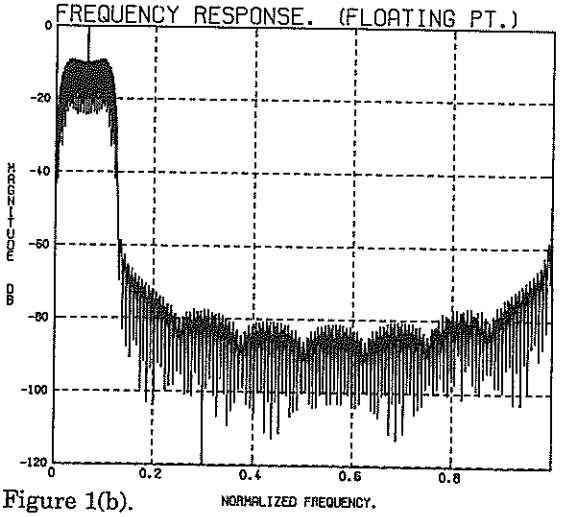


Figure 1(b).

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

DIFFERENTIALLY CODED MULTI-FREQUENCY MODULATION FOR DIGITAL COMMUNICATIONS

Paul H. Moose

Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943
and
Mercury Digital Communications, Inc.
243 Eldorado St., Suite 201,
Monterey, CA, 93940

Multi-frequency Modulation (MFM™), utilizing a multiplicity of orthogonal carrier tones simultaneously, produces a robust, bandwidth efficient signal for digital communications. Signals are generated at baseband or bandpass with minimal hardware requirements. Differential coding between adjacent carrier tones, providing the tones are closely spaced, eliminates the need for coherent carrier reference signals and for channel equalization. Encoding/decoding take place utilizing fast Fourier transforms.

1. INTRODUCTION

Multi-frequency Modulation (MFM™) is a new method for data communications that relys on digital signal processing capabilities resident in the host sending and receiving microcomputers to generate and demodulate the actual physical analog signals sent over the link. Interfacing to the link is via digital-to-analog (D/A), and analog-to-digital (A/D) converters. The frequency spectrum of the signal, either bandpass or baseband, is controlled by an externally supplied clock to the D/A and A/D. MFM is a packet oriented signalling format that sends K tones per baud for L bauds. These KL signals form an orthogonal signal set. Data are encoded in the amplitude and phase of each of the KL signals. In differentially encoded MFM, data are encoded as the change in amplitude and/or of phase between two adjacent tones within the same baud. Differential encoding of MFM signals is extremely effective when successive bauds or adjacent frequencies are subject to identical but unknown amplitude and/or phase changes between the transmit and receive computers. In this paper we describe the encoding, generation, demodulation, decoding and performance of Multi-Frequency Differential Quadrature Phase Shift Keyed (MFDQPSK), and of Multi-Frequency Differential 16-QAM (MFD16-QAM).

2. THEORY

MFM signals are generated inside the host transmit microcomputer using an Inverse Discrete Fourier Transform (IDFT). The DFT technique was first suggested by Weinstein and Ebert [1] and has subsequently been further described by others (see, for example, [2], [3], and [4]). Each baud consists of a digital signal of k_x real values. When clocked out upon command through an I/O port to a D/A converter at f_x samples per second, a baud of length $\Delta T = k_x/f_x$ seconds is sent over the channel. The signal consists of tones spaced at intervals $\Delta f = 1/\Delta T$ Hz. A bandpass signal is generated in the band $f_1 = k_1\Delta f$ to $f_2 = k_2\Delta f$ by assigning non-zero amplituded only to those digital frequencies between k_1 and $k_2 = k_1+K$. A baseband signal is generated by assigning all digital frequencies between one and $k_x/2-1$ non-zero amplitudes. In both cases, the actual frequency spectrum occupied by the signal is controlled by the clock frequency f_x. Concatenation of L signal bauds produced by an L-fold repetition of this process creates a signal packet of length $L\Delta T$.

From the discussion above, we see that in MFM the data to be transmitted with each baud are encoded directly in the frequency domain as complex numbers. In MFQPSK, two bits (a di-bit) are sent with each digital frequency using the state diagram of Figure 1. Since the signal occupies a bandwidth of $K/\Delta T$, the throughput rate is 2 bits per Hz of occupied channel bandwidth. In MF16-QAM, four bits are sent with each digital frequency using the state diagram shown in Figure 1. Three bits are encoded into

the 8 phases and one bit is encoded as amplitude of the digital frequencies. Using this constellation, data is transmitted at a throughput rate of 4 bits per Hz of channel bandwidth. An MFM packet is illustrated in Figure 2.

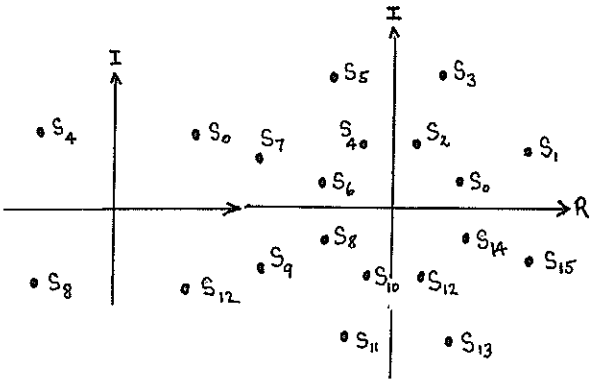


Figure 1
MFM Constellations

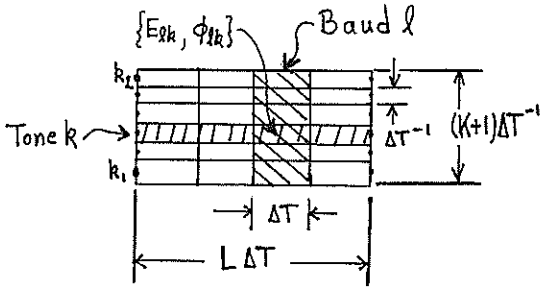


Figure 2
An MFM Signal Packet

The demodulation and decoding of MFM is accomplished as the inverse of the encoding and modulation process. That is, at the receiver L real valued sequences of k_x points are obtained from the packet of L bauds by sampling the received analog signal at f_x samples per second and storing the samples in the host receiver microcomputer's RAM. The k_x point DFTs are obtained of the L sequences, but only those complex coefficients that correspond to transmitted digital frequencies are retained for decoding. In an additive white gaussian noise memoryless channel, the 2KL values obtained in this manner are statistically independent gaussian random variables with identical standard deviations and with means that depend on the transmitted data values. Dividing each of these values by the standard deviation yields a set of 2KL statistically independent, unit variance, gaussian random variables that have mean values given by

$$E[R_l(k)] = \{2E_{kl}/N_o\}^{1/2} \cos\phi_{kl}$$
$$E[I_l(k)] = \{2E_{kl}/N_o\}^{1/2} \sin\phi_{kl} \quad (1)$$

where E_{kl} is the energy and ϕ_{lk} is the phase of the kth tone during the lth signal baud and the white noise has power spectral density $N_o/2$ [See the Appendix].

The bit error rate for MFQPSK is identical to the bit error rate of ordinary QPSK and the symbol error rate of the 3 phase bits of MF16-QAM is the same as the symbol error rate of ordinary 8 PSK given the same E_{kl}/N_o.

3. DIFFERENTIAL CODING IN THE FREQUENCY DOMAIN

Demodulation of MFM is strictly coherent and requires that phase synchronization between the transmitter and receivers be maintained for each of the multiplicity of carrier frequencies in the MFM signal. For links involving radio frequency (or acoustic) propagation between the transmitting and receiving microcomputers these requirements may be difficult or impossible to meet. In such cases, differential encoding should be employed.

In Multi-frequency Differential Modulation (MFDM), symbols are differentially encoded within each baud between adjacent tones. The differential encoding algorithims that we employ are given in Table I. The first digital frequency, k_1, is always assigned state S_0. K+1 digital frequencies are sent with each signal baud. At the receiver, following the DFT, the complex product between the DFT coefficient of digital frequency k and the complex conjugate of the DFT coefficient of digital frequency k-1 is formed. In the case of MFDQPSK, the result is multiplied by $\exp(j\pi/4)$; in the case of MFD16-QAM, the result is multiplied by $\exp(j\pi/8)$. Consideration of Table I shows that this realigns the differentially encoded phase-bits to the constellations of Figure 1. It is shown in the Appendix that the 2KL values thus obtained are approximately gaussian random variables and that after normalization by dividing by their standard deviations they have mean values given by

$$E[R_l(k)] = A_k \cos\phi_{kl}$$
$$E[I_l(k)] = A_k \sin\phi_{kl} \quad (2)$$

with,

$$A_k = \{2E_kE_{k-1}/N_o(E_k+E_{k-1}+N_o)\}^{1/2}. \quad (3)$$

Comparing (1) and (2), we see that when adjacent frequencies have equal E_k there will be a theoretical loss of 3 db in output signal-to-noise ratio compared to coherent demodulation of each tone separately. However, actual measurements on a prototype system show that MFDQPSK performs within one to two db of MFQPSK over an E_k/N_o range of 5 to 20 db [Ref 6]. The reasons for this are as of now not completely clear, but are believed to be due to positive correlation of the noise of adjacent tones.

As can be seen from (2), the signal-to-noise ratio for decoding the phase bits in MFD16-QAM depends on the amplitude bit. If the amplitude bit is a zero, then one amplitude is high and one is low; if the amplitude bit is one, then adjacent tones have equal amplitudes which are either both high or both low. Phase bit decoding errors are dominated by the case when both amplitudes are low. The probability of a differential phase symbol decoding error, given both amplitudes low, is [See, for example, Hakin [5], pg. 317]

$$P_{\phi e} = 2Q\{(E_{Lo}/N_o)^{1/2}\sin(\pi/8)\}. \quad (4)$$

For moderately high signal-to-noise ratios, the probability of an amplitude bit error is closely approximated by

$$P_{Ae} = 1.5Q\{(E_{Lo}/N_o)^{1/2}3/8\} \quad (5)$$

where we have used high energy symbols with 25/4 the energy of the low ones. Overall, the symbol error probability for MFD16-QAM is bounded by

$$P_{se} \leq P_{Ae} + P_{\phi e}. \quad (6)$$

Figure 3 shows theoretical upper bounds for 4-bit symbol error probabilities for MFDQPSK and MFD16-QAM versus average E_b/N_o. MFQPSK is shown for comparison.

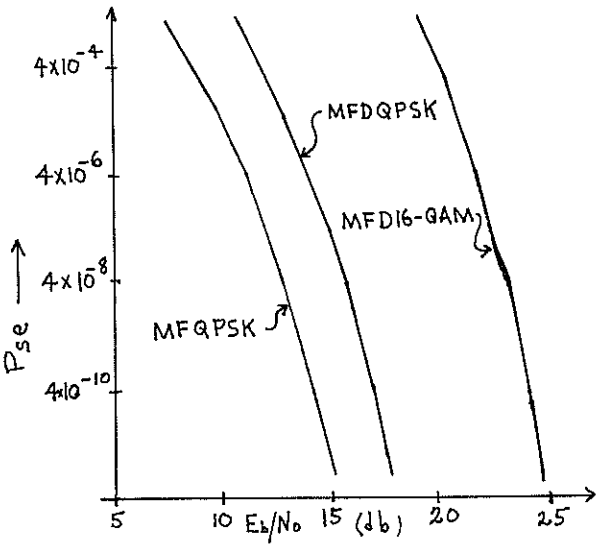


Figure 3
MFDM 4 Bit Symbol
Error Probabilities

INPUT DATA FOR MFD16-QAM/MFDQPSK	STATE FOR TONE k GIVEN TONE k−1 IS IN STATE S_{2n}/S_{2n+1}
0000	S_{2n+1}/S_{2n}
0001/00	S_{2n}/S_{2n+1}
0010	S_{2n+3}/S_{2n+2}
0011	S_{2n+2}/S_{2n+3}
0110	S_{2n+5}/S_{2n+4}
0111/01	S_{2n+4}/S_{2n+5}
0100	S_{2n+7}/S_{2n+6}
0101	S_{2n+6}/S_{2n+7}
1100	S_{2n+9}/S_{2n+8}
1101/11	S_{2n+8}/S_{2n+9}
1110	S_{2n+11}/S_{2n+10}
1111	S_{2n+10}/S_{2n+11}
1010	S_{2n+13}/S_{2n+12}
1011/10	S_{2n+12}/S_{2n+13}
1000	S_{2n+15}/S_{2n+14}
1001	S_{2n+14}/S_{2n+15}

TABLE I
(Index addition modulo 16)
Differential Encoding

4. CONCLUSIONS

Multi-frequency modulation is an extremely robust, bandwidth efficient technique for digital data communications. It relys on DFT algorithims to modulate and demodulate the data. The practical application of MFM is greatly enhanced by differentially encoding the information to be transmitted between adjacent digital frequencies. Differential coding\decoding algorithims have been described and theoretical performance results have been given herein for MFDQPSK and MFD16-QAM. The theoretical performance is approximately 3db lower than for non-differential MFM. Actual data suggest these results may in fact be too pessimistic; however, even with the 3 db reduction, differential coding should be employed because it reduces or eliminates the need for channel equalization.

APPENDIX

Let the lth baud of an MFM signal at the receiver be given by

$$y(t) = x(t) + w(t) \; ; \; 0 \le t \le \Delta T \qquad (7)$$

with,

$$x(t) = \Sigma (2E_k/\Delta T)^{1/2} \cos[2\pi kt/\Delta T + \phi_k] \qquad (8)$$

and w(t) white gaussian lowpass noise in the band 0 to $k_x/2\Delta T$ with power spectral density $N_o/2$. E_k is the received signal energy and ϕ_k is the phase of tone k during the lth baud. Sampling (7) at $\Delta T/k_x$ samples per second produces the discrete time signal

$$y(n) = x(n) + w(n) \; ; \; 0 \le n \le k_x - 1 \qquad (9)$$

with,

$$x(n) = \Sigma (2E_k/\Delta T)^{1/2} \cos[2\pi kn/k_x + \phi_k] \qquad (10)$$

and white noise sequence w(n) with zero mean and variance $N_o k_x/2\Delta T$. The k_x point DFT of (10) is the frequency domain random complex sequence

$$Y(k) = X(k) + W(k) \; ; \; 0 \le k \le \tfrac{1}{2} k_x - 1 \qquad (11)$$

with mean value,

$$X(k) = \tfrac{1}{2} (2E_k/2)^{1/2} k_x \exp[j\phi_k] \qquad (12)$$

and white gaussian complex noise sequence with uncorrelated real and imaginary parts and with

$$\mathrm{Var}\{\mathrm{Re}[W(k)]\} = \mathrm{Var}\{\mathrm{Im}[W(k)] = \tfrac{1}{4} k_x^2 N_o/\Delta T. \qquad (13)$$

The properties of the noise components of the DFT coefficients follow directly from the properties of w(n) and the definition of the DFT. Dividing (12) by the square root of (13) yields a frequency domain random sequence

$$F(k) = R(k) + jI(k) \qquad (14)$$

with unit variance, gaussian uncorrelated real and imaginary parts and with data dependent mean values given by (1).

Differential decoding the phase bits is accomplished by computing the random data sequence

$$D(k) = F(k)F^*(k-1)\exp[j\pi/2^m] \; ; \; k_1+1 \le k \le k_2 \qquad (15)$$

for the m bits that are differentially encoded as phase between adjacent digital frequencies in each MFDM baud. From the properites of W(k), it follows directly that the mean value of D(k) is given by

$$E[D(k)] = 2(E_k E_{k-1})^{1/2} N_o \exp[j\theta_k] \qquad (16)$$

where θ_k is the phase of the differentially encoded phase bits realigned to the constellations of Figure 1. The real and imaginary parts of D(k) are uncorrelated and have equal variances

$$\mathrm{Var}\{\mathrm{Re}[D(k)]\} = \mathrm{Var}\{\mathrm{Im}[D(k)]\} = 2(E_k + E_{k-1} + N_o). \qquad (17)$$

Dividing the real and imaginary parts of (15) by the square root of (17) yields the unit variance differentially demodulated random variables with mean values given by (2).

™ A trademark of Mercury Digital Communications, Inc.

REFERENCES

[1] S. B. Weinstein and P. M. Ebert, " Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform", **IEEE Trans. on Comm. Tech., Vol. Com-19, No. 5,** Oct. 1971.

[2] M. Alard and R. Halbert, "Principles of Modulation and Channel Coding for Digital Broadcasting for Mobile Receivers", **EBU Review, No. 224**, August 1987.

[3] L. J. Cimini, Jr. "Analysis and Simulation of a Digital Mobile Channel Using Orthogonal Frequency Division Multiplexing", **IEEE Trans. on Comm.,Vol. Com-33, No. 7**, July 1985.

[4] B. Hirosaki, "An Orthogonaly Multiplexed QAM System Using the Discrete Fourier Transform", **IEEE Trans. on Comm., Vol Com-29, No. 7**, July 1981.

[5] Simon Hakin, **Digital Communications**, John Wiley and Sons, New York,1988.

[6] T. K. Gantenbein,Implementation of Multi-Frequency Modulation on an Industry Standard Computer, **MSEE Thesis**, NPS, Monterey, CA., March 1990.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

The Application of Digital Signal Processing in Mobile Radio Transceiver Design

W.J. Whitmarsh, A. Bateman and J.D. Marvill
University of Bristol
Centre for Communications Research
Queens Building, University Walk,
Bristol BS8 1TR, United Kingdom
Tel: +44 272 303104, Fax: +44 272 255265

Abstract

Digital signal processing technology is playing an increasingly important role in the design of mobile communications equipment. This paper describes the algorithms used in the implementation of a narrowband linear modulation radio system. Pilot-aided correction of mobile channel distortions is outlined and a discussion of some of the practical implications of the algorithms, such as processor loading is given.

Introduction

In many engineering applications, digital signal processing has revolutionised design philosophy, a prime example being radio communications systems, where DSP is helping to meet the demand for small, efficient, low cost equipment. In many radio designs, emphasis is now being placed on simplifying RF circuitry by making use of complex baseband signal processing techniques which can potentially be implemented at low cost. As a result, channel filtering, spectral estimation, signal detection, speech coding, channel equalisation, waveform generation and amplifier linearisation are all part of a growing list of functions being implemented through DSP in a typical radio system.

A great deal of interest is shown at present in mobile communications systems such as cellular and cordless telephony, radio paging, private mobile radio and mobile satellite communications. Such topics form the motivation behind the work described here, although applications in other areas of radio technology are numerous.

At the University of Bristol, major advances have been made in the exploitation of linear modulation methods, that is modulation schemes that utilise both the amplitude and phase of the RF carrier to transfer information. This has involved extensive use of signal processing technology. Linear modulation exhibits a high spectral efficiency, but equipment has historically proved too costly and complex to compete in a commercial marketplace. By redesigning the radio architecture to maximise the contribution of the baseband signal processing, equipment cost has been dramatically reduced to fall in line with that of conventional mobile radio equipment.

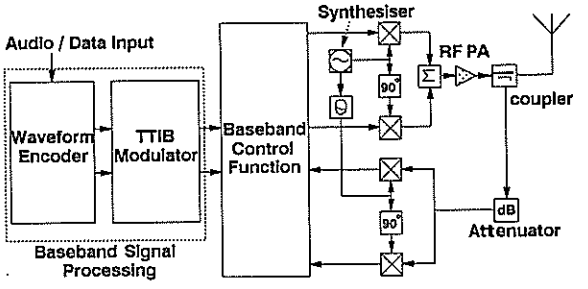


Figure 1: Linear Modulation Transmitter

This paper describes the processing techniques exploited in the new radio architecture, ranging from adaptive predistortion of the modulation signal to reduce nonlinearities in the RF power amplifier, through to channel acquisition and equalisation techniques used in the receiver to improve performance in a mobile channel.

The Transmitter

A block diagram of the transmitter is given in figure 1. It consists of low complexity direct conversion RF circuitry to translate the modulation up to the operating frequency. Amplifier non-linearities are reduced by the baseband control function, which in turn is driven from a waveform encoder and transparent tone in band (TTIB) modulator. The system is designed for use with pilot-aided linear modulation, which justifies the inclusion of TTIB technology [1,2], TTIB provides the most effective means of adding a pilot tone without interference to the message signal.

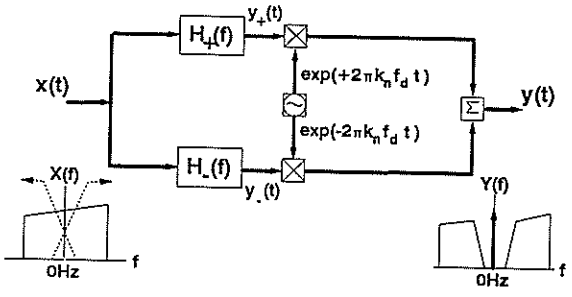


Figure 2: A TTIB Modulator

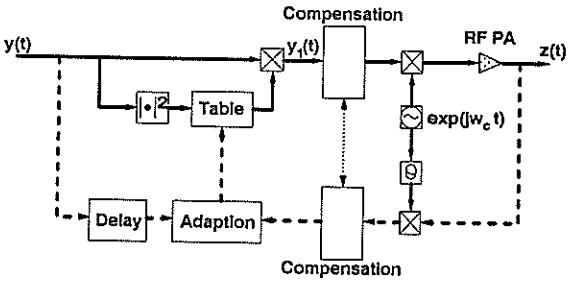


Figure 3: Linearisation of an RF Amplifier by Adaptive Predistortion

Baseband Signal Processing

The waveform encoder shown in figure 1 implements any speech coding and data modulation techniques that are used. The remainder of the transmitter is intended to be transparent to the signal format employed in the waveform encoder, so the details of this part of the design will not be considered further.

The structure of the TTIB modulator is shown in figure 2, with all signal designations representing complex baseband signals. A pair of complex filters, $H_+(f)$ and $H_-(f)$ select primarily positive and negative frequencies respectively. These filters are mirrored so that:

$$H_+(f) = H_-(-f) \tag{1}$$

To prevent intersymbol interference on data transmissions, filters $H_\pm(f)$ must exhibit a linear phase response. To achieve this, FIR techniques are used in their design.

A gap is created at 0Hz by mixing with the complex carriers $\exp(\pm j2\pi k_n f_d t)$. f_d is the data bit clock (if transmitting data) and k_n is a constant scaling factor selected to give a suitable TTIB gap width. As shown later, f_d can be recovered in the receiver to provide bit synchronisation. A pilot, P_k is added to the TTIB signal in the gap by adding a DC level to the output. The frequency domain representation of the resulting signal, $y(t)$, is included in figure 2 and is given by:

$$\begin{aligned} Y(f) \quad &= X(f - k_n f_d)H_+(f - k_n f_d) + P_k\delta(f) \\ &\quad + X(f + k_n f_d)H_-(f + k_n f_d) \end{aligned} \tag{2}$$

$\delta()$ is the Dirac delta function. Alternatively, if $y_+(t)$ and $y_-(t)$ are the upper and lower sub-bands respectively then:

$$y(t) = y_+(t) + P_k + y_-(t) \tag{3}$$

Amplifier Linearisation

The baseband control function attempts to minimise the distortions caused by the RF power amplifier. Any non-linearities will cause a broadening of the transmitted signal in the frequency domain which will result in adjacent channel interference. In many system designs, considerable effort is dedicated to minimising such interference. Typically, the radiated spectral density in the adjacent channel would need to be 80dB below the average transmitted carrier power.

Successful correction of the RF amplifier has been achieved at Bristol using feedback amplifier linearisation techniques. In a feedback system, the baseband control function may take a number of forms, the simplest being the cartesian feedback system [3] in which the control function consists of a pair of lowpass differential amplifiers. The gain-bandwidth product of the differential amplifiers must be very high and as a result there is little point in digitising the signals. Thus the control loop need not enter the DSP domain at any point.

A well known alternative is adaptive predistortion, which exhibits greater stability than cartesian feedback, but requires considerable processor bandwidth. This method involves mapping the input signal, $y(t)$, into a predistorted signal, $y_1(t)$, which is used to drive the amplifier. The mapping is done so that the amplifier output, $z(t)$ matches a frequency translated version of $y(t)$ with as little distortion as possible. The characteristics of the amplifier may alter with time, so the mapping is made adaptive to track any changes.

Many forms of predistorter have been published, each of which exhibits a different trade-off between processor bandwidth, memory requirements and adaptation speed. A gain-based system [4] is considered here as both the processor load and memory requirements are reasonably low. Figure 3 shows such a predistorter, again all signal designations refer to either complex baseband signals or to the complex envelope of bandpass signals.

The amplifier can be modelled by a level dependent complex gain, $G()$ so that the output, $z(t)$ is related to the input, $y_1(t)$ by

$$z(t) = y_1(t)G(|y_1(t)|^2) \quad (4)$$

The predistorter mapping, $F()$, is given by:

$$y_1(t) = y(t)F(|y(t)|^2) \quad (5)$$

The predistorter table shown in figure 3 consists of a list of complex gain factors used to generate $F()$. Ideally the predistorter and amplifier will combine to form a nominal constant amplitude gain, G_a. To achieve this, we must meet:

$$F(|y(t)|^2)G(|y(t)|^2\left|F(|y(t)|^2)\right|^2) = G_a \quad (6)$$

Different adaptation algorithms exist for obtaining $F()$ to approximate 6. These are carried out by the block marked ADAPTION in figure 3. Cavers gives a particularly fast algorithm in [4]. He also shows that adequate performance can be obtained with less than 100 table entries when using a class AB power amplifier.

The gain and phase match of the complex downconverter limits the image rejection at the transmitter output. 40 dB image rejection requires a phase match of 1 degree and an amplitude match of 0.1 dB. This can be achieved over a reasonable bandwidth using RF techniques alone. To extend the bandwidth further, an automatic calibration procedure can be carried out in production to calculate correction factors for any gain and phase mismatch. In service, the baseband processing can use these to correct the downconverted signals prior to use for adaptation. In the transmitter, DC offsets and carrier feedthrough will coincide with the pilot tone and can easily be made small enough not to dominate (offsets 45 dB down on PEP can be achieved, while the pilot is often around 10 dB down). Gain and phase matching and DC nulling are carried out by the functions marked COMPENSATION in figure {preblk

As a final point of interest, it should be noted that the downconverter circuitry can be shared between the transmitter and receiver when operating a system in half-duplex mode. In this way a considerable cost saving can be made.

The Receiver

A block diagram of the receiver is given in figure 4. It consists of a Weaver method downconverter [5], followed by translation to the digital domain using a delta-sigma ADC, followed by a number of baseband processing algorithms. A brief description of each algorithm is given in this section of the paper.

Channel Filtering and Signal Capture

The spurious free dynamic range of the receiver is limited by the ADC employed to digitise the baseband signal. To

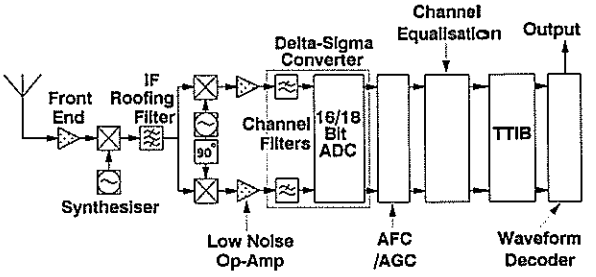


Figure 4: The Linear Modulation Receiver

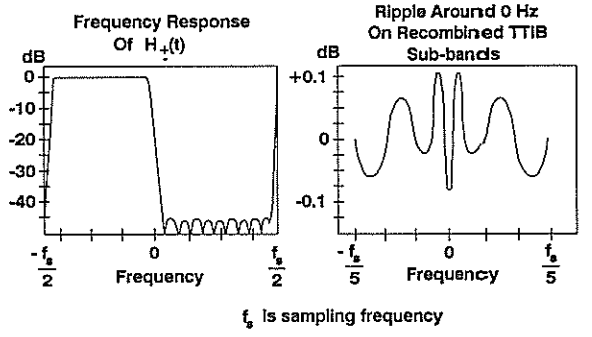


Figure 5: Comparison of Gain and Group Delay Characteristics of Digital and Crystal Channel Filters

maximise the dynamic range, a dual 16bit delta-sigma converter, the Crystal CS5326 has been used. This device decimates internally from a high sample rate, thus removing the need for tight analogue anti-alias filters. The decimation process yields a fast rolloff lowpass filter on each input signal which is suitable for use as a channel filter.

Figure 5 compares such a digital channel filter with a more conventional crystal filter (as used at IF in many narrowband radio systems). It can be seen that the digital version exhibits a more well defined selectivity performance than the crystal filter. The digital version also has a linear phase response which improves performance when data modulation is received.

Automatic Gain Control and Automatic Frequency Control

If all signals are to be accommodated at least 12 dB above the noise floor then the spurious free dynamic range provided by a 16 bit ADC is $96 - 12 = 86$ dB. Typically, a receiver will be expected to operate over a full dynamic range of 120 dB. To achieve this, some form of gain control will be needed in the downconverter. This can be achieved using continuous feedback AGC [6] under processor con-

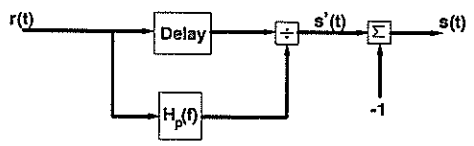


Figure 6: Principle of Feedforward Signal Regeneration

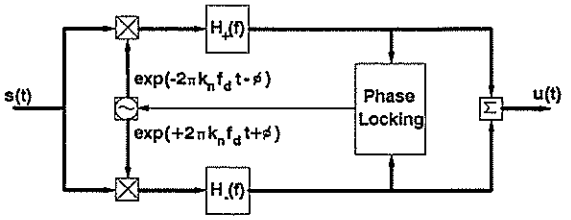


Figure 7: TTIB Demodulator

trol, however the dynamics of such a system are very slow. We have maximised the speed of the gain control by using a simple switched scheme. The gain of the low noise op-amps shown in figure 4 is switched (with sufficient filtering to prevent a wideband transient) between a high gain setting and a low gain setting. The absolute level of the two settings differs by 30 dB. Switching is dependent on the average received signal strength.

The receiver is used with pilot-aided linear modulation. As such, it is important to accurately track the frequency of the pilot and to control the output of the frequency synthesiser used on the system. This is achieved with an automatic frequency control loop (AFC). The purpose of AFC is to measure the average frequency of the pilot, and to control the synthesiser so as to hold that average frequency in one spot. This is achieved using a frequency locked loop [7] which has been found to perform well when the pilot is fading.

Channel Equalisation

If we assume that multipath distortions are identical across the channel bandwidth (*ie* signal fading is frequency flat) then feedforward signal regeneration (FFSR) [8] is a simple and effective means of removing such distortion. Figure 6 shows the principle. All designations are of complex baseband signals.

A mobile channel can be modelled by multiplying the transmitted signal by complex noise, $x_n(t)$, and adding thermal noise, $n(t)$, both of which are stationary Gausian random variables. The received signal, $r(t)$, is given by:

$$r(t) = x_n(t)y(t) + n(t) \tag{7}$$

Combining (3) and (7):

$$r(t) = x_n(t)y_+(t) + x_n(t)P_k + x_n(t)y_-(t) + n(t) \tag{8}$$

The filter, $H_p(t)$, is designed to isolate the pilot term in (8) leaving $z_p(t)$. Thus:

$$z_p(t) = x_n(t)P_k + n_p(t) \tag{9}$$

$n_p(t)$ is a contribution from the thermal noise. For this explanation we assume $n(t)$ and $n_p(t)$ are insignificant so that the signal, $s'(t)$ is given by:

$$s'(t) = \frac{x_n(t)y_+(t) + x_n(t)P_k + x_n(t)y_-(t)}{x_n(t)P_k} \tag{10}$$

Simplifying and subtracting one (the corrected pilot) leaves $s(t)$, a bandsplit and scaled version of the input signal with channel distortions removed:

$$s(t) = \frac{y_+(t)}{P_k} + \frac{y_-(t)}{P_k} \tag{11}$$

FFSR can remove any multiplying distortions experienced by the signal that lies within the bandwidth of $H_p(f)$, thus it reduces the effect of synthesiser phase noise, and removes Doppler shift and multipath distortions introduced by the mobile radio channel.

TTIB demodulator

The signal $s(t)$ is still in a bandsplit form, and a TTIB demodulator is required to recombine the two sub-bands. A diagram of such an algorithm is given in figure 7. It is largely a reverse of the TTIB modulator:

$s(t)$ is simply shifted in either direction by the oscillators $\exp(\pm 2\pi k_n f_d t - \phi)$ and then the relevant sub-bands are filtered off by $H_\pm(f)$ before recombination. ϕ is the resulting recombination phase error. For data transmission, ϕ, may result in unacceptable distortion and so the phase locking function shown in figure 7 is included. This controls the oscillator so as to minimise ϕ. Full details of the algorithm will not be given here. Consult [2,9] for an analysis of phase locking operation.

The filters $H_\pm(t)$ are the same as in the transmitter. To produce a flat output spectrum, they should be designed to meet:

$$|H_+(f)|^2 + |H_-(f)|^2 = 1 \tag{12}$$

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

TWO DSP METHODS FOR BANDWIDTH EFFICIENT OQPSK-TYPE TRANSMISSION THROUGH NONLINEAR AMPLIFIERS

A. Gusmão, N. Esteves

CAPS, IST, Lisboa, Portugal

In this paper, we describe two methods of digital signal processing for dealing with nonlinear amplification of variable-envelope OQPSK-type signals. In both methods, the encoding rules governing the digital signal processing are based on the ENCAP-OQPSK format proposed elsewhere by the authors.

1. INTRODUCTION

OQPSK-type (MSK-type) digital modulations are often recommended for applications in land mobile and satellite communication systems [1-3]. With most of these modulations (e.g., GMSK, recently adopted for the Pan-European public land mobile system), the modulated wave has a constant envelope, thus allowing an efficient (nonlinear) power amplification without any spectral spreading. However, the constant envelope constraint turns out to be a serious limitation when searching for better power-bandwidth tradeoffs. Better tradeoffs can be achieved if some envelope fluctuation is allowed, but, so far, variable-envelope schemes have generally been considered as precluding the use of nonlinear power amplifiers. Stated in another way, the usual preference for constant-envelope schemes is not due to any inherent superiority of those schemes, but rather to their easy compatibility with nonlinear RF amplifier technology.

In this paper, we present two methods of digital signal processing specially designed for the transmission of nonlinearly amplified OQPSK-type signals with a variable envelope. In both methods, the transmitted signals not only yield a high bandwidth efficiency (towards 2 bps/Hz if desired), but also allow a good detection efficiency, provided the encoding procedure is adequately employed. The proposed methods for efficient use of power amplifiers can be thought of as applications of digital signal processing to "transmitter linearization", since they circumvent the nonlinear distortion problem. In both methods, the encoding rules governing the digital processing are based on the ENCAP-OQPSK format (ENCoded-Amplitude-and-Phase) [4,5]. The digital implementation of these encoding rules, through ROM-based, table look-up processing, is straightforward, and allows a simple, flexible and cost effective design for a programmable transmitter.

Sec. 2 deals with the method described in detail in [6]. In Sec. 3, we explain the encoding rules for the other method. To conclude, some final remarks are made in Sec. 4.

2. PREDISTORTION METHOD

In this method, the transmitter follows the model shown in Fig. 1 (see also [6]). For a variable-envelope transmission, the digital baseband encoding must be a modification of the ENCAP-4^N encoding reported in [5]; the modified encoding rules can be regarded as a predistortion compensating for the AM/PM and AM/AM conversions due to the power amplifier, described as a bandpass memoryless nonlinearity (Of course, the amplifier characteristics, as well as the driving level, must be known to the encoder.) Whenever the compensation of the AM/AM conversion is possible, it is also possible to derive from the amplifier, even operating close to saturation, an OQPSK-type signal belonging to the ENCAP-OQPSK format, which not only exhibits a compact power spectrum but also allows a good detection efficiency [6].

In the transmitter of Fig. 1, the modulator generates a bandpass signal $s(t) = A(t)\cos[2\pi f_c t + \theta_c + \phi(t)] = I(t)\cos(2\pi f_c t + \theta_c) - Q(t)\sin(2\pi f_c t + \theta_c)$ $(A(t) \geq 0)$

By applying $s(t)$ to a power amplifier, an output signal $s_o(t) = G\, s_T(t)$ is obtained, in which G is a constant gain factor and

$$s_T(t) = A_T(t)\cos[2\pi f_c t + \theta_c + \phi_T(t)] \qquad (1)$$

Let $f_{NL}(.) = G\, f_{NL}^{(0)}(.)$ and $g_{NL}(.)$ be respectively the AM/AM and AM/PM conversion functions of the amplifier. If we want $s_T(t)$ to be an ENCAP-OQPSK signal [4,5], the envelope and phase functions of $s(t)$ must be such that

$$A(t) = f_{NL}^{(1)}[A_T(t)] \qquad (2a)$$

$$\phi(t) = \phi_T(t) - g_{NL}\left[f_{NL}^{(1)}(A_T(t))\right] \quad (2b)$$

where $f_{NL}^{(1)}(.)$ denotes the inverse function of $f_{NL}^{(0)}(.)$.

In view of this, the I and Q components of s(t) must be given by

$$I(t) = I^{(1)}(t)\cos\left\{g_{NL}\left[f_{NL}^{(1)}(A_T(t))\right]\right\} + Q^{(1)}(t)\sin\left\{g_{NL}\left[f_{NL}^{(1)}(A_T(t))\right]\right\} \quad (3a)$$

$$Q(t) = Q^{(1)}(t)\cos\left\{g_{NL}\left[f_{NL}^{(1)}(A_T(t))\right]\right\} - I^{(1)}(t)\sin\left\{g_{NL}\left[f_{NL}^{(1)}(A_T(t))\right]\right\} \quad (3b)$$

with

$$I^{(1)}(t) = f_{NL}^{(1)}[A_T(t)]\cos[\phi_T(t)] \quad (4a)$$

$$Q^{(1)}(t) = f_{NL}^{(1)}[A_T(t)]\sin[\phi_T(t)] \quad (4b)$$

As explained in [6], the baseband signals $I^{(1)}(t)$ and $Q^{(1)}(t)$ can be regarded as the baseband I and Q components of the ENCAP-OQPSK signal that would result from passing $s_T(t)$ through the "inverse nonlinearity" defined by $f_{NL}^{(1)}(.)$, with $g_{NL}^{(1)}(.) = 0$. Therefore, s(t) can be generated by means of a digital encoder structurally identical to an ENCAP-4^N encoder [5]; the ENCAP-4^N encoding rules must be modified, of course, according to eqs (3) and (4), that is, according to the characteristics of the nonlinearity and the driving level [6].

It should be pointed out that the compensation of the AM/AM conversion is possible if $f_{NL}^{(0)}(.)$ - and, consequently, $f_{NL}^{(1)}(.)$ - are single-valued, increasing functions over the intervals of variation of the envelopes. Moreover, by adjusting the value of the gain factor G, it may be possible to simplify the implementation of the digital encoder. In fact, the function $f_{NL}^{(1)}(.)$ affecting the data to be stored in the encoder ROMs is not the inverse of $f_{NL}(.)$, but the inverse of $f_{NL}^{(0)}(.) = f_{NL}(.)/G$.

3. COMBINING METHOD

Whenever the compensation of the AM/AM conversion is not practical (e.g., if the AM/AM conversion function of the amplifier is very close to that of a hardlimiter) an alternative method of transmitter linearization can be used. This second method (see Fig. 2) can be thought of as a simultaneous application of the ENCAP-OQPSK format and the LINC technique (Linear amplification with Nonlinear Components) reported in [7]. With this method, two constant-envelope signals, generated through baseband processing, are separately amplified - of course, without any nonlinear distortion, either of them - and then combined to produce the signal to be transmitted. Provided the desired signal is ENCAP-4^N, these encoding rules can be derived from the ENCAP-4^N encoding rules reported in [5]. With this combining method, the transmitter structure is as shown in Fig. 2. The signal to be transmitted is

$$s_0(t) = I_0(t)\cos(2\pi f_c t + \theta_c) - Q_0(t)\sin(2\pi f_c t + \theta_c) = A_0(t)\cos[2\pi f_c t + \phi_0(t) + \theta_c] \quad (5)$$

where the envelope is such that $0 < A_0(t) \leq A_M$. Hence, a function $\theta_0(t)$, such that $0 < \theta_0(t) \leq \pi/2$, can be defined, for

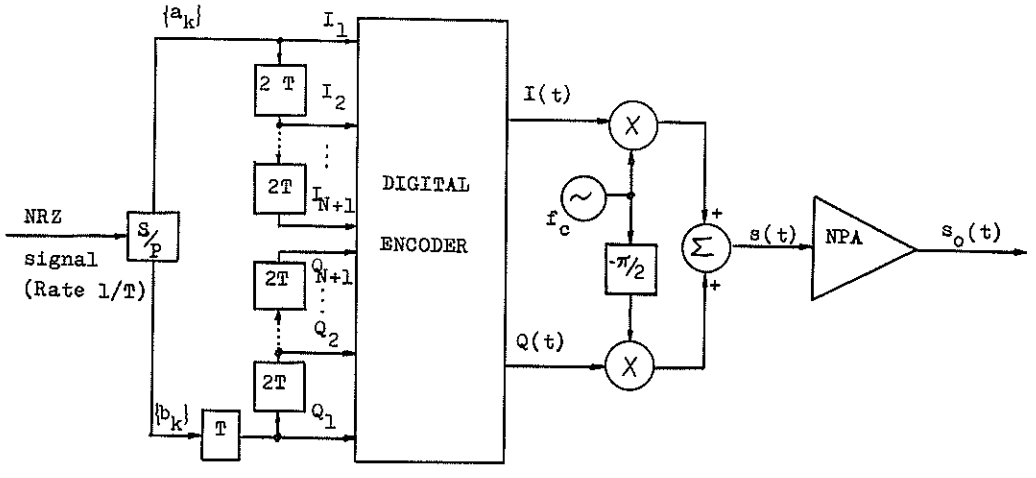


Fig. 1 - Block diagram of an OQPSK-type transmitter using the predistortion method (a_k, b_k = ± 1; NPA : nonlinear power amplifier).

which

$$A_0(t) = A_M \sin [\theta_0(t)] \qquad (6)$$

From eqs (5) and (6), it follows that

$$s_0(t) = G_A[s_X(t) + s_Y(t)] \qquad (7)$$

with

$$s_X(t) = \frac{A_m}{2} \sin [2\pi f_c t + \phi_0(t) + \theta_c + \theta_0(t)] \qquad (8a)$$

$$s_Y(t) = -\frac{A_m}{2} \sin [2\pi f_c t + \phi_0(t) + \theta_c - \theta_0(t)] \qquad (8b)$$

The constant envelope $A_m/2$ is such that $A_m = A_M/G_A$, where the gain factor $G_A = \frac{2}{A_m} f_{NL} (A_m/2)$ is assumed to have the same value for both power amplifiers. Another implicit assumption is that $g_{NL} (A_m/2) = 0$ for both amplifiers. For any other value $g_{NL} (A_m/2)$, common to both amplifiers, $s_0(t)$ would simply suffer a constant phase shift. Further manipulation of the above expressions will give :

$$s_X(t) = I_X(t) \cos (2\pi f_c t + \theta_c) - Q_X(t) \sin (2\pi f_c t + \theta_c) \quad (9a)$$

$$s_Y(t) = I_Y(t) \cos (2\pi f_c t + \theta_c) - Q_Y(t) \sin (2\pi f_c t + \theta_c) \quad (9b)$$

in which

$$I_X(t) = \frac{1}{2G_A} [I_0(t) + Q_0(t) \operatorname{cotg} [\theta_0(t)]] \qquad (10a)$$

$$Q_X(t) = \frac{1}{2G_A} [Q_0(t) - I_0(t) \operatorname{cotg} [\theta_0(t)]] \qquad (10b)$$

$$I_Y(t) = \frac{1}{2G_A} [I_0(t) - Q_0(t) \operatorname{cotg} [\theta_0(t)]] \qquad (10c)$$

$$Q_Y(t) = \frac{1}{2G_A} [Q_0(t) + I_0(t) \operatorname{cotg} [\theta_0(t)]] \qquad (10d)$$

Eq (6) implies that

$$\operatorname{cotg} [\theta_0(t)] = \sqrt{\frac{A_M^2}{A_0^2(t)} - 1} \qquad (11)$$

Note that, for a constant-envelope signal $s_0(t)$, that is, for $A_0(t) = A_M$, we would have : $I_X(t) = I_Y(t) = I_0(t)/(2G_A)$; $Q_X(t) = Q_Y(t) = Q_0(t)/(2G_A)$.

Suppose now that we want $s_0(t)$ to be an ENCAP-4^N signal, with $I_0(t)$ and $Q_0(t)$ obeying the encoding rules presented in [5] for a given set of 4^N generating functions, $\{y_i(t), i = 0, 1, ..., 4^{N-1}\}$, defined in the interval (0,T). In view of eqs. (6-11), the encoding rules can be stated as follows, with $A_M = \max_{i,i',t \in (0,T)} \sqrt{y^2_i (t-2nT) + y^2_{i'} [(2n+1)T-t]}$,

$i = (\gamma_{2N} \gamma_{2N-1} ... \gamma_1)_2$ and $i' = (\gamma_1 \gamma_2 ... \gamma_{2N})_2$:

$$I_{X,2n}(t) = \frac{1}{2G_A} a_n y_j (t-2nT) + \frac{1}{2G_A} b_n y_k [(2n+1)T-t] \sqrt{\frac{A_M^2}{A_0^2(t)} - 1} \qquad (12a)$$

$$Q_{X,2n}(t) = \frac{1}{2G_A} b_n y_k [(2n+1)T-t] - \frac{1}{2G_A} a_n y_j (t-2nT) \sqrt{\frac{A_M^2}{A_0^2(t)} - 1} \qquad (12b)$$

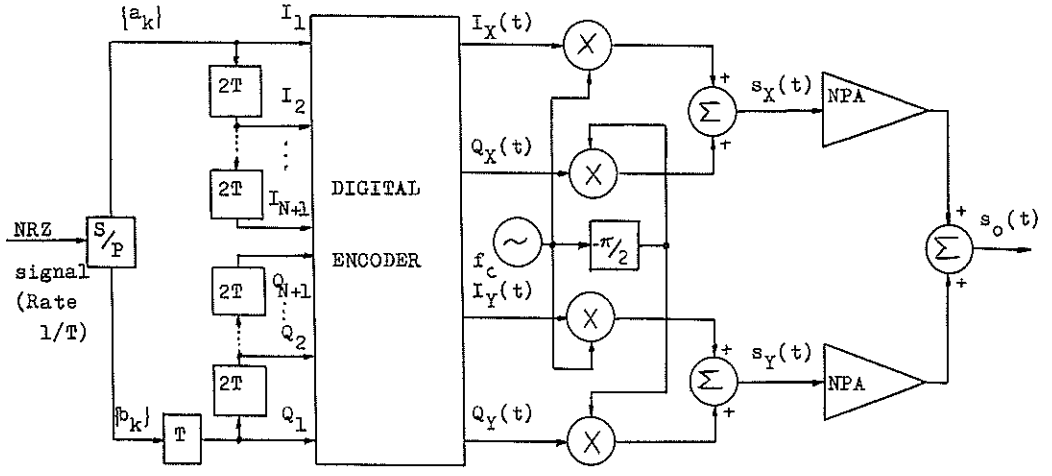


Fig. 2 – Block diagram of an OQPSK-type transmitter using the combining method (a_k, $b_k = \pm 1$; NPA : nonlinear power amplifier).

$$I_{Y,2n}(t) = \frac{1}{2G_A}\, a_n y_j\,(t-2nT) - \frac{1}{2G_A}\, b_n y_k\,[(2n+1)T-t]\sqrt{\frac{A_M^2}{A_0^2(t)} - 1} \qquad (12c)$$

$$Q_{Y,2n}(t) = \frac{1}{2G_A}\, b_n y_k\,[(2n+1)T-t] + \frac{1}{2G_A}\, a_n y_j\,(t-2nT)\sqrt{\frac{A_M^2}{A_0^2(t)} - 1} \qquad (12d)$$

with $A_0(t) = \sqrt{[y_j^2\,(t-2nT) + y_k^2\,[(2n+1)T-t]}$ and

$$I_{X,2n+1}(t) = \frac{1}{2G_A}\, a_{n+1}\, y_l\,[(2n+2)T-t] + \frac{1}{2G_A}\, b_n y_m\,[t-(2n+1)T]\sqrt{\frac{A_M^2}{A_0^2(t)} - 1} \qquad (13a)$$

$$Q_{X,2n+1}(t) = \frac{1}{2G_A}\, b_n y_m\,[t-(2n+1)T] - \frac{1}{2G_A}\, a_{n+1}\, y_l\,[(2n+2)T-t]\sqrt{\frac{A_M^2}{A_0^2(t)} - 1} \qquad (13b)$$

$$I_{Y,2n+1}(t) = \frac{1}{2G_A}\, a_{n+1}\, y_l\,[(2n+2)T-t] - \frac{1}{2G_A}\, b_n y_m\,[t-(2n+1)T]\sqrt{\frac{A_M^2}{A_0^2(t)} - 1} \qquad (13c)$$

$$Q_{Y,2n+1}(t) = \frac{1}{2G_A}\, b_n y_m\,[t-(2n+1)T] + \frac{1}{2G_A}\, a_{n+1}\, y_l\,[(2n+2)T-t]\sqrt{\frac{A_M^2}{A_0^2(t)} - 1} \qquad (13d)$$

with $A_0(t) = \sqrt{[y_l^2\,[(2n+2)T-t] + y_m^2\,[t-(2n+1)T]}$, the subscripts j, k, l and m being determined by the data sequence ("even" subsequence $\{a_k\}$ and "odd" subsequence $\{b_k\}$) as in [5].

4. CONCLUSION

In this paper, two DSP methods are presented, by means of which an efficient power amplification of OQPSK-type signals with a compact spectrum can be achieved. In both methods, an ENCAP-OQPSK transmitted signal $s_0(t)$ is assumed, which is allowed to have some envelope fluctuation. As pointed out in [6], a suitable choice for $s_0(t)$ is an ENCAP-4^N signal having an equivalent OQAM characterized by an even pulse with finite duration $2(N+1)T$.

If the adequate type of amplifier is available, the predistortion method is easier to implement and uses power amplification more efficiently than the combining method. The main advantage of the combining method lies in the fact that it allows the use of any type of power amplifier. However, a serious difficulty arising with this method is its high sensitivity to the phase error due to any difference in electrical length between the amplifying branches. A corrective feedback loop has been proposed to overcome the above difficulty [8].

- REFERENCES -

[1] - S. Gronemeyer and A. McBride, "MSK and Offset QPSK Modulation", IEEE Trans., COM-24, pp. 809-819, 1976.

[2] - H. Pham Van and K. Feher, "New Modulation Techniques for Low Cost Power and Bandwidth Efficient Satellite Earth Stations", IEEE Trans., COM-30, pp. 275-283, 1982.

[3] - K. Murota et al., "GMSK Modulation for Digital Mobile Radio Telephony", IEEE Trans., COM-33, pp. 1044-1050, 1981.

[4] - A. Gusmão and N. Esteves, "ENCAP-4 : an OQPSK-type Modulation Technique for Digital Radio", IEE Proc., Pt. F, pp. 105-110, 1988.

[5] - A. Gusmão and N. Esteves, "A Generalized ENCAP-4^N Digital Modulation Technique", to be published soon in IEE Proc.

[6] - A. Gusmão and N. Esteves, "Bandwidth Efficiency with OQPSK-type Digital Modulation and Nonlinear Amplification", ICC'90, Atlanta, April 1990.

[7] - D. Cox, "Linear Amplification with Nonlinear Components", IEEE Trans., COM-22, pp. 1942-1945, 1974.

[8] - S. Tomisato et al., "Phase Error Free LINC Modulator", Electronics Letters, Vol. 25, Nº. 9, pp. 576-577, 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

OUTAGE TIME ESTIMATION FOR MICROWAVE RADIO

Igor OZIMEK, Jurij TASIC

Jozef Stefan Institute, Jamova 39, 61000 Ljubljana, Yugoslavia

In the article the joint effect of imperfect microwave channel and transmit/receive filters is studied and expressed in terms of expected outage time. Also, the effectiveness of the use of adaptive transversal equalizer as a means to combat signal distortion is shown.

1. INTRODUCTION

Neither the microwave channel alone nor the transmit and receive filters alone are not the cause of fatal distortion of the signal. It is the joint effect of both that causes ISI (Inter-Symbol Interference) large enough to interrupt transmission of the signal. With the aid of computer simulation we will study the influence of the channel, the filters, modulation and the equalizer on the outage time with respect to the CCITT recommendations.

2. THE RUMMLER'S MODEL

The Rummler's model [1], [2], [4] originates from a common three ray model:

$$H(j\omega) = 1 + a_1 e^{-j\omega\tau_1} + a_2 e^{-j\omega\tau_2} \qquad (1)$$

After some simplifications (which can be made due to the limited bandwidth of the signal) and suitable changes of variables it gets its final form:

$$H(j\omega) = a\left[1 - b\,e^{-j(\omega-\omega_0)\tau}\right] \qquad (2)$$

In eq (2) the following three variables are used:

a represents the flat fading depth. In the ideal case (no flat fading) it has the value of 1.

b represents the selective (frequency dependent) fading depth (notch depth) and has values between 0 and 1. In the ideal case (no selective fading) it has the value of 0.

ω_0 represents the frequency of the minimum in the amplitude spectrum, caused by selective fading. Typical (amplitude) transfer characteristic of the Rummler's model is shown in fig. 1.

An example of the Rummler's model transfer function (amplitude characteristics only) is shown in fig. 1.

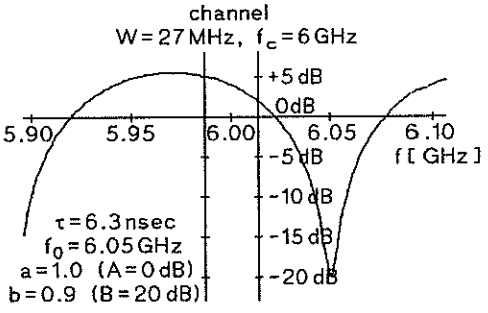


Fig. 1: selective fading characteristic

3. THE SIGNATURE CURVE

To describe the behaviour of a communication equipment in the selective fading environment, the signature curve is used. It describes the critical value of notch depth b versus notch frequency ω_0, which causes outage of a communication link.

Knowing the signature curve of a communication equipment and the statistics of the Rummler's model parameters (in selective fading case only a and ω_0) we can predict the outage time of the communication link:

$$P(1-b > X) = X^{2.3} \qquad (3)$$

$$p(f_0) = \begin{cases} \frac{5\tau}{3} & |f_0| \le \frac{1}{4\tau} \\ \frac{\tau}{3} & \frac{1}{4\tau} < |f_0| < \frac{1}{2\tau} \end{cases} \qquad (4)$$

$$(\omega_0 = 2\pi f_0)$$

The worst month outage time is then:

$$P_{OUT} = P_F \int_{-\frac{1}{2\tau}}^{\frac{1}{2\tau}} e^{-\frac{S(f)}{3.8}} p(f)\, df \qquad (5)$$

$S(f)$ is signature curve in [dB].

P_F is probability of communication channel being nonideal in the worst month of the year. Its value depends on the length of the microwave hop and on the meteorological and geographical conditions [3]. For the Rummler's case the cumulative time of nonideal channel equals 8100 seconds in the worst month.

4. THE TRANSVERSAL EQUALIZER

The choice of the equalizer is limited by the fact that with digital microwave links the trasmission rates are very high and all the computing that the equalizer requests must be done in hardware. Because of that, more sophisticated filters and algorithms (for instance lattice equalizers) are not affordable. Besides, the microwave channel is changing relatively slowly, so the speed of convergence, which is the main benefit of more complicated equalizers, is not very important.

According to these limitations the SG (Stochastic Gradient) equalizer was chosen. It uses LMS criterion for adaptation, it has relatively slow convergence (which is not so very important for our application) and it is simple to realize [5]. Eq. 6 represents the adaptive algorithm of SG equalizer.

$$\mathbf{c}_{T+1} = \mathbf{c}_T + \beta e_T \mathbf{y}_T^* \qquad (6)$$

5. THE JOINT EFFECT OF MICROWAVE CHANNEL IMPERFECTION AND TRANSMIT AND RECEIVE FILTERS

Symbol rate of the transmitting signal is limited by the channel bandwith according to the Nyquist theorem. With channel bandwith being 40 MHz maximum, symbol period is limited to 25nsec, which is much greater then average secondary ray delay according to Rummler's model (6.31 nsec). In this way a symbol could penetrate only partially to his neighbouring symbols (about 25%) and could not cause fatal ISI. These nevertheless happens because of the transmit and receive filter oscillations ("tails"). These filters are normally designed so as to achieve zero crossing of the oscillations at the neighbouring symbols sampling points (Nyquist criterion). When selective fading appears the tails begin to jitter along the time axis, no more satisfying the zero crossing criterion and in this way causing great ISI. Thus, the transfer characteristic of the transmit and the receive filter is of great importance for the end effect, which a certain degree of channel deterioration would have on the signal distortion at the receiver.

6. THE COMPUTER SIMULATION

Computer simulations have been used to simulate a typical (Rummler's) microwave hop. The filter used is Nyquist (equally divided between the transmitter and the receiver) with various roll-off factor (α).

Two different sampling criteria have been employed: maximum response criterion and minimum distortion criterion. The first one selects sampling so as to achieve maximum response at the reference symbol sample regardless of the ISI appearing at the neighbouring symbol samples. The latter one begins the same way but then trims the sampling phase so as to achieve minimum ISI at the neighbouring symbol samples at the expense of smaller response at the reference symbol sample. It minimizes the following expression:

$$D_i = \frac{\sum_{j \neq i} y_j}{y_i} \qquad (7)$$

Modulation schemes used were 4-, 16-, 64- and 256-QAM.

Symbol rate was 23 MHz (which in the case of 64-QAM corresponds to 140 Mbit/s).

The first step in simulation was to calculate signature curves for various (fixed) values of parameters. Four such curves are shown in fig. 2 (frequency f_0 is normalized with respect to the symbol frequency).

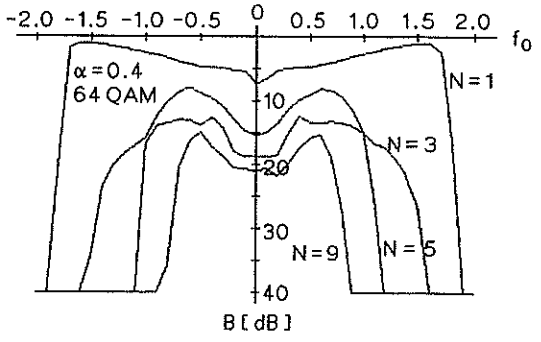


Fig. 2: signature curves

The next step was to calculate outage time from each signature curve. Figs. 3-6 show dependence of outage time on the roll-off factor (α) of the filter. Filter with smaller α has sharper transition between the pass and the stop region. This causes longer tails of the signal ("ringing") and thus greater ISI. Minimum and nonminimum-phase channel gives the same results because of the symmetry of the time response of the Nyquist filter.

The 8100s mark corresponds to the worst-month cumulative time of channel deterioration. The 22s mark corresponds to the limit imposed by CCITT G.821 recommendation for outage time of the digital transmission equipment.

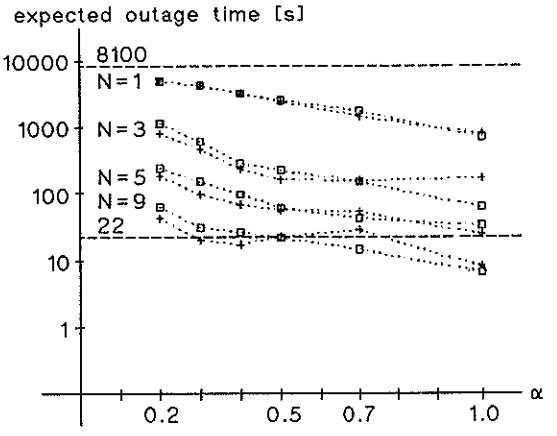


Fig. 5: outage time vs. α, 64-QAM

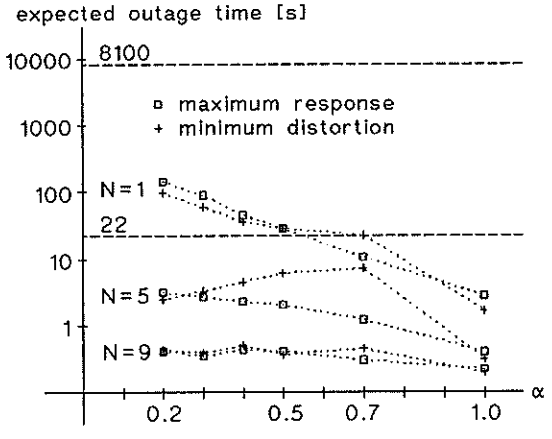


Fig. 3: outage time vs. α, 4-QAM

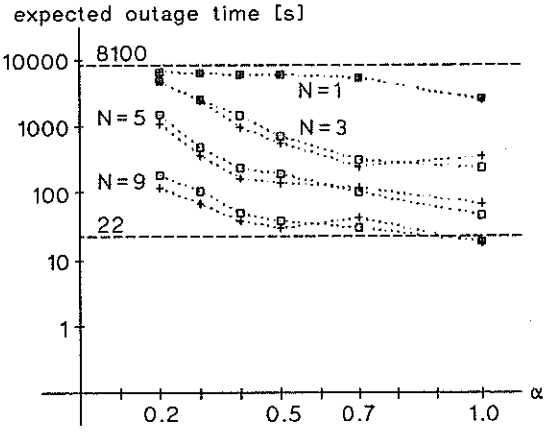


Fig. 6: outage time vs. α, 256-QAM

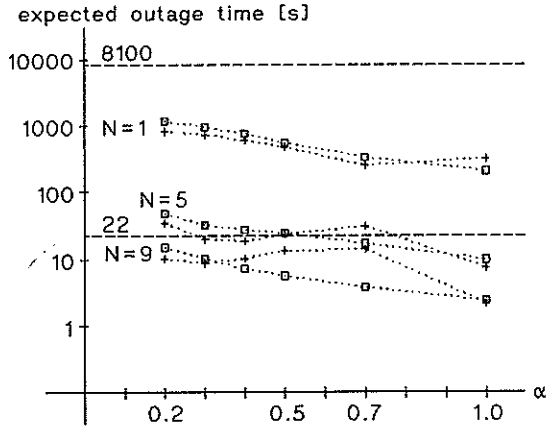


Fig. 4: outage time vs. α, 16-QAM

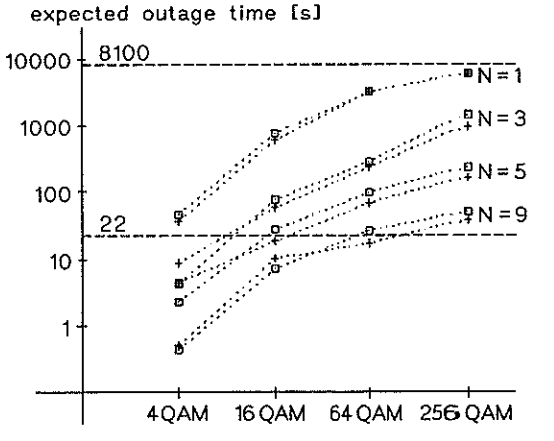


Fig. 7: outage time vs. modulation, $\alpha = 0.4$

Fig. 7 shows dependence of outage time on the modulation. Higher level modulations are much more susceptible to channel imperfection.

Fig. 8 shows dependence of outage time on the order of the transversal equalizer.

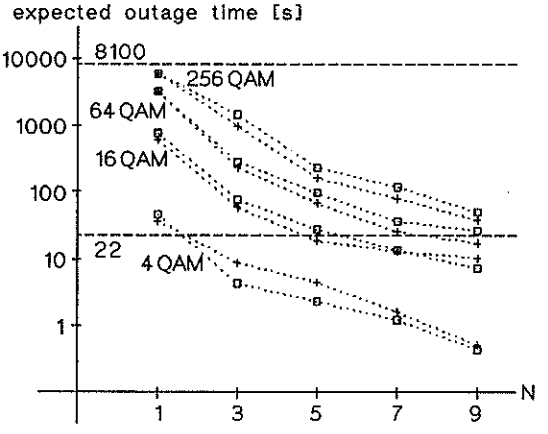


Fig. 8: outage time vs. equalizer order, $\alpha = 0.4$

7. CONCLUSIONS

The results of simulations show that outage time is strongly influenced by the filter type and parameters and they must be selected with great care especially when high-level modulation schemes are used.

The results of simulations also give insight into how much the transversal equalizer can combat the distortion of the signal due to the microwave channel deterioration. For a Rummler-like (42 km) microwave hop and 64-QAM modulation at least 9-tap transvesal equalizer is needed to satisfy the CCITT recommendations unless space or frequency diversity is used.

REFERENCES

[1] W. D. Rummler, "A New Selective Fading Model: Application to Propagation Data", The Bell System Technical Journal, No. 5, may-june 1979

[2] C. W. Lundgren, W. D. Rummler, "Digital Radio Outage Due to Selective Fading - Observation vs Prediction From Laboratory Simulation", The Bell System Technical Journal, No. 5, may-june 1979

[3] W. T. Barnett, "Multipath Propagation at 4, 6, and 11 GHz", The Bell System Technical Journal, vol. 51, februar 1972, num. 2

[4] W. D. Rummler, R. P. Coutts, M. Liniger, "Multipath Fading Channel Models for Microwave Digital Radio", IEEE Communications Magazine, Vol. 24, No. 11, november 1986

[5] Michael L. Honig, David G. Messerschmitt, Adaptive Filters (Kluwer Academic Publishers, 1984)

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Joint Carrier Recovery and Data Equalization Using Frequency Domain Techniques

Steven Goldberg, Michael Ready, and Ron Ibaraki
Applied Signal Technology, 160 Sobrante Way
Sunnyvale, California 94086, USA

Abstract

A T/2 fractionally-spaced equalizer implemented in the frequency domain is presented here. Carrier synchronization is accomplished jointly with the equalizer weight update by using a multimode technique. Simulation results are presented which demonstrate the versatility of the design.

1 Introduction

This work considers the problem of adaptive equalization for digital modem signals that have been distorted by a frequency-selective, slowly fading channel. We define digital modem signals as signals that can be described by one of the many variations of two-dimensional amplitude and phase modulation, e.g. PSK or QAM modulation formats. The majority of current equalizer designs for this application are adaptive finite impulse response (FIR) filters that provide some compromise between reducing signal distortion, interference suppression, and maximizing signal-to-noise ratio. Based on linear filter theory, an equalizer can be realized either in the time domain or the frequency domain. If properly designed, both realizations should provide equivalent steady-state bit-error-rate performance. However, as the equalization requirements become severe, as in the case of long delay multipath distortion, the computational complexity of performing the equalization in the time domain becomes excessive. It has been shown [5] that for filter requirements of greater than 32-64 taps it may be more efficient to implement the filtering in the frequency domain. Additional advantages of frequency domain equalization include potentially increased convergence speed, and for some specific applications, a more direct method for defining an appropriate cost function to drive the equalizer weight update.

However, before adaptive frequency-domain equalizers can be used in real-world systems it is important that the problem of carrier frequency and phase tracking be considered [2]. In the case of point-to-point digital microwave modems both symbol timing and carrier recovery are issues. We have assumed, for this work, that symbol timing recovery is accomplished prior to equalization and independent of carrier recovery. We also assume that the carrier recovery loop follows equalization and does not include the equalizer delay. Section 3 discusses this subject in more detail.

This work will provide some insight into the operation of the carrier frequency and phase tracking when used in conjunction with a fractionally-spaced frequency domain equalizer (FDE).

The paper is organized by first briefly describing the equalizer operation in Section 2. Section 3 describes the synchronization techniques involved. Simulation results are presented in Section 4. Conclusions and a summary are contained in Section 5.

2 T/2-Spaced Frequency Domain Equalization

Figure 1 shows the block diagram of the frequency domain equalization (FDE) architecture used in the simulations. All of the processing in the FDE is performed on blocks of input data rather than on a sample-by-sample basis. It represents an implementation of a linear convolution as described by [1]. At this point we note the two extra FFT's in the weight update operation. By exploiting the FFT the block average gradient can computed more efficiently.

All of the weights are updated on a block basis so that the equalizer weights are held constant for a time equal to the length of each block. At the end of each

block, the error signal is computed and the weights are updated for the next block. To filter the signal in the frequency-domain, the conventional *overlap-save* method was used [4]. The equalizer has the equivalent of N, T/2 -spaced taps, where N is a power of 2. An equalizer of length $M < N$ can be used by constraining the last $N - M$ taps to zero. As per the overlap-save procedure, the input signal, $x(n)$, is buffered in blocks of $2N$ samples with each new block overlapping the last block by N samples. The input sampling rate is assumed to be exactly two samples per symbol in keeping with the T/2 fractionally-spaced design. Each input block, denoted by $\mathbf{x}(lN)$ is Fourier transformed using a $2N$-point FFT giving the transformed signal, $\mathbf{X}(lN)$, where $l = (0, 1, 2, ...)$ is the block index. Next, the transformed signal, $X(lN)$, is multiplied, term-by-term, by the $2N$-point FFT of the complex filter taps, denoted by $\mathbf{H}(lN)$. The equalized signal is inverse transformed using a $2N$-point inverse FFT. The first N samples are discarded and last N samples form the actual equalizer output. Finally, the output is decimated by a factor of two so that only one sample per symbol time is actually outputted. This equalized signal is then passed through a quantizer which makes a decision as to which symbol has been transmitted.

We now consider the weight update procedure. It is assumed that the steady-state symbol-error rate (SER) has been reduced to better than 10^{-1} so that a minimum mean-squared error weight update algorithm can be used. Our approach is to first use a constant modulus based weight update [3] which is data and carrier phase independent. Although, this algorithm has not been shown to guarantee convergence to the optimum Wiener solution it will generally improve performance so that the decision-directed algorithm described below will function.

The update begins by computing a loop error signal. It is computed in the time domain due to the fact that standard minimum mean-squared error criteria are based on characteristics of the equalized time-domain signal. The frequency-domain weight update can be updated in the time-domain and then transformed to the frequency-domain or, equivalently, the average gradient can be transformed to the frequency-domain and the weights updated directly. The latter approach is more computationally efficient and was used here.

The update equation for an individual weight is given by the scalar expression:

$$h_j(lN+N) = h_j(lN) + 2\mu \sum_{m=0}^{N-1} x^*(lN+j+m)\epsilon(lN+m) \tag{1}$$

where h is the weight, ϵ is the error (the difference between the despun quantizer output and the raw equalizer output), and μ is the adaptive step size. The summation term is the correlation of the error signal with the conjugate of the input signal and can be efficiently computed using FFT's. Ferrara [1] is credited with deriving this expression. First, define the $2N$-element error vector prepended with N zeros as

$$\mathbf{e}(lN) = [\underbrace{0, ..., 0}_{N}, \epsilon(lN), \epsilon(lN+1), ..., \epsilon(lN+N-1)]. \tag{2}$$

Next, compute, $\mathbf{E}(lN)$, the $2N$-point FFT of $\mathbf{e}(lN)$. The time averaged gradient is given by

$$\bar{\mathbf{d}}(lN) = \text{first N terms of } FFT_{2N}^{-1}[\mathbf{E}(lN) \odot X^*(lN)]. \tag{3}$$

Finally, the frequency-domain weights are updated with the formula

$$\mathbf{H}(lN+N) = \mathbf{H}(lN) + 2\mu\bar{\mathbf{D}}(lN) \tag{4}$$

where $\mathbf{D}(lN)$ is the transform of the gradient.

3 Synchronization

As was mentioned earlier, the equalization block is only a part of a complete receiving system. With the exception of certain specialized modulation formats (e.g. OQPSK) the symbol timing recovery can be accomplished completely independent of the equalization and carrier recovery process. All of the simulations which follow assume that timing recovery has been achieved before equalization and carrier recovery.

Returning to Figure 1 we now consider carrier recovery loop. The carrier recovery design presented here will be based on a second-order phase-locked loop (PLL) located inside the adaptive equalizer weight update loop. It is capable of tracking both frequency and phase offsets during each filter weight update cycle. The output of the equalizer is quantized and the angle is computed representing the instantaneous phase error. This error is then filtered by the second-order loop. The loop filter is followed by mixer which is fed by a numerically-controlled oscillator which generates the proper signal to advance or delay the phase of the raw equalizer output. This *despun* equalizer output is then used to compute the appropriate filter weight update.

The overall carrier tracking actually uses a multimode strategy. The error calculation described above implicitly assumes that the quantizer will link the equalizer output with the correct signal constellation point. This decision-directed carrier tracking approach functions quite well with relatively high SNR at error rates below 10^{-2}. To guarantee that this mode will acquire and track the FDE first uses a so-called 'blind' or gated carrier tracking mode. It generally is used in conjunction with the blind equalization update mode. It simply makes phase updates on constellation symbols which are outside a given constellation dependent radius. This technique is much less sensitive to noise and distortion but does limit tracking bandwidth. A predetermined number of symbols are processed in this gated mode while equalization improves the output SNR. At this point most of the equalizer decisions are assumed correct and a decision-directed update and can be used to direct the PLL to provide the proper frequency and phase offset to stop the signal constellation from spinning (the effect of improper carrier frequency recovery at the receiver) [5].

4 Simulation Results

To demonstrate the performance of the FDE, a simulation was run using a 16QAM modem signal. A 50% square-root raised cosine shaped transmit filter was used. The SNR was set at 30dB. Approximately 50,000 samples were generated spanning 25,000 symbols. The equalizer was 128 taps (64 symbols) long implying the use of 256 point FFT's. Figure 2 shows the impulse response of the channel that was used. The delay spread covers approximately 30 symbols. A conventional time-domain tapped-delay line equalizer will be used as a baseline for comparison and measure of performance. This time-domain equalizer uses a sample-by-sample update. The adaptive step size was adjusted to .0001 for both the time and frequency domain equalizers. The carrier offset was initialized to .125% of the sampling frequency.

Figure 3(a) shows the completely distorted and unequalized constellation. Figure 3(b) shows the constellation after 25,000 points where decision-directed carrier recovery and equalization commenced. Although the carrier frequency offset has not been completely removed, several symbol radii are clearly visible implying acceptable eye opening. After 50,000 samples Figure 3(c) shows the distortion is almost completely removed and the 16QAM constellation is shown with tight clusters corresponding to approximately 25dB. Performance consistent with the input SNR of 30dB could be achieved by allowing the equalizer to continue to converge and by using a smaller step size to minimize the steady-state misadjustment error.

Figure 4 shows a comparison of the convergence of the FDE operating on the 16QAM signal with frequency offset versus both a time and frequency domain equalizer operating on the same signal with no frequency offset. Note that 4(a) and (b) show almost identical performance. Figure 4(c) clearly shows the effect of the frequency offset. Although it is not apparent from this figure, the carrier recovery loop is slowly incrementing the instantaneous phase until the constellation stops spinning. This is clearly seen to happen at approximately 37,000 samples where the cluster variance begins to rapidly decrease. In this case, at 50,000 samples, the FDE with frequency offset had a cluster variance differing by only 1-2 dB. The adaptive step-size, the carrier tracking loop parameters, and the length of the blind equalization mode all affect this performance.

5 Summary

It has been shown that practical techniques for carrier recovery can be implemented jointly with a T/2-spaced frequency domain equalizer. Convergence time and tracking bandwidth are both complicated functions of a number of variables but can be designed to practically acceptable values for a wide range of modulation formats and multipath channels.

References

[1] Earl Ferrara. Fast implementation of lms adaptive filters. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 474–475, 1980.

[2] Richard Gooch and Michael Ready. An adaptive phase lock loop for phase jitter tracking. In *Proceedings of the Twenty First Asilomar Conference on Signals, Systems, and Computers*, 1987.

[3] John R. Treichler and Brian G. Agee. A new approach to multipath correction of constant modulus signals. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, ASSP-31(2):459–471, 1983.

[4] Alan Oppenheim and Ronald Schaefer. *Digital Signal Processing*. Prentice Hall, 1975.

[5] Michael Ready, Steven. H. Goldberg, and Richard Gooch. Architecture considerations for frequency domain adaptive equalizers. In *Proceedings of the Asilomar Conference on Circuits, Systems, and Computers*, 1989.

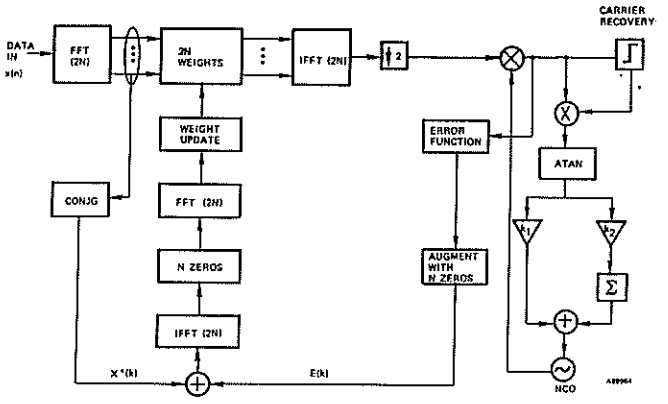


Figure 1: Frequency Domain Equalizer with Carrier Recovery

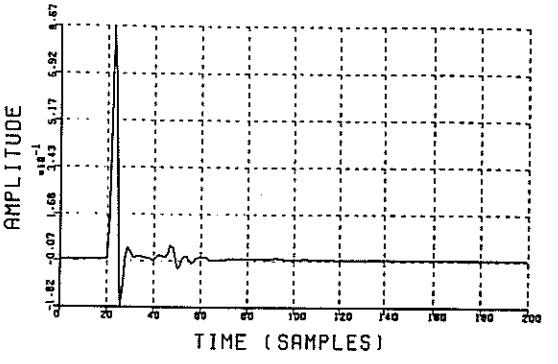


Figure 2: Channel Impulse Response

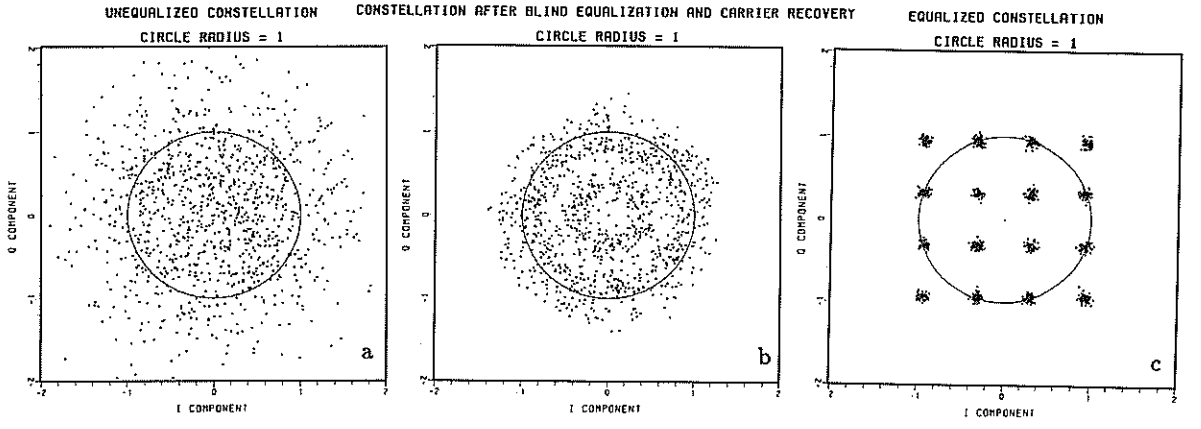
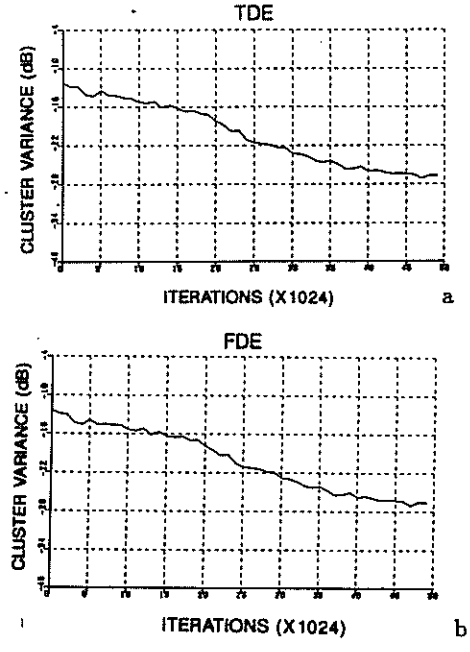


Figure 3: Signal Constellation during Equalization



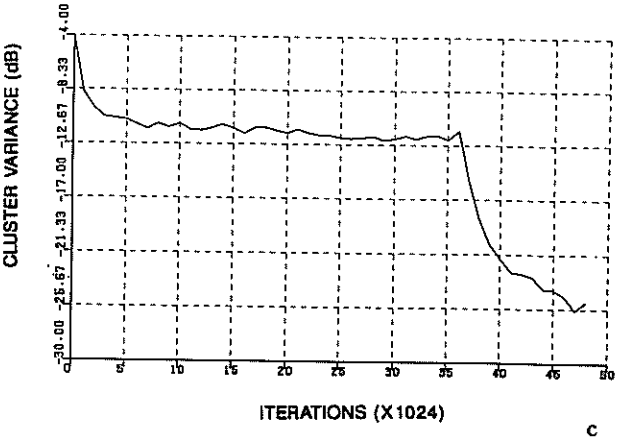


Figure 4: Convergence Rate Comparison

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Tree based synchronization algorithm applied to satellite communications

Roberto Viola & Javier Ventura-Traveset Bosch
European Space Agency (E.S.A.)
European Space Research and Technology Center (ESTEC)
Postbus 299, 2200 AG Noordwijk, The Netherlands

Abstract

A new algorithm for maximum likelihood synchronization is discussed. A likelihood function updated on a symbol by symbol basis is derived for linear modulation formats. The optimum search strategy is accomplished in a ternary tree describing all possible evolutions of timing off-set. Simulation results confirm the conformance with ML estimation.

1 Introduction

The introduction of fully digital modems has renewed the interest for the basic estimation theory. It is clear in fact that to fully exploit the advantages provided by Digital Signal Processing (DSP), a simple digital implementation of analog synchronizers is highly suboptimal [1].

In satellite communications, the efficiency of the link is more and more attained by using channel encoding which forces the modem to work at low Eb/No (< 6 dB). It is therefore important to guarantee optimal performances of synchronizers at low power/noise ratios. On the other hand, the operations in burst mode require quick and reliable synchronization with minimum overhead.

The use of the Maximum Likelihood algorithm (ML) for digital synchronizers and its flexibility for different modulation formats, has been widely studied and discussed [1], [2], [3] and [4].

In this paper we will derive the ML estimator for a vast class of linear modulation signals (MPSK, MQAM) and a computationally efficient tree search scheme for its decision-aided implementation. The derivation will be based on a likelihood function which is updated on a symbol by symbol basis, the so called "cumulative estimation" –similar to the approach in [5]– The advantages of this approach are various [5], i.e., the transition from acquisition to tracking mode is smooth, it provides an offset detection on symbol-by-symbol basis like conventional DA circuits and can be easily translated in efficient computational structures. The decision directed approach is followed to achieve good performances at low Eb/No.

We will show that in digital systems, the evolution of timing off-set can be fully described as a path in a tree of possible values, and that the cumulative maximum likelihood function can be used to identify the most likely path at the receiver side. A deterministic behavior of the timing-offset evolution, e.g. doppler, "forces" preferred paths. This can be exploited by properly smoothing the logic in the search algorithm.

In section 2, we will explain the derivation of the ML tree estimator, in section 3, simulated results will be compared to the Cramer-Rao-Bound, and finally, in section 4, conclusions will be drawn.

2 Derivation of the ML tree estimator

Let us consider the transmission scheme illustrated in figure 1.

Indicating with $\tilde{s}$ the complex envelope of a modulated signal QAM

$$\tilde{s}(t,\tilde{\tau},\tilde{\varphi}) = A_0 e^{j\tilde{\varphi}} \sum_{k=-\infty}^{\infty} [a_k g_t(t-kTs-\tilde{\tau}) + j b_k g_t(t-kTs-\tilde{\tau})] \tag{1}$$

where:

- a_k, b_k : A sequence of symbols belonging to the transmitter alphabet.
- $\tilde{\tau}, \tilde{\varphi}$: Time varying phase and timing offsets respectively.
- $g_t(t)$: Impulse response of the transmitter symbol shaping filter.
- A_0 : Signal envelope amplitude.
- $1/Ts$: Symbol rate.

the received signal will be:

$$\tilde{x}(t) = \tilde{s}(t,\tilde{\tau},\tilde{\varphi}) + \tilde{n}(t) \tag{2}$$

with:

$$\tilde{n}(t) = n_c(t) + j n_s(t) \tag{3}$$

Let us consider $n_c(t)$, $n_s(t)$ a couple of AWGN process with a one-side spectral density of N_0.

At the receiver, after matched filtering, phase rotator and sampling at time $ti = i\frac{T_s}{M} + \hat{\tau}$ the signal will be as follows:

$$\begin{aligned} r(i\frac{T_s}{M}) &= A_0 e^{j\varphi} \sum_{k=-\infty}^{\infty} [a_k \bar{g}((i-kM)\frac{T_s}{M} - \tau) \\ &\quad + jb_k \bar{g}((i-kM)\frac{T_s}{M} - \tau)] \\ &\quad + n(i\frac{T_s}{M}) \end{aligned} \tag{4}$$

where:

- $\frac{T_s}{M}$: Sampling interval at the output of the matched filter.
- Ts : Symbol duration.
- M : Number of samples per symbol.
- $\bar{g}(i\frac{T_s}{M}) = [g_t(t) \otimes g_r(t)]_{t=i\frac{T_s}{M}}$: output of the matched filter.
- $\tau = \tilde{\tau} - \hat{\tau}$: Residual timing error.
- $\varphi = \tilde{\varphi} - \hat{\varphi}$: Residual phase error.

Let us now consider vectors $\underline{r}_i$ and $\underline{s}_i$ of N symbols, and consequently with N.M samples:

$$\underline{r}_i = (r_i, ..., r_{i+M-1}, ..., r_{i+NM-1}) \tag{5}$$

$$\underline{s}_i = (s_i, ..., s_{i+M-1}, ..., s_{i+NM-1}) \tag{6}$$

Three parameters have to be known from the received signal: The transmitted symbols (a_k, b_k), the timing error τ and the phase offset φ.

The ML estimator of these three parameters is given by definition as:

$$ML \equiv max_{\tau,\varphi,\underline{a}}\{p(\underline{r}|\tau,\varphi,\underline{a})\} \tag{7}$$

where:

- $\underline{a}$: Vector of transmitted symbols.

Using tentative decisions $\hat{a}_k, \hat{b}_k$ (data aided approach) we can reduce the three search parameters to an approximated two dimensional (τ, φ) estimation.

$$ML \simeq max_{\tau,\varphi}\{p(\underline{r}|\tau,\varphi)\} \tag{8}$$

Considering the ideal AWGN channel we have

$$p(\underline{r}|\tau,\varphi) = \frac{1}{(2\pi N_0)^{N/2}} exp(-\frac{1}{N_0} \sum_{q=1}^{N} \sum_{i=(q-1)M+1}^{qM} |r_i - \hat{s}_i(\tau,\varphi)|^2) \tag{9}$$

where $\hat{s}_i(\tau,\varphi)$ are elements of the vector approximation of (6).

If we call

$$p_q(\underline{r}|\tau,\varphi) = \frac{1}{(2\pi N_0)^{m/2}} exp(-\frac{1}{N_0} \sum_{i=(q-1)M+1}^{qM} |r_i - \hat{s}_i(\tau,\varphi)|^2) \tag{10}$$

(8) can be rewritten as:

$$max_{\tau,\varphi}\{p(\underline{r}|\tau,\varphi)\} = max_{\tau,\varphi}\{\Pi_{q=1}^{N} p_q(\underline{r}|\tau,\varphi)\} \tag{11}$$

Developing (10)

$$\begin{aligned} p_q(\underline{r}|\tau,\varphi) &= \gamma_q exp[-\frac{1}{N_0}(\sum_{i=1}^{M} |\hat{s}_{q+i}(\tau,\varphi)|^2 - \\ &\quad -2\sum_{i=1}^{M} Re[r_{q+i}\hat{s}^*_{q+i}(\tau,\varphi)])] \end{aligned} \tag{12}$$

with:

$$\gamma_q = \frac{1}{(2\pi N_0)^{M/2}} exp(-\frac{1}{N_0} \sum_{i=1}^{M} |r_{i+q}|^2) \tag{13}$$

which does not depend upon τ.

Defining the log-likelihood function :

$$L_N(iNTs) \equiv ln[p(\underline{r}|\tau,\varphi)] \tag{14}$$

the symbol log-likelihood:

$$\Lambda_q(qTs) \equiv ln[p_q(\underline{r}|\tau,\varphi)] \tag{15}$$

and the cumulative-log-likelihood:

$$C(iTs) \equiv \sum_{q=-\infty}^{i} \Lambda_q(qTs) \tag{16}$$

it follows:

$$L_N(iNTs) = \sum_{q=i}^{(i+N)} \Lambda_q(qTs) = C(iTs) - C((i-N)Ts) \tag{17}$$

From (12) we can express (15) as follows:

$$\Lambda_q(qTs) = -\frac{1}{N_0}(\sum_{i=1}^{M} |\hat{s}_{q+i}(\tau,\varphi)|^2 - 2\sum_{i=1}^{M} Re[r_{q+i}\hat{s}^*_{q+i}(\tau,\varphi)]) + ln\gamma \tag{18}$$

Recalling (1), $\hat{s}_i(\tau,\varphi)$ can be expressed as follows:

$$\hat{s}_i(\tau,\varphi) = \bar{A}_0 e^{j\varphi}\hat{s}_i(\tau) = e^{j\varphi}[\hat{c}_i(\tau) + j\hat{w}_i(\tau)] \tag{19}$$

where $\bar{A}_0$ is the amplitude of the signal estimation.

r_i can be decomposed in its phase and quadrature components

$$r_i = r_i^c + jr_i^w \tag{20}$$

Now, with (19) and (20), the two significant terms of (18) can be developed as follows:

$$\sum_{i=1}^{M} |\tilde{s}_{q+i}(\tau,\varphi)|^2 = \bar{A}_0^2[\sum_{i=1}^{M}(\hat{c}_{q+i}^2(\tau)+\hat{w}_{q+i}^2(\tau))] \quad (21)$$

and

$$\sum_{i=1}^{M} Re[r_{q+i}\tilde{s}_{q+i}^*(\tau,\varphi)] =$$

$$= A_0\bar{A}_0[\sum_{i=1}^{M}(\hat{c}_{q+i}(\tau)r_{q+i}^c + \hat{w}_{q+i}(\tau)r_{q+i}^c)cos\varphi -$$

$$- \sum_{i=1}^{M}(\hat{c}_{q+i}(\tau)r_{q+i}^w - \hat{w}_{q+i}(\tau)r_{q+i}^w)sin\varphi] \quad (22)$$

From (21) and (22) we can find, with different notations, a similar result of [4], i.e, under the current hypothesis, the residual phase estimation can be derived by the ML estimation of τ. The ML becomes then a one-dimensional search.

Let us now subdivide the timing offset τ in small integer steps of minimum duration Δ with $\Delta = \frac{T_s}{bM}$ and $\hat{\tau} = n\frac{T_s}{bM}$, such as $|\tau - n\frac{T_s}{bM}| << \frac{T_s}{bM}$, with $n, b = 1, 2, ...$

Starting with a "known" estimate of τ, we can easily derive all the possible evolutions of τ in the interval of estimation NTs.

The trellis of possible paths is depicted in fig 2. For simplicity of illustration the case $\Delta = \frac{T_s}{4}$ is shown.

At this point, it is clear that a set of strategies are possible to find the closest path to the original evolution of τ_q. The simplest one is as follows:

Early-late search

For each symbol we calculate $\Lambda_q^{\Delta}(qTs)$ (early), $\Lambda_q^0(qTs)$, $\Lambda_q^{-\Delta}(qTs)$ (late) and their respective "cumulative" $C^{\Delta}(qTs)$, $C^{-\Delta}(qTs)$ and $C^0(qTs)$. At this point the best metric is selected and the operation repeated with the survivor node.

A more reliable search can be obtained by properly exploiting a tree search mechanism, that introduces us to the (N,L) search:

(N,L) search

For each estimated symbol new values of $C(iTs)$, are computed. Decisions are taken with a delay of N symbols, having selected the best C(iTs) (or $L_N(iNTs)$) over L-th path of 3^N possible ones. The path metrics are given by the value of the "cumulative" maximum likelihood function.

In a practical situation, time off-set will be constant over more than one symbol, expecially when tracking conditions are reached. In this case, transitions in the tree can take place in a time interval of more than one symbol. Therefore, the depth of the tree will not necessarily be equal to the number of transmitted symbols.

If the log-likelihood function is unimodal over the search range, we can use the derivative function as well. Nevertheless the derivative should be used with care for certain modulation formats and certain pulse shaping functions.

From the previous formulas (18), (22) and (14) we obtain a simplified version of the derivative-symbol-partial-log-likelihood:

$$\frac{\delta\Lambda_q(qTs)}{\delta\tau} \simeq$$

$$\simeq \frac{2}{N_0}[A_0\bar{A}_0[\sum_{i=1}^{M}(\hat{c}'_{q+i}(\tau)r_{q+i}^c + \hat{w}'_{q+i}(\tau)r_{q+i}^c)cos\varphi -$$

$$- \sum_{i=1}^{M}(\hat{c}'_{q+i}(\tau)r_{q+i}^w - \hat{w}'_{q+i}(\tau)r_{q+i}^w)sin\varphi] \quad (23)$$

and the derivative log-likelihood function:

$$L'_N(iNTs) = \sum_{q=i}^{(i+N)} \Lambda'_q(qTs) = C'(iTs) - C'((i-N)Ts) \quad (24)$$

where x' means derivative of x.

The search metrics criteria will be then modified accordingly.

In the next section some practical results are presented and discussed for a simple case of this last version of the tree.

One final remark is about the incremental steps: so far we have assumed that the law to increment Δ is linear, the procedure can actually be generalized to allow an arbitrary incremental rule, e.g. exponential laws, when it is necessary to achieve a better convergence rate or tracking capability.

3 Simulation results

In this section, a simple case of the Tree algorithm has been simulated as a forward estimator. The main intention of this simulation was to show that the proposed tree search achieve performance close to the Maximum likelihood estimator.

As has been explained before, the main application of the tree search algorithm should be in a closed loop structure, since both the advantages of forward and feedforward schemes, are kept.

The scheme is depicted in figure 3. Basically, a QPSK signal is transmitted and filtered by a square root raised cosine filter, after noise addition (AWGN) the signal is filtered and sampled. Using a derivative circuit, an interpolator and taking the decisions from a two level discriminator we implement the desired tree algorithm.

The obtained results are shown in figure 4, where the normalised rms values are plotted versus Eb/No for 10, 50 and 100 symbols of estimation length. These results confirm that despite working at low Eb/No ratios, the performance of the algorithm is quite close to the Cramer-Rao-Bound.

4 Conclusions

The general structure of a class of ML synchronizers has been discussed. Timing off-set quantization gives rise to a finite set of possible paths described in a ternary tree. Search strategies to increase the computational efficiency of ML have been discussed. Simulation results show that tree search does not impair the ML performance. Further work is envisaged to derive more practical synchronizer structures.

5 References

[1] F.M. Gardner,"Demodulator Reference Recovery Techniques suited for digital implementation,"ESA contract 6847/86/NL/DG,final report,Aug. 1988.

[2] M. Moeneclay, "A fundamental lower bound on the performance of practical joint carrier and bit synchronizers," IEEE Trans.Commun,,vol.COM-32,pp. 1007-1012, Sept. 1984.

[3] M. Moeneclay, "Two Maximum-Likelihood symbol synchronizer with superior tracking performances," IEEE Trans.Commun,,vol.COM-32,pp. 1178-1185, Sept. 1984.

[4] G.Ascheid et al,"An all digital receiver architecture for bandwidth efficient transmission at high data rates," IEEE Trans.Commun,,vol.COM-37,pp. 804-813, Aug.1989.

[5] A. Gubser,"Cumulative Maximum-likelihood synchronization in digital communications,",Ph.D. dissertation, ETH Zurich, 1989.

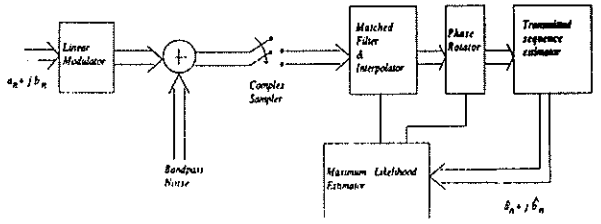


Figure 1: General transmission scheme for a linear modulated signal and ML estimation.

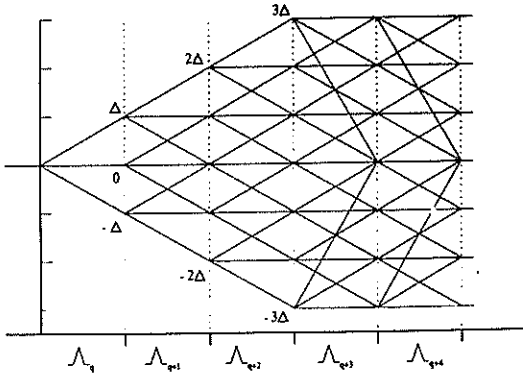


Figure 2: Trellis of possible paths for the simplified case $\Delta = \frac{T_s}{4}$

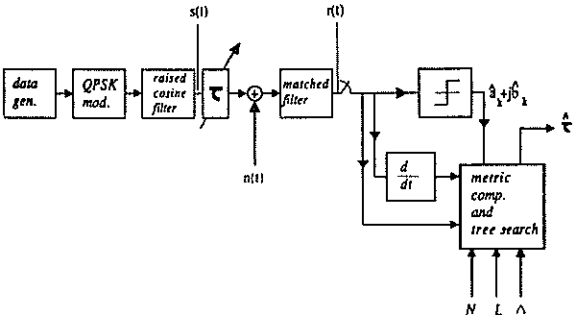


Figure 3: Diagram of the QPSK simulated transmission scheme.

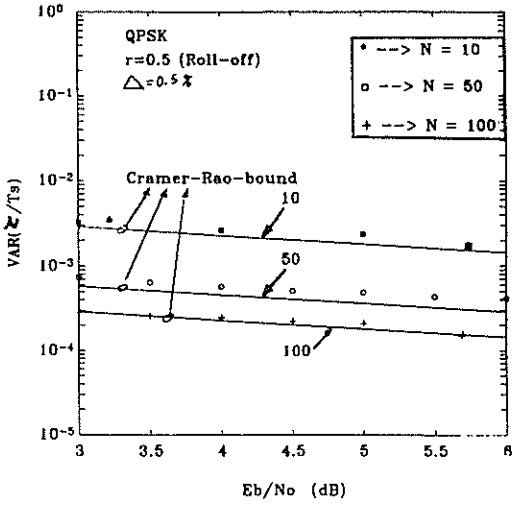


Figure 4: Normalyzed tracking error variance of the simulated scheme vs. Eb/No for (N=10, 50 and 100 symbols)

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Analysis of Baud-Rate Timing Recovery Techniques for a DSP-Based 2B1Q Digital Receiver

*M. Hage, *T. Aboulnasr, **B. Sayar, **S. Aly,

*Ottawa Carleton Institute for Electrical Engineering, Dept. of Electrical Engineering, University of Ottawa, Ottawa, Ontario, Canada K1N 3N6
**BNR, P.O. Box 3511, Station C, Ottawa, Canada K1Y 4H7

A baud rate timing recovery scheme suitable for ISDN transmission on digital subscriber loops (DSL) is investigated. Three timing estimates based on this scheme are studied for steady-state and convergence behavior. Theoretical analysis and simulation results are included. The performance of the timing estimates on realistic loops in a practical digital receiver using 2B1Q data symbols is characterized and compared.

1. Introduction

Digital transmission is currently widely used over subscriber loops. In such systems, it is essential to synchronize the local sampling clock to the far end transmit clock (i.e. recover timing) to achieve optimum receiver performance. For DSP-based receivers, numerous timing recovery algorithms are available to perform this synchronization requiring information at twice the sampling rate or more. However, full duplex transmission on these systems is achieved using echo cancellers. Since the receiver complexity is determined mainly by the echo canceller complexity which is in turn proportional to the sampling rate, there has been a great interest in using baud rate timing recovery algorithms [1], [2].

The classical paper by Mueller and Müller [1] on baud rate timing recovery is the basis for most of the current work. First, a timing function is selected, then an estimate of this function is constructed and implemented. The timing function should be based on a characteristic of the channel pulse response which is insensitive to the specific loop make-up such as length, gauge size, and bridged taps. This would enable the same timing recovery algorithm to be effective for a wide range of loop characteristics.

In this paper, the timing function $f(\tau)$ used is based on the first precursor zero crossing of the single channel pulse response such that $f(\tau) = h_{-1}$, where h_0 is the main cursor of the channel response. This timing function has been shown to satisfy the above requirements [2]. As well, it results in reducing precursor intersymbol interference (ISI) since the first precursor is forced to be zero. Numerous estimates of this specific timing function can be derived to satisfy the conditions given in [1]. In this paper, we study three such estimates derived for the 2B1Q line code. Two of the estimates $z_1(k)$ and $z_2(k)$ are based on a linear combination of the signal vector x(k), and a weighting vector g(k) such that $z_{1,2}(k) = g_{1,2}(k).x(k)$, where x(k) is a function of the samples of the received signal after equalization, and g(k) is a function of the received data symbols a(k) after quantization. The third timing estimate is of the same general form $z_3(k) = g_3(k).e(k)$, where e(k) is the error signal $e(k)=x(k) - h_0.a(k)$. In each case, the weighting vector is chosen such that the expected value of the weighted sum equals the timing function. The optimum phase τ_0 is defined as the sampling point at which $f(\tau=\tau_0)=0$. The timing estimates are given below in Table 1.

Table 1: Three Timing Estimates Based on $f(\tau) = h_{-1}$

i	$z_i(k)$
1	$\frac{(a(k)^2-5)}{16}$ (a(k)x(k-1) - a(k-1)x(k))
2	$\frac{1}{(\sqrt{5}\,a(k)+5)}$ (-a(k)x(k-1) + a(k-1)x(k))
3	$\frac{1}{E\{a(k)^2\}}$ a(k) (x(k-1) - h_0 a(k-1))

The performance of these three estimates is studied in detail by theoretical analysis as well as simulation. We show that even though all estimates satisfy the requirements in [1] for the same timing function, they differ significantly in their performance on realistic subscriber loops. The performance will be evaluated principally by the receiver convergence, steady state timing jitter, and noise margin allowed before convergence is lost.

In section 2, the timing recovery loop used in the characterization of the timing estimates is introduced. The variances of the three estimates are then derived in the presence of white gaussian noise. This shows the dependence of each timing estimate on the channel distortion and additive noise. Finally, the effect of the dead-zone threshold, which is used to reduce timing jitter, is discussed.

In section 3, a simulation is setup to verify the above theoretical results. A 2B1Q transceiver for digital subscriber loop applications, running at the baud rate of 80 KHz, is modelled using the CAPSIM/BLOSIM software simulation package [3]. The effect of the various system parameters on the performance of each of the timing estimates is studied for the half-duplex mode for two Bellcore test loops, one with bridged taps, B3 loop which is 16 Kft long, and one without, B4 loop which is 17 Kft long [4]. For this, the peak to peak and RMS phase jitter, percentage of phase jumps, and offset from optimum phase are compared for all estimates for the steady state

behavior. The convergence time is also characterized using the same loops. Conclusions based on the theoretical and simulation results are given in section 4.

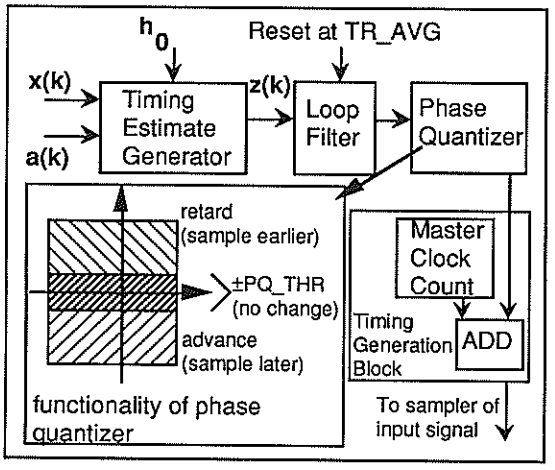


Figure 1: Timing Recovery Block

2. Theoretical Analysis

2.1. Timing Recovery Loop

The block diagram for the timing recovery loop is shown in figure 1. The timing estimate generator implements any of the three timing estimates. It receives the input signal x(k) and output of the slicer a(k) as inputs, as well as the output of an automatic reference control (ARC) block which approximates the channel gain h_0 when determining $z_3(k)$. The output of the timing estimate generator consists of the timing estimate exactly as given in Table 1, calculated at every baud. The loop filter performs a time average over a fixed length of data symbols (TR_AVG). Since $E\{z(k)\}$ is the timing information we seek, we need to minimize the error in the approximation of the ensemble average of z(k) by time averaging by using a long average length. In contrast, a long average is unpractical in that phase corrections are performed once every TR_AVG so that convergence time of the timing recovery loop is directly proportional to the TR_AVG length. This is a first order loop where all system variables and states are reset every TR_AVG. The phase quantizer block compares the output of the loop filter to a threshold (dead-zone) and depending on the outcome, a NO CHANGE, ADVANCE or RETARD clock flag is issued to the timing generation block. The functionality of the phase quantizer is illustrated in figure 1. The threshold in the phase quantizer (PQ_THR) is normalized by the channel gain as follows:

$$PQ\_THR = DZT \cdot ARC(\tau(k))$$

The DZT is a fixed dead-zone factor, pre-selected by the designer, and $ARC(\tau(k))$ is an estimate of the main pulse h_0 at the current input sampling phase. The normalization by the channel is necessary since the DZT value is not variable and must be applicable to a wide range of loops. The output of the phase quantizer is fed back to the timing generation where the input sampling phase is adjusted accordingly. The size of the phase corrections is fixed at $\frac{1}{128}$ baud period.

The timing recovery loop setup here implies that the steady state phase will hover around the optimum phase τ_0 such that, on average, it will exactly equal τ_0 .

2.2. Effect of Channel Distortion and Additive Noise

The effect of additive white gaussian noise and ISI on the performance of the timing estimate is quantified in its variance. The expressions for the variance of z(k) as specified in Table 2 are functions of the mean square distortion (MSD) of the channel and of the noise power σ^2. Since $E\{z(k)\} = f(\tau)$ is the information we seek, then the variance of z(k) $var\{z(k)\} = E\{z(k)^2\} - E\{z(k)\}^2$, by definition, will give us a measure of the error in z(k) in estimating $f(\tau)$. Consequently, the timing estimate with the lowest variance is the better one.

Table 2: Expressions for the Variance of $z_{1,2,3}(k)$

i	$VAR\{z_i(k)\}$
1	$3.125 \sum_{(j\neq 0)} h_j^2 + (h_1^2 - 3.125 h_1 h_{-1}) + 0.625\sigma^2$
2	$7.25 \sum_{(j\neq 0)} h_j^2 - 8.25 h_1 h_{-1} + 2(h_1^2 + h_{-1}^2) + 1.4\sigma^2$
3	$2 \sum_{(j\neq 0)} h_j^2 - 0.36 h_{-1}^2 + 0.2\sigma^2$

If we assume a noiseless distortion free channel, a perfect zero-variance estimate is obtained in all three cases. For the more realistic case where both noise and ISI are present, the best timing estimate will be determined by the relative contribution of each of the two error sources. For the estimates considered here, the ratio of contribution of noise and ISI is the same. From Table 2, $z_3(k)$ results in an MSE that is 9 dB lower than that of $z_2(k)$ which is the worst case, and 5 dB lower than $z_1(k)$. These results represent the margin for noise and channel distortion of each estimate. In our comparison, we ignore the extra terms comprised of h_1 and h_{-1} samples. This is valid since at steady state, both values are zero. The noise power term in the variance is ~7dB lower than the MSD value. This becomes an advantage during steady state since most of the channel distortion will be cancelled by the equalization blocks, and the channel noise term will be the dominant influence.

2.3. Effect of the Dead-Zone Threshold

The dead-zone threshold in the phase quantizer allows a reduction of the timing jitter by allowing a certain amount of phase jitter to be undetected. This factor should be chosen to ensure minimum steady state phase jitter with insignificant offset. The allowable peak to peak phase jitter θ_{0pp} for a certain DZT can be directly determined from the loop's functionality as follows:

$$\theta_{0pp} \approx \frac{2.DZT.h_0}{S.T.TR\_AVG}$$

S is the slope of the timing function in the vicinity of τ_0 and T is the baud period. The above was used initially to set the range of applicable DZT factors that would satisfy the above requirements.

3. Simulation Results

3.1. Simulation Setup

The architecture for the half-duplex receiver used in the simulation is shown in figure 2. At the input, 2B1Q symbols are linearly convolved with the channel pulse response at the baud rate, at the input sampling phase provided by the timing recovery, with white gaussian noise samples added to every signal sample. The channel pulse response includes some filtering for pulse shaping, decimation, precursor shaping for timing recovery and pulse tail reduction. Equalization of the received signal is performed using a transversal decision feedback filter (DFE) with N taps. The ARC acts as the zeroeth tap of the DFE except that its output does not directly affect the signal, and is used by the error, timing recovery and ATQ blocks for normalization of the threshold. The automatic threshold quantizer (ATQ) takes the 'clean' signal after ISI cancellation and slices it to output 2B1Q data symbols which are in turn used by the timing recovery, DFE and ARC blocks as inputs. The same feedback error is used by all adaptive blocks, based on the least-mean square (LMS) adaptation algorithm:

$$ERR(k) = x(k) - ARC(k-1) \cdot a(k)$$

The system runs at the baud rate with all the functions converging simultaneously at the start of the run. The simulation results are all based on steady state behavior which is defined as the point at which the RMS jitter is equal to or less than 3% of the baud period.

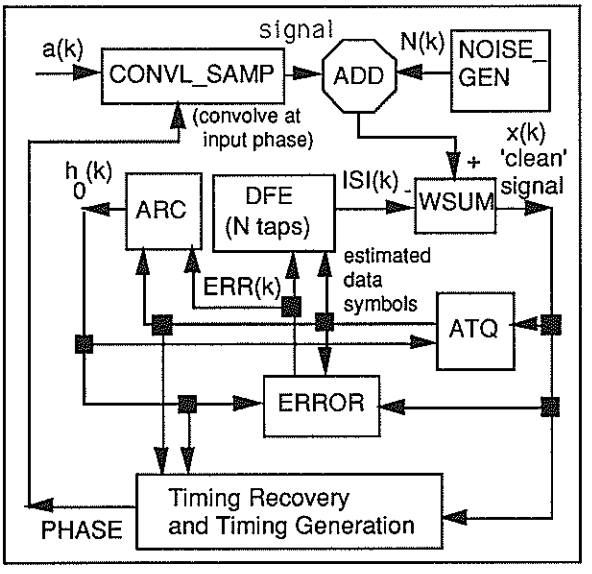


Figure 2: Simulation Setup for Half-Duplex Receiver

3.2. Simulation for Channel Distortion and Noise

The above receiver was simulated for various signal to noise ratios (SNR) of ~24,16,13 and 10 dB, for some arbitrary initial phases for the three estimates, using a DZT = 5, and different values for N such that some ISI distortion is included. The average values normalized with respect to the baud period, for the B3 and B4 loops are given in Table 3. Looking at the overall results, $z_3(k)$ shows the best performance with jitter values of up to 4 times lower than $z_2(k)$ and 3 times lower than $z_1(k)$. The typical behavior of each estimate is shown in figure 3. For the fraction of phase jumps, $z_3(k)$ results in values of up to 9 times lower than $z_1(k)$ and 16 times lower than $z_2(k)$. The phase offset is insignificant in most cases, with worst results for high SNR levels, especially $z_3(k)$. Given that DZT=5, this can be explained by the higher sensitivity of $z_3(k)$ to the dead-zone effect, which will be discussed in the next section. The phase offset at steady state can be eliminated by reducing the DZT factor.

Table 3: Simulation Results for ISI and Noise Effects

SNR	$z_1(k)$	$z_2(k)$	$z_3(k)$
Average peak to peak phase jitter			
24 dB	0.008	0.016	0.04
16 dB	0.016	0.04	0.012
13 dB	0.028	0.06	0.024
10 dB	0.072	0.096	0.024
Average RMS phase jitter			
24 dB	0.0034	0.0032	0.0018
16 dB	0.0045	0.0069	0.0049
13 dB	0.0068	0.012	0.009
10 dB	0.014	0.018	0.0042
Average fraction of phase jumps			
24 dB	0.0015	0.013	0.00045
16 dB	0.012	0.12	0.0019
13 dB	0.046	0.21	0.005
10 dB	0.2	0.38	0.025
Average phase offset			
24 dB	0.02	0.004	0.024
16 dB	0	0	0.012
13 dB	0	0	0
10 dB	0	0	0

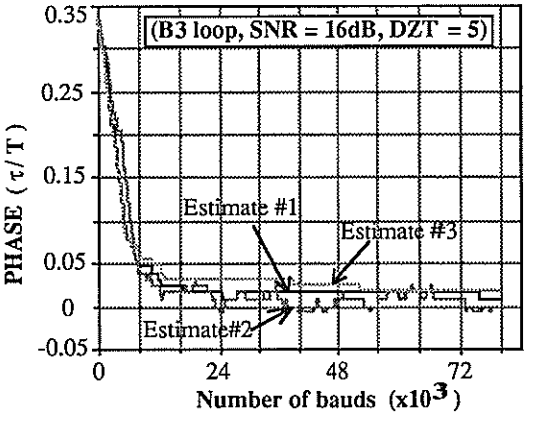


Figure 3: Example of Phase Behavior

3.3. Simulation for the Effect of the Dead-Zone

In this section, we fixed all system parameters and ran the receiver for the DZT values of 0, 5 and 7 for an SNR = 14dB and N made long enough to cancel any significant ISI. The average results for both B3 and B4 loops are given in Table 4. Again, $z_3(k)$ outperforms both of the other two estimates with $z_1(k)$ following closely, and finally, $z_2(k)$ with the worst results. However, as the DZT becomes larger, the performance of $z_3(k)$ deteriorates faster than the other two compared with lower DZT values. As can be seen, a compromise value for the DZT can be found where the phase jitter and phase offset are both acceptable. For very high DZT, the steady state phase offset becomes unacceptable such that the optimum phase is not closely reached, on the other hand, very low DZT values result in very high steady phase jitter with the optimum phase exactly reached on average. The DZT choice depends on the characteristics of the system and channel.

Table 4: Simulation Results for Dead-Zone Effect

DZT	$z_1(k)$	$z_2(k)$	$z_3(k)$
Average peak to peak phase jitter			
0	0.07	0.066	0.054
5	0.03	0.06	0.022
7	0.036	0.048	0.028
Average RMS phase jitter			
0	0.013	0.013	0.0094
5	0.0073	0.012	0.009
7	0.0098	0.11	0.010
Average fraction of phase jumps			
0	1.0	1.0	1.0
5	0.046	0.225	0.005
7	0.0125	0.072	0.0055
Average phase offset			
0	-0.0043	-0.00624	0.00312
5	0.00196	0.00196	0.0097
7	0.007	0.00604	0.0318

3.4 Comparison of Convergence Times

This section examines the behavior of the three estimates by measuring the convergence time for each for a variety of system parameters. B3 and B4 loops are considered, each taken at four different initial phase offsets for five SNR levels. Histograms of the results are shown in figure 4. $z_3(k)$ shows the fastest convergence times on average followed by $z_1(k)$ and finally $z_2(k)$. Note, in our simulation, all the blocks adapt simultaneously, however, in practice some training sequence could be used to converge the timing recovery, with equalization performed only after a certain error threshold is reached by the timing recovery. Various methods exist to improve convergence when the adaptation of the blocks is done blindly. Our results show that the timing estimates converge even in worst case startup conditions. The channel response greatly affects the convergence behavior, and, in our case, convergence occurred at a much faster rate for the B4 loop than the B3 loop. However, the relative performance of the three estimates for each loop remains as described in the average results.

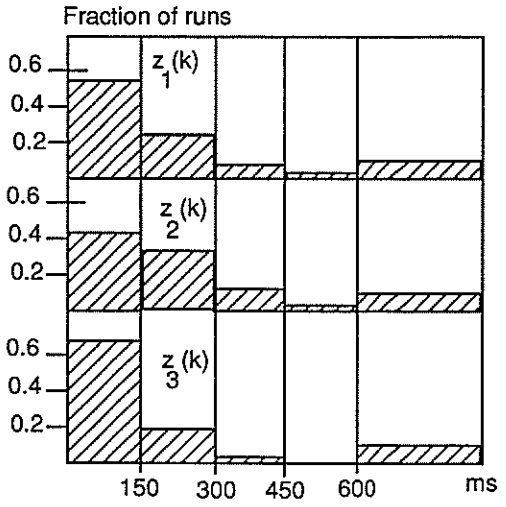


Figure 4: Histogram of Convergence Times

4. Conclusions

A theoretical and simulative comparison was made between three baud rate timing estimates suitable for digital receivers. Even though all satisfy the conditions set by [1] for the same timing function, it was found that their performance differs with the varying system parameters. $z_3(k)$ was found to be the least dependent on channel distortion and additive noise. It performed better overall during steady state behavior as well as in speed of convergence for both B3 and B4 loops. However, it was found to have a higher sensitivity to the dead-zone effect with its performance greatly deteriorating for high DZT factors. This estimate depends on the channel gain which is estimated using the ARC block and this interaction between the two adaptive functions did not impair its performance in any way. $z_1(k)$ was second best with results very close to $z_3(k)$ for high SNR levels. $z_2(k)$ is the worst with the highest steady state phase jitter and longest convergence times consistently for the various system parameters. The theoretical and simulation results corresponded in every instance.

REFERENCES

[1] K.H. Mueller and M. Müller, "Timing Recovery in Synchronous Data Receivers", IEEE Trans. COM-24, No.5, May 1976

[2] C. P. Tzeng, D.A. Hodges, D. Messerschmitt, "Timing Recovery in DSL Using Baud Rate Sampling", IEEE Journal on Selected Areas of Communications, Vol. JSAC-4, November 1986.

[3] J.T. Stonick, L.J. Faber, S.H.Ardalan, "Capsim User's Guide V2.0 ", CCSP, Dept. of Elect. and Comp. Eng., North Carolina State University, November 1988.

[4] ANST, "Integrated Services Digital Network (ISDN) - Basic Access Interface for Use on Metallic Loops for Application on the Network Side of the NT (layer 1 spec.)", ANSI T1.601, 1988

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SIMULTANEOUS PARAMETERS ESTIMATION OF DIGITAL MODULATED SIGNALS

M. Cabrera, M. A. Lagunas

Dept. of Signal Theory and Communications
ETSIT-UPC
P.O. Box 30.002-Barcelona, Spain

A maximum likelihood estimate for frequency carrier, phase leakage and amplitude of digital PSK modulated signals is presented. The three optimum parameters obtained are analyzed and compared with suboptimum realizations in the synchronization process. In order to achieve an efficient system a final structure is closed where each method is selected in order to get not only appropriated performance as well as a relatively low computational load.

1. INTRODUCTION

When digital phase and frequency modulated signals are transmitted in a satellite communications environment, the principal characteristics of the communications system have to be reviewed. When the signal is received in Time Division Multiple Access (TDMA) mode, loop systems as Phase locked loops and adaptive estimation algorithms are not adequate to be used because of the short slots of signal received periodically. The convergency of the algorithms to estimate parameters of synchronization could not be got inside each slot of signal arriving at the system.

Here, different kind of estimation methods will be analyzed, as in an optimum sense as in a suboptimum mode in the parameters estimation part. First of all the environment and kind of signal will be studied. A presentation of the general system will be done with two different stages. In the first one, the signal is filtered and down converted to a low pass band signal. In the second stage the synchronization is implemented by mean of estimating parameters of the carrier signal as amplitude, residual frequency and phase. The complete system can be summarized by the blocks diagram of figure 1.

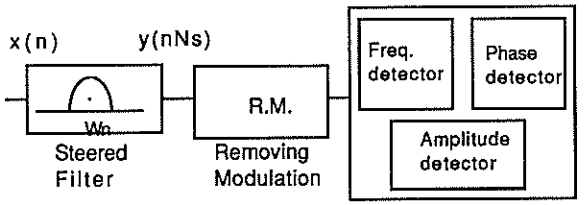


Figure 1. Basic scheme of the filtering/estimation process.

The three parameter detectors can be implemented jointly or separately inside the second stage.

Regardless our objective is the simultaneous estimation of the magnitude, phase and frequency, we may conclude that frequency represents the main difficulty and, for this reason, we will discuss initially the design of the filter of figure 1, in terms of frequency estimation only.

After modulation of the signal has been removed, the estimation stage is done, selecting a function objective to be minimized. The signal is modeled as a single tone with frequency, phase and magnitude to be estimated. The parameters are obtained in every burst or slot of signal, minimacing the Mean Square Error (MSE) between the real signal and its model, and they result optimum in a maximum likelihood sense, which allow to deal with modulated carriers and also provides narrowband interference rejection.

Phase and magnitude process are used in the synchronization of the system here presented, but optimum frequency obtained results high computationally inefficient. Because of this other kind of frequency estimation [2] with considerable lower computational load, is used here and it is compared with optimum methods to show that with Eb/No (bit energy to noise spectral density ratio) above 0 dB, both process give as result the same performance.

The experiments have been done over PSK-4 and PSK-8 modulated signals in coloured noise, and they confirm the statements on robustness and Eb/No threshold effects done previously.

Taking as base the optimum estimation method, other kind of suboptimal estimation for frequency and phase have been also proved. In a final system, each parameter detector, will have to be selected, and the particular solution will usually depend on a tradeoff between low computational load and optimum performance required in the system.

2. SIGNAL ENVIRONMENT

The type of signal used in this work is described. The kind of modulation is PSK. The information of the symbols is contained in the phase, added to the phase of the carrier signal. For N Symbols by burst, the received form will be as is shown in (1), where p(.) is the resulting pulse in the receiver without intersymbol interference.

$$x_r(t) = A.\sum_{s=1}^{N}\cos[(\omega_d+\omega_N)t+\theta+\theta_s]p(t-sT_s) \quad (1)$$

T_s is the symbol period, θ_s is the modulation phase and θ the leakage phase. It is assumed that a previous translation from the real carrier frequency ω_c to the nominal frequency ω_N has been done about the signal. There is a residual unknown frequency ω_d. This uncertainty has to be estimated to get appropriated accuracy.

The signal is sampled with a sampling period T normalized to be one. Taking in consideration the modulation process, the sampling is done to set $f_N + f_d$ around 0.25 normalized value at the sampling frequency. N_{SS} will be the number of samples by symbol. In every burst there are N symbols to process the signal (2).

$$x(i)=\sum_{s=1}^{Q}A.\cos[(\omega_d+\omega_N)n+\theta+\theta_s]p(n-s.N_{SS}) + n(i) \quad (2)$$

With this kind of signal, we have N symbols and $N.N_{SS}$ samples in every burst.

3. FILTERING STAGE

In the first stage of the receiver the signal is filtered to obtain a sample by symbol for each group of N_{SS} samples. It is assumed than accurate timing synchronization is got in this stage. With a vector notation at the output filter it will be

$$y(n)=\underline{A}^H.\underline{X}(n) \quad (3)$$

where $\underline{A}^H$ is the filter response and $\underline{X}(n)$ is the signal vector corresponding to a single symbol. Taking as the filter order Q, the same as the number of samples by symbol N_{SS} , there will be N samples of signal y(n) at the filter output.

Inside each symbol, the signal can be considered as a single sinusoid without any phase change. Receiving the signal with the steering vector at the nominal frequency ω_N, at the output of the filter, there will be the signal y(n), where the frequency will be just the unknown doppler frequency.

$$y(n) = \frac{1}{Q}\sum_{i=0}^{Q}x(n-i)\exp(j\omega_N\frac{i}{Q}) \quad (4)$$

In these conditions, the filter response $\underline{A}(n)$ is the steering vector $\underline{S}$

$$\underline{A}=\underline{S}=[1,\exp(-j\omega_N),.....(-j\omega_N(Q-1)] \quad (5)$$

Analysing y(n), analytically will be as (6)

$$y(n)= A.\exp[j\varphi(n)] + n(n) = A(n).\exp[j\psi(n)] \quad (6)$$

The instantaneous magnitude A(n) is the signal amplitude with noise and $\psi(n)$ is the instantaneous phase. $\varepsilon(n)$ is the output noise contained in $\psi(n)$.

$$\psi(n)=\omega_d.Q(n-1) + \theta_n + \theta + \omega_d\frac{Q-1}{2} + \varepsilon(n) \quad (7)$$

An approximation is assumed in (7). The phase of each sample is the central phase of the measure snapshot. Really in the case of no interferences and only additive Gaussian noise (7) will be the exact phase at the filter output.

Being our objective to estimate A and $\varphi(n)$ from a single snapshot $\underline{X}(n)$ there is an alternative to filter the signal x(n). It is found, deriving the maximum likelihood estimate for parameters A and ψ. The filter response is then obtained as a result of minimacing the difference between the signal vector and a sinusoid model for it, and it results as a data dependent vector. But in ideal conditions, we mean, without other sinusoid interferences, white noise, stationarity and signal autocorrelation matrix without estimation errors, the maximum likelihood filter results just as the filter presented in (5), the FFT processing of the current snapshot [1]. In other words, for the above mentioned assumption the optimum filtering is not longer data dependent and can be implemented as a DFT processor of length Q.

4 ESTIMATION STAGE

The available signal samples in this stage, are shown in (6) and (7). There are N_{SS} samples of y(n), a sample by symbol, and the objective is to estimate the residual frequency carrier ω_d, the phase leakage θ and also the magnitude A. The modulation phase θ_n is removed from the signal. In order to remove the modulation, we need to multiply the measured phase by the number of modulation levels M. This will remove from $\varphi(n)$ the contribution of modulation steps between

symbols but it will introduce 2π steps in the corresponding phase. To remove the 2π steps existing in the new phase, a phase unwrapping algorithm can be used when it is necessary. Considering the above explained operations, the new available signal to process parameters will be as it is shown in (8)

$$y(n)=A_n^H.\exp j(M\omega_d n + M\theta + M\frac{n-1}{2}\theta\omega_d + \varepsilon'(n)) =$$

$$A_n'.\exp j\varphi(n) \qquad (8)$$

Being the Mean Square Error (MSE) the selected objective, the problem is formulated in (9)

$$\xi = \sum_{n=1}^{N} |\, y(n) - A'(n).\exp(j\psi'(n)|^2) \quad (9)$$

where $\psi'(n)$ is equal to $\beta + \alpha(n-1)$. Identification of phase θ and frequency ω_d with the parameters α and β is trivial. It should be noted that the above objective is not an optimum criteria if the phase noise is not gaussian; but, even in the non gaussian case the associated performance is recognized. Due to the way y(n) is formed, if there are no wrong phase step correction in the unwrapping, the associated noise to the phase remains gaussian, when the Eb/No there is high enough, above 2 or 3 dB and the MSE estimation for magnitude A, frequency ω_d and phase θ remain optimum.

Taking derivatives of (9) with respect A'and φ'in terms of β and α, and setting to zero, the maximum likelihood estimations obtained are the following.

Magnitude estimation:

$$A=\frac{1}{N}\sum_{i=1}^{N} A(i).\cos\varepsilon'(n) \qquad (10)$$

the classical estimate for magnitude (11) results the optimum estimate MSE whenever $\varepsilon'(n)$ is small enough to assume the second term of the sum as one for all the measurements done

$$A \cong \frac{1}{N}\sum_{i=1}^{N} |\, y(n)| \qquad (11)$$

This estimate can be used to validate frequency and phase estimates, depending on the A level.

Concerning the phase derivative, and taking first the case of derivative with respect the phase leakage β (12) is obtained to compute the optimum phase.

The obtained optimum phase is just the same presented by Viterbi in [3] also as the optimum, but for the case of no doppler frequency, α=0. It is just the phase obtained, in the Fourier transform of the signal evaluated at α frequency.

$$\text{Phase: } \theta = \frac{1}{M}\, tg^{-1}\, \frac{\text{real}\left(\sum_{m=0}^{N-1} y(mT_s)\exp(-j\alpha(m-1))\right)}{\text{imag}\left(\sum_{m=0}^{N-1} y(mT_s)\exp(-j\alpha(m-1))\right)} \qquad (12)$$

The frequency estimate is obtained from the derivative of (9) with respect to parameter α. After some algebra it can be shown that the optimum α just sets out the condition to find an extrema of the periodogram of y(.) or the square magnitude of its Fourier Transform. Thus, the optimum way to determine the frequency leakage α, will be a DFT of the filter output samples or measurements available.

Three optimum solutions for magnitude, frequency and phase estimates have been given. They represent the maximum likelihood estimates with the only drawback for the frequency computes of its computational load. With the DFT method it would require a lot of samples to get a BER of the order of 10^{-4} .

The precedent method give the optimum ω_d estimation but with a considerable computation cost. As an alternative for frequency estimation, Kay's approach [2] minimizes an objective function between a finite consecutive samples phase differences vector (Δ) and $\alpha 1$. The computational load of the method is very low. It is synchronous with the symbol period, and gets the Cramer-Rao bound when Eb/No is above 6 dB, possible threshold for its use.

The resulting algorithm, yet preserving the mentioned properties, becomes optimum even for signal contaminated with coloured noise. This is of capital importance taking in mind that many currently TDMA systems in communications will require to track amplitude, phase and frequency from symbol to symbol, with a maximum of 16 symbols in the burst and with no more than 4 samples per symbol.

The determination of amplitude estimates provides an useful quality index of the phase and frequency estimates. This is due to the role played by carrier, or sinusoid, magnitude in the threshold effect. Note that, in order to obtain gaussian noise in the phase estimates, the input signal to noise ratio is above 0 dB is required. Once this constraint holds, the frequency and phase estimates achieve the Cramer- Rao bound. At this point, it is clear that narrow band interferences, coloured noise or sinusoid modulation destroy this property and, as a consequence, the over-all system fails.

As a suboptimum alternative to the optimum phase and frequency estimations, linear regression can be used with the phase of the removed modulation signal, to obtain α and β estimates. In this case the algorithms work directly with the signal phase samples.

Summarizing the final structure for the receiver, the general process to synchronize the TDMA signal is shown in figure 2.

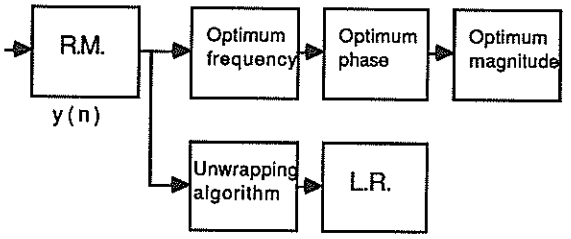


Figure 2. General Receiver

5. RESULTS AND COMMENTS

Simulations results were obtained over PSK-4 and PSK-8 modulated signals in coloured noise.

In the figure 3, for a 4-PSK signal with N=18 symbols, N_{ss}=5 samples/symbol, ω_d=.005, θ= 10º, the normalized errors are compute for Eb/No from -10 dB to 10 dB, with 50 randomized trials for each Eb/No value. For each signal, the maximum, average and minimum estimate errors are compute for parameters ω_d, θ and A.

Thresholds of Eb/No for work, could be estimated from figure 3. In these conditions 1 dB for Eb/No is appropriated to get low values of errors. For other values of parameters to estimate, the appropriated thresholds can also be processed and they always result better than the required Eb/No in PSK transmission if f_d is greater than 0,01 (5% of the carrier frequency).

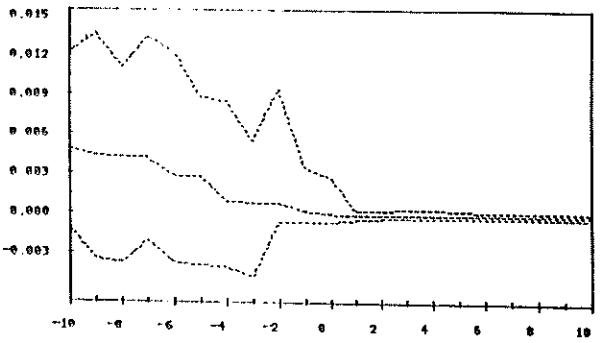
6. REFERENCES

[1] M. A. Lagunas, M. Cabrera. "Open Loop Joint Parameter Estimation for Burst Communication Mode". ESA (European Spatial Agency) Report, ESA-ESTEC Noorwijk (Netherlands), February 1990.

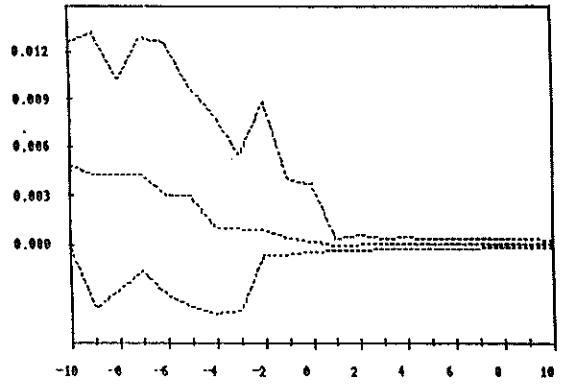
[2] S. Kay, "Statistically/computationally Efficient Frequency Estimation". IEEE-ICASSP 88, paper E3.5, New York 1987.

[3] A. J. Viterbi, A. M. Viterbi, "Nonlinear Estimation of PSK Modulated Carrier Phase with Applications to Burst Digital Transmission". IEEE Tr. on IT, Vol IT_29 nº 4, July 1983.

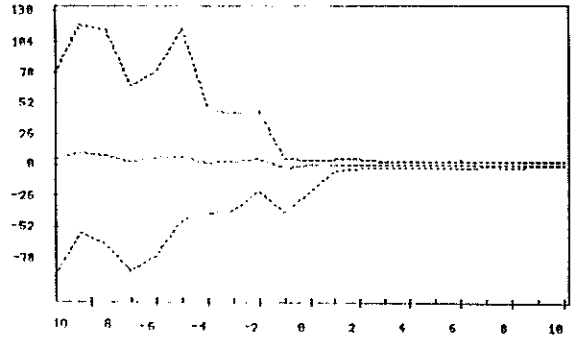
This work has been supported by ESA and PRONTIC



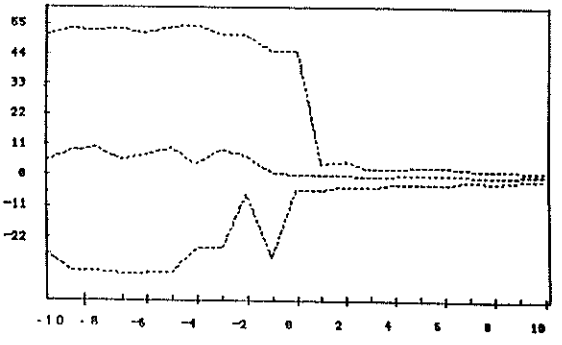
a) Linear regression frequency estimate



b) Kay frequency estimate



c) Linear regression phase estimate



d) ML phase estimate

Figure 3. 4PSK signal, ω_d=.005, θ=10º, 18 symbols and 5 samples/symbol. Values of Eb/No from -10 dB to + 10 dB.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

SYNCHRONIZATION IN DEEP NOISE OF COMMUNICATION SIGNALS

Dr. James W. Bond

Naval Ocean Systems Center, Code 83
271 Catalina Boulevard, San Diego, CA, 92152-5000

1. BACKGROUND

The properties of military communications impose constraints on the design of the communication signal and on the processing needed to communicate under adverse conditions. Important military messages are often short. Therefore, the synchronization process should be designed so that synchronization can be achieved on a single short message and its copies.

Traditionally, military communication requirements have often been met by spreading the information bits through use of a pseudo-random chip sequence. Recent developments in microprocessor technology allow digital radios to be built with significantly more processing capability and memory than earlier radios. Signal processing techniques, which can be used to suppress interference, can be implemented in such a radio, as well as techniques to identify and/or correct bit errors. As a result, the performance of a communication mode in a noisy channel need not be provided by bandspreading alone.

The motivation for designing a communication mode which uses these new techniques is that they allow for faster message delivery on fixed bandwidth channels than communication modes which just use bandspreading.

These considerations led us at NAVOCEANSYSCEN to consider the design of communication modes involving a moderate degree of bandspreading, 16 to 128 chips/symbol.

The use of automated copy code combining and convolutional encoding with a moderate amount of bandspreading leads to a communication signal with Es/No values far lower than a mode with similar signal-to-noise performance using bandspreading alone. This paper treats chip synchronization for modes with Es/No -6 dB, i.e., chip synchronization in deep noise.

Synchronization will be treated for the following mode designed to allow transmission of messages of 10 to 100 characters. Each character will be represented by 6 bits, which are encoded in 12 symbols by a rate 1/2 convolutional encoder, and spread by 40 chips. The chips are broadcast at 2400 baud and transmitted using a Minimum Shift Keyed (MSK) signal. The received signal will be coherently demodulated, Viterbi decoded, and the copies automatically combined.

The author was one member of the team which evolved the chip synchronization approach described in this paper. The author has made use of technical memorandum provided to the NAVY by Dr. Per Kullstam, Mr. Arnie Michelson, and Mr. Scott Little.

2. FORMULATION OF THE CHIP SYNCHRONIZATION PROBLEM

Each symbol is spread by a pseudo-random chip sequence and the symbols are recovered within the receiver by correlating the known chip sequence with the received signal. Chip synchronization consists of aligning the internal chip sequence timing with that of the received signal.

To avoid operator input, it might be desirable to allow for a timing offset between the transmitted signal and the receiver for reception anywhere on the surface of the earth, i.e., 13,000 miles/186,000 miles per second $\cong$.07 seconds, which corresponds to approximately (2400)(.07) $\cong$ 168 chip durations.

For an MSK signal, a timing offset of 1/2 chip leads to a coherent chip demodulation loss of 2.4 dB. Chip synchronization consists of a gross synchronization step (synchronization to within 1/2 chip), followed by fine synchronization (synchronization to within 1/8 chip). Fine synchronization to 1/8 chip suffices since it results in a maximum coherent chip demodulation loss, incurred by a 1/16 chip offset between the internal and received chip timing, of about .2 dB.

To achieve gross synchronization, the received signal is correlated with internally generated chip sequences for 168 timing offsets, each one chip apart. The offset yielding the highest correlation is then chosen as the best estimate of the correct timing offset. It is easy to see from these observations why gross chip synchronization can become one of the most computationally intensive processing tasks performed by a receiver.

A message will usually be lost if synchronization is declared for the wrong offset, so it is desirable to synchronize with a high probability of correct synchronization, say .9999, with a low probability of false synchronization of 10^{-4}. We took these as design goals.

Copy combining depends on the proper alignment of copies of the same message so that corresponding symbols can be combined. It is natural to include this task with the task of gross synchronization.

3. NEED FOR A SYNCHRONIZATION HEADER

There are two ways to achieve gross synchronization: (1) synchronize on the message symbols which are unknown, and (2) synchronize on known symbols provided for synchronization.

If synchronization is achieved by processing unknown symbols and message data is not to be lost while achieving synchronization, then either (1a) the predemodulation chip samples need to be saved until the correct timing offset is identified or (1b) symbol level inphase and quadrature data needs to be saved for each of the 168 timing offsets until the correct timing offset is identified. For option (1a) the symbol inphase and quadrature samples have to be generated from the saved chip data after synchronization is achieved and before message processing can begin.

If synchronization is achieved through use of known symbols within the message, these known symbols should be placed at the beginning of the message to minimize the requirements to save message data and the time to declare synchronization.

Synchronization based on processing message data gives no information on when the message started or which copy of a multicopy message is being received. In contrast, declaring synchronization based on known symbols incorporated in a header provides the start time of the message and as will be seen in the next subsection, it can also provide the copy number.

To compare the different options we formulate gross synchronization as a classical detection problem. The development which follows is based on a memo by Dr. Kullstam in support of the mode design effort on which this paper is based. "The general mathematical formulation of this problem is to consider a set {u(m)} of M hypotheses test metrics of which one contains the signal and the others not. The correct signal acquisition decision probability

$$\Pr\{\text{correct}\} = E[\Pr\{\text{all}^{o}u(m) < u\}|\text{signal}]$$

where the 'zero exponent' (^{o}) denotes that we have not included the metric for which the signal is present and E[u|signal] is the expectation given the signal present for the metric u."

Dr. Kullstam then makes use of the union bound to obtain

$$\Pr\{\text{correct}\} \geq 1 - \text{sum}[^{o}E[\Pr\{u(m) \geq u\}|\text{signal}].$$

The probability densities underlying the problem are given by

$$p(u|Eo) = [1/\sqrt{(\pi No)}\,]\exp[-(u\mp\sqrt{Eo})^2/No]$$

for signal present ($Eo \neq 0$) and

$$p(u|Eo) = [1/\sqrt{(\pi No)}\,]\exp[-u^2/No]$$

for signal absent ($Eo = 0$), where

Eo is the total energy over the observation interval and No is the noise variance.

Then for known data Dr. Kullstam obtains

$$\Pr\{\text{miss}\} \leq [(M-1)/2]\text{erfc}\sqrt{(Eo/2No)} \cong [(M-1)/\sqrt{(2\pi Eo/No)}\,]\exp(-Eo/2No)$$

for M hypotheses. Here the detection variable is

$$\Sigma(sI_s + Q_s),$$

where (I_s, Q_s) is the inphase and quadrature sample pair for the symbol s given by chip demodulation for a given timing offset and the summation is over the symbols of the header. The "s" inside the summation represents the sign of the symbol "s".

Thresholds to provide a $P_{FA} = 10^{-4}$ for noise alone are calculated for $\hat{N} = 12\hat{N}$, with $\hat{N}$ an integer. Then the probability that the detection variable would exceed the threshold is calculated for each $\hat{N}$ given that the signal was present and Es/No = -6 dB. In this manner, we found that 156 symbols are not quite enough to achieve a $P_D = .9999$ and 168 symbols more than enough to achieve a $P_D = .9999$ (.99992 is achieved).

Next Dr. Kullstam calculated a "non-coherence loss" for combining absolute values and squares of the symbols. Here, combining absolute values is equivalent to using the detection variable

$$\Sigma(\,|sI_s| + |sQ_s|\,)$$

to declare gross chip synchronization and combining squares is equivalent to using the detection variable

$$\Sigma(\,|sI_s|^2 + |sQ_s|^2\,)$$

to declare synchronization.

The formulas for the losses are

$$L_{av} = 10\log[2(1+No/Es)]$$

and

$$L_{sqr} = 10\log[(\pi-2)(1+No/Es)]$$

respectively. These formulas showed that 10 times as many unknown symbols as known symbols are necessary if non-coherent summation uses absolute values and 5.7 times as many when sums of squares are used for Es/No = -6 dB. This would mean that 10(168) = 1680 symbols would be required to achieve synchronization if the energy of the unknown symbols was combined using absolute values. If the sum of the squares method is used, then (5.7)(168) = 958 symbols (rounded to the nearest integer) are required. After obtaining these results, we restricted our attention to messages with synchronization headers and focused on how to effectively design the headers.

4. USE OF SYNCHRONIZATION HEADER INFORMATION TO FRAME MESSAGES

Given that propagation delays can only shift the initial time of the header sequence by a fraction of a second (about .07 seconds assuming no knowledge of the transmission path for a 2400 chips/second), detection of the presence of a synchronization header allows determination of the start time of a message to the nearest second.

Modular arithmetic provides the analytical technique to provide all the required information. Protocols are considered which allow transmission of 2^k copies of a message in such a way that the number of copies yet to be transmitted can be determined from knowledge of the start time of the first received message header time. The required protocol follows: the first copy transmission time is a multiple of 2^h seconds, 2^k copies are broadcast, with $k \leq h$ and the messages are an odd integer number of seconds in length.

Then the copy number of any message can be determined by the remainder obtained by dividing the message start time by 2^h. To see this, suppose that the i-th and j-th copies of the message have the same remainder after division by 2^h and let L denote the length of the message in seconds. Then we have

$$iL - m_i 2^h = r = jL - m_j 2^h$$

which implies that

$$(i-j)L = (m_i - m_j)2^h$$

and since L is an odd number 2^h must divide $|i-j|$, a number less than 2^k, so therefore $|i-j| = 0$.

Note that if k = h in the protocol then some message copy starts on each second mark so that the protocol can be very efficient.

The requirement that the message be an exact number of seconds in length places a constraint on the allowable lengths of the header unless filler symbols are used. For convolutional encoding by a rate 1/2 constraint length 7 code, 6 flush bits are necessary, which leads to 12 symbols in the message. Let N be the number of symbols in the header. Then, for a 2400 baud mode with 40 chips/symbol bandspreading:

$$N + 12(M \text{ message characters} + 1) = 60T$$

with N, M, and T being positive integers. It immediately follows that 12 must divide N. Conversely, if $N = 12N^{\wedge}$ then

$$12N^{\wedge} + 12(M + 1) = 60T$$

which means that the message will have integer second length if the number of characters in the message satisfies

$$M = 5T - N^{\wedge} - 1$$

for some choice of T. Imposing the requirement that the duration be an odd number of seconds implies that all message lengths must change in increments of 10 characters to be broadcast for a given length header without need of filler bits.

The protocols described in this section have been formulated in terms of seconds. The amount of time available to perform synchronization processing depends on the protocol. For example, If messages are transmitted on 8 second tick marks rather than 4 second tick marks, the available time doubles. It is not necessary to start on second marks. For example, protocols could be developed for time units of 1/2 and 1/4 of a second, which would reduce the amount of time available for synchronization processing by factors of 1/2 and 1/4, respectively.

5. HEADER LENGTH AND MODE EFFICIENCY

We define mode efficiency in the following way:

$$100(6M/60L) = 10(M/L) \text{ percent}$$

with M the number of characters and L the length of the message in seconds.

If the message consists of text alone, then 6M = 60L and the efficiency would be 100 percent. For such a message, whose bits are encoded by a rate 1/2 convolutional code, the efficiency would be somewhat less than 50 percent since 12(M+1) = 60L because of the need for 6 tail bits to initialize the Viterbi decoding process. If a header is introduced then the efficiency is further reduced:

$10(M/L) = 600M/(12N^{\wedge}+12(M+1)$

$= 50M/(M+N^{\wedge}+1)$ if $12N^{\wedge}+12(M+1) = 60L$

6. SYNCHRONIZATION ON MULTIPLE HEADERS

The increasing loss of mode efficiency for short messages with increasing synchronization header length motivated the consideration of efficient ways to use multiple synchronization headers for gross chip synchronization.

The timing protocol allows the synchronization processing to be tailored to the number of copies which might be combined. The only fundamental constraint on the synchronization header length N is that kN symbols should suffice to support k copy combining at the k-th copy Es/No design point for values of k less than or equal to the number of copies of a message transmitted.

The penalty for using multiple headers to declare synchronization is not in delivery time, at least for steady state channels, but rather in the requirement to save either chip samples (or bit samples) for the 168 different chip sequence timing offsets until synchronization is declared. This suggests the question: can a first threshold be set to decide a few offset times for which bit data needs to be saved with synchronization declared after processing the synchronization headers of subsequent copies of the message? Arnie Michelson suggested the appropriate analytical formulation of gross synchronization as a detection problem so that a precise answer could be given to this question.

Consider the case of using the information in two synchronization headers. (It is easy to generalize to the multiple copy case from this case.) Multiple thresholds are set and used as follows:

1. Given a synchronization header of length N set a first threshold so that $P_{FA} = 10^{-2}$ for noise alone and save bit data for up to K threshold crossings so that the probability of one of them being the correct timing offset is $P_d = .9999$ or greater at the 2^k copy combining Es/No design point.

2. Continue to search for gross chip synchronization at other possible header start times until synchronization is declared.

3. Set the second threshold using the combined energies of the first and second headers so that $P_{FA} = 10^{-4}$ for noise alone and the probability of declaring synchronization for one of the K hypotheses for which bit data has been saved is $P_D = .9999$ when the signal is present.

This approach provides a mechanism to separately meet the requirements for P_D and P_{FA}. It provides a way to trade-off synchronization header length (mode efficiency), the number of synchronization headers for which bit data needs to be saved, and the memory required to save bit data prior to declaring synchronization. It can be used in various ways to determine an overall synchronization approach. Note that if a large number of messages are to be transmitted with a small but fixed number of symbols, then the integer odd second length requirement leads to allowable header lengths, differing by 60 symbols so that very few calculations are required to investigate all of the feasible options. If the short messages lengths are variable, then the number of headers that are to be combined determine the design.

For 4-copy combining using 4 headers, the natural starting point is to choose N to work at 0 dB and check if 2N can support 2-copy combining, 3N 3-copy combining, and 4N 4-copy combining. The multiple threshold performance is very close to that given by a single threshold test. The following single threshold results for $P_{FA} = 10^{-4}$ indicate that this approach would be feasible: $P_D = .99998$ for $E_s/N_o = 0$ dB and $N = 48$; $P_D = .999998$ for $E_s/N_o = -3.0$ dB and $N = 96$; $P_D = .999997$ for $E_s/N_o = -4.8$ dB and $N = 144$; $P_D = .999995$ for $E_s/N_o = -6.0$ dB and $N = 192$.

For 4-copy combining using 2 headers, N should be chosen about half the length required for 4-copy combining, i.e., $168/2 = 84$. Then the natural thing to check is whether 84 bits is enough to support 2-copy combining. This almost works, as indicated by the result $P_D = .9997$ and $P_{FA} = 10^{-4}$ for $E_s/N_o = -3$ dB and $N = 84$. Thus using a header of 96 symbols would support 2-copy combining at the design point.

If messages need to be transmitted ranging from very short to moderate length, then our analysis suggests using multiple header processing for the few shortest message lengths and single header processing for all the remaining allowable message lengths.

7. SUMMARY

The designs suggested by our analysis effectively provide gross synchronization while providing the information needed to provide copy framing. This is achieved through use of integer odd message lengths and prescribed multi-message transmission times depending on the number of messages to be broadcast. Use of multiple headers with multiple thresholds allows for greater mode efficiency for the transmission of the shortest messages. We feel that we have described an approach that can be used to design a gross synchronization process suitable for performing synchronization in deep noise.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

LINEAR PHASE ADAPTIVE LINE ENHANCER FOR IMPROVING THE PERFORMANCE OF PHASE SYNCHRONIZERS

F.Castro. J.Castells. J.Sánchez. G.Vázquez.

Department of Signal Theory and Communications
E.T.S.E. Telecomunicació.Apto 30002, (08080) Barcelona (Spain)

Abstract:

This paper deals with the problem of symbol recovery under low SNR conditions and nonlinear channel effects. In order to improve the performance of phase synchronizers the use of some linear ALE methods is proposed: classic ALE, CMA, and ML filtering.
Comparisons among them are made from different signal processing aspects such as capacity of noise removal and speed properties.
A linear regression is shown to be a good quick form for phase and frequency adquisition. Finally two types of ALE structures are suggested.

I - INTRODUCTION

The main goal of this paper is the inclusion of advanced DSP methods in classical communication structures. In our study we specially consider open loop diagrams to achieve good solutions for burst PSK modulation.

As it is well known, in communication systems good estimation of amplitude, phase and synchronize are required.

An easy method for symbol recovery without demodulation process can be implemented with a bank of matched filters to each possible waveform. Only a good symbol synchronization is required.

Unfortunately, this structure is is not flexible and cannot adapt itself to the variant characteristics of the incoming signal. Therefore a Doppler deviation, for instance, forces the system to work in a different frequency which is not the nominal one, so the performance becomes worse.

Thus, this paper suggests the use of a linear regression method as frequency and phase estimator. In order to improve it, we deal with an ALE, CMA and ML filtering as a previous process.

II - THE LINEAR REGRESSION METHOD.

A linear regression is an easy and quick form for obtaining the phase and frequency of an unmodulated carry signals from its phase samples.

The next picture shows the block diagram of a frecuency and phase estimator based on linear regression.

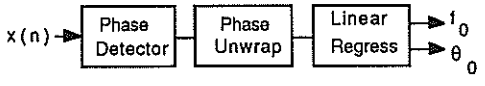


Figure 1

If an arctangent phase detector is used, (fig 2), the phase unwrapping reconstructs a segment of the phase straight line.

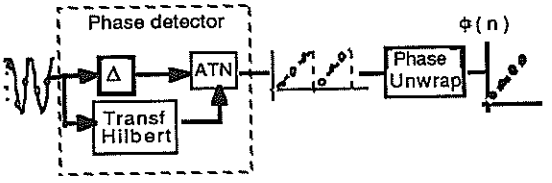


Figure 2

Then, a linear regression from the noisy samples $\hat{\phi}(n)$ is performed :

If $\hat{\phi}(n)$ represents the least square linear, then

$$E(\phi(n) - \hat{\phi}(n))^2 \rightarrow \min \qquad (1)$$

where $\hat{\phi}(n) = \hat{\omega}_0 nT + \hat{\theta}_0$

Minimizing the cost function (1) we achieve the optimal solution for $\hat{\omega}_0$ and $\hat{\theta}_0$, that is

$$\hat{\omega}_0 = \frac{E(n\phi(n)) - E(n)E(\phi(n))}{E(n^2) - E(n)E(n)}$$

$$\hat{\theta}_0 = \frac{E(\phi(n))E(n^2) - E(n)E(n\phi(n))}{E(n^2) - E(n)E(n)}$$

Graphically $\hat{\omega}_0$ and $\hat{\theta}_0$ can be seen as the slope and the origin ordinate of the straight line that matches the sample set $\phi(n)$.

The sample window length becomes a tradeoff between the ability of the system to adap itself to frecuency shifts and the noise sensitivity.

This work was granted by the CICYT National Research Plan on Information and Communication Technologies of Spain.

The structure presents a thereshold level in noise and a maximum estimated frequency since a nonlinear operation (the phase unwrapping) is employed; a high level of noise masks the 2π crossings of the sawtooth phase signal.

We can improve the properties of the linear estimator placing a previous equalizer that removes both noise and modulation from the signal.

An adaptive equalizer can adjust its behavior to the statistics of the incoming signal.

III - ADAPTIVE SYSTEMS

- Introduction

In the next sections we'll suppose that symbol synchronization is known.

Furthermore only digital angular modulation is employed, due to its immunity in front of noise, channel nonlinearities, etc. For this reason the use of linear phase adaptive systems is desired (information "travels" on the phase).

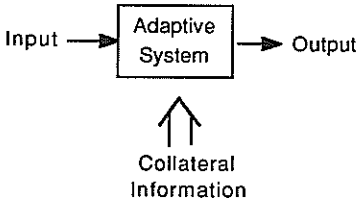


Figure 3

An adaptive system obtains from the incoming signal a specific information (output signal), thanks to an additional or collateral external information.

We can consider two types of collateral information:

- Independent information from the input and output signals.
For example, the modulus of the output signal (CMA algorithm), the nominal carry frequency (ML filtering), etc.

- Dependent information.
For example, a reference signal (ALE).

The type of collateral information and its use determine the different methods of adaptive filtering.

- The Adaptive Line Enhancer (ALE)

The ALE [1], which basic scheme can be seen in figure 4, was initially designed as a degenerate form of a noise canceller.

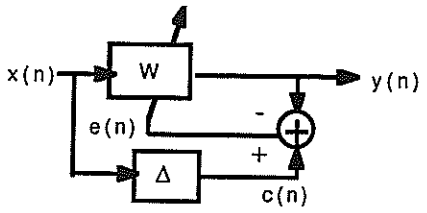


Figure 4. Classic ALE structure

The ALE uses the LMS adaptive algorithm to recursively adjust the weights of the filter.

So $W(n) = W(n-1) + 2\,\mu\, e(n)\, X(n)$
with $e(n) = x(n) - y(n)$

The collateral information employed is an appropiate delayed replica of the incoming signal.

The delay Δ is chosen to obtain a high level of correlation at the input of the comparator.

Therefore if
$x(n)= A \sin(\omega_0 nT + \theta_0) + No(n)$
with $\omega_0=2\pi/T_o$

then Δ is forced to be $\Delta= m\, T_o$ with m= 1,2,...

Thus the error signal e(n) has several incorrelated components, basically noise, and it's used to properly adjust the frecuency response of the filter.

ALE reachs a high level of noise cancelling, even in very low SNR cases, as shown below.

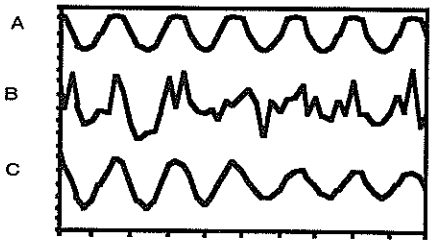


Figure 5. (A) Ideal signal
(B) Noisy signal
(C) Ouptut signal of the filter
SNR= 0 dB

For a fast convergence a weight vector
$W(0)=1/LF\ (1,\ \cos(\omega_0),...\cos(\omega_0(LF-1)))$
is used, so initially the algorithm performs a DFT.

The μ parameter controls the stability, convergence speed and filter capacity of the ALE.

In the classic ALE scheme it can be shown the reference is noisy because it's obtained from the incoming signal. An improved scheme extracts the reference from the filter output.

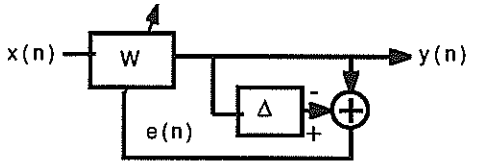


Figure 6. Improved ALE scheme.

When the input signal is a modulate signal the performance of the ALE becomes worse because no Δ produces a high level of correlation. Then the CMA algorithm must be used.

- Constant Modulus Algorithm (CMA)

The objective of this adaptive filtering [2] [3] is to restore the incoming signal to a form wich, on the average, has a constant instantaneous modulus.

The collateral information used is the knowledge about the constant modulus of the output signal.

The error function to minimize is

$$\zeta = E\left(\left(\|y(n)\|^2 - 1\right)^2\right)$$

Thus, using a gradient search algorithm

$$W(n+1) = W(n) - \mu\left(\|y(n)\|^2 - 1\right) y(n) \quad X^*(n)$$

This CMA was initially introduced as a method of correcting the multipath effects:

If $x(n) = A \exp(j(\omega_0 nT + \theta_0 + \theta(n))) + No(n)$

then y(n) defined as $y(n) = x(n) + \alpha\ x(n-\Delta)$

has not constant modulus.

In the next figure the correcting process is shown

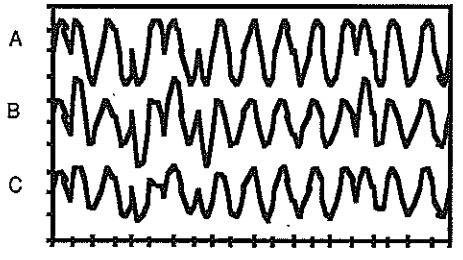


Figure 7. (A) Ideal BPSK signal
(B) Multipath corrupted signal
(C) Output signal of CMA.

With unmodulated signals the CMA performance becomes worse than ALE, since ALE works with a reference signal, which supplies speed and noise removal capability.

- Maximum Likehood (ML) Filtering

The ML filter [3] uses the knowledge of the nominal frequency of the carry signal as collateral information.

The filter is designed under the assumption on minimizing the output power.

$$\min\ (P_y) = W^H R W$$

under the constrain equation

$$S^H W = 1$$

where

$$S = (1,\ \exp(-j\ \omega_0 T)\ ,.\exp(-j\ \omega_0 (N-1)\ T))^T$$

and R = Autocorrelation matrix of X data.

The function to be minimized is

$$\zeta = P_y - \lambda\ (S^H W - 1)$$

being λ a Lagrange multiplier.

Taking the gradient $\nabla \zeta = 0 = R\ W_{opt} + \lambda\ S$

Since $S^H W_{opt} = 1$

then $W_{opt} = R^{-1} S\ (S^H R^{-1} S)^{-1}$

If, as usually, the R matrix is unknown an adaptive algorithm can be used.

Thus, if $W(n+1) = W(n) - \mu\ \nabla \zeta$

then $W(n+1) = P\ (W(n) - \mu\ y(n)\ X(n)) + F$

where

$P = I - S\ (S^H S)^{-1}\ S^H$ (Proyection vector)

$F = S\ (S^H S)^{-1}$ (Shift vector)

The filter works respecting a single tone steered at ω_0 and removing the rest components as possible.

In the particular case of considering white noise then

$$R^{-1} = \text{constant } I$$

Then $Wopt = R^{-1} S\ (S^H R^{-1} S)^{-1} = S/L$

where L = DFT length

This is equivalent to perform a DFT of the input signal steered at the frecuency ω_0.

$$DFT(x(n)) = \sum x(n) \exp(-j\ \omega_0\ n\ m\ T))$$

with m=1

The DFT filtering provides information about the amplitude and phase of the signal.

Let's assume a MPSK transmission.

Then

$$x(n) = A \exp(j\ \omega_0\ n\ T + \theta_0 + \theta(n)) + No(n)$$

We can estimate the phase of each incoming symbol taking only one sample per symbol of the DFT argument output.

The modulation must be removed from the set of DFT samples. To do it, the phase samples are multiplied by M. Thus the modulation phase term become 2π m jumps and they are removed by the phase unwrapping system.

Symbol synchronization can be achieved making a L length DFT over the whole data string (being L = number of samples per symbol). When the DFT window is centered just on a symbol the modulus is maximum. In other cases it clearly decreases.

Therefore adjusting the sampling symbol instants (L samples separated) to the modulus of the DFT output the symbol synchronization can be achieved.

The DFT results in an attractive usefull method in burst transmission mode thanks to its speed and simplicity.

IV - CONCLUSIONS

The analysis of the previous methods suggests the implementation of two different structures:

- Structure based on ALE filtering

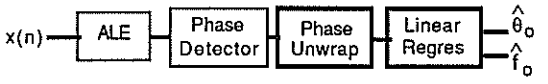


Figure 8

The adaptive characteristics adjusts the behavior of the filtering according to the variations of signal parameters (noise, frequency). No information about nominal frequency is required, so a wide frequency range can be tracked.

The convergence time makes it not fast enough for burst transmission mode cases. However an initial DFT vector supplies a high speed improvement.

Under modulated signals ALE only provides a first approximation for frequency estimation.

- Structure based on ML filtering (DFT)

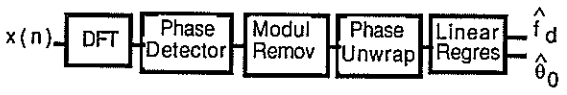


Figure 9

The quick processing time is very usefull in burst mode thanks to the open loop process employed.

Furthermore the structure can be used both in modulated and unmodulated cases.

This system requires the knowledge of the nominal carry frequency, so only Doppler deviations can be tracked.

REFERENCES

[1] John R. Treichler, "Transient and Convergent Behavior Of the Adaptive Line Enhancer" IEEE Trans Acoust, Speech, Signal Processing, vol. ASSP-27, NO.1, FEBRUARY 1979.

[2] John R. Treichler and Brian G. Agee,"A New Approach to Multipath Correction of Constant Modulus Signals". IEEE Trans Acoust, Speech, Signal Processing, vol. ASSP-31 .NO.2, APR 1983.

[3] John R. Treichler and Michael G. Larimore, "New Processing Techniques Based on the Constant Modulus Adaptive Algorithm". IEEE Trans Acoust, Speech, Signal Processing, vol ASSP.33, NO.2, APR.1985.

[4] Otis Lamont Frost, "An Algorithm for Linearly Constrained Adaptive Array Processing". IEEE Proceedings, vol 60, NO 8, August 1972.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

REJECTION OF MULTI-TONE INTERFERENCE IN PN SPREAD SPECTRUM SYSTEMS USING LINEARLY CONSTRAINED LMSE FILTERS

Cheng Zhong
Dept. of Electr. Eng.
Shenyang Inst. of Aero. Eng.
Shenyang 110031, China.

Zhengmao Li
Inst. of Inform. Syst.
Univ. of Sci. & Tech. of China
Chengdu 610054, China.

Fuhua Lin
Dept. of Radio Eng.
Southeast Univ.
Nanjing 210018, China.

This paper presents a novel method for the rejection of multi-tone interference (MTI) in pseudo-noise (PN) spread spectrum systems using linearly constrained least mean square error (LC-LMSE) filter. It is shown that much more complete rejection of the MTI can be achieved by the LC-LMSE filter than by the conventional LMSE filter. When the power of the MTI and/or the length of the above filters are large enough, the SNR improvement (a criterion-of-goodness) of the LC-LMSE filter is comparable to that of the LMSE filter. Besides, being insensitive to the input power, the LC-LMSE filter is suitable to the case where the received signal varies intensely in amplitude, such as in mobile communications.

1. INTRODUCTION

In recent years, various methods have been proposed for the interference rejection in pseudo-noise (PN) spread spectrum systems[1]-[3]. The most important and widely used one is the least mean square error (LMSE) filtering. The performance of multi-tone interference (MTI) rejection has been analyzed with this method[4]. It should be noted, however, that (1) the LMSE filter cannot reject the MTI perfectly unless the length of the filter is large enough; (2) the LMSE filter is hard to work optimally, especially when the input varies intensely in amplitude. In order to overcome these drawbacks, the linear constraints scheme[5] is used here to reject the MTI.

In this paper, a linearly constrained LMSE (LC-LMSE) filter is employed to reject the MTI and its performance analyzed. Different from the narrow-band interference (NBI) case[6], in order to reject the MTI completely, a certain condition concerning the length of the filter should be met, and proper constraints be chosen. The results obtained in the MTI case are also true of the NBI case. The depth of the notches provided by the LC-LMSE filter is significiently improved compared with the LMSE filter when using the same length of the filter. It is also shown that the SNR improvement (a criterion-of-goodness) of the LC-LMSE filter is comparable to that of the LMSE filter when the power of MTI and/or the length of the filter are large enough. Besides, being insensitive to the input power, the LC-LMSE filer is suitable to the case where the received input varies intensely in amplitude, such as in mobile communications.

2. PERFORMANCE ANALYSIS

The LC-LMSE filtering can be regarded as a linearly constrained random optimization problem. The general statement is as follows[5]:

$$\min_{W} E\{|d(k) - W^T X(k)|^2\} \tag{1a}$$

$$\text{Subject to: } C^+ W = f \tag{1b}$$

where
- X(k): N x 1 vector of input sequence x(k)
- W : N x 1 vector of filter tap weights Wi
- d(k): desired signal
- C : N x 1 vector of constraints parameters
- f : a complex scalar of constraints parameters
- N : length of the LMSE and the LC-LMSE filters
- M : number of the MTI
- * : conjugate operation symbol
- T : transpose operation symbol
- + : conjugate transpose operation symbol.

If the input sequence x(k) is wide-sense stationary, and X(k) and d(k) are jointly stationary, then the solution to problem (1) is given by

$$W_{LC,opt} = R_{xx}^{-1}(r_{dx} + \lambda^* C) \tag{2}$$

where R_{xx} is a N x N dimensional matrix of the auto-correlation of X(k), and r_{dx} is a N x 1 vector of cross-correlation between d(k) and X(k). λ is the Lagrange multiplier, which is given by

$$\lambda = (f^* - r_{dx}^+ R_{xx}^{-1} C) / (C^+ R_{xx} C) \tag{3}$$

The minimum mean square error of the LC-LMSE filter is given by

*This work was supported in part by the National Natural Science Foundation of China under Grant 68620**44 and 6872018.

$$\varepsilon_{LC,min}=E(|d(k)|^2)-W_{LC,opt}^{+}r_{dx}+\lambda^{*}f \qquad (4)$$

If $\lambda=0$, then $W_{LC,opt}$ is reduced to W_{opt}, which is the Wiener solution of the LMSE filter.

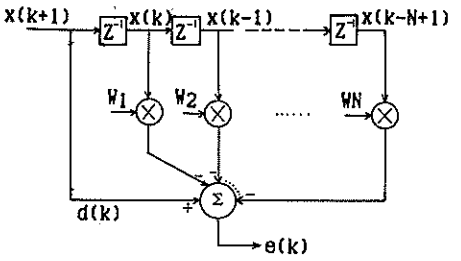


Fig. 1. LC-LMSE filter

$$\begin{bmatrix} S+\sigma_n^2+\sum_{m=1}^{M}J_m & \sum_{m=1}^{M}J_m e^{-j\omega_m} & \cdots & \sum_{m=1}^{M}J_m e^{-j(N-1)\omega_m} \\ \sum_{m=1}^{M}J_m e^{j\omega_m} & S+\sigma_n^2+\sum_{m=1}^{M}J_m & \cdots & \sum_{m=1}^{M}J_m e^{-j(N-2)\omega_m} \\ \vdots & \vdots & & \vdots \\ \sum_{m=1}^{M}J_m e^{j(N-1)\omega_m} & \sum_{m=1}^{M}J_m e^{j(N-2)\omega_m} & \cdots & S+\sigma_n^2+\sum_{m=1}^{M}J_m \end{bmatrix} \begin{bmatrix} \sum_{r=1}^{M}B_r e^{j\omega_r} \\ \sum_{r=1}^{M}B_r e^{j2\omega_r} \\ \vdots \\ \sum_{r=1}^{M}B_r e^{jN\omega_r} \end{bmatrix} = \begin{bmatrix} \sum_{r=1}^{M}(J_r+\frac{\lambda^*}{M})e^{j\omega_r} \\ \sum_{r=1}^{M}(J_r+\frac{\lambda^*}{M})e^{2\omega_r} \\ \vdots \\ \sum_{r=1}^{M}(J_r+\frac{\lambda^*}{M})e^{jN\omega_r} \end{bmatrix} \qquad (11)$$

The LC-LMSE filter used in PN spread spectrum systems for the rejection of MTI is shown in Fig.1. It is essentially the configuration described in [4] and [6]. In this case, d(k) in the above formulae is replaced by x(k+1). The samples at the input of the LC-LMSE filter can be expressed as

$$x(k)=s(k)+n(k)+i(k) \qquad k=1,2,\ldots \qquad (5)$$

where s(k), n(k) and i(k) are mutually independent sequences of the desired PN spread spectrum signal, additive white Gaussian noise (AWGN) and the MTI, with power S, σ_n^2, and J, respectively. The MTI i(k) can be further expressed as

$$i(k)=\sum_{m=1}^{M}A_m e^{j(\omega_m k+\theta_m)} \qquad (6)$$

where A_m, ω_m, and θ_m ($m=1,2,\ldots,M$) are amplitudes, offset radian frequencies from the carrier and random phases uniformly distributed over $[0,2\pi]$, respectively. With (5) and (6), $\phi_{xx}(l)$, the auto-correlation fuction of x(k), is given by

$$\phi_{xx}(l)\triangleq E\{x(k)x^{*}(k+l)\}=(S+\sigma_n^2)\delta_{0l}+\sum_{m=1}^{M}J_m e^{j\omega_m l} \qquad (7)$$

where δ_{0l} is the Kronecker symbol, $J_m=A_m^2/2$, $m=1,2,\ldots,M$, are the power of the individual interference of the MTI.

In order to make the LC-LMSE filter behave as a deeply notched filter at the frequencies of the MTI, we choose C and f as

$$C^{+}=\frac{1}{M}\left(\sum_{m=1}^{M}e^{-j\omega_m},\sum_{m=1}^{M}e^{-j2\omega_m},\ldots,\sum_{m=1}^{M}e^{-jN\omega_m}\right)^{T} \qquad (8)$$

and $f=1$ (9)

Let $W_{LC,opt}(l)=\sum_{r=1}^{M}B_r e^{j\omega_r l}$ $l=1,2,\ldots,N$ (10)

where B_r, $r=1,2,\ldots,M$, are undetermined coefficients. Substituting (10) into the matrix form of (2), we obtain (11)

Comparing both sides of any row in (11), for example, the first row for convience, we obtain

$$\sum_{r=1}^{M}NJ_r B_r e^{j\omega_r}+\sum_{r=1}^{M}B_r(S+\sigma_n^2)e^{j\omega_r}+\sum_{\substack{m=1\\ m\neq r}}^{M}\sum_{r=1}^{M}J_m B_r\gamma(\omega_r-\omega_m)e^{j\omega_r}=\sum_{r=1}^{M}(J_r+\frac{\lambda^*}{M})e^{j\omega_r} \qquad (12)$$

where $\gamma(\omega)$ is defined by

$$\gamma(\omega)\triangleq\frac{1-e^{jN\omega}}{1-e^{j\omega}} \qquad (13)$$

The properties of $\gamma(\omega)$ have been discussed in detail in [7], so we will use the results directly. It is shown that $\gamma(\omega_r-\omega_m)$ is perfectly zero when the difference between ω_m and ω_r $m\neq r$ is integral multiples of $2\pi/N$ (excluding 0 and 2π). Further, if N is large enough so that

$$2\pi/N\ll|\omega_r-\omega_m|\ll 2\pi-2\pi/N \qquad (14)$$

then all $\gamma(\omega_r-\omega_m)$ can be approximately regarded as zero. In practice, the frequencies of the tones are most probably arbitrary, thus it is reasonable to take them to be uniformly distributed. Therefore, for large N we can assume that these frequencies satisfy the condition (14).

When $\gamma(\omega_r-\omega_m)=0$, $m\neq r$, (12) becomes

$$B_r=\frac{J_r+\lambda^*/M}{S+\sigma_n^2+NJ_r} \qquad (15)$$

Using (8)-(10), and (15), we can express λ as

$$\lambda=\frac{M}{N}(S+\sigma_n^2) \qquad (16)$$

From (4), we obtain

$$\varepsilon_{LC,min}=(1+\frac{M}{N})(S+\sigma_n^2) \qquad (17)$$

From (10), (15), and (16), the optimal tap weights are given by

$$W_{LC,opt}(l)=\frac{1}{N}\sum_{r=1}^{N}e^{j\omega_r l} \qquad l=1,2,\ldots,N \qquad (18)$$

Performance evaluation of the LC-LMSE filter is taken as follows:

(a) The frequency response of the LC-LMSE filter

can be written as

$$H(e^{j\omega}) = 1 - \sum_{l=1}^{N} W_{l,LC,opt} e^{-j\omega l}$$

$$= 1 - \frac{1}{N}\sum_{m=1}^{M}\sum_{l=1}^{N} e^{j(\omega_m - \omega)l} \qquad (19)$$

For $\omega = \omega_m$ $(m=1,2,\ldots,M)$,

$$H(e^{j\omega_m}) = \sum_{\substack{r=1 \\ r\neq m}}^{M} \gamma(\omega_r - \omega_m) \qquad (20)$$

According to the properties of $\gamma(\omega)$, for

$$|\omega_r - \omega_m| = \frac{2\pi}{N}k,\ k = 1,2,\cdots,N-1;\ \forall m \neq r,\ m,\ r = 1,2,\cdots,M \qquad (21)$$

we have $H(e^{j\omega_m}) = 0$ $\quad m=1,2,\ldots,M$ (22)

This indicates that the MTI has been completely rejected. In general, if (14) is met, then the MTI is thought to be suppressed. The typical curves of frequency response about the LC-LMSE and the LMSE filter are shown in Fig. 2.

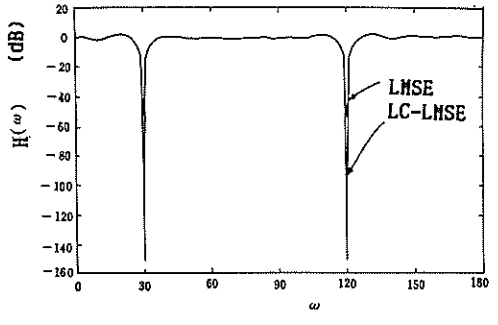


Fig. 2. The frequence response of LC-LMSE and LMSE filters. (N=20, M=2). For LMSE: S=1,σ_n^2=0.1, J=20 (m=1,2),ω_m=30°, 120°.

(b) The SNR at the input and output of the LC-LMSE filter are given by

$$SNR\,i = \frac{S}{J+\sigma_n^2} \qquad (23)$$

and

$$SNR^0 = \frac{S}{(1+\frac{M}{N})\sigma_n^2 + \frac{M}{N}S} \qquad (24)$$

Hence the SNR improvement factor (a criterion-of-goodness[1]) can be written as

$$\eta_{LC-LMSE} = \frac{J+\sigma_n^2}{(1+\frac{M}{N})\sigma_n^2 + \frac{M}{N}S} \qquad (25)$$

Fig. 3 shows the typical curves of the SNR improvement factor given by (25).

3. DISCUSSIONS

(a) We note from (24) that, theoretically speaking, with the LC-LMSE filter the MTI has been completely rejected as there is no MTI component left at the output of the filter. In practice, however, the MTI cannot be rejected perfectly due to the use of approximate condition $\gamma(\omega)=0$. From (20) or Fig. 2, it is easy to see that the depth of the LC-LMSE notch filter is only affected by the filter length and the number of tones. With the same N and M, the notches provided by the LC-LMSE filter are significantly improved compared with the LMSE filter.

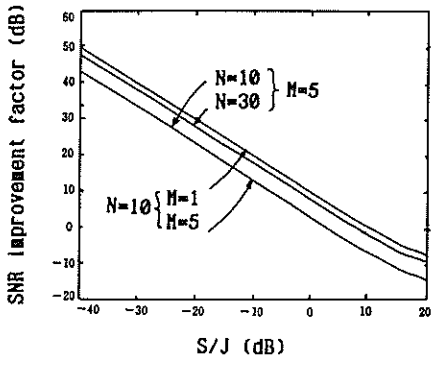


Fig. 3. The SNR improvement factor of LC-LMSE filter versus S/J. (S=1,σ_n^2=0.01).

(b) From Fig. 3, it follows that the SNR improvement factor will somewhat decrease with the number of MTI increasing; It will increase as the filter length increases and/or the MTI becomes stronger.

(c) From (25), we obtain

$$\lim_{N\to\infty} \eta_{LC-LMSE} = 1 + \frac{J}{\sigma_n^2} \qquad (26)$$

This means that when the length of the filter is large enough, $\eta_{LC-LMSE}$ is never less than 0dB, and it will increase when the power of AWGN decreases, as is shown in Fig. 4.

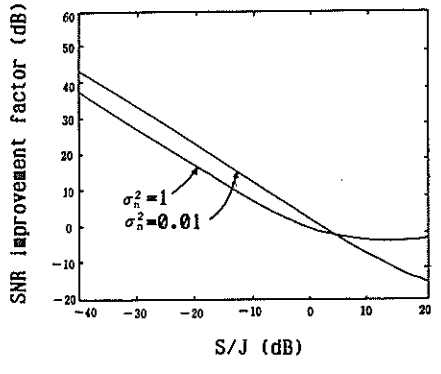


Fig. 4. The SNR improvement factor of LC-LMSE filter versus S/J. (N=10, M=5, S=1).

(d) For comparing purpose, we rewrite the SNR improvement factor of the LMSE filter[4] as

$$\eta_{LMSE}=\frac{\sigma_n^2+J}{\sigma_n^2+(S+\sigma_n^2)\sum_{m=1}^{M}\frac{J_m}{NJ_m+S+\sigma_n^2}} \tag{27}$$

It has already been proved that for constant total power of the MTI and fixed number of tones, equal power of each tone is the worst-case jamming condition. In this case, (27) can be reduced to

$$\eta_{LMSE}=\frac{\sigma_n^2+J}{\sigma_n^2+\frac{(S+\sigma_n^2)MJ}{NJ+(S+\sigma_n^2)M}} \tag{28}$$

Consider the ratio of $\eta_{LC-LMSE}$ to η_{LMSE}, we get the following expressions

$$\lim_{N\to\infty}\frac{\eta_{LC-LMSE}}{\eta_{LMSE}}=1 \tag{29}$$

and

$$\lim_{J\to\infty}\frac{\eta_{LC-LMSE}}{\eta_{LMSE}}=1 \tag{30}$$

It is shown in Fig. 5 that the SNR improvement factor of the LC-LMSE filter can be compared to that of the LMSE filter, provided that the length of both filters and/or the power of the MTI are large enough.

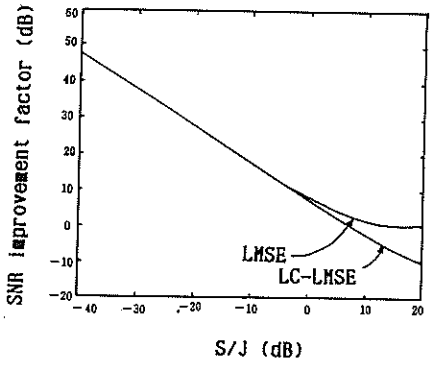


Fig. 5. Comparison of the SNR improvement factor between the LC-LMSE and the LMSE filters. (N=10, M=5, σ_n^2=0.01, S=1).

(e) From (20), we can see that the frequency response of the LC-LMSE filter is independant of the power of the input. Therefore, the LC-LMSE filter is especially suitable to the case where the received signal varies intensely in amplitude.

(f) When the number of the MTI is equal to one, i.e., the NBI case, the results obtained here are in accordance with those in [6]. However, it is unnecessary for the length of the LC-LMSE filter to satisfy (14) or (20) in order to obtain (21).

4. CONCLUSION

In this paper, a linearly constrained LMSE filter is proposed for the rejection of multi-tone interference in PN spread spectrum systems and its performance is analyzed. It is demonstrated that nearly perfect rejection of the MTI can be achieved if the constraints are chosen properly. With the same filter length, the notches set by the LC-LMSE filter are significantly deeper than by the LMSE filter. It is also shown that the SNR improvement factor of the LC-LMSE filter is comparable to that of the LMSE filter without constraints when the power of the MTI and/or the filter length are large enough. In addition, the LC-LMSE filter is suitable to the case where the received signal varies intensely.

ACKNOWLEDGEMENT

The authors would like to thank their many friends for helping them prepare the final manuscript.

REFERENCES

[1] L.B.Milstein,"Interference rejection techniques in spread spectrum communications," Proc. IEEE, vol.76, pp.657-671, June 1988.

[2] Z.Li,"Spread spectrum anti-jamming multiple access (SS-AJMA) communications: Concepts, techniques, and applications," Ph.D. dissertation, Southeast Univ., Nanjing, China, May 1988. (in Chinese)

[3] C.Zhong,"Adaptive interference rejection techniques in spread-spectrum systems: A unified theory and implementations with SAW devices," M.S. thesis, Southeast Univ., Nanjing, China, May 1989. (in Chinese)

[4] Z.Li et al.,"Rejection of multi-tone interference in PN spread spectrum systems using adaptive filters," in Proc. IEEE ICC, pp.874-878, Seattle, USA, June 1987.

[5] N.L.Oweley,"A recent trend in adaptive spatial processing for sensor arrays: Constrained adaptation," in Proc. of NATO Advanced Inst. on Signal Processing, pp.591-603, Loughborough, England, Aug. 1972.

[6] Z.Li et al.,"Rejection of narrowband interference in PN spread spectrum systems using linearly constrained LMSE filters," in Proc. IEEE PCCC, pp.161-164, Pheonix, USA, Feb. 1987.

[7] J.R.Zeidler et al.,"Adaptive enhancement of multiple sinusoids in uncorrelated noise," IEEE Trans. Acoust., Speech, Signal Processing, vol. ASSP-26, pp.240-254, June 1978.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A simple Doppler-corrector and metric processor for an MDPSK receiver using CORDIC elements

F. Kocsis*† and J.F. Böhme

Department of Electrical Engineering
Ruhr University, Bochum
D-4630 Bochum, West Germany

A new, simple digital signal processing circuitry is proposed for correcting the Doppler-phase and computing the branch metric values in MDPSK receivers. The solution is based on the CORDIC method that is suitable for simple ASIC-type realizations. The nonlinear, open loop estimation of the Doppler-phase, the smoothing (filtering) of phase noise, the open loop phase correction, and the branch metric computations can be implemented using only planar rotations that are the basic operations of a CORDIC processor. With respect to the circuit complexity, the element count can be decreased significantly by the solution proposed.

1. Introduction

A recent trend in communications is the growing interest in digital implementation of receivers based on digital signal processing (DSP). The advanced, bandwidth and power efficient modulation methods demand sophisticated signal processing algorithms. A possible way to implement the required DSP functions is the application of general purpose processing elements (e.g. DSP μP's). But these building blocks—being general purpose elements—contain a lot of redundancy. A possible alternative is the use of special purpose building blocks to capitalize on the special structure of the signal processing tasks to be solved. In many cases this way leads to some simplifications in the receiver structure and to savings in the number of necessary building elements or in the silicon area required in the case of a VLSI realization.

The proposed solution uses CORDIC processor for the implementation of the Doppler-corrector and the metric processor computing the initial metric values in a Viterbi-processor based MDPSK receiver The distortion due to the Doppler-phase can be significant e.g. in the case of receivers in moving vehicles. Using ASIC-type CORDIC element a very simple architecture can be derived. The Doppler-correction and the metric computation involve the estimation of the Doppler-phase, a phase rotation for correction and the computation of the initial metric values for the decision processor. Using open loop estimation and correction strategy in the correlative receiver, the necessary operations are the evaluation of the arctan() function and expressions as $(x \sin\alpha + y \cos\alpha)$, and multiplications with predetermined constants in the [0,1] interval. The latter operation can also be interpreted as multiplication with cos() and sin() values. The CORDIC instruction set is very suitable for this task. The method can be even used in the case of greater M-values.

In **Section 2** the necessary signal processing steps and algorithms are briefly discussed. **Section 3** summarizes the CORDIC method. **Section 4** covers the implementational issues of the CORDIC-based receiver.

2. Estimation and correction of the Doppler-phase, and computation of the initial metric

The input signal at the receiver is ([1]):

$$r(t) = A \sin\{2\pi(f_0 + \Delta f)t + \phi(t)\} + n(t) \qquad (1)$$

with

- $A \sin\{2\pi f_0 t + \phi(t)\}$ is the transmitted signal,
- $n(t)$ is a Gaussian noise process,
- A characterizes the signal power,
- f_0 denotes the carrier-frequency,
- Δf is the Doppler-shift,
- $\phi(t)$ is the information carrying $MDPSK$-phase with symbol rate of $1/T$.

In certain cases the Doppler-effect deterioriates the data transmission, and coherent demodulation is essential to achieve satisfactory data detection. Part of the signal processing tasks is the estimation and correction of the Doppler-phase in each symbol period before computing the initial metric. The known methods—as a first

*On leave from Research Institute for Telecommunications, Budapest, Hungary

†This research was supported by the Alexander von Humboldt Foundation, West Germany

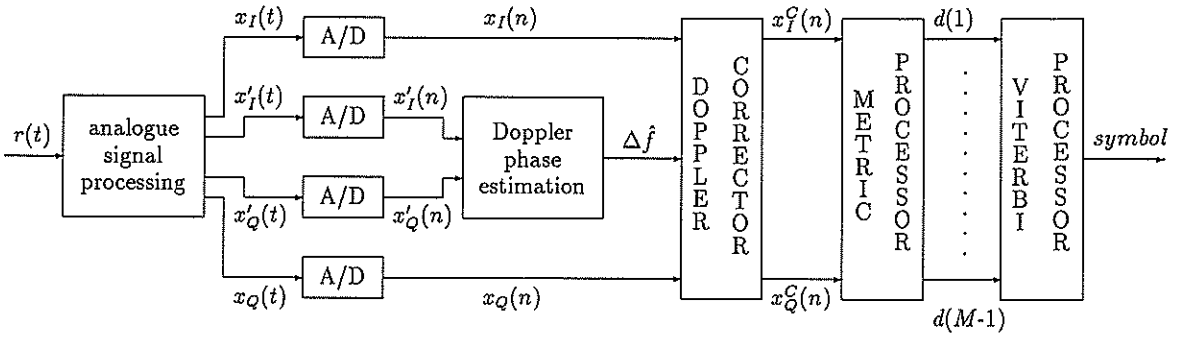


Figure 1. MDPSK receiver

approach—can be classified into two classes. The conventional closed loop estimation schemes ([2]) use tracking loops, mostly phase locked loops (PLL's) which involve relatively complex, partly analog circuitry as VCO to estimate and correct the deviations in carrier frequency. The other group of solutions are the open loop methods and use mainly nonlinear techniques ([3],[4],[5]) for the direct estimation and correction of the phase error. The main advantage of open loop estimators is the absence of unstable equilibrium points and therefore the absence of hang-up problems. The receiver (Fig.1.) uses analog methods as the usual multiplicative demodulator [2]) for the derivation of four baseband auxiliary signals $(x_I(t), x'_I(t), x_Q(t)$, and $x'_Q(t))$ from the received signal. With

$$\varphi_1 = 2\pi(f_0 + \Delta f)T/2 + \phi(t) - \phi(t - T/2), \quad (2)$$
$$\varphi_2 = 2\pi\Delta fT + \phi(t) - \phi(t - T/2), \quad (3)$$

we have

$$x_I(t) = K\cos\{\varphi_1\} + N_I(t), \quad (4)$$
$$x'_I(t) = K\cos\{\varphi_2\} + N'_I(t), \quad (5)$$
$$x_Q(t) = K\sin\{\varphi_1\} + N_Q(t), \quad (6)$$
$$x'_Q(t) = K\sin\{\varphi_2\} + N'_Q(t), \quad (7)$$

where $N_I(t), N'_I(t), N_Q(t)$, and $N'_Q(t)$ are noise components originating from $n(t)$, and K is proportional to the signal energy. According to the properties of the $MDPSK$ modulation assuming rectangular baseband pulses, and sampling at the time instants $(n + 3/4)T$ and $(n + 1/2)T$, the phases are

$$\phi\{(n + 3/4)T\} - \phi\{(n + 3/4)T - T/2\} = 0,$$
$$\phi\{(n + 1/2)T\} - \phi\{(n + 1/2)T - T/2\} = \phi(nT),$$

respectively. We remark that applying a more realistic baseband pulse, namely in the case of 100% excess bandwidth root raised cosine type-pulse forming ([2]), the optimal sampling time instants remain the same. $\phi(nT)$ is the nth data symbol phase before encoding. $\{x'_I(n)\}$ and $\{x'_Q(n)\}$ can serve for Doppler-phase estimation, and $\{x_I(n), x_Q(n)\}$ for data symbol estimation. After digitally filtering the $\{x'_I(n)\}$ and $\{x'_Q(n)\}$ sequences to reduce the phase noise effects, the Doppler-phase can be easily estimated. Choosing a simple first-order recursive filter with coefficient a in the interval $(0 \le a \le 1)$ the Doppler-phase estimate ([3]) is

$$\pi\Delta\hat{f}T = \arctan\left\{\frac{\sqrt{1-a^2}\sum_{k=0}^{\infty} a^k x'_I(n-k)}{\sum_{k=0}^{\infty} a^k x'_Q(n-k)}\right\}. \quad (8)$$

By simple algebraic manipulations, it can be shown that the estimate is unbiased. In general and for M-ary modulation, the estimate has an M-fold ambiguity because it is derived from the modulated carrier. But in the case of differential encoding it is unambiguous. The estimate is used for the phase correction,

$$x^C_I(n) = x_I(n)\cos 2\pi\Delta\hat{f}T + x_Q(n)\sin 2\pi\Delta\hat{f}T \quad (9)$$
$$= K\cos\left\{2\pi(\Delta f - \Delta\hat{f})T + \phi(n)\right\} + N^C_I(n), \quad (10)$$
$$x^C_Q(n) = -x_I(n)\sin 2\pi\Delta\hat{f}T + x_Q(n)\cos 2\pi\Delta\hat{f}T \quad (11)$$
$$= K\sin\left\{2\pi(\Delta f - \Delta\hat{f})T + \phi(n)\right\} + N^C_Q(n), \quad (12)$$

where $\{N^C_I(n)\}$ and $\{N^C_Q(n)\}$ are noise components. Finally the initial decision metric $\{d(i)|0 \le i \le M-1\}$ can be computed multiplying the quadrature-components $\{x^C_I(n)\}$ and $\{x^C_Q(n)\}$ with all possible data symbol phases ([2]) in every time instant n:

$$d(i) = x^C_I(n)\cos\frac{2\pi i}{M} + x^C_Q(n)\sin\frac{2\pi i}{M}, \quad (13)$$

$$= K\cos\left\{2\pi(\Delta f - \Delta\hat{f})T + \phi(n) - \frac{2\pi i}{M}\right\} + N^i_d(n), \quad (14)$$

where $\{N^i_d(n)\}$ is a noise sequence. In uncoded case the decision rule is very simple ("select the greatest $d(i)$ value"), while in precoded case a more sophisticated decision strategy is necessary (e.g. Viterbi-processor). The detailed structure of the Doppler-corrector and the metric processor can be seen on the Fig.2.

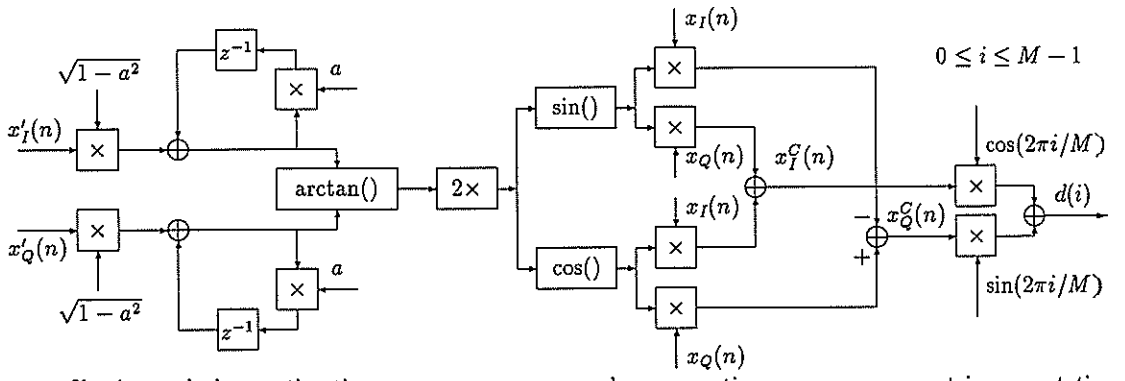


Figure 2. The Doppler-corrector and the metric processor

3. The CORDIC method

As far as the realization of the necessary computation is concerned, the basic operation is the *planar rotation* of a vector (x,y) by an angle α, defined by the following expressions,

$$x' = x\cos\alpha + y\sin\alpha, \quad y' = -x\sin\alpha + y\cos\alpha. \quad (15)$$

It can be implemented by the CORDIC method which is an iterative algorithm that can be used to calculate many well-behaved functions, for example magnitude and phase, or a rotated version of a vector in different coordinate systems by only a sequence of shift-add operations. The basic iterative equations are, in our case, the following,

$$x_{i+1} = x_i + \delta_i 2^{-i}, \qquad y_{i+1} = -\delta_i 2^{-i} x_i + y_i, \quad (16)$$

$$z_{i+1} = z_i + \delta_i \alpha_i, \qquad \alpha_i = \arctan 2^{-i}, \quad (17)$$

where δ_i=$\pm$1 and is chosen so that either the variable y or the variable z is forced to 0 after a sufficient number of iterations has been performed. For the details we refer to [6]. The CORDIC processors are regular and modular in structure, and are very suitable for ASIC-type realization. To satisfy the relatively modest speed requirements of our applications, a reduced complexity recursive or bit-serial architecture can be applied. A bit-serial solution consists of only 3×1-bit full adders, 2×1-bit half adders, 3 shift registers, some gates, an angle ROM and some control logic. In the following we denote a CORDIC processor symbolically as in Fig. 3.

4. CORDIC-based Doppler-corrector and metric processor

The task of the Doppler-corrector and the metric processor is the realization of the equations (10)-(16),(20). Comparing the form of the equations to the instruction set of a CORDIC processor, one can easily deduce the applicability of the CORDIC processors in the receiver

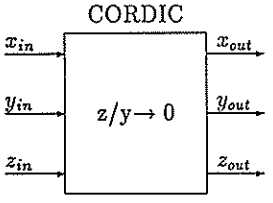


Figure 3. Trigonometric CORDIC processor

implementation. Additionally the difference-equation describing the noise smoothing,

$$x''_{I,Q}(n) = x'_{I,Q}\sin b + x''_{I,Q}(n-1)\cos b \quad (18)$$

where a=$\cos b$ (Fig. 2.), can be also realized by the CORDIC method. In the computation of the decision variables, one can exploit the symmetry properties of the trigonometric functions. Let M be a power of two (M=2^n, $n \ge 2$), $0 \le i \le M/4-1$, and k=0,1,2,3. Then,

$$d(i + \frac{kM}{4}) = \\ = x^C_I(n)\cos(\frac{2\pi i}{M} + \frac{k\pi}{2}) + x^C_Q(n)\sin(\frac{2\pi i}{M} + \frac{k\pi}{2}).$$

For k=0,1 the values of $d(i + kM/4)$ can be computed simultaneously by the two data outputs of a CORDIC processor, while for k=2,3 the required values are simply the values of $d(i)$ and $d(i + M/4)$ multiplied by -1. For i=0, $M/4$, $M/2$, $3M/4$ the metric values are trivial,

$$d(0), -d(M/2) = x^C_I(n) \quad d(M/4), -d(3M/4) = x^C_Q(n).$$

The number of necessary rotations (ROT) is ($M/4$+3) per sample period: 2 ROT's for the filtering, 1 ROT for the phase estimation, 1 ROT for the phase correction and ($M/4-1$) ROT's for the computation of the initial decision metric. The architecture of a receiver with CORDIC processors can be seen in Fig. 4. For

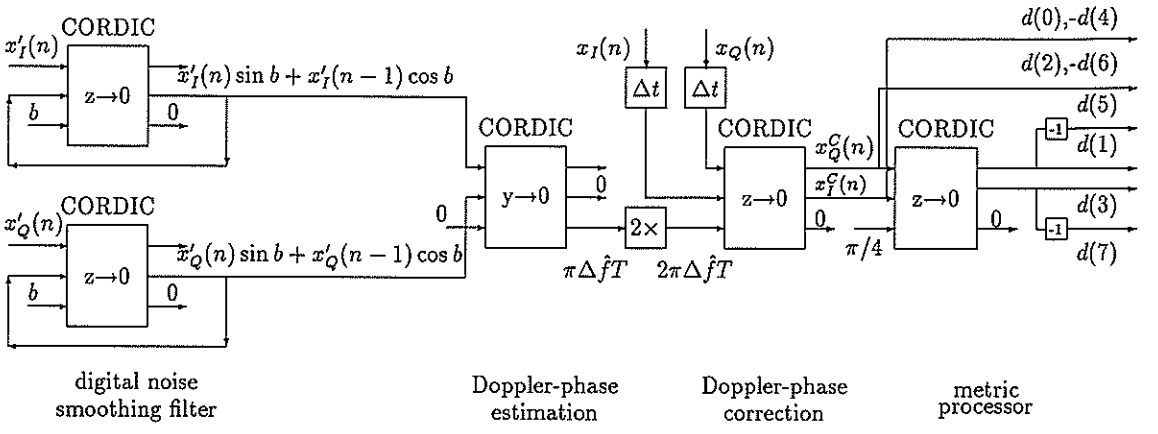


Figure 4. CORDIC-based Doppler-corrector and metric processor for M=8

M=8 the number of rotations is 5. At data symbol rate 2400(4800) symbol/s the symbol period is ~417 μs(~208,5 μs). Using only one simple recursive or bit-serial CORDIC processor, the maximal permissible execution time of one rotation is ~83,4 μs(~41,7 μs). For M=16 the corresponding values are ~59,57 μs(~29,8 μs). These rotation times can be realized using a simple CMOS ASIC or CORDIC gate-array IC. The circuit complexity of a recursive or bit-serial CORDIC processor is relatively low, so the necessary auxiliary circuits (storage registers for the samples, storage for the rotation angle values and the filter parameter b=arccos a, some timing logic, a shifter, multiplier with the value of -1) can also be integrated on a common chip with the CORDIC processor. The complexity of the CORDIC portion of the corrector-metric processor IC can be further decreased exploiting the fact that the necessary wordlength of the decision variables $\{d(i)\}$ is relatively low.

5. Conclusions

The Doppler-effect may seriously deterioriate the data transmission as in communication networks including mobile vehicles. The correction involves additional signal processing in the receiver, namely estimating of Doppler-phase and correcting with the estimated phase before deciding. In the case of a nonlinear open loop method the necessary operations are the computation of arctan() values and the evaluation of expressions as $(x\sin\alpha + y\cos\alpha)$, additions and multiplications with predetermined constants in the [0,1] range. All these computation fit well the instruction set of a CORDIC processor. The solution proposed in this paper uses CORDIC elements to implement also the metric processor computing the initial metric values for a decision device (e.g. Viterbi-processor).

REFERENCES

[1] M. SIMON and D. DIVSALAR, "Doppler-corrected differential detection of MPSK," *IEEE Transactions on Communications*, vol. COM-37., pp. 99–109, February 1989.

[2] J. PROAKIS, *Digital Communications.* McGraw-Hill, 1985.

[3] A. VITERBI and A. VITERBI, "Nonlinear estimation of PSK-modulated carrier phase with application to burst digital communication," *IEEE Transactions on Information Theory*, vol. IT-29., pp. 543–551, July 1983.

[4] B. PADEN, "A matched nonlinearity for phase estimation of a PSK-modulated carrier," *IEEE Transactions on Information Theory*, vol. IT-32., pp. 419–423, May 1986.

[5] P. KAM, "Maximum likelihood carrier phase recovery for linear suppressed-carrier digital data modulations," *IEEE Transactions on Communications*, vol. COM-34., pp. 522–527, June 1986.

[6] J. WALTHER, "A unified algorithm for elementary functions," in *Proceedings of SJCC*, pp. 379–385, 1971.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

UNCODED AND TRELLIS-CODED SIGNALS VIA THE DIGITAL RADIO RELAY CHANNEL DETECTED WITH DIFFERENT RECEIVER STRUCTURES

Eckard BOGENFELD and Werner RUPPRECHT

Institute for Communications Technology, University of Kaiserslautern, P.O. Box 3049
D-6750 Kaiserslautern, West Germany

Digital radio relay systems have to utilize the allocated frequency bands as fully as possible. For example multi-level modulation types satisfy this requirement. They are, however, more sensitive to disturbances and distortions. We compare uncoded and coded signals up to 256-QAM by using different receiver structures.

1. INTRODUCTION

In digital transmission systems, additive disturbances are superimposed to the useful signal. Provided the data rate is high, the signal additionally is subject to intersymbol interference (ISI). ISI is caused by the transmission channel, which spreads the received pulses. On radio links, such distortions have their main reason in multipath propagation, which in addition is time-variant.
In order to restore the transmitted symbol sequence from the disturbed and distored signal we distinguish two possibilities: first the classical method of equalization by an adaptive filter-type equalizer and a memoryless threshold detector; and second the method of detection, using a detector having memory. It is also possible to combine both methods.
The method of adaptive equalization is based on the minimization of a metric or error criterion [1,2]. In this paper for the radio channel [3], the criteria mean square error (MSE) and peak distortion (PEAK) are minimized by a fast iterative procedure and the results are compared.
The method of detection uses models of disturbance and of distortion to determine the transmitted symbols by a suboptimum sequential decoding algorithm [4]. Whereas the optimum Viterbi algorithm (VA) [5] takes all trellis-paths into consideration, a suboptimum algorithm selects only a subset of paths, which remain to be pursued. The results of our simulations illustrate the performance of the suboptimum M-algorithm (MA) [6].
The transmission quality of the radio channel by using trellis-coded modulation (TCM) technique [7] is investigated. A method to detect TCM signals in the presence of ISI and noise has been described in [8].

2. EQUIVALENT BASEBAND TRANSMISSION SYSTEM

For theoretical examinations and simulations it is expedient to consider the transmission of modulated data signals in the equivalent baseband. Inevitably, we have to use complex-valued time functions [1]. Fig.1 shows the block diagram of such transmission system, which consists of source, modulator, transmitter, channel, receiver and sink.

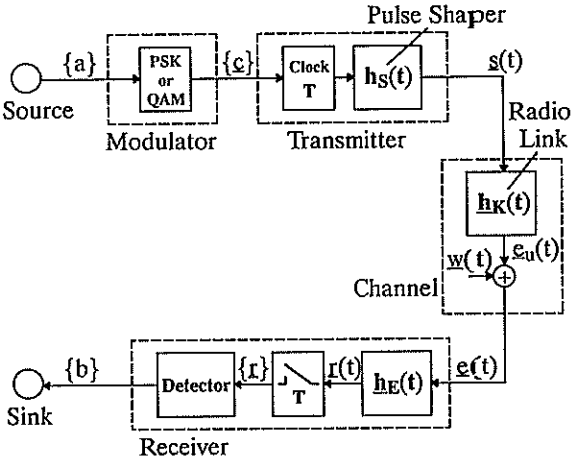


Fig.1 Equivalent baseband model

The source delivers a binary sequence ⟨a⟩ which is mapped to a complex-valued symbol sequence ⟨c⟩ in accordance with a chosen type of modulation, for example 64-QAM. Using ⟨c⟩ the pulse shaper generates the transmitted signal $\underline{s}(t)$ which is distorted and disturbed by the channel. The considered channel is the digital radio relay channel; we can simulate its characteristic even with the two-path model [3]. Its transfer function is given by

$$H_K(f) = a(1-b\cdot e^{-j2\pi(f-f_n)\tau}) \qquad (1)$$

where a is the amplitude of the direct ray, b is the relative amplitude of the delayed ray, f_n is the notch frequency and τ is the difference of time delay between the two rays. Fig.2 shows an example of a channel response (a=1,b=0.7,f_n= 7MHz,τ=100ns) to a 4-QAM symbol (roll-off 0.5).

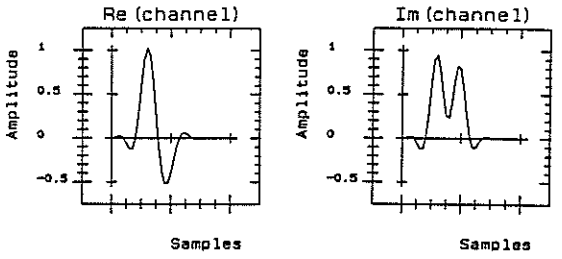


Fig.2 Channel response to a 4-QAM symbol

ISI is caused by the delayed ray. This kind of distortion can be eliminated by costly equalizers or it can be usefully considered in the detection having memory.

3. METHOD OF EQUALIZATION

Before attaining the memoryless threshold detector the output signal of the channel is equalized. The complex-valued system requires a cross-coupled equalizer structure, e.g., a transversal filter with complex coefficients [1]. As the radio channel causes time-variant distortions the equalizer should be adaptive.

3.1. Adaptive Equalization

The structure of the adaptive equalizer (Fig.3) consists of the adjustable filter, the signal evaluation and the adaptive algorithm. These blocks form an automatic digital control system. The adaptive algorithm [2] changes the set $\{\underline{q}\}$ in steps until the output signal $\underline{r}(t)$ is equalized optimally. In this case the metric F is minimum.

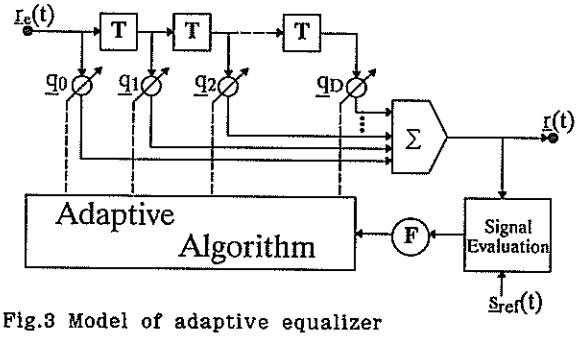


Fig.3 Model of adaptive equalizer

The definition of the metric F may contain or not contain a reference signal. An example for the second case is the PEAK criterion:

$$F_{PEAK} = \mathrm{Max}|\mathrm{Re}(\underline{r}(t))|^2 + \mathrm{Max}|\mathrm{Im}(\underline{r}(t))|^2. \quad (2)$$

The maximum of the distorted signal is larger than that of the undistorted signal having the same mean power (works also with one coefficient fixed). An example for a criterion using a reference signal is the mean square error (MSE):

$$F_{MSE} = E[\,|\underline{r}(t) - \underline{s}_{ref}(t)|^2\,] \quad (3)$$

where $E[\cdot]$ is the mean value.

3.2. The Eye Pattern as a Quality Criterion

The attainable eye aperture is very informative for the judgement of different error metrics. Fig.4 shows the eye pattern of the real components of the equalized 4- to 256-QAM signals by applying the PEAK criterion. We renounce to display the imaginary components because of their similarity. PEAK only equalizes 4- and 16-QAM signals sufficiently. Fig.5 shows the eye pattern by using MSE. In contrast to PEAK the MSE criterion yields equalization of high quality up to 256-QAM.

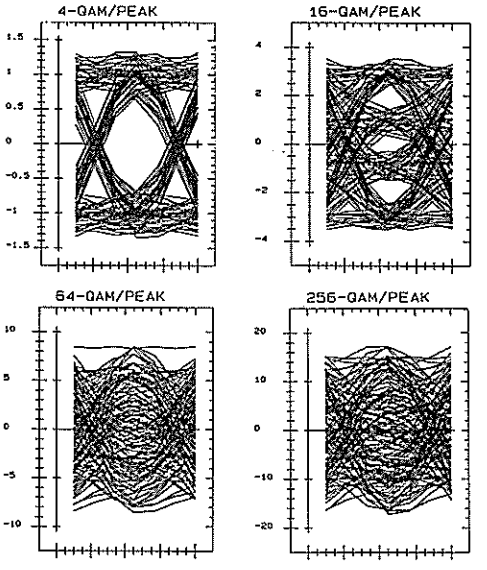


Fig.4 Eye pattern 4- to 256-QAM, PEAK

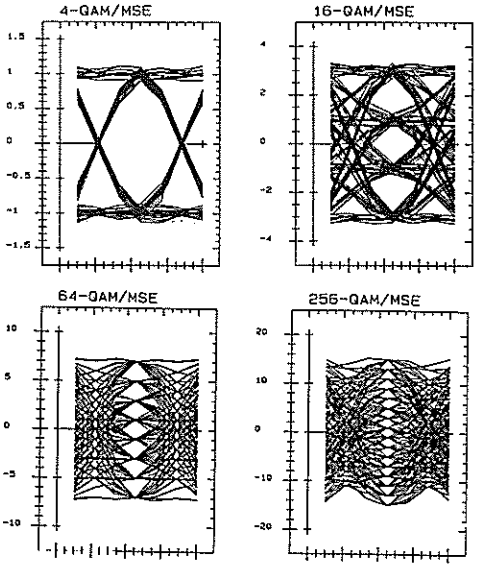


Fig.5 Eye pattern 4- to 256-QAM, MSE

4. METHOD OF DETECTION

For the method of detection the decision of the output signal is made by a detector having memory. It is assumed that all source symbols have equal a-priori probabilities. Consequently the maximum-likelihood (ML) detector is the optimum. To describe its principle the model in Fig.6 is used [6].

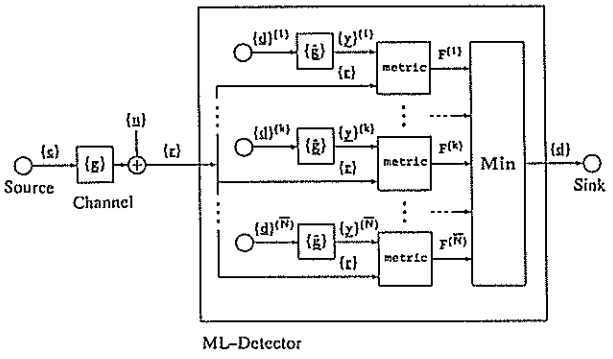


Fig.6 Time discrete model with ML-detector

The complex source symbol sequence $\langle \underline{c} \rangle \in \underline{A}$ has the length N. Its elements have the same probability and are statistically independent. Noise samples $\langle \underline{n} \rangle$ are uncorrelated and gaussian. The estimated pulse response $\langle \underline{\hat{g}} \rangle$ distinguishes from $\langle \underline{g} \rangle$ only in a constant delay time. The length of the pulse response is given by $\hat{\nu}$ respectively ν samples. The set $\underline{A}$ contains all possible symbol sequences which are considered by the detector:

$$\langle \underline{d} \rangle^{(\nu)}, \quad \nu = 1, \ldots, k, \ldots, \overline{N}, \quad \overline{N} = L^N. \qquad (4)$$

L is the number of different symbols, e.g., four in 4-QAM. The detector determines the sum of the partial error metrics $|\underline{r}_\mu - \underline{y}_\mu^{(k)}|^2$ of the received samples $\underline{r}_\mu$ and the possible undisturbed samples $\underline{y}_\mu^{(k)}$ over all sample points μ:

$$F^{(k)} = \sum_{\mu} |\underline{r}_\mu - \underline{y}_\mu^{(k)}|^2 \qquad (5)$$

The metric $F^{(k)}$ describes the cost of the symbol sequence $\langle \underline{d} \rangle^{(k)}$. The minimum of those costs

$$F^{(o)} = \min_{\nu} F^{(\nu)}, \quad \nu = 1, 2, \ldots, \overline{N} \qquad (6)$$

is the cost of the most likely transmitted sequence. An other manner to illustrate the principle of ML-detection uses the trellis diagram. A symbol sequence $\langle \underline{d} \rangle^{(k)}$ corresponds to a sequence of states in the trellis. Such a sequence is called a path with the metric $F^{(k)}$.
The Viterbi algorithm (VA) [5] represents an optimum procedure for the search of the smallest metric in the trellis. After the whole symbol sequence is transmitted and the last state is reached, the path with the lowest cost is known. The VA searches through all trellis states. Therefore the complexity increases with L^ν. With larger number of symbols in the signal set the cost of implementing the VA would be to high, even at short pulse responses as in the radio channel. In order to achieve less complexity suboptimum algorithms can be used. In doing so, degradation compared to the VA occurs. A survey of different suboptimum sequential decoding algorithms (SDA) is given in Fig.7. A detailed description can be found in [4].

	Depth-First	Metric-First	Breadth-First
Non-sorting	Single-Stack-A. Fano-A.		Metric-Bound-A.
	↑ Two-Cycle-A. ↑		
Sorting		Stack-A. Bucket-A. Merge-A.	*M-Algorithm*
	↑ Multiple-Stack-A. ↑		↑ Generalized-Stack-A. ↑

Fig.7 Survey of SDA's

The M-algorithm (MA) [6] is suited for the radio channel. Compared to the VA the MA examines in each time step only the $L \cdot M < L^\nu$ best paths. At the detector an equidistant symbol output is possible.

5. TRELLIS-CODED MODULATION

Trellis-coded modulation (TCM) was introduced by Ungerboeck. He demonstrated that compared to standard QAM, TCM offers a coding gain without reducing data rate or requiring more bandwidth. The modulator in Fig.1 is replaced by the TCM encoder (Fig.8).

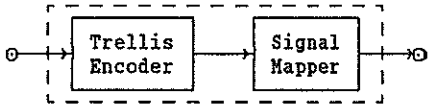


Fig.8 TCM encoder

It is investigated a simple 16-state convolutional encoder as trellis encoder [7]. For detection we use a receiver structure [8], which applies the VA and assumes that the pulse response of the channel is known.

6. SIMULATION RESULTS AND CONCLUSIONS

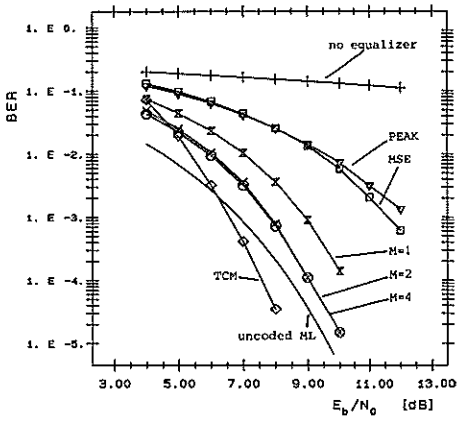


Fig.9 Uncoded 4-QAM / Coded 8-PSK

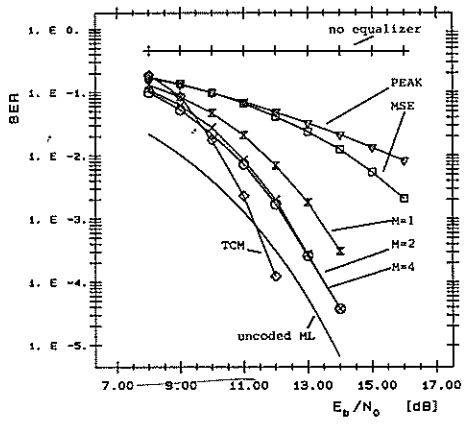


Fig.10 Uncoded 16-QAM / Coded 32-Cross

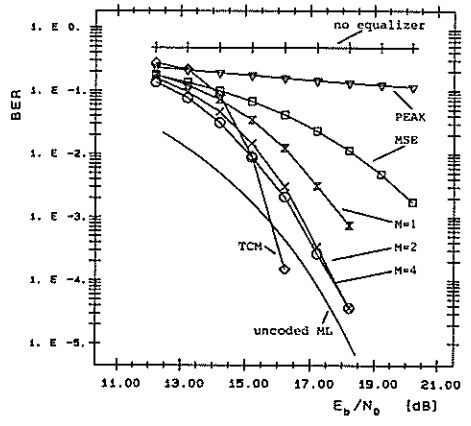


Fig.11 Uncoded 64-QAM / Coded 128-CROSS

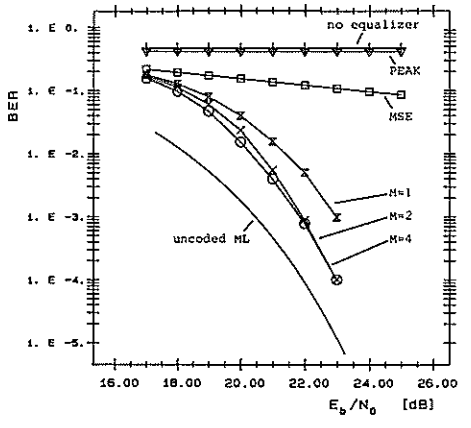


Fig.12 Uncoded 256-QAM

Figs.9-12 show the bit error rate for 4- to 256-QAM for different receiver structures. In case of uncoded modulation the curves are bounded by the memoryless threshold detector (no equalizer) and by the optimum ML-detector.

The adaptive equalization eliminates ISI extensively by using MSE (Fig.5). The adjusted filter, however, reduces the signal-to-noise-ratio (SNR here: E_b/N_0) at the threshold detector input. This is the reason for the losses compared to the optimum detection. PEAK is less suited for 16-QAM; it is not suited for 64- and 256-QAM.

For the method of detection we show the simulation results using the MA. With increasing M the error rate decreases and the typical satuaration effect arises. An increase $M > 2$ only yields low extra gains. Already for $M = 4$ and increasing SNR the error rate approximates the uncoded ML-detector for all types of modulation.

With TCM additional gains can be achieved. Figs. 9-11 show the error rates for coded 8-PSK, 32-CROSS and 128-CROSS. As reference for uncoded modulation the curve $M = 1$ can be used. The coding gain is up to 3 dB for the investigated radio channel and 16-state TCM.

REFERENCES

[1] Rupprecht, W., *Orthogonalfilter und adaptive Datensignalentzerrung*, München: Oldenburg, 1987

[2] Bogenfeld, E. and Rupprecht, W., "Untersuchung verschiedener Empfängerstrukturen zur Entscheidung von Funkkanal-Digitalsignalen," *will be puplished in Kleinheubacher Berichte*, vol. 33, 1990

[3] Rummler, W.D., "A new selective fading model: application to propagation data," *Bell Syst. Tech. Jour.*, vol. 58, pp. 1037-1071, 1979

[4] Bauer, D. and Rupprecht, W., "Sequential decoding algorithms for detection of severely distorted data signals," *Proc. of Eurocon-88*, pp. 114-117, Stockholm, June 1988

[5] Forney, G.D., "Maximum-likelihood sequence estimation of digital sequences in the presence of intersymbol interference," *IEEE Trans. Inform. Theory*, vol. IT-18, pp. 363-378, 1972

[6] Sauer, W. and Rupprecht, W., "A suboptimum maximum-likelihood detector for severely distorted data signals using a breadth-first strategy," *Proc. of ISCAS-88*, pp. 127-130, Helsinki, June 1988

[7] Ungerboeck, G., "Channel coding with multilevel/phase signal sets," *IEEE Trans. Inform. Theory*, vol. IT-28, pp. 55-67, Jan. 1982

[8] Wesolowski, K., "Efficient digital receiver structure for trellis-coded signals through channels with intersymbol interference," *Electron. Lett.*, pp. 1265-1267, Nov. 1987

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

CREATING OF DISCRETE POWER SPECTRA FOR FSK

Kittel, L., FernUniversität Hagen, FRG

Slominski, M., Warsaw University of Technology, Poland

Wysocki jr., T., Institute of Telecomm. ATR Bydgoszcz, Poland

Conditions are formulated which could allow the FSK signal to have in its power sprectrum a nonzero discrete component which can be utilized for synchronization. The methods investigated are preencoding and preencoding with unbalancing, respectively. Examples are given for MSK where the preencoding without and with unbalancing is performed by two different 3B4B line codes.

1. INTRODUCTION

In designing of a communication system, one of the problems is the choice of a modulation technique. In selecting the optimum technique, some key conditions must be satisfied as are: proper spectral characteristics, immunity from noise, and synchronization between transmitter and receiver.

For radio communications, narrow band angular modulations are often used. The spectral properties for the most of them are usually discussed under the assumption that the modulating signal consists of statistically independent data.

To satisfy the spectral requirements for a modulated signal, two methods are generally utilized:

(i) pulse shaping of the modulating signal, e.g. SFSK, GMSK [1],
(ii) preencoding combined with modulation [2], [5].

The objective of this paper is to formulate conditions for a preencoding which could allow the FSK signal to have a nonzero discrete component in its power spectrum.

2. SPECTRAL ANALYSIS

2.1. A model of the FSK modulated signal

Generally, the FSK modulated signal is an isochronous pulse stream. Thus, the FSK modulator can be treated as a conventional Moore's machine $M =< E, S, G, \delta, \lambda >$, where E is a set of data words, S is a set of states, G is a set of modulated words (signal pulse sequences), $\delta : E \times S \rightarrow S$ is a transition function, and $\lambda : S \rightarrow G$ is an output function.

On the basis of this model, it is easy to find a Markov chain specified by the triple

$$(S, P_{SS}, P_S), \tag{1}$$

where S is the set of states corresponding to the states of the modulator M, P_{SS} is the transition probability matrix, and $P_S = [p_1, ..., p_L]$ is the vector of stationary state probabilities. With regard to the model M, the FSK modulated signal, denoted by $x(t)$, is given by

$$x(t) = \sum_n g^{(n)}(t - nT_w), \tag{2}$$

where T_w is a word time slot and $g^{(n)} = \lambda[s^{(n)}]$.

2.2. Method of analysis

Since the FSK signal is described by the Markov model (1), the modulated signal $x(t)$ can be expressed by

$$x(t) = \sum_n \overline{g}(t - nT_w) + \sum_n h^{(n)}(t - nT_w), \tag{3}$$

where

$$\overline{g}(t) = E\{g(t)\} = \sum_{k=1}^{L} p_k \lambda(s_k),$$

and

$$h_k(t) = \lambda(s_k) - \overline{g}(t).$$

The first sum of (3) describes the cyclical signal which is responsible for the appearence of a discrete component in the power spectrum of the modulated signal according to the formula [3] :

$$S_d = \frac{2\pi}{T_w^2} |\overline{G}(\omega)|^2 \sum_r \delta(\omega - \frac{2\pi r}{T_w}). \tag{4}$$

The second sum of (3) corresponds to the continuous power spectrum of the modulated signal which is described by [3] :

$$S_c(\omega) = \frac{1}{T_w}[P_h H^* + C(\omega)], \quad (5)$$

where

$$C(\omega) = 2Re[\sum_{k=1}^{\infty} P_h (P_{SS})^k H^* exp(j\omega k T_w)].$$

The expressions in (5) have the following meaning:

$\overline{G}(\omega)$ is the Fourier transform of $\overline{g}(t)$,
$\delta(\cdot)$ is the Dirac delta function,
$P_h = [p_1 H_1(\omega), ..., p_L H_L(\omega)]$,
$H = [H_1(\omega), ..., H_L(\omega)]$,
$H_i(\omega)$ is the Fourier transform of $h_i(t)$, and
H^* is the vector conjugated with H.

2.3.Application to a binary FSK

For a binary FSK with angular frequencies ω_0 and ω_1, the elements of the set B of the modulated signal pulses are given by

$$\begin{aligned} b_0^i(t - iT) &= A \cdot h(t - iT) cos(\omega_0(t - iT) + \varphi_0^i), \\ b_1^i(t - iT) &= A \cdot h(t - iT) cos(\omega_1(t - iT) + \varphi_1^i), \end{aligned} \quad (6)$$

where

$$h(t) = \begin{cases} 1 & \text{for } 0 \le t < T, \\ 0 & \text{otherwise.} \end{cases}$$

For the rational values of a modulation index, $m = (\omega_1 - \omega_0)T/2\pi$, the set B is a finite one. If a modulating signal consists of statistically independet pulses, the initial values of phase angle φ_0^i and φ_1^i have a uniform distribution on the interval $< 0, 2\pi >$. Therefore,

$$\begin{aligned} \sum_{\varphi_0^i} p_i cos(\varphi_0^i) &= 0, \\ \sum_{\varphi_1^i} p_i cos(\varphi_1^i) &= 0, \end{aligned} \quad (7)$$

where p_i is the probability of a pulse with initial phase φ_0^i or φ_1^i appearence in the modulated signal, respectively. Thus, independent of the probabilities p and q of mark and space in the modulating signal, the average pulse of a modulated signal is equal to zero for all values of t. This indicates the absence of a discrete component in the power spectrum.

3. PREENCODED FSK

3.1 Condition for discrete power spectrum

As an example of preencoding of modulating data, a block code is considered. For codewords of lengh M, an average $\overline{g}(t)$ of modulated words (of the same lenght M) is given by the expression:

$$\begin{aligned} \overline{g}(t) &= \sum_{i=1}^{L} p_i \lambda(s_i) = \sum_{i=1}^{L} p_i g_i(t), \\ &= \sum_{i=1}^{L} p_i \sum_{j=1}^{M} b_{i_j}[t - (j-1)T], \\ &= \sum_{j=1}^{M} \sum_{i=1}^{L} p_i b_{i_j}[t - (j-1)T], \end{aligned} \quad (8)$$

where L is the power of modulated words set, and T is a single pulse time slot.

From (5) it follows that

$$\begin{aligned} b_{i_j}(t) &\equiv 0, \quad \text{for} \quad t \le 0 \quad \text{or} \quad t > T, \\ b_{i_j}(t) &\not\equiv 0, \quad \text{for} \quad 0 < t \le T. \end{aligned}$$

Thus, the average value of modulated words is identically equal to zero. The absence of a discrete component in power spectrum occurs if and only if, for each $j = 1, ..., M$,

$$\sum_{i=1}^{L} p_i b_{i_j}[t - (j-1)T] \equiv 0. \quad (9)$$

A nonzero discrete power spectrum component appears only in the case when, for any $j = 1, ..., M$, the condition (9) is not satisfied.

It follows that such $j = j^*$ exists for which

$$\sum_{\varphi_0^*} p_i cos(\varphi_0^*) \neq 0, \text{ or } \sum_{\varphi_1^*} p_i cos(\varphi_1^*) \neq 0, \quad (10)$$

where φ_0^* and φ_1^* are the initial phases of the pulses $b_{i_{j^*}}(t)$ corresponding to "0" and "1", respectively.

To satisfy the condition (10), line codes or error protection codes can be used as a preencoding scheme.

3.2 MSK combined with binary line encoding

To give an example of preencoding schemes for MSK, two different 3B4B codes will be analyzed. Spectral properties of the modulated signals will be discussed unter the following assumptions:

(i) the signal to be encoded consists of random equiprobable binary pulses,
(ii) the initial value of a phase angle is equal to zero for the first modulated pulse.

(a) MSK combined with 3B4B [4].

The encoding rule of the 3B4B code is given in Tab. 1. The distribution of phases φ_0^i and φ_1^i of the modulated signal is presented for this code in Tab. 2.

000	0101	
001	1001	
110	0110	
111	1010	
010	1110	0001
011	1101	0010
100	0111	1000
101	1011	0100

Table 1: The 3B4B [4] mapping

	Codeword phase								Total	
	1		2		3		4			
φ	0	1	0	1	0	1	0	1	0	1
0	3	5	0	0	3	5	0	0	6	10
0.5π	0	0	3	3	0	0	3	3	6	6
π	5	3	0	0	5	3	0	0	10	6
1.5π	0	0	5	5	0	0	5	5	10	10

Table 2 : Distribution of modulated pulses initial phases for MSK combined with 3B4B [4]

It can be easily verified here by inspection of Tab.2 that condition (10) is satisfied. Therefore, a nonzero discrete component in power spectrum is expected. Results of computations, carried out on the formulae (3) and (4), are plotted in Fig. 1.

(b) MSK combined with 3B4B [6]

The encoding rule of the 3B4B code is given in Tab.3. For this code, the distribution of phases φ_0^i and φ_1^i of the modulated signal is presented in Tab. 4.

By inspection of Tab.4, it can be easily verified that the condition (10) is not satisfied here. Thus, in contrast with the technique (a), a nonzero discrete component does not exist in the power spectrum of the modulated signal. Results of power spectra computations are presented in Fig. 2.

000	0001	0111
001	0011	
010	0101	
011	0110	
100	1001	
101	1010	
110	1100	
111	1000	1110

Table 3: The 3B4B [6] mapping

	Codeword phase								Total	
	1		2		3		4			
φ	0	1	0	1	0	1	0	1	0	1
0	4	4	0	0	4	4	0	0	8	8
0.5π	0	0	4	4	0	0	4	4	8	8
π	4	4	0	0	4	4	0	0	8	8
1.5π	0	0	4	4	0	0	4	4	8	8

Table 4: Distribution of modulated pulses initial phases for MSK combined with 3B4B [6]

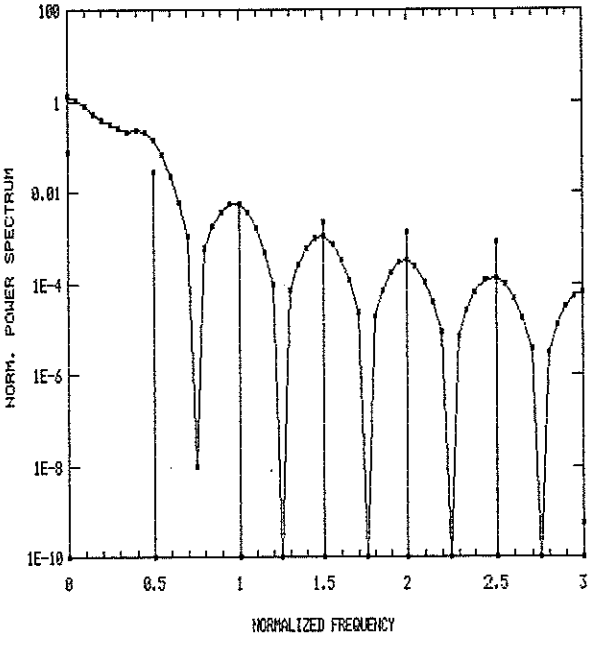


Figure 1: Presence of a nonzeore discrete component in power spectrum of MSK combined with 3B4B [4].

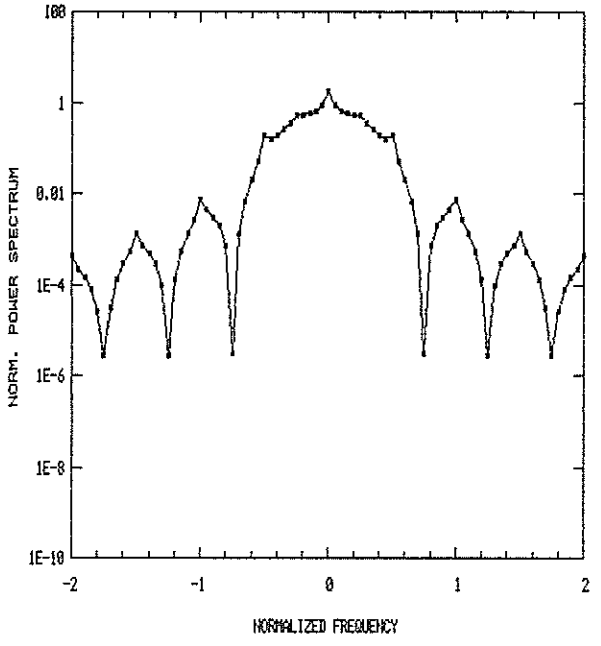


Figure 2: Absence of a nonzero discrete component in power spectrum of MSK combined with 3B4B [6].

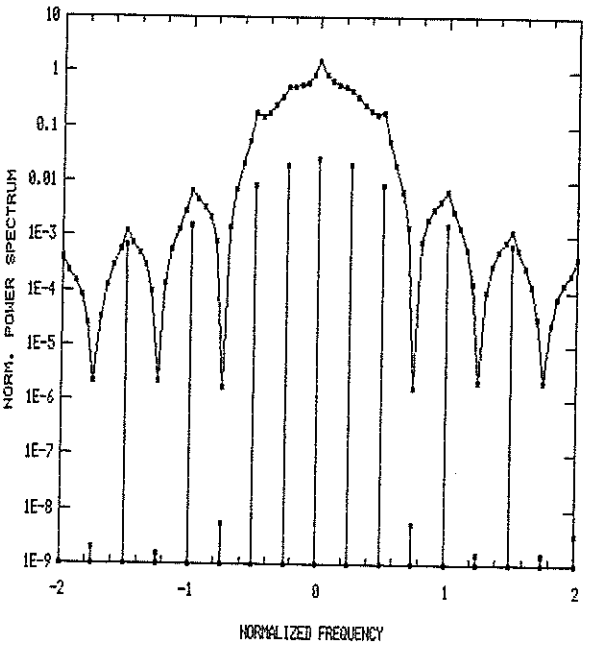


Figure 3: Presence of a nonzero discrete component in power spectrum of MSK combined with 3B4B [6]; unbalanced modulating signal. ($q = 0.4; p = 0.6$)

Continuing this example, it can be interesting to discuss spectral properties of the modulated signal for a modulating signal being unbalanced, i.e. when the probability of mark differs from the probability of space. For this case, the spectral analysis was done. The results (see Fig.3) confirmed that the unbalanced data stream has caused a nonzero discrete component in the power spectrum of the modulated signal.

4. CONCLUSION

In the paper, a method of creating discrete power spectra for FSK signals is presented. It is based on changing the initial values of modulated pulse phase angle distribution (uniformity violation). To achieve such distributions, preencoding of the modulating data was proposed. Another method discussed was preencoding of modulating data in combination with unbalancing.

The presence of a nonzero discrete component can be utilized for extracting a carrier frequency at the receiver. The method of creating discrete power spectra for FSK presented in this paper can be applied to a synchronous communication system, for which the untreated modulated signal - without preencoding - has the zero discrete power spectrum component.

References

[1] Miki T., Hota M.: Performance of 16 kbit/s GMSK transmission with postdetection selection diversity in land mobile radio, IEEE Journ. on Selected Areas in Commun. vol.SAC-2, No.4, pp.512-517, 1984.

[2] Chung K.-S.: Generalized tamed frequency modulation and its application for mobile radio communications, IEEE Journ on Selected Areas in Commun. vol.SAC-2, No.4, pp.487-497, 1984.

[3] Wysocki T. jr., Rydel M.: A method of evaluating power spectral density of digitally modulated signals (in Polish), Proc. of Polish Academy of Science Rozprawy Elektrotechniczne, vol. XXXIII, No.2, pp.401-421, 1987.

[4] Pietroiusti R.: Binary line codes for digital transmission on optical fibres, Contr. CCITT, COM XVIII, 1979.

[5] Wysocki T., Slominski M.: Efficient narrow band modulation for digital communications, 2nd EURASIP Workshop on Medium- to Low- Rate Speech Coding, Hersbruck, FRG, Sept. 20-22, 1989.

[6] Wysocki T.: Easy encodable mBnB codes, Proc. Int. URSI Sympo. ISSSE'89, Erlangen, FRG, Sept. 18-20, 1989.

[7] Korn I.: Generalized MSK, IEEE Trans. Inf. Theory, vol. IT-26 no 2, pp. 234-238, 1980.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

PRACTICAL MEASUREMENTS OF BEAMPATTERNS FOR CONCURRENT TRANSMISSIONS

Ding, S. and Griffiths, J.W.R.

University of Technology, Loughborough, U.K.

Earlier reports have shown by computer simulation that it is possible to transmit a set of beams each pointing in a different direction and each of a different frequency. This report presents the results of some practical measurements on concurrent transmission using one of the flexible sonar transmitters which have been developed at the University.

1. INTRODUCTION

In the sector scanning sonar based on transmitter scanning TRANSCAN [1], a succession of pulses are transmitted each of a different frequency and each with the beam pointing in a different direction. The bandwidth available limits the minimum length of the pulse and hence for a system with 15 or 16 channels the duration of the transmission is very significant, limiting the minimum range that can be observed. Thus it would be very desirable if all of the coded beams could be transmitted simultaneously. An earlier report [2] has shown by computer simulation that in fact, as linear theory would predict, it is possible to transmit all the signals concurrently and this paper reports some practical experiments confirming the predictions.

In the earlier practical systems which have been built, limitations in the memory meant that each sample of stored data was represented by only four bits and thus the effect of this quantization must be considered. A further report [3] showed that quantization to 4 bits does indeed cause some deterioration of the beams but surprisingly little and with 8 bit quantization the effects were hardly discernible.

It was decided to carry out a series of tests on the practical equipment to see if the results predicted by the simulation could be obtained in practice.

2. MEASURING EQUIPMENT

All measurements were made in the department test tank which measures approximately 9 x 6 metres by 2 metres deep.

The SONAR TRANSMITTER used was the low power version of the two equipments which have been developed at LUT [4]. It is based on a Nascom Microcomputer which generates the required waveforms and stores them in 16 memory channels with only four bits per data sample. The outputs from these memories after D/A conversion are, under system control, fed to the individual power amplifiers and so to the array.

The ARRAY used for these tests is only a single line of 15 elements spaced at 37.5 mm. centres which is equal to 1λ at 40 kHz. The transmitting sensitivity of this array has a 3 dB bandwidth of about 12 kHz.

However, over the range of frequencies to be used in the experiments there is a significant variation in response and this will have an effect on the intensity of the different beams.

The AUTO BEAMPATTERN MEASUREMENT SYSTEM is a unit developed at LUT which can measure the pattern of an array automatically and store the results. It includes a Pulse Gate which allows sampling of the direct wave and avoids errors from the multipath signals. This enables the measurements to be carried out in a relatively small tank.

The BANDPASS FILTER, GENERATOR and FREQUENCY CHANGER. These provide a narrow band receiving channel to isolate a particular frequency in the multifrequency transmission. The centre frequency of the bandpass filter is 2 kHz and the bandwidth is 200 Hz. Adjusting the frequency of the generator allows each frequency to be selected. The transient response time of the filter, determined by the bandwidth of 200 Hz, is approximately 7 ms. and thus the pulse length of the transmitted signal must be greater than this to ensure stable conditions. Such a long pulse makes it difficult to separate the wanted signal from the interference. It was necessary to reduce the strength of the multipath signals and this was achieved using a rubber tube as a simple shield. This reduced the interference by about 10 dB which was sufficient to allow the measurements to proceed.

However even with these precautions it was necessary to operate with the receiving hydro-

phone close to the array (about 2.5m) and hence the transmitted waves had to be focussed at this distance to simulate the far field conditions. One of the advantages of the flexible transmitter is that it is relatively easy to carry out this focussing by suitable phasing of the signals in each channel.

3. RESULTS

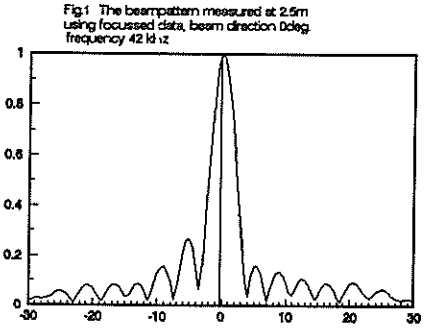
Figure 1 shows the beampattern measured at 2.5m using focussed data, the steered direction is 0^{o} and the frequency 42 kHz. The pattern is not quite the same as would be predicted by theory but the array is known to have slight phase errors. This measurement serves to test the system and the sensitivity and was used before each new set of results were taken. The maximum amplitude of the received signal can be observed on the oscilloscope so as to set the reference direction to the normal of the array.

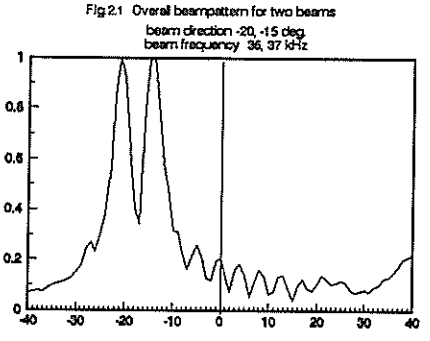
Figure 2.1 shows the beampattern for two beams of different frequencies steered to -20^{o} (36 kHz), and -15^{o} (37 kHz). Figures 2.2 and 2.3 show the beampatterns measured after filtration and it can be seen that the individual beams can be separated with some small leakage due to the filter response.

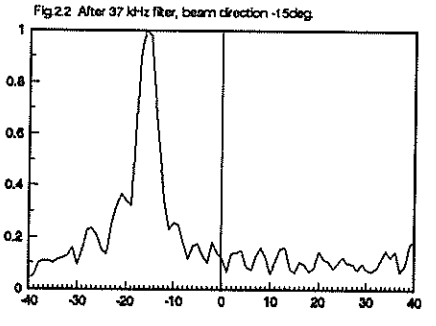
Figure 3.1 shows the overall pattern when five beams are transmitted and in figures 3.2 to 3.6 we see the results after appropriate filtration. There are obviously some unwanted sidelobes but in most cases this is associated with the diffraction secondaries rather than being due to the simultaneous transmission. In order to obtain as narrow a transmitted beam as possible with a fixed number of elements a decision had been made to use an array with 1λ spacing even though this can lead occasionally to some ambiguity due to diffraction secondaries. Thus when the beam is steered to $+30^{o}$ there is an equal amplitude diffraction secondary at -30^{o} and vice versa.

In the results presented so far the calculations to obtain the beams had assumed the elements were point sources and hence had not taken into account the effect of the beam pattern of the individual elements. Thus although the beams were designed to have the same amplitude it is clear from the results that as the beam is steered further from the normal its amplitude reduces.

Figure 4 and figure 5 show the resulting beampatterns when 15 beams are transmitted concurrently. In figure 4 the frequency range is from 36.5 to 43.5 kHz, spaced at 0.5 kHz; similarly figure 5 shows the results when the frequencies transmitted range from 36 to 50 kHz with a spacing of 1 kHz. The spacing of the beams in angle is based on $\sin(\alpha)$ rather than α. Thus they are set at intervals of $\sin\alpha=1/14$ between -1 to +1 i.e., between -30^{o} and $+30^{o}$.







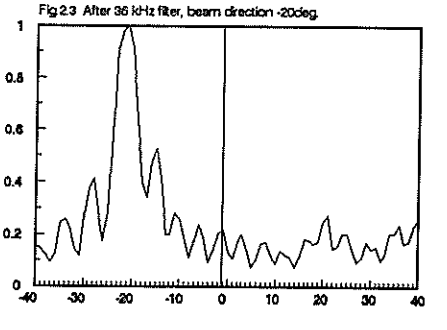
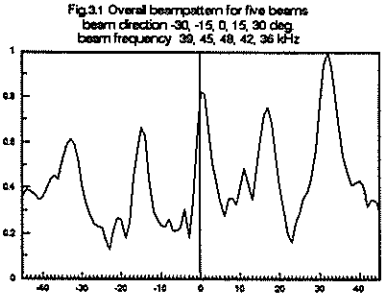
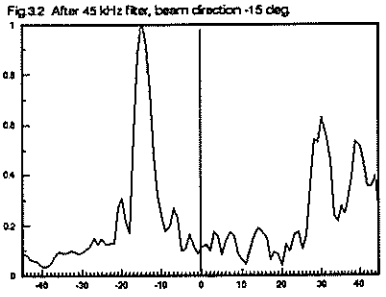
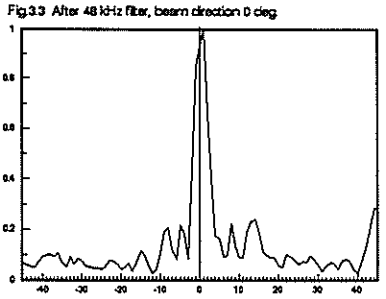
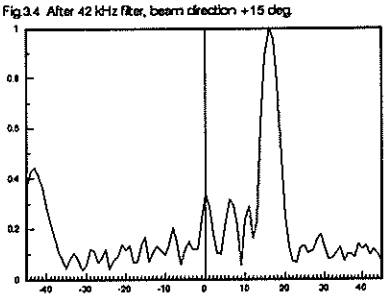


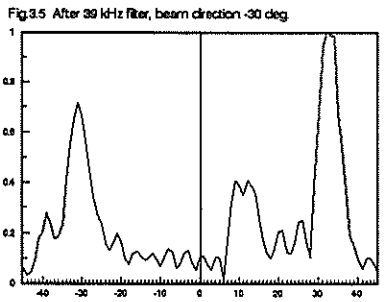
Fig.3 The beampattern for five beams of different frequency steered to five different directions and results after filtration (pulse length 8.2ms)

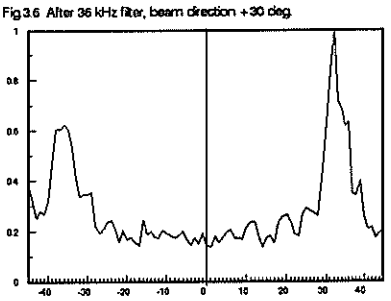


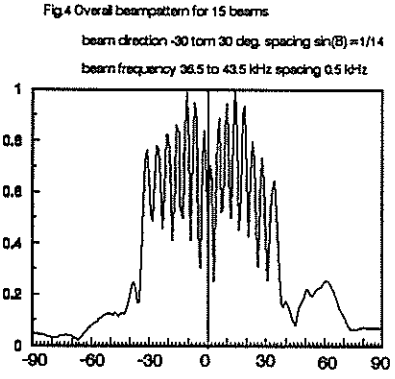


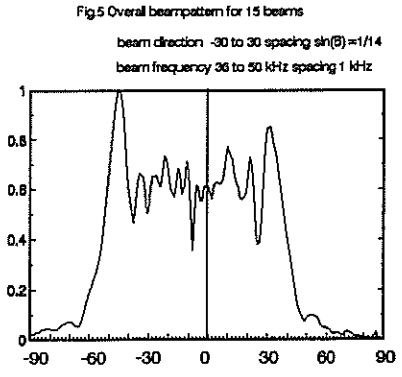












It will be noticed that there is a clear difference between the results for the two frequency ranges.

4. CONCLUSION

It is fairly clear from the results obtained in these experiments that even with the limitation of the 4 bit resolution in the memory it is possible to transmit a number of beams of different frequencies at the same time and to separate out each beam on reception by suitable filtration.

ACKNOWLEDGEMENT

A number of people in the Sonar Group assisted in the carrying out of these experiments and the authors would like to express their sincere thanks for their help.

REFERENCES

[1] Cook, J.C. et al, 'A Sector Scanning Sonar System using Transmitter Scanning', Proc. of UDT 1988, 26/28 Oct. 1988.

[2] Ding, S. and Griffiths, J.W.R., 'Computer simulation of multisignal transmission', Proc. UI'89, Madrid, Spain, July 1989.

[3] Ding, S. and Griffiths, J.W.R., 'Concurrent transmission: the effect of quantization', Aug. 1989, Sonar Res.Grp. Internal Rep. No. 30, Loughborough Univ. of Technology.

[4] Goodson, A.D. et al, 'A Flexible Sonar Transmitter', Proc. IOA, Vol. 8, Part 3, pp. 197-205, Salford, April 1986.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

STUDY AND FABRICATION OF INSTRUMENTATION INTENDED TO MEASURE THE BIOMASS OF A RESERVOIR

Salvetat R.* / Garandel Y.* / Mayet A.* / Aragon B.*/ Tourenq J.N.**

*Laboratoire d'Acoustique de Métrologie et d'Instrumentation
38,rue des 36 ponts 31400 TOULOUSE.- FRANCE
**Laboratoire d'Hydrobiologie - CNRS UA 695 -
118,route de Narbonne 31000 TOULOUSE.- FRANCE

Abstract: To set up a predictive model for the management of a (natural or artificial) reservoir, it is necessary to estimate the fish stock as accurately as possible. Work has shown the interest o echosounding methods for estimating the biomass. However, the instrumentation is cumbersome and exploitation of the recordings uncertain. This led us to envisage designing more efficient, more accurate instrumentation which would also be transportable. In the development of this instrumentation, we have made a model centred on performing a census of the aquatic population and numerically processing the signals received.

1. INTRODUCTION

Biologists and hydrobiologists working on water bodies to obtain a better understanding of the ecosystem have encountered the following problems:

- knowledge of the fish population from a purely qualitative and not quantitative point of view;
- poor understanding of the dynamics of water masses and the ensuing movements of microparticles;
- localization of the thermocline enabling lake stratification to be studied.

Studies [1][2] have shown the interest of an echosounding method for partially solving the above problems. Up to now the bulk and weight of the instruments needed to implement this technique have prevented ecologists from applying the techniques used for investigations on the sea. Our intention is to develop easily transportable instrumentation fulfilling the specific needs of the biologists and allowing the results to be exploited almost immediately. Since the field of action is reservoirs, certain limits have been defined concerning the depth, and the fish population and density. This makes high sampling rates possible while keeping the number of samples within a range which can be processed in real time by a portable system.

2. ECHOSOUNDING

2.1. General:

Echosounding consists of performing acoustical sounding in an aquatic medium to determine the depth of water, the nature of the bottom, or detect shoals of fish.

A reflection index TS is defined for each type of target:

$$TS = 10.\log(\sigma_{bs})$$

σ_{bs} is the monochromatic backscattering cross section: $\sigma_{bs}(f) = 4.\pi. \mid C(f) \mid^2$
where C(f) is the transfer function of the target for the case we are interested in, a fish.

Fishes are all the more easily detectable as they have a bag under the abdomen, called a swim bladder, which is full of oxygen and nitrogen and is used for floating, among other things. It represents 7% of the volume of fresh water fish (5% for sea fish). This air filled bag has a characteristic impedance clearly different from those of the flesh of the fish and the water, and can thus be considered as an important acoustic reflector. Studies carried out by Foote [3] show that 90% of the energy is due to the swim bladder, i.e. the reflection index, TS, of the fish without its swim bladder is 10 to 15 dB smaller than with the bladder. C.S. Clay defines an acoustic model of the fish based on the swim bladder [4] where the fish's back scattering cross section is defined as the sum of a coherent component and a distributed incoherent component:

$$\sigma_{bs} = \sigma_c + \sigma_d$$

one can also define

$$\Gamma = \sigma_c/\sigma_d$$

Γ depends on the behaviour of the fish and the ratio of the size of the fish to the emitted wavelength.

As the signal to be processed results from the detection of an envelope, it seems necessary to

consider the probability density of the envelope $r_x(t)$ as a Rice-Nakagami distribution:

$$p_r(r_x) = \frac{r_x}{(\sigma_n)^2}.\exp\left[-\frac{r_x^2 + r_s^2}{2.(\sigma_n)^2}\right].J_0\left[\frac{r_x.r_s}{(\sigma_n)^2}\right]$$

Where J_0 is the modified Bessel function
In this case the signal to noise ratio (σ_n)is defined by Γ.

2.2. Application to estimation of the fish stock:

2.2.1. Numerical estimate

The first stock estimation studies consisted of counting the echos on the recording paper to obtain a density in individuals per unit volume, knowing the beam volume and the number of emissions. (Cushing 1952) [5] . Since then progress in computing has made it possible to count the echos and evaluate their intensity automatically.

The beam volume for a given sounding device depends only on the depth.

$$V_c = f(H^3) \quad \text{where H is the depth}$$

To obtain the volume sampled during the complete sounding campaign, the volumes of individual emissions are added together to give a density which can be considered as the result of drawing a sample of a population, and a statistical process can then be applied to give a probability density with a confidence interval.
A study performed by Marchal [6] has shown that this can be as accurate as echo-integration and, above all, far less costly, if the fish are well dispersed and the area studied not too large.

2.2.2. Echo-integration

During the echo sounding campaign the sounding signals are recorded on magnetic tape. The cassettes are then passed through a device called an echo-integrator. This device first transforms the signal amplitude U into its square $f(U^2)$ for each pulse, thus giving a value proportional to the intensity. Each pulse is then cut into layers of water having a thickness e = c.Tb/2. Next, values are averaged for each layer of thickness.

$$(U^2) = \frac{1}{t_2 - t_1}.\int_{t_1}^{t_2} f(U^2).dt$$

with t_2-t_1=Tb/2

The values obtained for each elementary layer can be summed for each pulse so as to obtain macro-layers. After this, the sum of the values obtained per layer is taken for a number of pulses corresponding to a unit distance.

The results are presented in the form of a table showing the value of U^2 per distance covered and per layer. The correlation between U^2 and the biomass depends on the model; some take account of beam overlapping which, from a certain depth onwards, introduces a bias by over-estimation. However, this technique is very cumbersome to use: problems of calibration of the echosounder, exploitation of the results delayed.
Studies [2] have shown that this estimation technique can be used for shoals of fish but is not suitable for isolated fish.

3. STUDY OF THE INSTRUMENTATION:

Echosounding can be envisaged as a fast, practical - and in some cases the only - means of obtaining information on the biology, ecology, and dynamics of fish populations and the biomass. The apparatus to be made should enable the biomass of a reservoir to be estimated. For this, the user will divide up the water body to be explored into a grid. The instrument must classify fish by depth layer and by size. Statistical processing will have to be used for this classification since only a part of the water body will be sounded. Improvement will be made to the estimate by making the sampling grid denser and improving the geostatistical exploration strategy.

3.1. Hardware realization:

We start with an echosounder head on which we perform suitable signal processing. A preliminary study enabled us to know the shape and nature of the signals available at the sounder receiver. The echosounder we use is of the monobeam, monofrequency (192 kHz) type with programmable emission durations (40 - 2000 μs). We also have a narrow beam sounder (angle at -3dB θ= 8°).
Within the development of the instrument, we have made a model structured around a 16-bit microprocessor (MC68000) working under the OS9 operating system. We possess a transportable version which lets us acquire and analyze data on the experimental site. We have made an interface card for envelope acquisition and numerical processing by Analog/Digital conversion. The converter used is of the parallel type, the conversion being performed on 8 bits and the execution time not exceeding 660 ns. This short conversion time allows us to use a sampling frequency higher than 200 kHz. Because of the processing time requirements, we decided to work in assembly language. As the processing is performed between two emission bursts, we have 100 ms to carry out the acquisition and processing. We therefore limited the depth sounded to 50 metres, which is sufficient for our field of application. We also chose to take a sampling frequency of the order of 150 kHz, which represents a quantity of samples of 10 kBytes for one emission,

to be processed by a signal processor (ADSP 2100 - card in course of development) before the next emission. This model allows a processing algorithm to be developed to solve a detection and estimation problem.

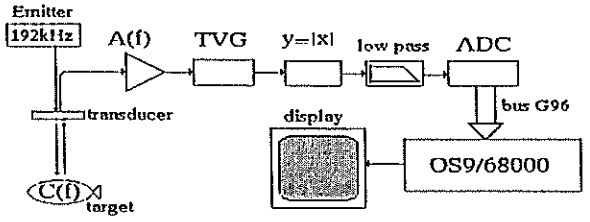


figure 1: Hardware realization

3.2. Software realization:

Studies by Ehrenberg [8] and Marchal [6] lead one to think that, for low densities of fish and medium sized areas to be explored, the counting technique is more accurate than echo-integration. It should be noted that, for these theories to be respected it is important for the size of the fish to be sufficiently large relative to the wavelength ($\Gamma > 50$), which is our case, given the species living in the lakes and the recurrence frequency of the sounder. Within the framework of the development of processing algorithms, we chose to take the above works as a basis.

Our algorithm first corrects attenuation and geometrical spread (Time Gain Varied). Then echos are detected with respect to a minimum value (or threshold value). Numerical integration is then performed on the envelope, which gives us the acoustic intensity of the reflected echo. From this intensity we can work back to the back scattering cross section and thus obtain information on the size of the fish. The use of a narrow beam allows us to collect only the echos from targets situated along the acoustic axis; the directivity problem is thus partly solved and the overlap phenomenon minimized.
The use of a microprocessor for signal processing gives us hope firstly of an improvement in our statistical processing algorithm and secondly of an extension of the field of application of our instrumentation to following the movement of masses of water (image processing); this second phase is currently under way.

4. EXPERIMENTS:

The experimental part was carried out at the Pareloup site (Aveyron) and in the lock on the Canal du Midi at Castanet (Haute Garonne) in collaboration with the Laboratoire d'Hydrobiologie of the Université Paul Sabatier, Toulouse. The model was finalized in the lock at Castanet, which has the advantage of being sufficiently deep and wide for reflections from the walls not to be a problem.
Another series of experiments was performed at Pareloup so that we could familiarize ourselves with the technique. The following facts became clear:
- problem of number of bursts to localize a fish while one is moving;
- influence of the setting of the reception signal sensitivity threshold, echosounder calibration technique;
- problem connected with the variation of the TVG function.

5. CONCLUSION:

The ultimate utility of a project such as this is interesting, particularly for the development of industrial fishing on large lakes such as Pareloup. This kind of instrument would allow hydrobiologists and fishing authorities (Federations, Conseil Supérieur, those fishing hill lakes, research organizations: CNRS, INRA, IFREMER) to gain a better idea of the total exploitable biomass and the distribution of an aquatic ecosystem.

REFERENCES:

[1] BURCZYNSKI J. - Introduction to the use of sonar systemes for estimating fish biomasse - 1979 F.A.O. Fisheries Technical Paper, N°191

[2] LABOULLE H. - Maitrise des méthodes acoustiques en vue de l'évaluation de stocks piscicoles-
D.E.A. Ecologie, 1988 - UPS TOULOUSE III

[3] FOOTE K.G. - Effects of fish behaviour on echo energy: the need for measuring of orientation distribution -
J.Cons.CIEM, 1980 - v.39(2)

[4] CLAY C.S. / STANTON T.K. - Sonar echo statistics as a remonte-sensing tool: volume and seafloor.
IEEE Journal of Oceanic Engineering, vol OE-11, n°1, January 1986.

[5] CUSHING D.H. - Echo surveys of fish - J.Cons.CIEM, 1952 - v.18 (1)

[6] MARCHAL E. - Comptage des écho-traces & integration des signaux, deux méthodes acoustiques appliquées a l'évaluation de la biomasse du stock de Harengs de Strait of Georgia (USA) Symp.Fisheries Acoustics FAO, 1982 n°300

[7] SALVETAT R./GARANDEL Y./ARAGON B. GUILHOT J.P. -Etude et réalisation d'une instrumentation à base d'échosondeur en vue de l'évaluation de la biomasse d'une retenue - Journée Scientifique en ASM, 1989 Note du LMA

[8] EHRENBERG J. - Echo counting and echointegration with a sector scanning sonar - Journal of Sound and Vibration, 1980 - v.73(3) p.321-332.

[9] MACLENNAN D.N. - Acoustical measurement of fish abundance. J.Acoust.Soc.Am 87 (1),January 1990

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

An Acoustical Measurement and Modelling Approach for the Remote Sensing of Stratified Marine Geological Systems

Luc PEIRLINCKX, Leo P. VAN BIESEN, Serge MASYN and Stanislas WARTEL*

Vrije Universiteit Brussel, Department of Fundamental Electricity and Instrumentation, Pleinlaan 2, 1050 Brussels, Belgium
*Koninklijk Belgisch Instituut voor Natuurwetenschappen, Vautierstraat 29, 1040 Brussels, Belgium

In order to derive the geological properties of a marine system from the measured acoustical parameters, models which describe the physical phenomena taking place are required as well as modern digital signal processing methods to assure a fast mode of operation. Therefore, a model is proposed which takes into account absorption and dispersion effects. In order to model the behaviour of mud deposits, a density gradient is simulated. Furthermore, a parametric approach is presented to estimate the PSD of the colored noise sources. Finally, the synthetic reflectograms generated with this model are compared with the measured reflectograms.

1. INTRODUCTION

Many scientifical institutes, as well as private companies, are involved in the study of the composition and the displacement in function of the time of mud deposits and other sediments in estuaries and seas ([1]-[3]). This is because the deposits contain an important source of information for the marine geologists, marine biologists, ecologists and chemists, but also because the silting up phenomenon acts as a practical, economical problem in many European harbors. Many great efforts are put in every year in having the European seaports accessible to their respective tonnage or even to improve the latter. Therefore, it is necessary to maintain navigational routes by dredging operations at very regular instants, but when quay walls, dams, barrages or locks have to be constructed, one has to gain understanding of the behavior of sandbanks and mud layers in order to enable the prediction of their evolution in short and medium length terms. An accurate mapping of silted areas is furthermore of extreme importance with respect to the environment. It is very well known that mud represents a reservoir for many toxical waste products and pollutants, such as heavy metals [4].

Nowadays, the possibility to obtain in real-time information with high accuracy on the content and the precise location of the different sediment layers and the bottom is not present yet. The depth and echo-sounder techniques in combination with commercial available chart-recorders on board of most ships do guarantee sufficient security to be on the safe side, e.g. alarm conditions in shallow waters, but, although very expensive instruments and advanced visual displays are used, do not allow an accurate and detailed description of the distinct layers, which is of fundamental importance for the explanation of the transport mechanisms of the sediments and for the monitoring of the seaways. The actual measurement instruments do not speed up the rather complex process of accurate mapping and ordnance survey either, and hence the real-time processing of the measured data on board of the vessel, performing the survey, would mean a serious scientifical, technological and economical advantage.

2. THE MEASUREMENT TECHNIQUE : reflectogram records obtained from time domain reflectometry.

The acoustical reflectometry is an appropriate method to solve some of the problems stated in a marine environment. One is only interested here in the study of the sediment layers and the near subbottom, and not in a geological prospecting of the underground of seas and rivers. This means that information up to a few meters of depth via a non-destructive or non-disturbing technique is envisaged, i.e. without having the composition and location of the sediment layers altered while measuring. This implies that the use of probes towed along the seabottom is excluded. The acoustical echosounding method is for this remote sensing task a far better candidate then the techniques based on seismics. This is due to the fact that it is possible to generate emitted pulses with a high power content electronically (up to a few kW) and to the short duration of an experiment, since only a low penetration is aimed, so that with a high repetition frequency the bottom can be scanned (e.g. 5 records/second). This high repetition rate, together with the relatively low speed of the vessel (up to 4 or 5 knots), confirm the assumption that it is possible to make a continuous scan of the bottom instead of a discrete one (the maximal horizontally measured distance between two records is then limited to about 0.5 m).

Although from metrological point of view, the acoustical reflectometry seems to be a good measurement method, it follows from practice that the interpretation of the echograms is a simple nor a trivial task! Several arguments can be used to explain this statement: the definition of mud

(but also other sediments), the used instrumentation and the physical environment where the experiments have to be carried out. The definition of mud is certainly one of the most difficult tasks because it is a very heterogeneous, cohesive sediment composed of a small fraction of sand and an important fraction of clay and organic material. It is known that the clay-particles are playing an important role in the storage of pollutants. Therefore, the thickness and the composition of the mud layer is extremely important to derive the content of pollutants in the marine environment. The currently used measurement set-up for subbottom profiling echosounders is shown in figure 1. A measurement probe is towed aside the ship. The probe contains a set of transmitter/receiver transducers (several piezo-electric/ceramic sensors to receive and/or more ceramic transducers to transceive), which is located in a dynamical acceptable housing, called a fish. The transmitter converts an electrical impulse into an acoustical wave.

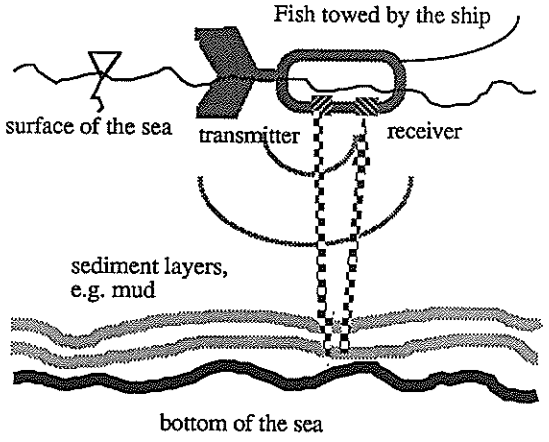


Figure 1: A typical subbottom profiling echosounder measurement set-up: the transceiver part.

The power of the emitted signals is a function of the observed attenuation of the media and of the depth and has a value ranging up to several kW. The emitted power can be seismic, sonic or ultrasonic (from a few mHz up to several hundreds of kHz). The actual emitters usually offer an impulse windowed amplitude modulated signal with a fixed carrier, e.g. 3.5 kHz or 12 kHz in the audioband or 300 kHz in the ultrasonic band. Their spectrum is therefore composed of the spectrum of the applied time window convolved with a Dirac-distribution located at the carrier frequency, mostly yielding into narrowband signals due to the sinc approximation of the windowed spectra. It has been shown, however, that this point of view is in contradiction with the optimal generation of echograms by the acoustical generating unit [5]. The reason that such signals are still popular is due to the fact that only a limited bandpass characteristic is needed and the ability to generate such pulses with high power relatively easy. Furthermore, a strong resemblance with seismic pulses can be observed.

Due to the poor definition of mud and to the disturbances invoked by the measurement environment (roll and beat of the vessel, superposition of periodic noise generated by the propellers, engines and power generators, to the background noise, noise produced by waves and the turbulences of the fish etc.) a practical echogram is obtained, where the unique determination of the structure and the parameters of the measured medium using a visual interpretation has become almost impossible. A typical echogram, recorded in the Kalo lock, is shown in figure 2.

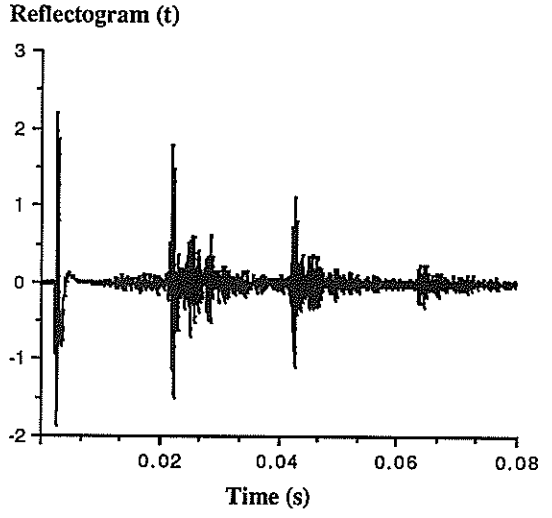


Figure 2: A reflectogram recorded in the Kalo lock

One recognizes the emitted pulse, interwoven reflections on a mud layer containing a gradient of the density and the primary as well as secondary reflections on the sea-floor.

3. MODELLING OF THE ACOUSTICAL PROPAGATION PHENOMENA

The outlined model is based upon a layered structure representation of the seabed ([5],[6]). Furthermore, the effects of absorption and dispersion are taken into account, using a linear wave propagation theory ([5]-[8]). Making these assumptions, the noisy reflectogram is given by:

$$r(t) = \mathfrak{I}^{-1} [H_m(\omega) . H(\omega) . S(\omega)] + n(t) \qquad (1)$$

with: $\mathfrak{I}^{-1}$: the inverse Fourier transform

$H_m(\omega)$ the transfer characteristic of the measurement equipment

$H(\omega)$: the transfer function of the marine system

$S(\omega)$: spectral content of the acoustic input signal

In a first approach, a noiseless simulation is carried out for the measured echogram shown in fig. 2, without taking into account the measurement equipment. The geological parameters involved in the model are determined from

samples taken on the same location where the acoustic reflectogram was recorded (density profile, sediment type). The acoustic parameters (quality factor and velocity of sound) are obtained from the work of Hamilton [9]. Furthermore, it is readily seen from the measured reflectogram that 3 major reflections occur. An accurate estimate for the thickness of the water layer is given by the cepstrum of the measured reflectogram (fig. 3).

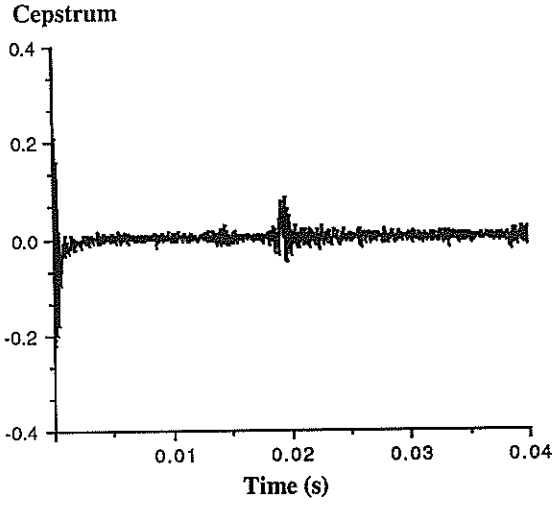


Figure 3: Cepstrum of the measured reflectogram.

With this set of parameters the marine system is described in a first order approximation with a 4-layer model (seawater - mud - clayey sand - sand). The simulated reflectogram is shown in figure 4.

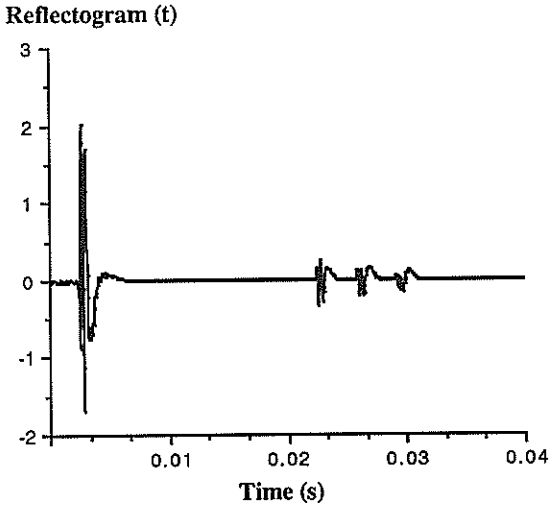


Figure 4: Simulated reflectogram

4. MODELLING OF THE COLORED NOISE SOURCES

In order to study the colored noise sources corrupting the measured echograms, a dynamic signal analyser (DSA-720-SPINNOV [10]) is used to digitize the noise records without aliasing errors (anti-alias filter: 11th order Cauer elliptic filter, pass band ripple of ± 0.3 dB, stopband attenuation 96 dB). The PSD (Power Spectral Density function) of the correlated noise sources, generated by propellers, engines, stabilizers, the fish incorporating the transducers, the water waves and the marine background noise, is estimated using a non-parametric as well as parametric approach [11],[12]. The estimated PSD obtained with the autocorrelation method and with an ARMA modellisation are compared with the periodogram (fig. 5 and 6).

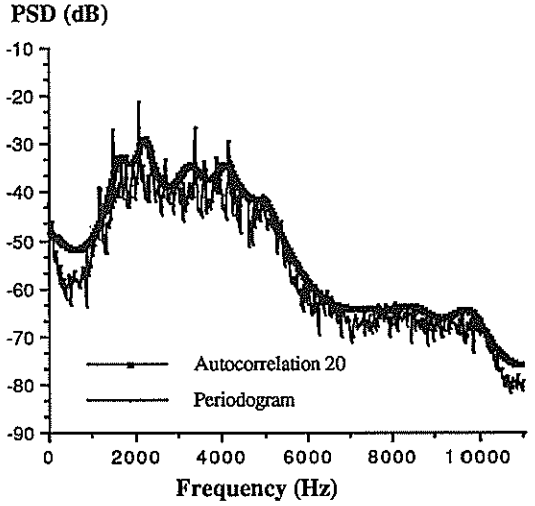


Figure 5 : Estimated PSD with autocorrelation method and with the periodogram

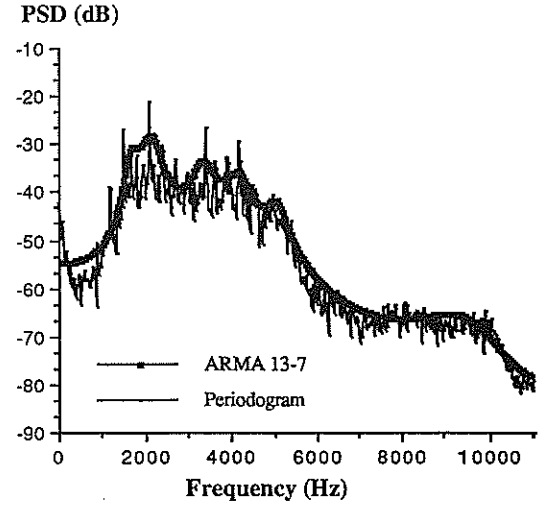


Figure 6 : Estimated PSD with ARMA modellisation and with the periodogram

This study demonstrates that the shape as well as the fundamental components appearing in the estimated PSD correspond with the periodogram. Furthermore, this parametric approach makes it possible to incorporate the influence of the colored noise sources into the model which describes the acoustical propagation phenomena. Calculating the time signal n(t) in (2) from the estimated PSD with ARMA-model and taking into account the transfer characteristics of the measurement equipment (receiver filter, Time Varying Gain amplifier) the synthetic reflectogram can be validated.

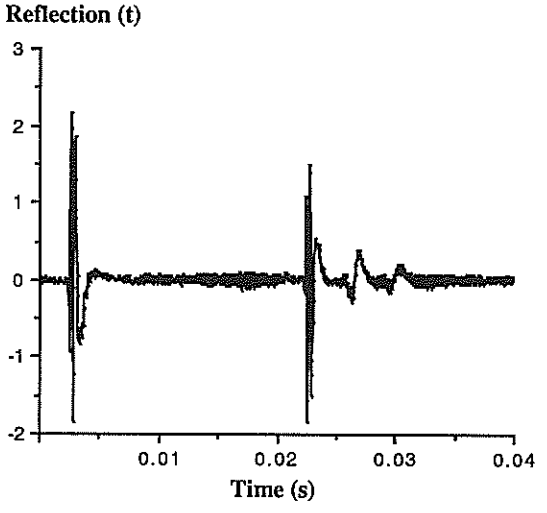


Figure 7 : Synthetic reflectogram including the effect of the receiver network, TVG amplifier and the colored noise sources.

5. CONCLUSION

In this paper, the acoustic reflectometry is discussed as a measurement technique for the remote sensing of stratified marine systems. A modelling approach is introduced, which takes into account the acoustic wave propagation phenomena as well as the correlated noise sources present in the marine environment and the transfer characteristic of the measurement set-up. The complete model is validated using a set of parameters obtained from geological sampling.

6. ACKNOWLEDGEMENTS

The authors would like to express their gratitude towards the University of Brussels and the Belgian government for supporting the research and want to thank the members of the crew of the Belgian Oceanographic Ship the "BELGICA" for their outstanding help during the campaigns along the Belgian coast and the Scheldt.

7. REFERENCES

[1] Faas R. W., Wartel S.,"Sedimentology and Channel Slope Morphology of an Anoxic Basin in Southern Netherlands", Estuarine Processes . II, Wiley (Editor), Academic Press, New-York, pp 136-149,1976

[2] Faas, R. W., Wartel S.,"The Effect of Gas Bubble Formation on the Physical and Engineering Properties of Recently Deposited Fine-Grained Sediments", Geologie en Mijnbouw, 56 (3), pp 211-218,1977

[3] Wartel S., Faas R.W.,"Calciumcarbonate in Schelde Estaury Bottom Sediments",Bull. K. Belg. Inst. Nat Wet., Aardwetenschappen, 56, pp 383-389, 1986

[4] Baeyens W., Gillain G., Gjenidi S., Hoenig M., Wartel S., Dehairs ,"Metal Flows In, Out and Through the Belgian Coastal Waters", Belgian Research on Metal Cycling in the Environment , Rondia D. (editor), Proceedings of the Symposium on Belgian Oceanography, Brussels 11-12 October 1985, Koninklijke Academie voor Wetenschappen, Letteren en Schone Kunsten van België, SCOPE Committee pp 113-124, 1985

[5] L. Van Biesen, L. Peirlinckx, S. Masyn and S. Wartel,"Modelling of Multi-Layer Marine Geological Deposit Systems and the Computer Simulation of the Acoustic Propagation Phenomena by Synthetic Generation of Echograms", Proceedings of the European Simulation Congress, p.87-93, Edinburgh, 5-8 September, 1989.

[6] D.C. Ganley,"A method for calculating synthetic seismograms which include the effects of absorption and dispersion.", Geophysics, Vol.46. No.8, aug. 1981.

[7] P.R. Gutowski and S.Treitel,"The generalized one-dimensional synthetic seismogram", Geophysics, Vol.52, No.5, may 1987

[8] Walter I. Futterman,"Dispersive Body Waves.", J. of Geophysical Research, Vol.67,No.13,dec.1962.

[9] E.L. Hamilton, "Compressional-Wave Attenuation in Marine Sediments", Geophysics, Vol. 37, No.4, August, 1972.

[10] "Operating Manual", Dynamic Signal Analyser Spinnov, DSA-700 family, First Edition, January 1988, Triomflaan 190, 1060 Brussels, Belgium

[11] L. Van Biesen, L. Peirlinckx, S. Masyn and S. Wartel, "On the Measurement and the Modeling of the Correlated Noise Sources Corrupting the Marine Geological Echosounding Experiments", Proceedings of the IMTC'90 Conference, IEEE Catalog No. 90CH2735-9,pp. 353-359, February 14-16, San Jose, 1990.

[12] Steven M. Kay,"Modern Spectral Estimation.", Prentice-Hall, 1988.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

INVERSE Q-FILTERING APPLIED TO HIGH FREQUENCY SEA BOTTOM ECHOGRAMS

Pedro Cobo

Instituto de Acústica, CSIC. Serrano 144. 28006 Madrid, Spain

Absorption effects on 30 kHz sea bottom echograms are investigated. First, a constant-Q sea subbottom is modeled. In this step regression equations relating amplitude attenuation and time delaying with the Q factor are obtained. After that, the Q profile of the explored column is estimated from the spectral ratio of non overlapped replicas of the input sonar pulse. The estimated Q profile is then used together with the regression equations to compensate for absorption effects in the whole echogram. The method is illustrated with echosounding data collected from the bottom of the Buendia water reservoir.

1. INTRODUCTION

Acoustical Imaging is a rather powerful technique to visualize the interior of either optically opaque or hardly accessible media. In these applications, an acoustic wavefield is set up at the surface of the medium. The wavefield propagates inward the medium and becomes reflected at each impedance discontinuity. The whole reflected wavefield is then used to retrieve information about the medium interior.

The sea bottom, and specially the subbottom, is a typical example of a hardly accessible medium. Acoustical Imaging of sea subbottoms is called *subbottom profiling*. Here, a transducer sends a sonar pulse towards the sea bottom which penetrates inwards the medium and reflects back at each interface. Replicas of the input sonar pulse come back to the surface to build up the reflection response or *trace*. When the transducer scans a line above the sea bottom, the cascade representation of adjacent traces draws an image of the explored area, the so-called *echogram*.

Processing techniques are required to translate this qualitative image into quantitative information. Hopefully, after processing all propagation effects are removed, so that only reflectivity information is left in the echogram. Well known techniques include:

1. SNR improvement
2. Resolution improvement
3. Presentation of results.

SNR techniques aim to go from the multi-trace echogram to a single-trace representation with improved signal-to-noise ratio. Here, this task is carried out in two steps, Figure 1. First, a *tracking* algorithm developed by Geerlings and Berkhout [1] is applied to pick up the more important events in the echogram. Once the main reflectors are lined up, *lateral stacking* is the tool to calculate a single-trace representation of the whole explored area. A Q profile is then estimated from the spectral ratio of non overlapped replicas of the trace [2]. Vertical resolution is the ability of discriminating close

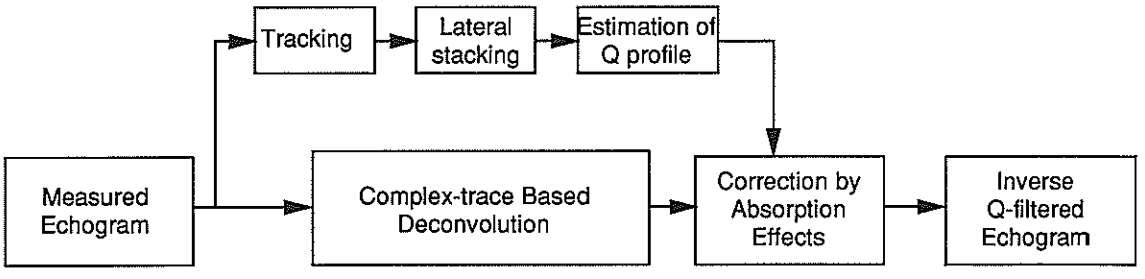


Figure 1. Block diagram of the inverse Q-filtering algorithm.

reflectors in the propagation direction and depends on the spectral contents of the input sonar pulse. In this processing scheme, complex-trace based *deconvolution* is applied to optimally compress replicas of the input waveform into spikes [3]. These spikes are corrected by absorption effects to produce the final image. Thus, the result is a picture of the subbottom reflectors with strengths corrected for amplitude attenuation and positions corrected for time delaying.

Representation of the sea subbottom requires a model. In Section 2, the *constant-Q* layered model is described. Methods used to solve equations of the model are discussed in Section 3. The procedure is then illustrated in Section 4 by applying it to a 30 kHz echogram collected from the subbottom of the water reservoir in Buendía (Spain).

2. THE CONSTANT-Q MODEL

Following to Cobo and Berkhout [4] the influence of absorption in a propagating pulse is threefold: (1) amplitude attenuation, (2) modified travel time, and (3) pulse spreading. Amplitude attenuation is due to the energy decaying as consequence of a variety of dissipation processes. Time delaying and pulse spreading are owing to the distorsion of the Fourier spectrum.

I assume a geoacoustic model of the sea subbottom characterized by:

1. Linearity and time invariancy
2. Horizontally layered medium
3. Constant-Q absorbing layers.

Linearity and time invariancy means that the transfer function can be synthesized as a summation over all frequencies (superposition principle). For high Q, the next relation between the Q factor and the attenuation coefficient, a(f), can be established

$$\frac{\pi}{Q} = \frac{a(f)}{f} \qquad (1a)$$

If Q is constant within each layer then attenuation is linear with frequency, so that

$$a(f) = \begin{cases} \frac{\pi}{Q} f & f \le f_c \\ 0 & f > f_c \end{cases} \qquad (1b)$$

According to Strick [5] amplitude attenuation implies phase distorsion if the propagating pulse have to be causal. Here, minimum-phase distorsion is considered (Paley-Wiener condition) so that

$$\hat{a}(f) = \frac{f}{Q} \ln\left[\left(\frac{f_c}{f}\right)^2 - 1\right] \qquad (1c)$$

where f_c is an upper frequency bound. Horizontal layering means that both the Q factor and the acoustic impedance only vary through the interfaces (piecewise profiles).

The impulse response for such a system can be written as [4]

$$r(t) = \sum_{n=1}^{N} r_n \int_{-\infty}^{\infty} e^{-[\pi f(\tau_n - \tau_1)/Q_n]} e^{j[2\pi f(t-\tau_n) - (2f/Q_n)\ln(f_c/f)]} df \qquad (2)$$

where τ_n is the two-way travel time of the n-th interface and Q_n is the effective quality factor of the n-th layer; f_c has been taken as twice the Nyquist frequency, so that $\ln[(f_c/f)^2 - 1] \approx 2\ln(f_c/f)$ for all frequencies. Notice that internal multiples are not included in this formulation.

3. METHODS

Inverse Q-filtering is based on the estimation of the Q-profile and the restoration of the correct amplitude and travel time for each peak event. A Q-profile is estimated by means of the spectral ratio method [3, 4]. According to this method, the ratio between the log-spectra of subsequent echoes in the trace is a straight line with slope, β, given by

$$\beta = \frac{2\pi(\tau_m - \tau_n)}{Q_{mn}} \qquad (3)$$

where Q_{mn} is the Q factor of the layer sandwiched by interfaces with travel times τ_m and τ_n. From these slopes the Q factor of each layer can be easily calculated.

The estimation of a Q-profile requires a single-trace representation of the whole explored area. This is accomplished by lateral stacking. Since 30 kHz are very sensitive to lateral variability (λ=5 cm for v=1500 m/s) lateral stacking might be preceded by some optimal alignment of reflectors. Here, a heuristic tracking algorithm is applied prior to lateral stacking. This algorithm, developed by Geerlings and Berkhout [1], is a combination of pattern recognition and picking techniques. A candidate window is positioned around each main reflector in the first trace. From this starting point, the problem is to find the optimal tracks which delineate the major events. The algorithm proceeds by computing the cumulative cost of each node in the next trace. When the last trace is reached, the algorithm runs

back delineating the minimum cost tracks.

Once the Q-profile has been estimated the echogram can be corrected by absorption effects. For this purpose, equations relating amplitude attenuation and time delaying with the Q factor are needed

$$A_p = f(Q) \quad (4a)$$

$$\tau_p = g(Q) \quad (4b)$$

where A_p and τ_p are the amplitude and travel time, respectively, of each peak event. For the peak amplitude the next equation is guessed

$$A_p = e^{-K\tau/Q} \quad (4c)$$

where K is a constant to be fitted by direct modeling. Notice that A_p goes to unity wether Q goes to infinity (non absorption) or τ goes to zero (non propagation).

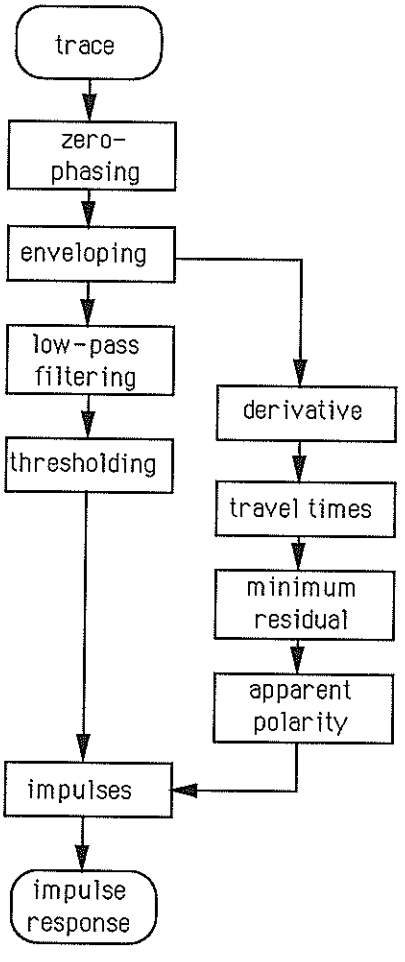


Figure 2. Complex-trace based deconvolution

Should Eqs. (4) be obtained, the peak events in the echogram could be easily corrected for attenuation and time delaying. Thus, deconvolution should be applied to compress replicas of the input sonar pulse into spikes. The complex-trace based deconvolution method is sketched in Figure 2. Zero-phasing is applied first to optimally shape the pulse waveform within the frequency band. From this shaped trace, the envelope is then calculated. Low-pass filtering smooths the envelope. Zeroes of the derivative of the envelope helps to locate the spikes. A threshold is used to discriminate weak events. The apparent polarity is finally estimated by minimizing the residual between real and synthetic traces within a candidate window around the spikes.

4. RESULTS

An experiment was carried out by the Hydroacoustic Laboratory of the Acoustic Institute (Madrid) to explore the sea bottom of a water reservoir at Buendía (Cuenca, Spain). The aim of this survey was to obtain a high resolution image of the sediment filling coming from the washing produced by the rainfall in the area around the reservoir. An ELAC LAZ 721 echosounder emitting 30 kHz ping sonar pulses (band ratio ≈ 1.2) was used as source. Figure 3a shows the image of the old basin river beneath the water reservoir. Six major reflectors on this echogram were tracked and aligned first. From the corresponding stacked trace, a Q-profile was then estimated. Since successive echoes were compared with the first one coming from the water-sediment interface, the effective Q-profile was indeed estimated. Table I shows the result including the regression parameter of the straight line from whose slope the Q factor was calculated. The whole echogram deconvolved and inverse Q-filtered appears in Figure 3b. Notice the drastic vertical resolution improvement gained with deconvolution. The apparent loss of lateral coherence is due to the heave component of the ship which has not been corrected for. The main effect of inverse Q-filtering is to stand up deeper reflectors which appeared weakened because of absorption.

Table I. Effective Q-profile of the Buendía subbottom.

Travel time (ms)	Effective Q	Regression parameter
0	∞	1
3.73	295	0.82
4.33	155	0.99
5.43	85	0.99

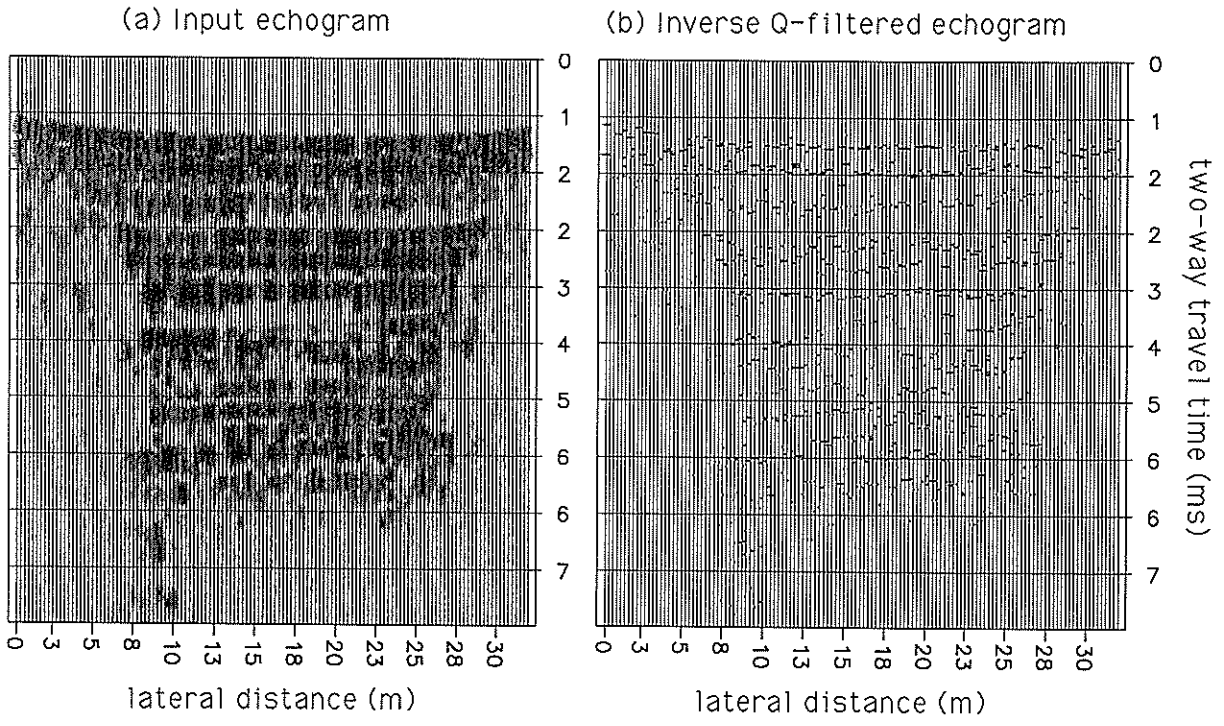


Figure 3.

5. CONCLUSIONS

Quantitative information of echograms is contained in the amplitude and phase spectra of the echoes. The amplitude spectrum contains information about the reflection strength whereas the phase spectrum informs about the two-way travel time from the source to the target. Both the amplitude and phase spectra are distorted by absorption. Since absorption increases with frequency, high resolution acoustical imaging requires inverse Q-filtering.

A layer-cake model with Q-constant absorbing layers affords a good frame for inverse Q-filtering. The spectral ratio method allows to estimate a Q-profile from the stacked representation of the whole explored area. Once the Q factor of each layer is known the amplitude and travel time of its bottom interface can be compensated for absorption.

Inverse Q-filtering combined with complex-trace based deconvolution yields a true-amplitude and correct-positioned image of the shallowest sediment layers of the sea subbottom.

ACKNOWLEDGEMENTS

This research was carried out at the Laboratory of Seismics and Acoustics, Delft University of Technology, The Netherlands, under a post-doctoral stay supported by the MAST Program of the European Communities. The author is grateful to Professor Berkhout, from the Delft University of Technology, for very estimulating and fruitful discussions.

REFERENCES

[1] Geerlings, A.C. and Berkhout, A.J. "Heuristic event tracking linked to linear discriminant analysis". In: Seeman and Aminzadeh, (eds.), **Advances in Geophysical Data Processing**. (JAI Press ,Connecticut,1989), pp. 201-233.

[2] Cobo, P. and Ranz, C. "Direct and inverse problems in layered sea bottoms including attenuation: Synthetic data ". ACUSTICA 69, (1989), 81.

[3] Cobo, P. "Processing scheme for high frequency reflection data from the shallowest sea subbottom". Technical Report, (TU Delft, 1990).

[4] Cobo, P. and Berkhout, A.J. "Constant-Q absorption model for high frequency acoustic exploration of sea subbottoms". ACUSTICA 70, (1990), in print.

[5] Strick, E. "A predicted pedestal effect for pulse propagation in constant-Q solids" .Geophysics 35,(1970), 387.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

TARGET MOTION ANALYSIS USING DOPPLER MEASUREMENTS AND SENSORS SHAPE CALIBRATION

Jean-Luc Nicolas, Frédérique Ywanne, Francis Martinerie

THOMSON SINTRA ASM, 1, av. A. Briand
94117 Arcueil Cedex, FRANCE

ABSTRACT:
The purpose of this paper is to track a target from two sets of doppler measurements provided by a pair of sensors. When the sensors locations are not known precisely, the target motion estimation is biased. This paper presents a method, which estimates simultaneously the target kinematics parameters and the sensors positions.

1. INTRODUCTION

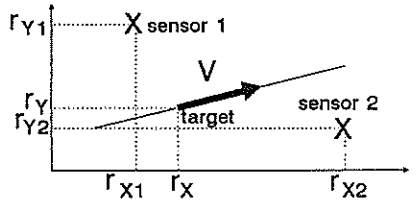
The problem of target motion analysis from doppler measurements when a single sensor is used has already been exposed in the literature [1]. If the frequency emitted by the target is known, the distance to CPA, the time of CPA and the speed are identifiable. Additional information are necessary to eliminate the remaining circular ambiguity.

The purpose of this paper is to track the target from two sets of doppler measurements provided by a pair of sensors. The case when the receivers locations are not known exactly is also examined.

2. SYSTEM DEFINITION

2.1 Geometric description

The system consists of a target with constant velocity and two fixed sensors. The target trajectory and receiver-pair are assumed to lie in a common plane (picture 1).



picture 1: system notations

A complete description of target motion is given by its position at time t_r and its speed components. The corresponding parameter vector in cartesian coordinates is:

$$(1)\quad X_T = (r_X, r_Y, v_X, v_Y)^t$$

In the same way, the sensors are completely defined by their position, i.e. for the receiver #i:

$$(2)\quad X_{Si} = (r_{Xi}, r_{Yi})^t$$

The target parameter vector at time t is given by:

$$(3)\quad X_T(t) = \Phi(t, t_r)\, X_T(t_r)$$

where Φ denotes the transition matrix:

$$\Phi(t,t_r) = \begin{bmatrix} I_2 & (t-t_r)\, I_2 \\ 0 & I_2 \end{bmatrix}$$

2.2 Measurements definition

The omnidirectionnal sensors measure frequencies corresponding to a tone f_0 emitted by the target. Assuming f_0 to be unknown, it must be introduced in the target parameter set, which becomes:

$$(4)\quad X_T = (r_X, r_Y, v_X, v_Y, f_0)^t$$

Each frequency measurement obtained with the ith sensor , denoted $f_{mi}(t)$, is modelized by:

$$(5)\quad f_{mi}(t) = f(X_T(t), X_{Si}) + n(t)$$

where the measurement noise n(t) is assumed to be zero-mean, white, gaussian with standard deviation $\sigma_i(t)$.
The functional relationship between the target localization parameter set X_T and the noise-free frequency measurement at the ith sensor at time t is as follows:

$$(6)\quad f(X_T(t),Xs_i) = f_0\ (1 - \dot{d}_i(t)/c)$$

where c denotes the sound velocity and $\dot{d}_i(t)$ the time derivative of the range:

$$d_i(t) = ((r_X(t)-r_{Xi})^2+(r_Y(t)-r_{Yi})^2)^{1/2}$$

The measurements at each sensor are made at discrete times. Define N the total number of measurements, the set of observations and the model can be written in vector notation as:

$$(7)\quad F_m = (f_{m1}(t_1),\ldots,f_{m2}(t_P))^t$$

$$(8)\quad F(X_T,X_S) = (f(X_T(t_1),Xs_1),\ldots,f(X_T(t_P),Xs_2))^t$$

Assuming the two sets of measurements to be independent, one can define the noise covariance matrix Σ_F:

$$(9)\quad \Sigma_F = \mathrm{diag}(\sigma_i(t_j))$$

3. SYSTEM ANALYSIS AND PERFORMANCES

3.1 Identifiability condition

It has been shown [1] that the doppler track at a single sensor allows us to observe the target speed, the distance to CPA and the time of CPA. When the data of the receiver-pair are mixed, the five components of X_T can be determined up to a reflection with regard to the sensors axis.

3.2 Cramer-Rao lower bounds

The likelihood function of the parameter vector X_T can be written:

$$(10)\quad p(X_T/F_m)=(2\pi^N\ \det(\Sigma_F))^{1/2} .\exp(-1/2\ \|F_m-F(X_T,X_S)\|^2_{\Sigma F})$$

According to the Cramer-Rao theorem, upper-bounds on localization performance can be determined from the Fisher Information Matrix by [2]:

$$(11)\quad CRLB = FIM^{-1}$$

where

$$(12)\quad FIM = E\{\ (dL/dX_T)(dL/dX_T)^t\ \}$$

L denotes the logarithm of (10).

3.3 Algorithms

Two approaches have been investigated. The first one consists in computing for each sensor the target speed, distance to CPA and time of CPA. Then, the cartesian components of X_T are obtained with an analytical transformation.

The second algorithm is based on the maximum likelihood criterion and is equivalent to a weighted least square estimator, which minimizes:

$$(13)\quad J(X_T) = \|F_m-F(X_T,X_S)\|^2_{\Sigma F}$$

A common way to achieve the minimization is to use a gradient search method (Gauss-Newton).

The two pictures (one for each algorithm) (2-a) and (2-b) represent Monte-Carlo results (points and thick dashes) and the position uncertainty ellipse (thin dashes) as predicted by CRLB. Note that Gauss-Newton is asymptotically efficient.

4. UNKNOWN SENSOR LOCATIONS

4.1 Estimation bias

When the sensors are not located as assumed in the model, the target parameter estimations are biased. The bias expression is given by a first order development of the gradient:

$$(14)\quad b(X_T)=FIM^{-1}(dF/dX_T)\Sigma_F^{-1} (F(X_T,X_S)-F(X_T,X_{MOD}))$$

where X_S denotes the true sensor locations and X_{MOD} the supposed ones.
The bias is in evidence in the Monte-Carlo simulations of both algorithms represented in the two pictures (3-a,b). In these simulations, only one receiver has not been properly modelized.

4.2 Sensor localization

A way to decrease the bias consists in estimating one sensor position. A geometric analysis shows that only the distance between receivers is identifiable with doppler measurements. To localize completly the sensor, it is necessary to use some *a priori* information about its position. The sensor is assumed to be located with an uncertainty modelized by a gaussian process of covariance S.

If the *i*th receiver has to be located, the parameter vector X_P becomes:

$$(15)\quad X_P = (X_T, X_{Si})^t$$

and the sensor location is introduced in the observation vector Z_m as follows:

$$(16)\quad Z_m = (F_m, X_{iMOD})^t$$

The covariance on Z is then given by:

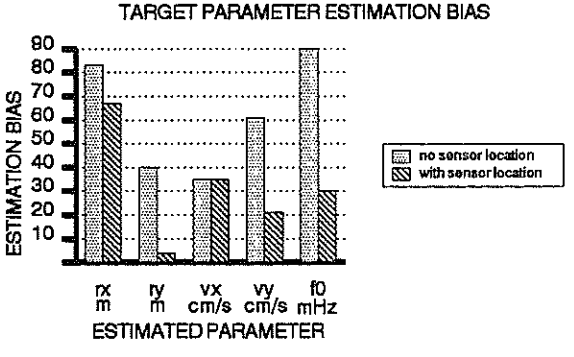
$$(17)\quad \Sigma_Z = \mathrm{diag}(\Sigma_F, S)$$

The system analysis can be studied in the same order as in the third chapter. We obtain the same expressions as (10) and (13), with X_T, F_m and Σ_F respectively replaced by X_P, Z_m and Σ_Z.

Monte-Carlo simulations have been computed with a single draw for sensor location and a hundred draws for frequency measurements. The target parameter and sensor location estimations are still biased. The analytical expression of this bias is:

$$(18)\quad b(X_P) = FIM^{-1} \begin{bmatrix} 0 & 0 \\ 0 & S^{-2} \end{bmatrix} \begin{bmatrix} 0 \\ X_S - X_{mod} \end{bmatrix}$$

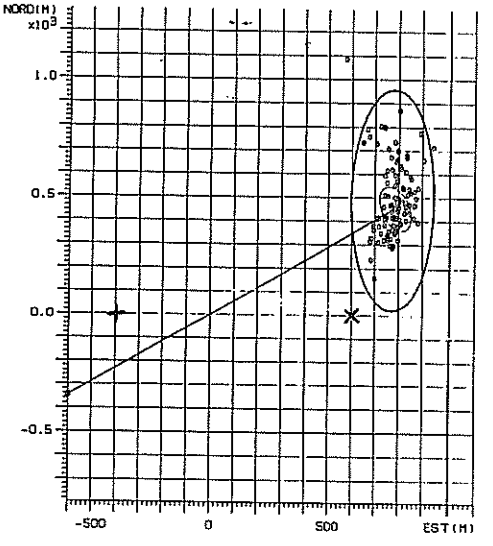
The results are presented on picture 4. We can notice that the bias is lower than it was on picture 3-b. The theorical bias have been computed from the expressions (14) and (18) and are represented in the following bar chart.



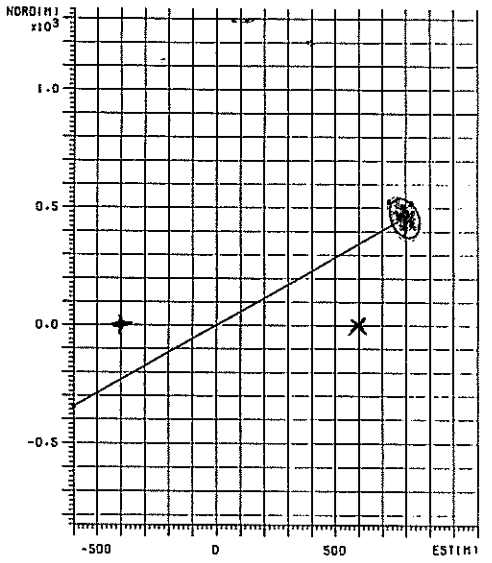
5. CONCLUSIONS

We have examined the problem of target motion analysis from doppler measurements when a sensor-pair is used. Among the two examined solutions, only the least square algorithm is an efficient estimator. When the positions of the sensors are not exactly known, a bias appears on target parameters estimation. To reduce this error, we have estimated in the same time the target parameters and the position of one sensor. In this case, it is necessary to assume some *a priori* information about the sensor location so that the system becomes identifiable. This method can be extended if both sensors should be localized.

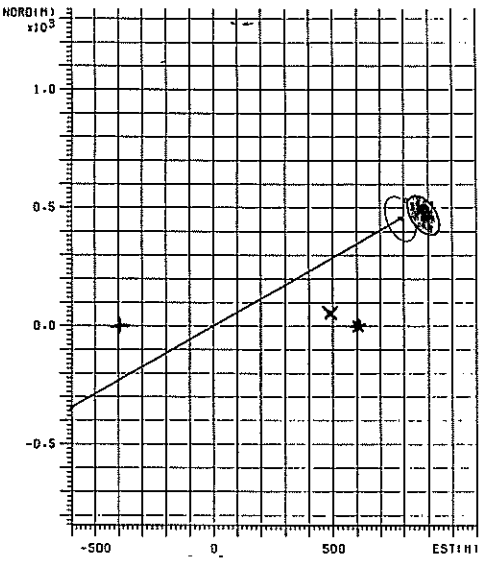
References:
[1] M.J.SHENSA
"On the uniqueness of Doppler tracking"
J.Acoust.Soc.Am.70(4), Oct.1981
[2] VAN TREES
"Detection, Estimation and Modulation Theory", Tome 1
Wiley 68



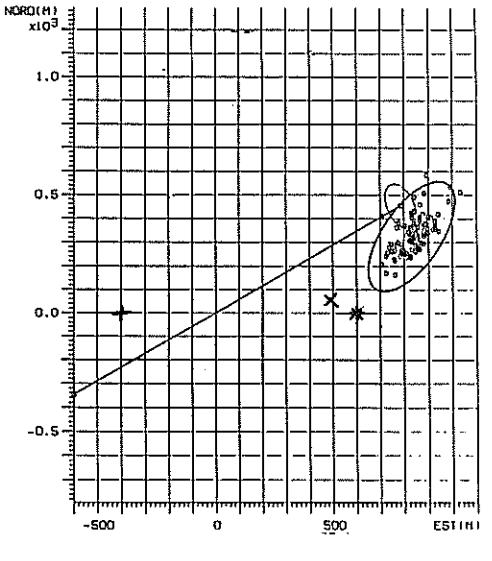
picture 2-a
known sensor locations - algorithm 1



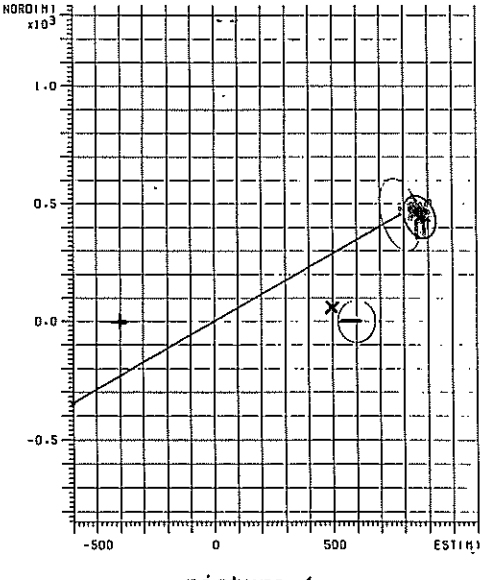
picture 2-b
known sensor locations - algorithm 2



picture 3-b
unknown sensor locations - algorithm 2



picture 3-a
unknown sensor locations - algorithm 1



picture 4
sensor location estimation

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Performance Analysis of Passive Location with Stochastic Wideband Signals

M. João D. Rendas* José M. F. Moura†

*CAPS, Depart. Eng. Elect. Comp., Instituto Superior Técnico, 1096 Lisboa Codex, Portugal.
†LASIP, Depart. Elect. and Comp. Eng., Carnegie Mellon University, Pittsburgh, PA, 15213.

1 Introduction

We compute the Cramér-Rao Bound for the location problem, assuming that the source signal spectrum must be estimated along with the source location. Two different situations are considered. In the first, the source spectrum is completely unknown, and its value at each frequency must be estimated. We conclude that in this case, estimation is not possible with a single sensor, revealing the fact that no spectral-based information retrieval is possible. Secondly, we consider that the source signal has a spectrum of known parametric form and we compute the Cramér-Rao bound for the problem of joint estimation of source location and spectral parameters. In this case, spectral-based information is available, the performance being determined by the relation of the dependency of the observed spectrum in the propagation (location dependent) parameters, and the unknown parameters in the spectrum. We derive conditions for either *(i)* no information loss (same performance as for the compeletely known source spectrum) and *(ii)* no additional information (same performance as for the completely unknown source case).

2 Known Source Spectrum

In this section, we present expressions for the Cramér-Rao bound on the location parameters of a single source of known spectral density. These expressions will be used when deriving the Cramér-Rao bound for signals with unknown source spectra. They have been previously presented in [1] and are discussed in detail in [2].

Under the assumption of large time bandwidth product, the Fourier components of the received signal at sensor k, $(k = 1, \ldots, K)$ are:

$$r_k(\omega) = \sum_{p=1}^{P} a_{kp} e^{j\omega\tau_{kp}(\alpha)} s(\omega) + w_k(\omega), \qquad (1)$$

where P is the number of paths; a_{kp} is the attenuation from source to sensor k through path p; τ_{kp} is the delay from source to sensor k through path p; α is the vector of parameters that describes the source location; $s(\omega)$ is the Fourier component of the source signal; $w_k(\omega)$ is the Fourier component of the observation noise at sensor k. The source signal and the observation noise are both zero-mean stationary Gaussian processes, independent of each other. We further assume that the noise is temporally and spatially white.

The Cramér-Rao bound (**CRB**) for the location parameters is the inverse of the Fisher Information Matrix (**FIM**), $\mathcal{F}(\alpha)$, which, for large sample size, is given (Whittle's formula) by [3]

$$[\mathcal{F}(\alpha)]_{ij} = \frac{N}{4\pi} \int \mathrm{tr}\left\{ \frac{\partial S_r(\omega)}{\partial \alpha_i} S_r^{-1}(\omega) \frac{\partial S_r(\omega)}{\partial \alpha_j} S_r^{-1}(\omega) \right\} d\omega. \qquad (2)$$

where $S_r(\omega)$ and $S_s(\omega)$ are the spectral densities of the observations and the source signal, respectively. Using model (1), the following expression is derived:

$$\begin{aligned} \mathbf{CRB}(\alpha)^{-1} &= \frac{N}{2\pi} \int \left\{ K_1 \mathrm{Re}\left\{ \frac{\partial h}{\partial \alpha}^H P_h^{\perp} \frac{\partial h}{\partial \alpha} \right\} \right. \qquad (3) \\ &+ \left. K_2 \mathrm{Re}\left\{ \frac{\partial h}{\partial \alpha}^H h \right\} \mathrm{Re}\left\{ h^H \frac{\partial h}{\partial \alpha} \right\} \right\} d\omega \end{aligned}$$

where

$$K_1 = \frac{S_s^2}{S_n E} \|h\|^2, \qquad K_2 = 2\frac{S_s^2}{E^2}, \qquad (4)$$

and we have further defined $E \triangleq S_n + S_s\|h\|^2$. The vector h appearing in this equation is the coherent combination of all the received steering vectors. Its generic element is given by:

$$[h(\omega)]_k = \sum_{p=1}^{P} a_{kp} e^{j\omega\tau_{kp}} \qquad k = 1, \ldots, K. \qquad (5)$$

We note that the first term in (4) is zero when $K = 1$, i.e., for a single sensor (see [1]).

3 Nonparametric Spectra

When the source spectral density function, $S_s(\omega)$, is not known it must be estimated along with the location parameters of the source. The **FIM** for the vector of source locations still follows Eq.(2), but the **CRB** must now be computed from the **FIM** for the extended parameter vector $\beta^T = [\alpha^T \eta^T]$, where α is the vector of source location parameters, and η groups all the parameters that define the source spectral density.

Assumption 1 *The source signal spectral density function, $S_s(\omega)$ is completely unknown to the receiver.*

Consider a discretization of the frequency domain. Define element i of η as the value of the source spectrum at frequency ω_i:

$$\eta_i = S_s(\omega_i), \qquad i = 1, \ldots, L, \tag{6}$$

where L is the number of discrete frequencies. $\mathbf{CRB}(\alpha)$ is the upper square block of $\mathcal{F}(\beta)^{-1}$ of dimension $P(K+1)$ where:

$$\mathcal{F}(\beta) = \begin{bmatrix} \mathcal{F}(\alpha) & \mathcal{F}(\alpha,\eta) \\ \mathcal{F}(\eta,\alpha) & \mathcal{F}(\eta) \end{bmatrix} \tag{7}$$

where $\mathcal{F}(\alpha)$ is the **FIM** for α when the source spectrum is known, given by Eq.(4).

Using the formulae for the inverse of a partitioned matrix, we obtain, assuming $\mathbf{FIM}(\eta)$ is nonsingular,

$$\mathbf{CRB}(\alpha) = [\mathcal{F}(\alpha) - \mathcal{F}(\alpha,\eta)\mathcal{F}(\eta)^{-1}\mathcal{F}(\eta,\alpha)]^{-1}. \tag{8}$$

Denote by $\mathcal{L}(\alpha,\eta)$ the loss term in Eq.(8):

$$\mathcal{L}(\alpha,\eta) = \mathcal{F}(\alpha,\eta)\mathcal{F}(\eta)^{-1}\mathcal{F}(\eta,\alpha). \tag{9}$$

With this definition,

$$\mathbf{CRB}(\alpha) = [\mathcal{F}(\alpha) - \mathcal{L}(\alpha,\eta)]^{-1}. \tag{10}$$

To determine the new entries in $\mathcal{F}(\beta)$, we begin by noting that:

$$\frac{\partial S_r(\omega)}{\partial S_s(\omega_i)} = h(\omega_i)h^H(\omega_i)\delta(\omega-\omega_i), \quad i = 1, \ldots, L. \tag{11}$$

This fact implies that the summation over frequency in the **FIM** expression will degenerate into a single frequency contribution. We can now compute

$$[\mathcal{F}(\eta)]_{ij} = \frac{N}{4\pi}\frac{1}{E^2(\omega_i)}\|h(\omega_i)\|^2\delta_{ij}. \tag{12}$$

Similarly, for the cross terms we obtain:

$$[\mathcal{F}(\alpha,\eta)]_{ij} = \frac{N}{2\pi}\frac{S_s(\omega_j)}{E^2(\omega_j)}\|h(\omega_j)\|^2 \mathrm{Re}\left\{\dot{h}(\omega_j)_{\alpha_i}^H h(\omega_j)\right\} \tag{13}$$

resulting, for the (i,j) element of $\mathcal{L}(\alpha,\eta)$,

$$[\mathcal{L}(\alpha,\eta)]_{ij} = \frac{N}{2\pi}\sum_{\ell}\frac{2S_s(\omega_\ell)^2}{E(\omega_\ell)^2}\mathrm{Re}\left\{\dot{h}(\omega_\ell)_{\alpha_i}^H h(\omega_\ell)\right\} \mathrm{Re}\left\{h(\omega_\ell)^H \dot{h}(\omega_\ell)_{\alpha_j}\right\} \tag{14}$$

which in the limit tends exactly to the second term of Eq.(4). We conclude thus that this term is precisely the information which is lost when the source spectral density is not known.

Fact 1 *Consider model (1), where the source signal spectrum is not known. Then*

$$\mathbf{CRB}(\alpha)^{-1} = \frac{N}{2\pi}\int K_1 Re\left\{\dot{h}_\alpha^H P_h^\perp \dot{h}_\alpha\right\} d\omega. \tag{15}$$

Noting that for $K = 1$, $P_h^\perp$ in Eq.(15) is zero, and consequently the **CRB** will grow to infinity, we get the following

Fact 2 *The Fisher Information Matrix of the source location parameters for a single sensor observation ($K = 1$) of the multipath propagation of a single stochastic Gaussian source signal with unknown spectrum is zero.*

4 Parametric Spectra

We consider in this section the case where a parametric expression for $S_s(\omega)$ is available, i.e.,

Assumption 2 *The source spectrum is known to the receiver, except for the L-dimensional vector of unknown deterministic parameters η: η:*

$$S_s(\omega) = S_s(\omega : \eta) \tag{16}$$

As for the case of completely unknown source spectrum, η must be estimated along with the location parameters α. The **CRB** is still given by Eq.(8).

Using

$$\frac{\partial S_r(\omega)}{\partial \eta_i} = h(\omega)h^H(\omega)\frac{\partial S_s(\omega)}{\partial \eta_i} \tag{17}$$

and

$$\frac{\partial S_r(\omega)}{\partial \eta_i}S_r^{-1}(\omega) = \frac{1}{E}\frac{\partial S_s(\omega)}{\partial \eta_i}hh^H \tag{18}$$

leads to:

$$\mathcal{F}(\eta) = \frac{N}{4\pi} \int \frac{\|h\|^4}{E^2} \frac{\partial S(\omega)}{\partial \eta}^T \frac{\partial S(\omega)}{\partial \eta} d\omega \quad (19)$$

$$\mathcal{F}(\alpha, \eta) = \frac{N}{4\pi} \int 2 \frac{S_s \|h\|^2}{E^2} \mathrm{Re}\left\{ \frac{\partial h^H}{\partial \alpha} h \right\} \frac{\partial S(\omega)}{\partial \eta} d\omega \quad (20)$$

Single Unknown Parameter

Consider the case of a single parameter, i.e., $L = 1$, when $\mathcal{F}(\eta)$ becomes a scalar. Define

$$\phi(\omega) = \frac{1}{C} \frac{\|h(\omega)\|^2}{E(\omega)} \dot{S}_{s\eta}(\omega) \quad (21)$$

where

$$C^2 = \int \frac{\|h(\omega)\|^4}{E^2(\omega)} \dot{S}_{s\eta}^{\ 2}(\omega).d\omega \quad (22)$$

Define also,

$$X(\omega_1, \omega_2) = \phi(\omega_1)\phi(\omega_2). \quad (23)$$

Let $\mathcal{S}_\phi$ be the one-dimensional space spanned by the function $\phi(\omega)$. Then, the integral projection operator in $\mathcal{S}_\phi$is $P_{\mathcal{S}_\phi}$:

$$P_{\mathcal{S}_\phi}[v(\omega)] = \int X(\omega, \omega_1) v(\omega_1) d\omega_1 \quad (24)$$

Finally, we define

$$v(\omega) = \frac{S_s(\omega)}{E(\omega)} \mathrm{Re}\left\{ \dot{h}_\alpha^H(\omega) h(\omega) \right\} \in \mathbf{R}^2. \quad (25)$$

With these definitions, we can write the loss term as:

$$\mathcal{L} = \frac{N}{\pi} \int \int v(\omega_1) X(\omega_1, \omega_2) v(\omega_2) d\omega_1 d\omega_2 \quad (26)$$

Since $P_{\mathcal{S}_\phi}$ is a projection operator, $\mathcal{L}$ can be written:

$$\mathcal{L} = \frac{N}{\pi} \left\langle P_{\mathcal{S}_\phi}[v(\omega)], P_{\mathcal{S}_\phi}[v(\omega)] \right\rangle \quad (27)$$

where $< ., . >$ denotes the usual $\mathcal{L}^2$ inner-product.

We can now write the **CRB** for the case of parametric spectrum, which we denote by $\mathbf{CRB}_{par}$, as:

$$\mathbf{CRB}(\alpha)_{par}^{-1} = \mathcal{F}(\alpha) - \frac{N}{\pi} \left\langle P_{\mathcal{S}_\phi}[v(\omega)], P_{\mathcal{S}_\phi}[v(\omega)]^T \right\rangle \quad (28)$$

Remember that $\mathcal{F}(\alpha)$ is the **FIM** for known spectrum, and can be written as (see Eq.(4)):

$$\mathcal{F}(\alpha) = \mathbf{CRB}(\alpha)_{unk}^{-1} + \frac{N}{\pi} \left\langle v(\omega), v(\omega)^T \right\rangle \quad (29)$$

where $\mathbf{CRB}(\alpha)_{unk}^{-1}$ is the **FIM** for completely unknown spectrum (see Fact 1, Eq.(15)).

Using these two equations, we can write:

$$\mathbf{CRB}(\alpha)_{par}^{-1} = \mathbf{CRB}(\alpha)_{unk}^{-1} + \mathcal{G}(\alpha, \eta) \quad (30)$$

where $\mathcal{G}(\alpha, \eta)$ is the information gain with respect to the complete unknown spectrum case:

$$\mathcal{G}(\alpha, \eta) = \frac{N}{\pi} \left\langle P_{\mathcal{S}_\phi}^{\perp}[v(\omega)], P_{\mathcal{S}_\phi}^{\perp}[v(\omega)]^T \right\rangle. \quad (31)$$

From this form of $\mathcal{G}$ we can conclude that:

Fact 3 *Let $\mathcal{G}$ be defined by (31). Then:*

$$0 \leq \mathcal{G} \leq \frac{N}{\pi} \left\langle v(\omega), v(\omega)^T \right\rangle \quad (32)$$

The proof of this fact is trivial.

The two extreme cases in Fact 3 are particularly interesting, since they correspond to the two situations previously analyzed:

(i) $\mathcal{G} = 0$, i.e., $\mathbf{FIM}(\alpha)_{par} = \mathbf{FIM}(\alpha)_{unk}$.

To have the above relation, it is necessary that the vector of functions $v(\omega)$ be colinear with its projection in $\mathcal{S}_\phi$. Using their definitions:

$$\frac{S_s(\omega)}{E(\omega)} \mathrm{Re}\left\{ \dot{h}_{\alpha_i}(\omega)^* h(\omega) \right\} = \frac{C_1}{C} \frac{\|h\|^2}{E(\omega)} \dot{S}_\eta(\omega) \quad (33)$$

where C_1 is an arbitrary constant. This equation is equivalent to:

$$\frac{\partial}{\partial \alpha_i} \left[\|h\|^2 S_s(\omega) \right] = C_* \frac{\partial}{\partial \eta} \left[\|h\|^2 S_s(\omega) \right] \quad (34)$$

where C_* is an arbitrary constant.

We get the intuitive result that no spectrum-based information retrieval is possible when the variation of the observed signal energy is the same with respect to the location and the spectral parameters.

(ii) $\mathcal{G} = \frac{N}{\pi} \left\langle v(\omega), v(\omega)^T \right\rangle$, i.e., $\mathbf{FIM}(\alpha)_{par} = \mathbf{FIM}(\alpha)$.

Again, using the same kind of arguments, we can conclude that to have no information loss, the vector $v(\omega)$ must be orthogonal to the function $\phi(\omega)$. This condition is equivalent to:

$$\int \frac{\partial}{\partial \alpha_i} \left[\|h\|^2 S_s(\omega) \right] \frac{1}{E^2(\omega)} \frac{\partial}{\partial \eta} \left[\|h\|^2 S_s(\omega) \right] d\omega = 0, \quad (35)$$

This equation means that the variation of the signal energy with respect to the spectral parameter must be orthogonal to its variation with respect to the location of the source so that uncertainty about η does not affect the estimation of α.

Arbitrary number of parameters

The case of an arbitrary number L of unknown spectral parameters can be treated in the same way as we did for $L = 1$. Define the L-dimensional vector:

$$\phi_L(\omega) = \frac{\|h(\omega)\|^2}{E(\omega)} \dot{S}_{s_\eta}(\omega)^T. \tag{36}$$

Note that $\mathcal{F}(\eta)$ is the $(L \times L)$ Gram matrix of the functions $\{\phi_{L_i}\}_{i=1}^{L}$,

$$\mathcal{F}(\eta) = \Gamma_L = \left\langle \phi_L(\omega), \phi_L(\omega)^T \right\rangle. \tag{37}$$

which we assumed non-singular.

Define the equivalent to the kernel $X(\omega_1, \omega_2)$

$$X_L(\omega_1, \omega_2) = \phi_L(\omega_1)^T \Gamma_L^{-1} \phi_L(\omega_2) \tag{38}$$

Let $\mathcal{S}_{\phi_L}$ be the space spanned by the functions $\{\phi_{L_i}\}_{i=1}^{L}$. This subspace has dimension L. Let $v(\omega)$ have the same definition (25). Then, the loss term $\mathcal{L}$ is given by:

$$\mathcal{L} = \frac{N}{\pi} \int \int v(\omega_1) X(\omega_1, \omega_2) v(\omega_2)^T d\omega_1 d\omega_2 \tag{39}$$

which is exactly the same expression as (26). The difference between the two cases lies in the dimensionality of $\mathcal{S}_\phi$, which is 1 for a single unknown parameter, and $n \leq L$ in the general case. The gain term $\mathcal{G}$ is:

$$\mathcal{G} = \frac{N}{\pi} \left\langle P^{\perp}_{\mathcal{S}_{\phi_L}} [v(\omega)], P^{\perp}_{\mathcal{S}_{\phi_L}} [v(\omega)]^T \right\rangle \tag{40}$$

where $P_{\mathcal{S}_{\phi_L}}$ is the projection operator in the orthogonal complement of $\mathcal{S}_{\phi_L}$. Fact 3 still holds for this general case.

The analysis of the two extreme cases (i) and (ii) must now be done taking into consideration the new dimensionality of $\mathcal{S}_{\phi_L}$.

(i) Now, we must require that all the components fo $v(\omega)$ belong to $\mathcal{S}_{\phi_L}$, i.e., to have $\mathcal{G} = 0$, there must exist a $(2 \times L)$ matrix T such that

$$v(\omega) = T\phi_L(\omega). \tag{41}$$

Note that if for a given subset of the unknown parameters this conditions is satisfied, then it will be trivially satisfied for the complete vector η, showing that having additional unknown parameters cannot remove ambiguities, as it should be.

Consider the discrete frequency version of the FIM. Let the number of frequencies be denoted by W. Within this framework, the kernel $X(\omega_1, \omega_2)$ instead of being defined in an infinite dimensional space, would be defined in a W-dimensional space. All the above equations hold, with a reinterpretation of the inner-product. If $W = L$, the subspace $\mathcal{S}_{\phi_L}$ has full dimension W, and its orthogonal complement is trivially equal to the zero vector. In this case, condition (41) is trivially satisfied, and $\mathcal{G} = 0$ always. The case of unknown spectrum of section 3 can be considered as the limiting case $W, L \to \infty$.

(ii) The condition for no information loss is that all the elements of $v(\omega)$ belong to $\mathcal{S}^{\perp}_{\phi_L}$:

$$\int v_i(\omega) \phi_{L_j}(\omega) d\omega = 0, \qquad \forall i, j. \tag{42}$$

Consider a fixed spectral parameter vector η_0 of dimension L. If we add another unknown parameter η_{L+1}, the subspace $\mathcal{S}^{\perp}_{\phi_{L+1}}$ is a proper subspace of $\mathcal{S}^{\perp}_{\phi_L}$, and consequently $\mathcal{G}_L \geq \mathcal{G}_{L+1}$, as it should be expected.

Fact 4 *Consider model (1) and Assumption 2. Then*

$$\mathbf{CRB}(\alpha)^{-1} = \mathbf{CRB}(\alpha)^{-1}_{unk} + \mathcal{G}(\alpha, \eta) \tag{43}$$

where $\mathbf{CRB}(\alpha)_{unk}$ *is the* **CRB** *under Assumption 1 and* $\mathcal{G}(\alpha, \eta)$ *satisfies Fact 3, and is given by:*

$$\mathcal{G} = \frac{N}{\pi} \left\langle P_{\mathcal{S}_{\phi_L}} [v(\omega)], P_{\mathcal{S}_{\phi_L}} [v(\omega)]^T \right\rangle. \tag{44}$$

where $P^{\perp}_{\mathcal{S}_{\phi_L}}[.]$ *is the projection operator into the orthogonal complement of* $\mathcal{S}_{\phi_L} = \mathrm{Span}\{\phi_{L_i}; i = 1, \ldots, L\}$.

The functions ϕ_{L_i} *and* $v(\omega)$ *where defined in (36) and (25), respectively.*

References

[1] M. João D. Rendas and José M. F. Moura. Cramér-Rao bounds for passive range and depth in a vertically homogeneous medium. In *Proc. of the 1990 Int. Conf. on Acoustic, Speech and Signal Processing, Albuquerque.* IEEE, April 1990.

[2] M. João Rendas and José M. Moura. Cramér-Rao Bound for Location Systems in Multipath Environments *submitted* to IEEE Trans. on Acoustic, Speech and Signal Processing, March 1990.

[3] P. Whittle. The Analysis of Multiple Stationary Time Series. *J. Royal Statist. Soc.*, Vol.15:125–139, 1953.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

PASSIVE TRACKING OF A MANEUVERING TARGET : AN ADAPTIVE APPROACH

S.K. Katsikas[1],[2], A.K. Leros[3], and D.G. Lainiotis[2],[3]

(1) Technological Educational Institute of Athens, Dept. of Computer Science, Ag. Spiridona st., Egaleo 12210, Athens, Greece
(2) Computer Technology Institute, Kolokotroni 3, Patras 26221, Greece
(3) University of Patras, Rion 26500, Patras, Greece

The problem of tracking a maneuvering target using bearings-only measurements (passive tracking) is examined within the framework of adaptive estimation. A class of target maneuvers are modeled and an estimation algorithm, derived using the concept of multimodel partitioning, is presented and evaluated through simulation.

1. INTRODUCTION

The area of passive target tracking has received considerable attention in the context of applications such as sonar tracking, infrared missile guidance, passive radar localization and tracking etc. One specific application of interest is bearings-only (passive) tracking in the ocean environment. In this problem an observer obtains, by means of a passive sonar device, noisy bearing measurements from a moving target vessel. Measurements are processed in real time to produce estimates of the target position and velocity.

A variety of estimation algorithms have been developed to deal with the inherent nonlinearities of the problem. These include the Extended Kalman Filter (EKF) in various forms [1]-[3], pseudolinear algorithms [4]-[6], and adaptive algorithms based on multimodel partitioning [7],[8]. All the above algorithms, however, have been developed under the assumption that the target moves on a steady course with constant speed.

In this paper we address the problem of estimating the motion of a maneuvering target observed by a single sonar device (passive tracking). A target maneuver is treated as an abrupt change in the process model, as described in the following section. This approach has been successfully used to model maneuvering aircraft in radar tracking applications [9]. Following this consideration, a new algorithm has been developed, which takes into account possible target maneuvers. The algorithm, derived by using the multimodel partitioning approach and the corresponding Adaptive Lainiotis Filter (ALF), is presented in section 3. Finally, in section 4 the algorithm is evaluated and compared to a conventional one, namely the EKF, via computer simulation of a typical tracking situation.

2. PROBLEM FORMULATION

The modified polar coordinate system is used in the mathematical formulation of the problem. It must be noted, though, that the proposed algorithm is not restricted by the choice of model. The resulting state model, given in [2], is of the form :

$$x(k+1) = f(x(k),w(k)) \quad (1a)$$
$$z(k) = Hx(k) + v(k) \quad (1b)$$

where the state vector x contains the relative target to observer position and velocity expressed in modified polar coordinates, w is a known deterministic vector expressing the observer maneuvers, f is a nonlinear functional describing the state evolution, z is the scalar bearing measurement, and v is the measurement error. {v(k)} is assumed to be a white Gaussian sequence with mean zero and variance r. The model is derived from the problem geometry under the assumption that the target moves on a steady course with constant speed.

If the assumption of a constant target velocity and heading is relaxed, the model of (1) is no more valid. The class of target maneuvers that are considered here consist of abrupt course changes. This restriction is necessary in order to maintain observability [10], and is further justified by the fact that typical vessels travel on piecewise linear trajectories. Low data rate allows us to additionally assume that a single maneuver is completed within an observation interval. Under these assumptions the observation process is adequately described by (1) at all time instants, ex-

cept for those, at which a maneuver occurs. In the latter case, the state model is given by :

$$x(k+1) = f(x(k),w(k)) + Gu \qquad (2a)$$

where the additional input term accounts for the target's change of course.

The objective is to estimate the target state (position and velocity) at any time point k based on noisy bearing measurements up to and including time k. The problem is differentiated from classical bearings-only tracking by the uncertainty as to which of the models (1) and (2) actually represents the evolution of the target state at any given instant, and by the additional uncertainty introduced by the unknown input vector u in (2).

3. THE PROPOSED ADAPTIVE ESTIMATOR

Uncertainty in the target model of (1) and (2) can be summarized by the parameter vector

$$\theta(k) = [u(1)\ u(2)\ \cdots\ u(k)]^T \qquad (3)$$

Direct application of the ALF [11]-[15] yields the following estimate of the target state, given the set of measurements $Z_k = \{z(1),z(2),\ldots,z(k)\}$:

$$\hat{x}(k/k) = \int_{\Omega} \hat{x}(k/k;\theta)p(\theta/k)d\theta \qquad (4)$$

where $\hat{x}(k/k;\theta)$ are the state estimates conditioned on θ, Ω is the sample space of θ (a subset of R^{2k}) and $p(\theta/k)$ is the a posteriori pdf of θ given by :

$$p(\theta/k) = \frac{L(k/k;\theta)p(\theta/k-1)}{\int_{\Omega} L(k/k;\theta)p(\theta/k-1)d\theta} \qquad (5)$$

$$L(k/k;\theta) = \exp\{-1/2\|\tilde{z}(k/k-1;\theta)\|^2 P_*^{-1}(k/k-1)\} \qquad (6)$$

P_* is the measurement error covariance matrix and $\tilde{z}(k/k-1;\theta)$ are the conditional innovations sequences. Each of the parameter-conditional estimates $\hat{x}(k/k;\theta)$ is obtained by a nonlinear filter, such as an EKF, matched to a model containing the corresponding value of θ. A particular model is described by (1) at all time, except for certain time points, when u is nonzero.

Since the parameter space is continuous, an approximation, such as discretization [16], is required. Still, it can be seen from (3) that the parameter space dimensionality, and consequently the number of models and corresponding filters, grows exponentially with time. Because of the enormous computational load involved, the algorithm is not possible to implement in the above form.

Simplification is possible under the assumption that the target cannot execute more than one maneuver during a specific time interval, say N times the observation period. This assumption is not restrictive in the case under study, since typical vessels do not perform frequent maneuvers. The parameter θ is now given by $\theta(k) = [T_i\ u(T_i)]$, where $k-N+1 \le T_i \le k$. The case of no maneuver occurrence within the interval examined is denoted by $T_i=0$. Thus the number of filters is reduced to a constant depending on the quantization of u.

Further reduction is achieved by considering a single value of u instead of discretizing the sample space. This value is chosen as the maximum likelihood estimate of u conditioned on the hypothesis that a maneuver did occur at time T_i, i.e. for each T_i, $\hat{u}_i$ is obtained by setting $\partial p(T_i,u/k)/\partial u = 0$.

Because of the nonlinear functional f involved in the state model, it is not possible to derive a closed-form expression for $\hat{u}_i$, unless an approximation of f is used. If the conditional filters employed are EKF's, such an approximation, namely linearization of f around the point $\hat{x}(1-1/1-1;T_i,u)$ is inherent in the calculation of the conditional estimates and no additional computation is required. Furthermore, the conditional estimates $\hat{x}(k/k;T_i,u)$ can be expressed in terms of the nominal filter estimate $\hat{x}(k/k;0)$, i.e. the one conditioned on no maneuver occurrence, as shown in [17]. There is therefore no need to implement separately the N conditional filters. The final (unconditioned) estimate is given by the following equations [17] :

$$\hat{x}(k/k) = \hat{x}(k/k;0) + \sum_{i=1}^{N} C(k,T_i)G\hat{u}_i p(T_i,k)$$

$$\hat{u}_i = \left[\sum_{l=i+1}^{k} A(l,T_i)\right]^{-1} \sum_{l=i+1}^{k} B(l,T_i)\tilde{z}(l/l-1;0)$$

$$C(k,i+1) = \prod_{l=i+1}^{k} M(l)$$

$$A(l,T_i) = \|H\Phi(l-1)M(l-1,T_i)G\|^2 P_*^{-1}(l/l-1)$$

$$B(l,T_i) = [H\Phi(l-1)M(l-1,T_i)G]^T P_*^{-1}(l/l-1)$$

$$p(T_i/k) = \frac{\exp\{-1/2\|\tilde{z}(k/k-1;0)\|^2 P_*^{-1}(k/k-1) + 1/2\|B(k,T_i)\tilde{z}(k/k-1;0)\|^2 A(k,T_i)^{-1}\}}{\sum_{i=1}^{N} \exp\{-1/2\|\tilde{z}(k/k-1;0)\|^2 P_*^{-1}(k/k-1) + 1/2\|B(k,T_i)\tilde{z}(k/k-1;0)\|^2 A(k,T_i)^{-1}\}}$$

$$M(1) = [I-K(1)H]\Phi(1-1)$$

The quantities $\hat{x}(k/k;0)$, $K(k)$, $\Phi(k)$, $\tilde{z}(k/k-1;0)$, and $Pz(k/k-1)$ are obtained from a nominal EKF based on model (1).

4. SIMULATION RESULTS

In order to assess the proposed adaptive algorithm's performance, the following tracking scenario [2],[5] was simulated: The observer moves at a constant speed of 14.14 m/sec on a piecewise linear trajectory, periodically changing course to 45° every 4+17k min and to 135° every 12.5+17k min. The target moves at a constant speed of 10 m/sec and executes a single maneuver by changing course from 90° to 0°. The sampling interval is 20 sec. A number of runs were carried out for various values of the initial range R(0) and the measurement noise standard deviation σ.

The proposed filter was tested against a standard nonadaptive algorithm, namely the Extended Kalman Filter (EKF) formulated in modified polar coordinates. The latter has shown very satisfactory performance in the non-maneuvering target tracking problem. Representative results are shown in figs. 1-4. It is evident that the EKF presents unacceptable delays in responding to the target maneuver. In the most favorable case, i.e. short initial range and high SNR, the modified polar EKF requires about 25 min to converge to the true target state in comparison to about 7 min required by the adaptive modified polar filter.

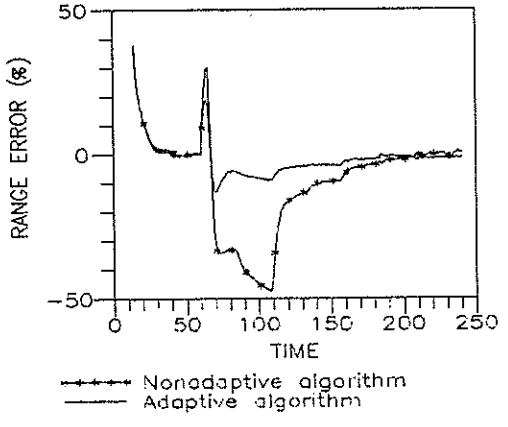


Fig. 1. Error in range, R(0)=2.5km, σ=1°

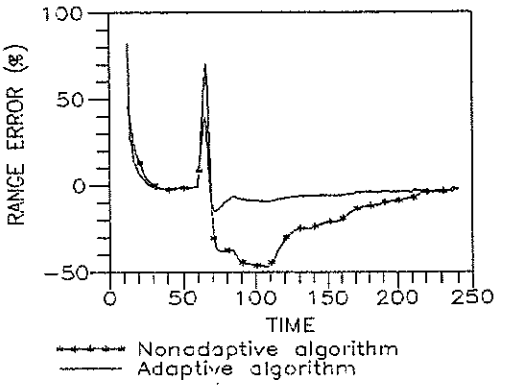


Fig. 2. Error in range, R(0)=2.5km, σ=4°

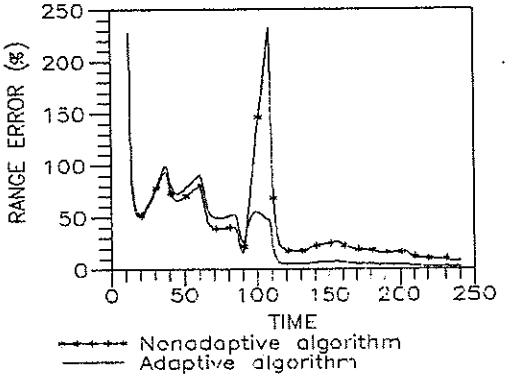


Fig. 3. Error in range, R(0)=25km, σ=1°

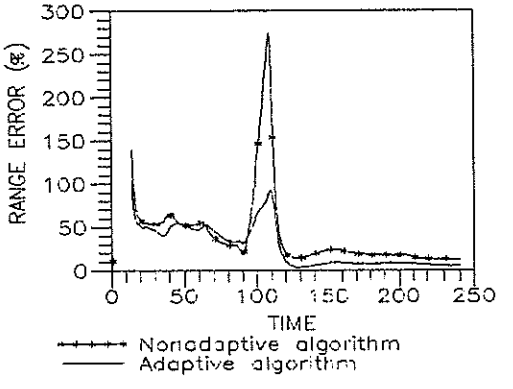


Fig. 4. Error in range, R(0)=25km, σ=4°

It can be seen that long initial range and high measurement noise levels have the effect of delaying the convergence of the adaptive algorithm. Still, the convergence rate is much faster than that achieved by the non-adaptive algorithm (note the change of scale in the error axis). Furthermore, the error overshoot following the maneuver is far less pronounced in the adaptive filter. The above results demonstrate the efficiency of the proposed filter in coping with a target maneuver in comparison to traditional nonadaptive techniques.

REFERENCES

[1] Weiss, H., and Moore, J.B.,"Improved EKF design for passive tracking", IEEE Trans. Autom. Control, vol. AC-25, no. 4, Aug. 1980.

[2] Aidala, V.J., and Hammel, S.E., "Utilization of modified polar coordinates for bearings-only tracking", IEEE Trans. Autom. Control, vol. AC-28, no. 3, March 1983.

[3] Song, T.L., and Speyer, J.S., "A stochastic analysis of a modified gain EKF with application to estimation with bearings-only measurements", IEEE Trans. Autom. Control, vol. AC-30, no. 10, Oct. 1985.

[4] Lindgren, A.G., and Gong, K.F., "Position and velocity estimation via bearing observations", IEEE Trans. Aerosp. Electron. Syst., vol. AES-14, no.4, July 1978.

[5] Aidala, V.J., "Kalman filter behavior in bearings-only tracking applications", IEEE Trans. Aerosp. Electron. Syst., vol. AES-15, no. 1, Jan. 1979.

[6] Aidala, V.J., and Nardone, S.C., "Biased estimation properties of the pseudolinear tracking filter", IEEE Trans. Aerosp. Electron. Syst., vol. AES-18, no. 4, July 1982.

[7] Petridis, V., "A method for bearings only velocity and position estimation", IEEE Trans. Autom. Control, vol. AC-26, no. 2, April 1981.

[8] Watanabe, K., "Application of a pseudolinear partitioned filter to passive vehicle tracking", Intern. J. of Syst. Sci., vol. 15, no. 9, Sept. 1984.

[9] Chan, Y.T., Hu, A.G.C., and Plant, J.B., "A Kalman filter based tracking scheme with input estimation", IEEE Trans. Aerosp. Electr. Syst., Vol. AES-15, no. 2, March 1979.

[10] Nardone, S.C., and Aidala, V.J., "Observability criteria for bearings only target motion analysis", IEEE Trans. Aerosp. Electron. Syst., vol. AES-17, no. 2, March 1981.

[11] Lainiotis, D.G., "Joint detection, estimation, and system identification", Info. and Control J., vol. 19, no. 8, 1971.

[12] Lainiotis, D.G., "Optimal adaptive estimation: structure and parameter adaptation", IEEE Trans. on Auto. Control, vol. AC-16, no. 2, 1971.

[13] Lainiotis, D.G., "Partitioning: A unifying framework for adaptive systems, I: Estimation", Proc. of the IEEE, vol. 64, no. 8, 1976.

[14] Lainiotis, D.G., "A unifying framework for estimation: Partitioned algorithms", Advances in Systems Theory, N.Y.: Springer Verlag, 1978.

[15] Lainiotis, D.G., "Partitioning filters", J. of Info. Sci., vol.17, 1979.

[16] Shengbush, R.L. and Lainiotis, D.G. "Simplified parameter quantization procedure for adaptive estimation", IEEE Trans. on Autom. Control, vol. AC-14, no. 8, Aug. 1969.

[17] Katsikas, S.K., Leros, A.K., and Lainiotis, D.G., "An adaptive estimation algorithm for nonlinear systems with model uncertainty", submitted for publication in the IEEE Trans. on Acoustics, Speech, and Signal Processing.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Application of Maximum Likelihood Estimation to Passive Sonar Tracking*

JOOST H. de VLIEGER and ROBERT H.J. GMELIG MEYLING+

Physics and Electronics Laboratory FEL-TNO, P.O. Box 96864 2509 JG The Hague, The Netherlands

The tracking problem of unknown marine platforms using bearing and frequency measurements is often poorly conditioned. Especially in long-range scenarios inaccurate and noisy measurements form an additional problem. The Maximum Likelihood estimates of the position and velocity parameters are obtained by using a Newton-type method with suitable numerical properties.

1. INTRODUCTION

The aim of Target Motion Analysis (TMA) is to estimate position, course and speed of a (maneuvering) marine platform, given a time sequence of sonar measurements. In the long-range case, passive sonar measurements are obtained by a ship equiped with a towed array. Generally, the data consist of bearing and frequency values as a result of processing the hydrophone signals. The frequency measurements contain an unknown doppler shift. Long-range TMA problems are often poorly conditioned due to low rates in bearing and radial velocity of the unknown platform as well as inaccurate and noisy measurements. This is the main reason for using a non-recursive estimation method with favourable numerical properties. Maximum Likelihood (ML) estimation is an efficient solution method for TMA. By using a proper numeric optimization method to obtain an ML estimate the disadvantages of Kalman filters [1, 2] can be overcome at the expense of some more computational effort.

This paper is focussed on the description and analysis of the ML method related to long-range scenarios with low signal-to-noise ratios. Attention is paid to the conditioning aspects of the TMA problem. A modified Newton scheme is applied to ensure a fast convergence and bounds on the estimated parameters are incorporated by using an active set strategy [4]. The effect of high noise levels on the ML estimates is shown by means of Monte Carlo simulation results.

2. PROBLEM FORMULATION

The position of an observed target ship (TS) and the own ship (OS) are denoted in Cartesian coordinates by (X_{TS}, Y_{TS}) and (X_{OS}, Y_{OS}). The TS is assumed to move according to a piecewise linear track with uniform motion. Each piece of the track is referred to as a leg. For m legs and N observations the position of the TS at time t_k is related to the position at t_N by

$$X_{TS}(t_k) = X_{TS}(t_N) - \sum_{i=1}^{m} T_i(t_k) V_{Xi}$$

$$Y_{TS}(t_k) = Y_{TS}(t_N) - \sum_{i=1}^{m} T_i(t_k) V_{Yi} \qquad (1)$$

$$k = 1, \ldots, N$$

with

$$V_{Xi} = \dot{X}_{TS}(t_k) \qquad \tau_{i-1} \leq t_k \leq \tau_i$$

$$V_{Yi} = \dot{Y}_{TS}(t_k) \qquad \tau_{i-1} \leq t_k \leq \tau_i$$

where $\{t_k \mid t_1 \leq t_2 \leq \ldots \leq t_k \leq \ldots \leq t_N\}$ denotes the set of observation times and τ_{i-1}, τ_i indicate the beginning and end time of leg i. The target maneuver times τ_i are assumed to be known, e.g. as a result of a maneuver detection procedure. The time periods $T_i(t_k)$ are defined as

$$T_i(t_k) = \max(\tau_i - t_k, 0) - \max(\tau_{i-1} - t_k, 0) \qquad (2)$$

The own ship position and velocity $(X_{OS}(t_k), Y_{OS}(t_k), \dot{X}_{OS}(t_k), \dot{Y}_{OS}(t_k))$ are assumed to be known for all k.
The target parameters in Cartesian coordinates are denoted in vector form by

*This study was supported by the Royal Netherlands Navy (assignment number A87KM172).
+Currently with the Royal Dutch/Shell Exploration and Production Laboratory, Rijswijk, The Netherlands.

$$y(t_k) = [X_{TS}(t_k)\ Y_{TS}(t_k)\ V_{X1}\ V_{Y1} \cdots V_{Xm}\ V_{Ym}\ F_o]^T \quad (3)$$

where the additional parameter F_o represents the emitted source frequency of the target platform. Obviously equation (1) can be rewritten into matrix form as follows:

$$y(t_k) = \Phi(N,k)y(t_N) \quad (4)$$

where $\Phi(N,k)$ denotes the transition matrix from t_N to t_k (backwards in time).

In order to describe the measurement model, polar position and velocity coordinates are introduced:

$$x(t_k) = [B(t_k)\ R(t_k)\ C_1\ V_1 \cdots C_m\ V_m\ F_o]^T \quad (5)$$

with $B(t_k)$ the bearing angle relative to the Y-axis (North), $R(t_k)$ the range, C_i, V_i the absolute course and speed of the platform at leg i. The transition from $x(t_N)$ to $x(t_k)$ is described by performing nonlinear transformations from polar to Cartesian coordinates and vice versa.

$$x(t_k) = f^{(-1)}(\Phi(N,k)f(x(t_N),t_N),\ t_k) \quad (6)$$

where $y(t_k) = f(x(t_k),t_k)$ denotes the polar-to-cartesian transformation and $f^{(-1)}$ denotes the inverse transformation.

The set of measured data generally consists of bearing angles and frequency data which result either from a direct acoustical path (DP) or from bottom bounce reflections (BB). The acoustical path is assumed to be known, e.g. by using an acoustical propagation prediction model. Here we only consider the DP case for simplicity. However the model can be easily extended for the BB case. If BB measurements are introduced, the line of sight is projected onto a horizontal plane. In that case the measurement equations also depend on the range and the sea depth. The measured DP bearing and frequency are now given by

$$\begin{aligned} z_B(t_k) &= B(t_k) + v_B(t_k) \\ F(t_k) &= F_o - \frac{\dot{R}(t_k)}{c} F_o \\ z_F(t_k) &= F(t_k) + v_F(t_k) \end{aligned} \quad (7)$$

where $v_B(t_k) \sim N(0,\sigma_B^2)$, $v_F(t_k) \sim N(0,\sigma_F^2)$ represent Gaussian noise and c is the sound velocity. Additional measurements about range, course and speed are denoted by

$$\begin{aligned} z_R(t_k) &= R(t_k) + v_R(t_k) \\ z_C(t_k) &= C_i + v_C(t_k) \\ z_V(t_k) &= V_i + v_V(t_k) \end{aligned} \quad (\tau_{i-1} \le t_k \le \tau_i) \quad (8)$$

where $v_R(t_k) \sim N(0,\sigma_R^2)$, $v_C(t_k) \sim N(0,\sigma_C^2)$ and $v_V(t_k) \sim N(0,\sigma_V^2)$.

3. MAXIMUM LIKELIHOOD ESTIMATION

In this tracking problem the purpose is to estimate the state vector $x(t_N)$ from a set of measurements Z_N which may contain all types of measurements as described above. The general form of a measurement is denoted by

$$\begin{aligned} z_d(t_k) &= d(x(t_N),t_k) + v_d(t_k) \\ Z_N &= \{z_d(t_k),\ k=1,\ldots,N\} \end{aligned} \quad (9)$$

where $d(x(t_N),t_k)$ denotes one of the variables $B(t_k)$, $F(t_k)$, $R(t_k)$, C_i, V_i.

The Maximum Likelihood (ML) estimate of the target parameters $x(t_N)$ is obtained by minimizing the negative log-likelihood function $L(Z_N,x)$ which is obtained from the conditional pdf $p(Z_N|x(t_N))$. The following nonlinear least squares problem results:

$$\text{minimize } L(Z_N,x) \quad (10)$$
$$x \in R^n$$

subject to feasibility constraints

$$\ell \le x \le u\ , \qquad \ell,\ u \in R^n. \quad (11)$$

with

$$L(Z_N,x) = \frac{1}{2}\, r(x)^T r(x) \quad (12)$$

where $r(x)$ represents the N-vector of weighted residuals

$$r_k(x) = \frac{d(x,t_k) - z_d(t_k)}{\sigma_d},\ k = 1,\ \ldots,N \quad (13)$$

In practical TMA problems, sonar performance limits, local acoustic conditions, maximum target speed, geometric constraints or other a priori knowledge can often be used to bound the target parameters in advance. Moreover, the constraints (11) prevent the optimization method to search for local minima related to nonrealistic solutions.

4. MODIFIED NEWTON METHODS

Newton-type methods are well-established iterative techniques designed to solve general

minimization problems [4]. Newton methods use the first and second order derivatives (or their approximations) of $L(Z_N,x)$. The Jacobian, gradient and Hessian are denoted by

$$J(x) = \nabla_x r(x)$$
$$g(x)=J(x)^T r(x) \qquad (14)$$
$$H(x)=J(x)^T J(x)+Q(x)$$

with

$$Q(x) = \sum_{k=1}^{N} r_k(x)\, \nabla_x^2 r_k(x)$$

Let $x^{[k]}$ represent a *feasible point* at iteration k, i.e. constraints (11) are satisfied. The active set strategy [4] is a convenient method to incorporate bounds on the target variables. A sequence $\{x^{[k]}\}$ will converge to a (constrained) local minimum x^* of the function $L(Z_N,x)$, provided that $x^{[0]}$ is sufficiently close to x^* and $\tilde{H}(x^*)$ is positive definite. Here, the *projected Hessian* $\tilde{H}(x^*)$ is composed of those rows and columns i of $H(x^*)$ for which $\ell_i < x_i^* < u_i$. The search vector $s^{[k]}$ is defined as the solution of a linear system

$$\mathcal{H}(x^{[k]})s^{[k]} = -g^{[k]}, \qquad (15)$$

with $\mathcal{H} = H$ for a Newton-method (N) or $\mathcal{H} = J^TJ$ for a Gauss-Newton method (GN). Practical implementations of Newton-type methods usually consist of a combination of N/GN search vectors such that N-directions are only used in the vicinity of the minimum. Once $s^{[k]}$ has been fixed, the steplength is chosen as the value $\alpha^{[k]}$ minimizing $L(x^{[k]}+\alpha^{[k]}s^{[k]})$. The active set strategy ensures that those components of $s^{[k]}$ corresponding to the active bounds ($x_i^{[k]} = \ell_i$ or $x_i^{[k]} = u_i$) are set to zero.

Unfortunately, traditional Newton-type methods applied to TMA may converge very slowly or may even fail to converge due to ill-conditioning of matrix $\mathcal{H}$ in (15). Here, we discuss modified Newton-type methods [3] which use the Singular Value Decomposition (SVD) of matrix J in order to handle (almost) linear dependencies in the model [5]. It can be shown that the numerical properties of J are fully determined by the geometry of the TS and OS tracks and the number of measurements and therefore connected to the *observability of* x. By using SVD the search direction is modified such that the cost function is minimized over a subspace of target parameters. In this subspace, the minimization problem is uniquely solvable and well-conditioned. In addition, the explicit formation of J^TJ is avoided.

Let $J = U\Sigma V^T$ be the SVD of the Jacobian matrix $J \in \mathbb{R}^{N\times n}$ ($N \geq n$). The matrices $U \in \mathbb{R}^{N\times n}$, $V \in \mathbb{R}^{n\times n}$ are orthonormal and Σ is an n×n-diagonal matrix containing the *singular values* of J in descending order ($\sigma_1 \geq \sigma_2 \geq \ldots \sigma_n \geq 0$). The basic idea is to divide the set $\{\sigma_1, \ldots,\sigma_n\}$ into a set $\{\sigma_1, \ldots,\sigma_\rho\}$ of "large" and a set $\{\sigma_{\rho+1},\ldots,\sigma_n\}$ of "small" singular values [3]. Parameter ρ (the *grade* of matrix J) could for instance be chosen such that

$$\sigma_\rho/\sigma_1 \geq \varepsilon, \qquad \sigma_{\rho+1}/\sigma_1 < \varepsilon \qquad (16)$$

for a suitable small number ε. Taking only dominant singular values into account leads to a *Graded Gauss-Newton* (GGN) method. In addition the search vector can be augmented by including effects due to smaller singular values which leads to the *Corrected Graded Gauss-Newton* (CGGN) method [3].

In practical long-range TMA computations the ratio σ_1/σ_n may have order 10^4 (or larger). Using a combination of GGN and CGGN (similar to GN/N) ensures fast convergence even for difficult cases. GGN-steps are taken as long as the cost function is sufficiently reduced. Near a minimum, however, the CGGN-method has superior convergence properties.

Calculating the SVD of an Nxn matrix requires $O(2Nn^2+4n^3)$ arithmetic operations [5]. On the other hand, the SVD is also useful to determine the accuracy and validity of TMA solutions. Whenever the SVD of J has been computed, the *estimated covariance matrix*

$$C = (J^TJ)^{-1} = V\Sigma^{-2}V^T, \qquad (17)$$

is readily available.

5. MONTE CARLO SIMULATION RESULTS

The performance of the GGN/CGGN method for TMA is shown using artificial data. Every simulation is repeated 500 times with different (Gaussian) noise sequences for each run. Only those TMA solutions are recorded which satisfy proper convergence and optimality conditions [3]. M measurements are produced over a period of 1 hour. Every sample time t_k one bearing and one frequency value is generated (i.e. N = 2M). All frequency data belong to a source frequency F_o = 500.0 Hz. Disturbances due to wind, current and waves are simulated by adding zero-mean Gaussian noise to the OS course and speed (σ_{Cos}=1.0°, σ_{Vos}=0.2 knots).

The first scenario is a one-leg problem, where the own ship and the target move in opposite directions. Specifically, the OS direction is 0° (North) with speed 10 knots, whereas the TS course and speed are 180.0° and 6.0 knots. At

the end of the scenario, the exact values are B = 118.1° and R = 17.0 NMI (nautical miles). This example is characterized by high data rates (M = 30) but also by a high noise level (σ_B = 5.0°, σ_F = 0.2 Hz). Figure 1 shows a composite scatterplot with estimated TS parameters (B, R, C, V, F_o) obtained for 500 runs. The ML estimates form clusters centered at the exact TS parameters. Note the strong correlation between C and F estimates (correlation coefficient -0.9625) and between R and V (corr. coeff. 0.8724), while the C and V estimates are almost linearly independent (corr. coeff. 0.0001). We emphasize that these dependencies between target parameters were the main motivation for using modified Newton-type methods.

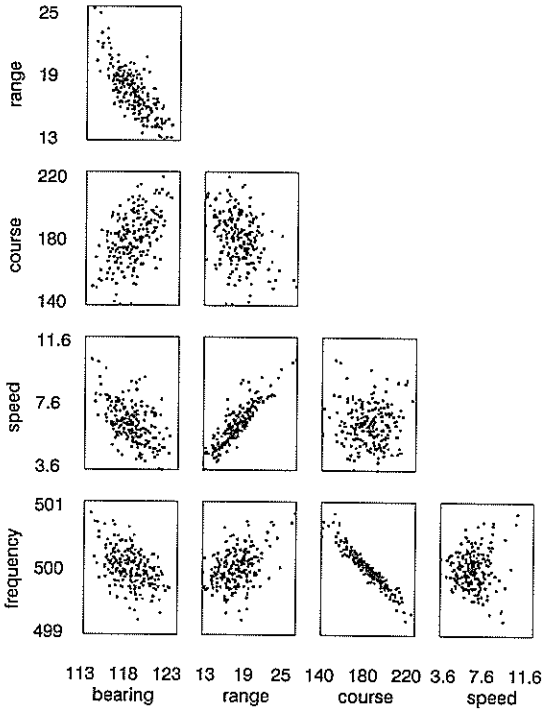


Figure 1.

As a final example, consider a two-leg scenario where the TS changes course after 30 minutes from 90.0° to 135.0°, at fixed speed of 10.0 knots. The OS course and speed are 270.0° and 5.0 knots. At the end of the scenario, the exact values are B = 0.0° and R = 20.0 NMI. The effect of data availability (i.e. M = 10, 20, 30) on the TMA solution quality is shown in Table 1. To avoid complications due to poor data quality, low noise levels (σ_B = 0.5°, σ_F = 0.02 Hz) are used. A two-leg scenario requires the estimation of 7 parameters (i.e. B, R, C_1, V_1, C_2, V_2, F_o). Even in the presence of a target maneuver and an unfavourable geometry, the ML estimated are quite accurate. The standard deviations $\bar{\sigma}$ in the estimates (as recorded over 500 runs) indicate that TMA solution quality is expected to improve whenever more B and F data become available.

	M=10	M=20	M=30	Exact	
$\bar{B}$	0.0	0.1	0.1	0.0	deg.
$\bar{R}$	20.3	20.0	20.1	20.0	NMI
$\bar{C}_1$	86.2	86.2	87.9	90.0	deg.
$\bar{V}_1$	10.2	10.0	9.8	10.0	kn.
$\bar{C}_2$	130.6	131.6	133.1	135.0	deg.
$\bar{V}_2$	10.0	9.8	9.8	10.0	kn.
$\bar{F}_o$	500.1	500.1	500.0	500.0	Hz.
$\bar{\sigma}_B$	0.5	0.3	0.2		deg.
$\bar{\sigma}_R$	2.0	1.4	0.9		NMI
$\bar{\sigma}_{C_1}$	16.5	13.9	7.5		deg.
$\bar{\sigma}_{V_1}$	1.5	1.3	0.7		kn.
$\bar{\sigma}_{C_2}$	16.4	11.7	5.8		deg.
$\bar{\sigma}_{V_2}$	2.0	1.9	1.1		kn.
$\bar{\sigma}_{F_o}$	0.5	0.4	0.2		Hz.

Table 1.

REFERENCES

[1] Aidala, V.J. "Kalman Filter Behaviour in Bearings-only Tracking Applications", *IEEE Trans. Aerosp. Electron. Syst.*, vol AES-15, No. 1, Januari 1979.

[2] Aidala, V.J. and S.E. Hammel, "Utilization of Modified Polar Coordinates for Bearings Measurement", *IEEE Trans. Automat. Contr.*, Vol. AC-28, No. 3, March 1983.

[3] P.E. Gill and W. Murray, Algorithms for the Solution of the Nonlinear Least-Squares Problem", *SIAM J. Numer. Anal.*, 15, pp. 977-992, 1978.

[4] P.E. Gill, W. Murray and M.H. Wright, "Practical Optimization", Academic Press, New York, 1981.

[5] G.H. Golub and C. Reinsch, "Singular Value Decomposition and Least Squares Solutions", in Linear Algebra, J.H. Wilkinson and C. Reinsch, Eds., Springer, Berlin, pp. 134-151, 1971.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A SUBOPTIMAL HIERARCHICAL APPROACH TO BEARINGS-ONLY TRACKING AND TRACK TO TRACK ASSOCIATION

J.M. PASSERIEUX, D. PILLON
Thomson Sintra ASM, Chemin des Travails BP 53
06801 Cagnes sur Mer Cedex FRANCE

Statistical procedures for Bearings-Only Track to Track Association in passive sonar are adressed. Usually this task was performed directly on bearing measurements. Here a simpler hierarchical approach is presented : first bearings tracks are processed in order to obtain local kinematic source parameters estimates, then Track to Track Association (TTA) is performed by using these local estimates. Without ownship maneuver a complete local estimation is impossible (unobservable situation) ; nevertheless the method still applies to association of estimated sets of possible source parameters.

INTRODUCTION.

In modern passive sonar, current trend is to use several sensors working in different frequency bands and if possible set on several widely separated platforms in order to perform acoustic surveillance on large area. Each sensor is equipped with its local reception chain and delivers bearing tracks related to every source it has detected. In this context Track to Track Association (TTA) has a great importance to obtain a synthetic view of the situation (a single source can be detected by several sensors) and an accurate source location (via triangulation and Target Motion Analysis (TMA)).

Due to parallax between sensors and possibly not time overlapping of bearing tracks, TTA cannot be performed directly on bearings measurements : it is necessary to use a model for source motion and to associate tracks by using kinematic source parameters instead of bearings. In this context classical association and localization technics ([4],[6]), which rely on Maximum Likelihood Estimation and Likelihood Ratio Tests, use whole bearing tracks which requires heavy computations. The aim of this paper is to develop a more practical approach (already introduced in [3] and [5]) which consists in performing TTA on local source motion parameters estimates instead of whole bearing tracks.

1. GEOMETRICAL ASPECTS

1.1 Basic Assumptions

Let's consider two sets of bearings measurements (i.e two bearings tracks) :

$$\underline{\Theta}_A^m = \{\theta_A^m(t_i), i=1,n_A\} \quad (\text{resp } \underline{\Theta}_B^m, t_j, n_B) \quad (1)$$

that have been collected by two different sensors A and B and/or over possibly not time overlapping intervals.

Positions and speeds of sensors A and B are assumed to be perfectly known at every time. Let's denote the 4-components vectors of these positions and speeds by

$$\underline{X}_A(t) = (x_A(t)\ y_A(t)\ \dot{x}_A(t)\ \dot{y}_A(t))'(\text{resp.B}) \quad (2)$$

Surrounding sources are assumed to move along Straight Lines with Constant Velocities (SLCV). So their trajectories are characterized by state vectors (with reference time t^*)

$$\underline{X} = (x(t^*)\ y(t^*)\ \dot{x}\ \dot{y})' \quad (3)$$

Bearings measurements $\theta_A^m(t_i)$ (resp. $\theta_B^m(t_j)$) are given by

$$\theta_A^m(t_i) = \tan^{-1}\left[\frac{x(t^*)+(t_i-t^*)\dot{x}-x_A(t_i)}{y(t^*)+(t_i-t^*)\dot{y}-y_A(t_i)}\right] + \eta_A(t_i) \quad (4)$$

where measurement errors $\eta_A(t_i)$ (resp $\eta_B(t_j)$) are assumed to be uncorrelated, zero-mean and gaussian with known standard-deviations σ_A and σ_B. Moreover it is also assumed that all bearing measurements in a same track are relative to the same source and that bearing tracks do not contain false alarm.

1.2 Bearings-Only Tracking - Observability.

With a single track it is well known [1,2] that an ownship maneuver is a necessary (but not sufficient) condition to make the trajectory of a SLCV source observable (ensure uniqueness of solution). At the opposite, in unobservable situations (one track , no ownship maneuver), the observer can only determine a set of possible homothetic source trajectories ([2]).

1.3 Geometrical ambiguity.

The simpler association problem is when two bearing tracks $\underline{\Theta}_A^m$ and $\underline{\Theta}_B^m$ have been collected. Then we have to decide between the two following hypotheses :

H : both tracks $\underline{\Theta}_A^m$ and $\underline{\Theta}_B^m$ are relative to the same source $\underline{X}$, or

$\bar{H}$: two different sources $\underline{X}'$ and $\underline{X}''$ correspond to tracks $\underline{\Theta}_A^m$ and $\underline{\Theta}_B^m$.

Particular geometrical situations, called ambiguous situations [6], may occur whenever source trajectories are unobservable from a single track. Examples of such situations are given figures 1-a and 1-b.

In these typical situations bearings collected on array A (resp. B) for $\underline{X}$ and $\underline{X}'$ (resp $\underline{X}''$) remain equal. Therefore using only bearing measurements it is impossible to decide between H and $\bar{H}$.

2. OPTIMAL METHODS FOR BOT AND TTA

2.1 Recall about estimation in BOT

The purpose of Bearings Only Tracking methods is to determine the source trajectory $\underline{X}$ from a set of bearings measurements (for instance $\underline{\Theta}_A^m$). With gaussian measurement errors [4], Maximum Likelihood Estimators $\hat{\underline{X}}$ of $\underline{X}$ minimizes the following criterion

$$L_A(\underline{X}) = \sigma_A^{-2} \, ||(\underline{\Theta}_A^m - \underline{\Theta}_A(\underline{X}))||^2 \qquad (5)$$

(with two tracks, criterion (5) becomes $L_{AB} = L_A + L_B$). When track duration includes an ownship maneuver, $\hat{\underline{X}}$ can be determined by several iterative algorithms [3]. At the opposite, in unobservable situations, it is impossible to estimate the whole vector $\underline{X}$ without additional information. From (4), we see that, in the case of a SLCV array A, bearings $\theta_A(t)$ can be rewritten :

$$\theta_A(t) = \tan^{-1}\left[\frac{1 + (t_i - t^*)\dot{\Delta x}/\Delta x}{\Delta y/\Delta x + (t_i - t^*)\dot{\Delta y}/\Delta x}\right] \qquad (6)$$

where $(\Delta x \; \Delta y \; \dot{\Delta x} \; \dot{\Delta y})' = \underline{X} - \underline{X}_A$ is the 4-components vector of relative source kinematic parameters (with respect to array A). So bearings history $\theta_A(t;\underline{X})$ only depends of three ratios $\Delta y/\Delta x$, $\dot{\Delta x}/\Delta x$, $\dot{\Delta y}/\Delta x$ which are obviously the only identifiable parameters. Therefore a simple approach to BOT in unobservable situation is as follows :

- first set one component of $\hat{\underline{X}}$ to an arbitrary value, for instance x to x_0, and estimate the three others, $\underline{X}^3$, using same algorithms as in the observable case,
- then deduce the set of possible solutions $\hat{\underline{X}}$ from $\underline{X}^3$ by homotheties in the relative array A axes.

To perform these homotheties let's partition $\underline{X}(x_0)$, $\underline{X}(x)$ or $\underline{X}_A$ as follows :

$$\hat{\underline{X}}(x) = \begin{bmatrix} x \\ \cdots \\ \underline{X}^3 \end{bmatrix} \qquad \hat{\underline{X}}_A = \begin{bmatrix} x_A \\ \cdots \\ \underline{X}_A^3 \end{bmatrix} \qquad (7)$$

and define $\hat{\underline{Y}} = ((\underline{X}^3(x_0) - \underline{X}_A^3)/(x_0 - x_A)$ (8)

Then $\hat{\underline{X}}^3(x) = \underline{X}_A^3 + (x - x_A)\,\hat{\underline{Y}}$ (9)

In BOT observable situations, ML estimators such as those of [3] have been proved to be unbiased, statistically efficient and approximately gaussian. In other words the covariance matrix of the estimation errors is given by the inverse of the Fisher Information Matrix :

$$F(\underline{X}) = A(\underline{X})' \, \Sigma^{-1} \, A(\underline{X}) \qquad (10)$$

(where $A(\underline{X}) = \partial\underline{\Theta}/\partial\underline{X}$ is the Jacobian matrix). This property still stands in unobservable situations. The only difference is in the size of matrices and vectors. As we now just estimate 3-components vectors $\underline{X}^3$ or $\underline{Y}$ we have to handle nx3 Jacobian matrix $A(\underline{X}^3)$ and 3x3 Fisher Information matrix $F(\underline{X}^3)$.

2.2 Bi-Hypotheses Track to Track Association.

The association test enables to decide wether the two bearing tracks $\underline{\Theta}_A^m$ and $\underline{\Theta}_B^m$ are related to a single source (H) or two different sources ($\bar{H}$). Each hypothesis is composite since it depends on source trajectories which have to be estimated. Therefore a generalized Likelihood Ratio Test ([7]) is used :

$$\Lambda = \frac{\underset{\underline{X}}{\text{SUP}} \; p(\underline{\Theta}_A^m, \underline{\Theta}_B^m / \underline{X}; H)}{\underset{\underline{X}', \underline{X}''}{\text{SUP}} \; p(\underline{\Theta}_A^m, \underline{\Theta}_B^m / \underline{X}', \underline{X}''; \bar{H})} \underset{\bar{H}}{\overset{H}{\gtrless}} \eta \qquad (11)$$

Under gaussian assumptions, by taking $-2.\log\Lambda$ instead of Λ, this test becomes [6] :

$$-2.\log\Lambda = L_{AB}(\hat{\underline{X}}) - L_A(\hat{\underline{X}}') - L_B(\hat{\underline{X}}'') \underset{\bar{H}}{\overset{H}{\lessgtr}} \eta \qquad (12)$$

where $\hat{\underline{X}}'$, $\hat{\underline{X}}''$, $\hat{\underline{X}}$ are Maximum Likelihood estimates of kinematic parameters from measurements $\underline{\Theta}_A^m$, $\underline{\Theta}_B^m$ or $\{\underline{\Theta}_A^m$ and $\underline{\Theta}_B^m\}$.

It has also been shown in [6] that the asymptotical distribution of $-2.\log\Lambda$ is a chi-square, central under H, non-central under $\bar{H}$, which number of degrees of freedom p is the difference between the number of parameters of models under the two hypotheses $p = N' + N'' - N$ (N', N'' and N are the number of observable components of $\underline{X}'$, $\underline{X}''$ and $\underline{X}$ under both hypotheses). According to observability situations, p is equal 2,3, or 4. This last result is useful to fix threshold η for a

given false alarm rate and to compute the power of the test (12).

3. ASSOCIATION PERFORMED ON LOCAL ESTIMATES

The basis of the proposed hierarchical approach (very similar to the one of [5]) is the fact that local estimates $\underline{X}$ can be approximately sufficient statistics for bearings tracks $\underline{\Theta}_m$. According to [10], when an estimator $\underline{X}$ is consistent and asymptotically efficient, the log-likelihood $f(\underline{X})$ can be approximated by

$$f(\underline{X}) = f(\hat{\underline{X}}) - \frac{1}{2}(\underline{X} - \hat{\underline{X}})\, F(\underline{X})\, (\underline{X} - \hat{\underline{X}}) + \cdots \quad (13)$$

First let's consider the observable case and assume that local estimates have been obtained for every bearings tracks prior to TTA. Let $\{\hat{\underline{X}}', V'\}$ and $\{\hat{\underline{X}}'', V''\}$ denote these local estimates and the corresponding estimated covariance matrices ($V=F^{-1}$) deduced from $\underline{\Theta}_A{}^m$ and $\underline{\Theta}_B{}^m$. Using (13) the log-likelihood f_{AB} of H becomes

$$f_{AB} = \sup_{\underline{X}} \left[f_A(\hat{\underline{X}}') - \frac{1}{2}(\underline{X} - \hat{\underline{X}}')' V'^{-1} (\underline{X} - \hat{\underline{X}}') + f_B(\hat{\underline{X}}'') - \frac{1}{2}(\underline{X} - \hat{\underline{X}}'')' V''^{-1} (\underline{X} - \hat{\underline{X}}'') \right] \quad (14)$$

whilst log-likelihood $\bar{f}_{AB}$ of $\bar{H}$ is obviously

$$\bar{f}_{AB} = f_A(\hat{\underline{X}}') + f_B(\hat{\underline{X}}'') \quad (15)$$

Therefore LRT (12) is approximately

$$-2.\log\Lambda = \inf_{\underline{X}} \left\{ (\underline{X} - \hat{\underline{X}}')' V'^{-1} (\underline{X} - \hat{\underline{X}}') + (\underline{X} - \hat{\underline{X}}'')' V''^{-1} (\underline{X} - \hat{\underline{X}}'') \right\} \quad (16)$$

Minimisation of (16) is straighforward, but tedious, and leads to a statistical distance (which is formally very close to the Mahalanobis distance [9]) between $\hat{\underline{X}}'$ and $\hat{\underline{X}}''$

$$-2.\log\Lambda = d(\hat{\underline{X}}', \hat{\underline{X}}'') = (\hat{\underline{X}}' - \hat{\underline{X}}'')'(V' + V'')^{-1}(\hat{\underline{X}}' - \hat{\underline{X}}'') \quad (17)$$

whilst the argument of extremum, $\hat{\underline{X}}$, is

$$\hat{\underline{X}} = (V'^{-1} + V''^{-1})^{-1} (V'^{-1}\hat{\underline{X}}' + V''^{-1}\hat{\underline{X}}'') \quad (18)$$

Finally, when local estimates are used, the bi-hypotheses LRT for TTA becomes :

$$d(\hat{\underline{X}}', \hat{\underline{X}}'') \underset{H}{\overset{\bar{H}}{\gtrless}} \eta \quad (19)$$

Moreover $d(\hat{\underline{X}}', \hat{\underline{X}}'')$ is also distributed according a chi-square law ([3]) with 4 degrees of freedom, central under H and non-central under $\bar{H}$.

Let's now consider the unobservable situation. On each array one can just estimate the 3-components vector $\underline{X}^3(x_0)$ or $\underline{Y}$. As previously let $\{\hat{\underline{Y}}', W'\}$ and $\{\hat{\underline{Y}}'', W''\}$ denote the local estimates of $\underline{Y}$ from $\underline{\Theta}_A{}^m$ or $\underline{\Theta}_B{}^m$ and the corresponding estimated covariance matrices (which are now 3-components vectors and 3x3 matrices).

Using (13), which still stands in unobservable situations, the log-likelihood ratio test (12) can be rewritten as

$$-2.\log\Lambda = \inf_{\underline{Y}', \underline{Y}''} \left[(\underline{Y}' - \hat{\underline{Y}}')' W'^{-1} (\underline{Y}' - \hat{\underline{Y}}') + (\underline{Y}'' - \hat{\underline{Y}}'')' W''^{-1} (\underline{Y}'' - \hat{\underline{Y}}'') \right] \quad (20)$$

where $\underline{Y}'$ and $\underline{Y}''$ are such that there are particular values of x and $\underline{X}$ for which

$$\underline{X} = \underline{X}_A + (x - x_A) \begin{bmatrix} 1 \\ \cdots \\ \underline{Y}' \end{bmatrix} = \underline{X}_B + (x - x_B) \begin{bmatrix} 1 \\ \cdots \\ \underline{Y}'' \end{bmatrix} \quad (21)$$

($\underline{Y}'$ and $\underline{Y}''$ are relative to the same source $\underline{X}$)

Further details about computations can be found in [3]. The main result is that, as in observable situation, the minimum of criterion (21) can be obtained analytically and does not require numerical iterative procedure (such as optimal technics of section II). Finally LRT of section II can also be approximated by a test on a distance d' between $\hat{\underline{Y}}'$ and $\hat{\underline{Y}}''$ (very similar to d) while $\hat{\underline{X}}$ is given by a linear fusion of local estimates.

4. EXPERIMENTAL RESULTS

Computer simulations of TTA have been conducted in the case of two separated arrays A and B. Conditions of simulations are :
- array A : initial position (x_A=0, y_A=-1000) (meters), speed 10 knots, course 180 degrees.
- array B : fixed in (x_B = 0, y_B = 1000).
- source S: initial position (x=20000, y=0), speed 10 knots, course 270 degrees.
- bearing measurements rate : 5 seconds with standard deviations equal to 0.5 degrees.

Source range estimation

Figure 2 shows the relative efficiency of both methods for source range estimation. Following quantities are plotted versus observation time (in seconds) :
- theoretical standard deviations of range estimation errors (by Cramer-Rao Lower Bounds) with dotted line "– – –",
- sampled standard deviations of range estimation errors for optimal 2-arrays BOT (section 2) with symbols "*",
- sampled s.d. of range error estimation for fusion of local estimates (section 3) with symbols "o".

(These last values have been obtained by

average of of range errors on 100 independant samples).
In these conditions performance of both methods is identical when the observation time is long enough (here about 15 minutes).

Track to Track Association

Comparisons of performance for both tests of sections 2 and 3 are given below. Probabilities that optimal LR and suboptimal test outputs are identical have been estimated on 100 independant samples for observation times equal to 650 or 950 secondes. For both tests threshold η corresponds to a false alarm rate equal to 0.1. After 650 seconds probabilities that outputs of both tests are identical are 0.72 (when optimal LRT decides H) or 0.67 (when optimal LRT decides $\bar{H}$). After 950 seconds these probabilities respectively become 0.89 and 1. It demonstrates the similar behaviour of both tests whenever observation time is long enough.

CONCLUSIONS

A simple hierarchical approach to Bearings-Only Track to Track Association has been proposed. This method still applies to association and fusion of sets of local estimates (in unobservable situations) when complete local estimates are not available and has performance very similar to optimal approach. Further studies will concern application to multi-targets situations when a hierarchical clustering (as in [5]) is performed.

REFERENCES

[1] S.C NARDONE, V.J AIDALA, "Observability for Bearings-Only Target Motion Analysis". IEEE Trans on AES, Vol 17 No 2, March 1981.

[2] C.JAUFFRET, D.PILLON, "New observability criterion in Target Motion Analysis". Proc of NATO ASI, Kingston, Canada, 1988

[3] J.M.PASSERIEUX, D.PILLON, P.VERVEUR "Track to Track Association Performed on Mahalanobis Distance in both Observable and Unobservable Situations" Proc of 23rd Asilomar Conf on SSC, Nov 1989

[4] S.C.NARDONE, A.G.LINDGREN, K.F.GONG "Fundamental Properties and Performance of Conventional Bearings-Only Target Motion Analysis",IEEE Trans on AC, Vol29 No 9, Sept 1984.

[5] A.G LINDGREN, K.F GONG, M.L GRAHAM, S.C NARDONE "Passive Localization and Trajectory Estimation in the Underwater Acoustic environment: the need for Hierarchical Estimation System", Proc of 21th Asilomar Conf on SSC, Nov 1987

[6] I. GUELLE, D. PILLON "Inter-Array Multi-Tracks Association", Proc of ICASSP, Glasgow, May 1989

[7] H.L.VAN TREES, "Detection, Estimation and Modulation Theory", Wiley, 1968

[9] S.KOTZ, N.L.JOHNSON, "Encyclopedia of Statistical Sciences", Wiley, 1982.

[10] D.V.LINDLEY "The use of Prior Probability Distributions in Statistical Inference and Decisions", Proc of 4th Symp. on math. stat and probability, Berkeley, 1961

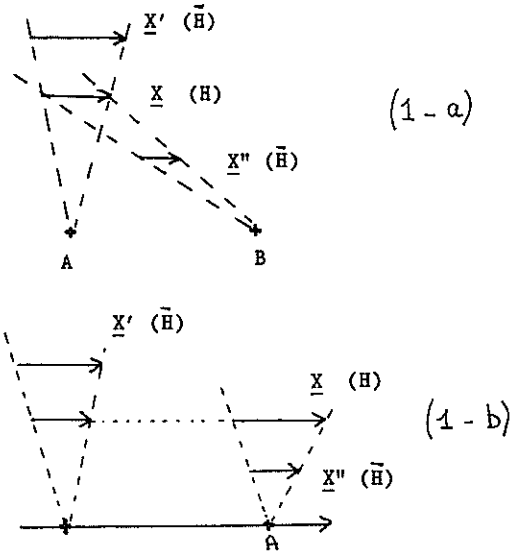


figure 1 : Ambiguous situations in TTA

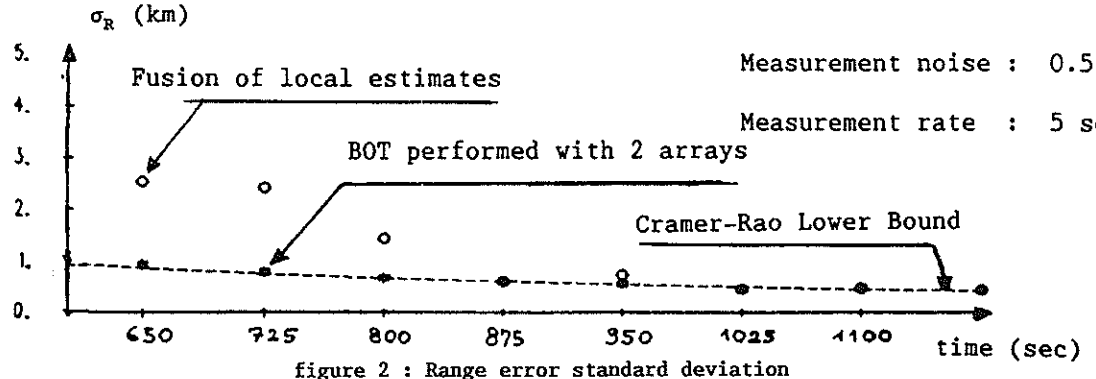


figure 2 : Range error standard deviation (unobservable situation)

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SOME SIMPLE AND EFFICIENT METHODS FOR BEARING-ONLY TARGET ESTIMATION

Dinh Tuan PHAM

Laboratory of Modeling and Computation, IMAG, C.N.R.S.
BP 53X, 38041 Grenoble Cedex, France

Two new estimation methods for the trajectory of a target with linear uniform motion, based on the only observation of its bearing, are proposed. Our linear estimator deals with the case where the observer also has a linear uniform motion so that one can only estimate the its relative trajectory up to a multiplicative constant, while our quadratic estimator permits the estimation of its absolute trajectory in the observable case where the observer has a non uniform motion. Both estimators are simple to compute and are highly efficient.

1. INTRODUCTION

This paper considers the estimation of the trajectory (position and velocity) of a target based only on the observation of its (noise corrupted) bearings, viewed from a single observer. The target is assumed to have a constant velocity and both it and the observer are moving in a fixed (horizontal) plane. The time evolution of the target's bearing thus contains information on its trajectory and is exploited to construct estimates of the latter.

Conventional methods for the above problem can be classified into two groups: (i) reccursive methods based on the extended Kalman filtering technique ([2], [5]) and (ii) non recursive batch processing methods such as the maximum likelihood method or some variation of it (see [3]). But the problem is instrinsically non linear so that the recursive methods may encounter serious convergence problem [1] and the batch processing methods require costly iterative computation. An exception to the last type of method is the pseudo-linear estimator (see [1],[3]) which requires only the solution of a linear system of equations, but this estimator is known to be biased and not efficient. In this work, we propose new estimation methods in the batch processing category, which are cheap and yet highly efficient, in the sense that the covariance matrix of the estimator closely approaches the Cramér-Rao (C-R) lower bound. Two different setups are considered, concerning the 'partial' and the full target estimation problems, respectively.

In the 'partial' target estimation problem, the observer is assumed to have a linear uniform motion. It is then well known that the system is unobservable (i.e.target estimation is impossible). However, it is still possible to estimate, up to a multiplicative constant, the relative trajectory of the target with respect to the observer. This is called the 'partial' target estimation problem. It has its own interest, for example in situations when the target's range may be obtained by other means. Our estimation method is based on the parametrization of the target's relative trajectory, up to a multiplicative constant, by three bearings at three different observation times. They are then estimated by certain weighted averages of the observed bearings. The target's relative trajectory can be recovered from them, up to a multiplicative constant, through simple trigonometric formula. By choosing the bearing times and the weighs judiciously, it may be shown that the above estimates are only slightly biased and have the covariance matrix closely approaching the C-R bound, if the relative displacement of the target during the total observation period, is small with respect to the target's range. This result has been confirmed by numerical computations.

In the full target estimation problem, the observer must have a non uniform motion. It is then possible, in general, to estimate the absolute position and velocity of the target, asumming that the observer's motion is known. Efficient estimators however would require non linear maximization and thus costly iterative computation. In this work, we introduce the quadratic estimator which is somewhat similar to the pseudo-linear estimator, but unlike the latter, does not have any bias. It is obtained from the eigen-vector associated with the smallest eigen-value of certain quadratic form, and hence is rather cheap to compute. The covariance matrix of our estimator is also close to the C-R bound, under the same condtion as above and when the noise variance is small with respect to 1. Our estimator is thus a cheap and viable alternative to the maximum likelihood estimator, or may be used as a starting value in an iterative procedure for its computation.

2. THE PARTIAL TARGET ESTIMATION PROBLEM

We assume here that the target and the observer both have constant velocities. The relative position of the former with respect to the latter at time t is thus given by $r_t = r_0 + vt$ where v denotes the relative velocity of the target with respect to the observer. For convenience, we shall represent 2-dimensional vector by a complex number, with real and imaginary parts being its x- and y- coordinates, respectively. Thus the bearing of the target at time t is

$$\beta_t = \pi/2 - \arg(r_t) \tag{2.1}$$

and the range is simply $|r_t|$. The problem is to estimate the target trajectory from n observed bearings at equi-spaced times, denoted for convenience by 1, ..., n :

$$z_t = \beta_t + e_t, \quad t = 1, ..., n, \tag{2.2}$$

e_t being the observation errors (assumed to be independent Gaussian random variables with zero mean and variance σ^2). Clearly, only the relative trajectory of the target up to a multiplicative constant can be estimated. Our idea is to parametrize such trajectory by three bearings at three distinct times t_0, t_1, t_2. It is shown in [4] that this parametrization is always possible and formula relating r_t (up to a constant factor) to the parameters $\theta_i = \beta_{t_i}$, i = 0, 1, 2, are provided. Thus β_t can thus be expressed as a function of these parameters.

In the next two subsections, we shall compute the C-R bounds for the above parameters, and introduce the linear estimator which closely attains this bound.

2.a) *The C-R bound*

It is well known that the covariance matrix of any unbiased estimate of the parameter of a statistical model cannot be greater the inverse of the Fisher information matrix, called the C-R bound. The model (2.2), with β_t being a function of the parameters θ_0, θ_1, θ_3, is a nonlinear regression model, and the Fisher's information matrix of can be shown to have the general (i,k) element

$$\frac{1}{\sigma^2}\sum_{t=1}^{n}\frac{\partial\beta_t}{\partial\theta_i}\frac{\partial\beta_t}{\partial\theta_k}. \tag{2.3}$$

In [4], it is shown that,

$$\frac{\partial\beta_t}{\partial\theta_i} = (|r_{t_i}|^2/|r_t|^2)\phi_i(t)\ , \ i = 0, 1, 2, \tag{2.4}$$

where ϕ_0, ϕ_1, ϕ_2 are the Lagrange interpolation polynomials based on t_0, t_1, t_2:

$$\phi_i(t) = \Big[\prod_{k\neq i,\ 0\le k\le 2}(t - t_k)\Big] / \Big[\prod_{k\neq i,\ 0\le k\le 2}(t_i - t_k)\Big].$$

Now, let ψ_3 be the orthogonal polynomial of third degree on {1, ..., n}, i.e. the third degree polynomial satisfying

$$\sum_{t=1}^{n} t^k\psi_3(t) = 0, \quad k = 0, ..., 2,$$

t_0, t_1, t_2 be its roots (known to be real and contained in the interval [0,n], see [4]) and ϕ_0, ϕ_1, ϕ_2 be the Lagrange interpolation polynomials based on them. Then one has an analogue of the three-point Gauss quadrature formula (see [4]), for the approximate computation of the sum (2.3):

$$\frac{1}{\sigma^2}\sum_{k=0}^{2}\|\phi_k\|^2\frac{\partial\beta_{t_k}}{\partial\theta_i}\frac{\partial\beta_{t_k}}{\partial\theta_j} \tag{2.5}$$

where $\|\phi_k\|^2 = \phi_k(1)^2 + ... + \phi_k(n)^2$ is the squared norm of ϕ_k. Note that the functions ϕ_0, ϕ_1, ϕ_2 are orthogonal (in the sense that $\phi_i(1)\phi_k(1) + ... + \phi_i(n)\phi_k(n) = 0$ for $i \neq k$) and that $\|\phi_k\|^2$ also equals $\phi_k(1) + ... + \phi_k(n)$, see [4].

Thus, taking t_0, t_1, t_2 in the the definition of θ_0, θ_1, θ_2 to be precisely the three roots of ψ_3 as above, the Fisher's information matrix, by (2.5), is approximately diagonal with diagonal elements $\|\phi_0\|^2/\sigma^2$, $\|\phi_1\|^2/\sigma^2$, $\|\phi_2\|^2/\sigma^2$. Hence the C-R bound for is also approximately diagonal, with diagonal elements $\sigma^2/\|\phi_0\|^2$, $\sigma^2/\|\phi_1\|^2$, $\sigma^2/\|\phi_2\|^2$. Explicit computation yields [4]

$$t_0 = (n+1)/2, \ \ t_1 = t_0 - \tau, \ \ t_2 = t_0 + \tau, \ \ \tau = (3n^2-7)/20,$$

$$\|\phi_0\|^2 = n\,\frac{4(n^2-4)}{3(3n^2-7)}, \quad \|\phi_1\|^2 = \|\phi_2\|^2 = n\,\frac{5(n^2-1)}{6(3n^2-7)}.$$

2.b) *The linear estimator*

Using again the anlogue of Gauss quadrature formula based on the points t_0, t_1, t_2, one has [4]

$$(2.6) \qquad \|\phi_i\|^{-2}\sum_{t=1}^{n}\phi_i(t)\beta_t \cong \beta_{t_i}.$$

This suggests the following estimator of θ_i, (i = 0, 1, 2):

$$\hat{\theta}_i = \|\phi_i\|^{-2}\sum_{t=1}^{n}\phi_i(t)z_t,$$

called the linear estimator, which thus has expectation approximatively θ_i. Further, from the properties of the polynomials ϕ_0, ϕ_1, ϕ_2 cited in previous sub-section, this estimator has the covariance matrix precisely the approximation to the C-R bound obtained there.

Table 1 Bias of the linear estimator : first row refers to the true bearings (central bearing taken as origin), second row to the biases, *multiplied* by 100, n = 50 observations.

Heading	Range variation ratio								
	.3			.4			.5		
π/12	-.034	0	.037	-.047	0	.035	-.062	0	.042
	.001	-.001	-.001	.003	-.003	-.002	.008	-.007	-.005
π/6	-.064	0	.014	-.089	0	.068	-.031	0	.083
	.001	-.001	.001	.003	-.003	-.002	.007	-.006	-.005
π/4	-.089	0	.076	-.122	0	.098	-.157	0	.120
	.000	.000	.000	.00	.000	.000	.000	.001	.000
π/3	-.106	0	.095	-.144	0	.124	-.184	0	.152
	-.001	.001	.001	-.003	.003	.002	-.007	.007	.005
5π/12	-.115	0	.108	-.155	0	.143	-.194	0	.176
	-.001	.001	.001	-.003	.002	.002	-.007	.005	.006
π/2	-.116	0	.116	-.154	0	.154	-.191	0	.191
	.000	.000	.000	.000	.000	.000	.000	-.001	.001

The above approximations is based essentially on the fact that the functions β_t and $|r_t|^{-2}$ vary slowly with t on its working range 1, ..., n. This holds if the range variation ratio (RVR), defined as $|r_n - r_0|/|r_{(n+1)/2}|$, is small. To see how this affects the performance of the linear estimator, we compute numerically its bias, and its covariance matrix and the C-R bound (divided by σ^2/n) and report the results in tables 1 and 2. The bias is computed from the difference between the left and the right hand sides of (2.6), the C-R bound from the inverse of the matrix with general element given by (2.3) and (2.4). The results depend little on the sample size n, only on the RVR and the heading, defined as $\arg[(r_n - r_0)/r_{(n+1)/2}]$. Therefore, we shall consider several values of the RVR and heading but only one value of n (= 50). Note that only headings from 0 to π/2 need to be considered, the tables may be completed for other headings by symmetry consideration. Indeed, changing the heading from ß to - ß ($0 \le ß \le \pi/2$) changes the sign of the biases but does not change the covariance matrix of the estimator, and changing from ß to π - ß interchanges the biases and the variance of $\hat{\theta}_1$ and $\hat{\theta}_2$ and the covaariance between them and $\hat{\theta}_0$.

Table 2 The C-R bound for the estimators: n = 50, the corresponding variances of the linear estimators are 2.25, 3.60 and 3.60 (all numbers are divided by σ^2/n).

Heading	Range variation ratio								
	.1			.2			.3		
0	2.24			2.22			2.18		
	.01	3.59		.03	3.58		.06	3.55	
	.01	-.01	3.59	.03	-.02	3.58	.06	-.05	3.55
π/12	2.24			2.22			2.19		
	.01	3.59		.02	3.58		.05	3.56	
	.01	-.00	3.59	.02	-.02	3.58	.05	-.04	3.56
π/6	2.25			2.23			2.20		
	.00	3.59		.02	3.58		.04	3.57	
	.00	-.00	3.59	.02	-.01	3.58	.04	-.03	3.56
π/4	2.25			2.24			2.22		
	.00	3.60		.01	3.59		.02	3.58	
	.00	-.00	3.60	.01	-.01	3.59	.02	-.02	3.58
π/3	2.25			2.25			2.24		
	.00	3.60		.00	3.60		.00	3.60	
	.00	-.00	3.60	.00	-.00	3.60	.01	-.00	3.59
5π/12	2.25			2.26			2.26		
	-.00	3.60		-.00	3.60		-.01	3.61	
	-.00	.00	3.60	-.00	-.00	3.60	-.01	.01	3.60
π/2	2.25			2.26			2.27		
	-.00	3.60		-.01	3.60		-.01	3.61	
	.00	.00	3.60	-.01	-.00	3.60	-.01	.01	3.61

It may be seen from table 1 that the bias is very small. Also, the C-R bound for the estimator, from table 2, is indeed close to a diagonal matrix and the variances of the linear estimators are close to its diagonal elements, as expected theoretically. The deviation increases for higher RVR but remains small at RVR = 0.3. It is also more pronounced for smaller headings. Note that the linear estimator may have smaller variance than the C-R bound (for RVR = 0.3, heading = π/2 for ex.) because it is not exactly unbiased so that the C-R inequality needs not hold.

3. THE FULL TARGET ESTIMATION PROBLEM

We now assume that the observer have a non uniform motion. Then, denoting by c_t its position at time t, the relative position of the target to the observer is given by: $r_t = r_0 + vt - (c_t - c_0)$ where v is the *absolute* velocity of the target. As before, we represent vectors in $\mathbb{R}^2$ by complex numbers. Thus the bearing is given by (2.1) and the range is $|r_t|$. The problem is to estimate the target trajectory (or equivalently r_0 and v) based on n observations of the bearing z_t, given by (2.2), at times t = 1, ..., n.

We now introduce a new estimator called the quadratic estimator (QE). It can be viewed as a modification of the well known pseudo-linear esimator (PLE) (see [3] for ex.), which is obtained by the minimization of

$$(4.1) \qquad \sum_{t=1}^{n} [|r_t|\sin(z_t - \beta_t)]^2.$$

It can be easily seen that the function in the above braccket is linear in the parameters

$$\theta = (\mathrm{Re}(r_s), \mathrm{Im}(r_s), \mathrm{Re}(v), \mathrm{Im}(v)),$$

s being any reference time. Thus the estimation of θ through the minimization of (4.1) leads directly to the solution of a linear system. The PLE is very easy to compute but is known to suffer a severe defect because it is biased. This comes from the inadequacy of the criterion (4.1): one minimise not only the discrepencies $\sin(z_t - \beta_t)^2$ between the model and the observations but also the squared ranges $|r_t|^2$ of the theoretical trajectory. Our idea is to consider the new criterion

$$(4.2) \qquad C(\theta) = \sum_{t=1}^{n} [|r_t|\sin(z_t - \beta_t)]^2 / (\sum_{t=1}^{n} |r_t|^2).$$

To minimize the above criterion, we introduce the new parameter $\mu = (\mu_0, ..., \mu_4)$, defined up to a constant factor by $\theta = (\mu_1, ..., \mu_4)/\mu_0$. Then

$$r_t = [\mu_0(c_t - c_s) + \mu_1 + j\mu_2 + (t - s)(\mu_3 + j\mu_4)]/\mu_0,$$

yielding $|r_t|\sin(z_t - \beta_t) = \mu\zeta_t/\mu_0$ where

$$\zeta_t = [\sin(z_t)c_{xt} - \cos(z_t)c_{yt}, \sin(z_t), -\cos(z_t), (t - s)\sin(z_t), (s - t)\cos(z_t)]',$$

c_{xt} and c_{yt} being the real and imaginary parts of $c_t - c_s$. Thus $C(\theta)$ can be written as $\mu S\mu'/\mu G\mu'$ where

$$S = \sum_{t=1}^{n} \zeta_t\zeta_t',$$

$$G = \sum_{t=1}^{n} \begin{bmatrix} |c_t - c_s| & c_{xt} & c_{yt} & (t-s)c_{xt} & (t-s)c_{yt} \\ c_{xt} & 1 & 0 & t-s & 0 \\ c_{yt} & 0 & 1 & 0 & t-s \\ (t-s)c_{xt} & t-s & 0 & (t-s)^2 & 0 \\ (t-s)c_{yt} & 0 & t-s & 0 & (t-s)^2 \end{bmatrix}.$$

The QE is obtained from the minimization of $\mu S\mu'/\mu G\mu'$. This yields (see[4]) the eigen-vector $\hat{\mu}'$ of S relative to G, i.e. the solution of $\det(S - \lambda G)\hat{\mu}' = 0$, corresponding to the smallest eigen-value λ.

It is shown in [4] that for small $\tilde{\sigma}^2 = E[\sin(e_t)^2]$, the QE is approximately unbiased with covariance matrix $\tilde{H}^{-1}(\tilde{K} + \tilde{L})\tilde{H}^{-1}$ where the matrices $\tilde{H}$, $\tilde{K}$, $\tilde{L}$ have the general element

$$\tilde{H}_{ik} = \sum_{t=1}^{n} \rho_t(1 - 2\tilde{\sigma}^2) \frac{\partial\beta_t}{\partial\theta_i} \frac{\partial\beta_t}{\partial\theta_k},$$

$$\tilde{K}_{ik} = \tilde{\sigma}^2\{1 - \tilde{\sigma}^2 - \mathrm{var}[\sin^2(e_t)]\} \sum_{t=1}^{n} \rho_t^2 \frac{\partial\beta_t}{\partial\theta_i} \frac{\partial\beta_t}{\partial\theta_i}$$

$$\tilde{L}_{ik} = \frac{1}{4}\mathrm{var}[\sin^2(e_t)] \sum_{t=1}^{n} \frac{\partial\rho_t}{\partial\theta_i} \frac{\partial\rho_t}{\partial\theta_k}$$

with $\rho_t = |r_t|^2/(|r_1|^2 + ... + |r_n|^2)$. This matrix is close to the C-R bound when ρ_t varies little with t.

REFERENCES

[1] ADAILA, V.J. Kalman filter behaviour in bearing-only tracking applications. IEEE Trans. AES-15, 1882, 29-39.

[2] CHANG, C.B., TABACZYNSKI, J.A. Application of state estimation to target tracking. IEEE Trans. AC-84, 1984, 98-109

[3] NARDONE, S.C., LINDGREN, A.G., GONG, K.F. Fundamental properties and performance of conventional bearing-only target motion analysis. IEEE Trans. AC-29, 1984, 775-787.

[4] PHAM, D.T. Some quick and efficient methods for bearing only target motion analysis. Technical Report No RR 762-M, 1989, TIM3, IMAG, Grenoble.

[5] SONG, T.L., SPEYER, J.L. A stochastic analysis of a moddified gain extanded Kalman filter with application to estimation with bearing-only measurements. IEEE Trans. AC-30, 1985, 940-949.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

High Resolution Channel Measurement for Mobile Radio

S. Hermann, U. Martin, R. Reng, H.W. Schuessler, K. Schwarz
Lehrstuhl für Nachrichtentechnik, Universität Erlangen-Nürnberg

Abstract - **Propagation measurements in the field provide the base for the design of communication systems, using the mobile radio channel. Furthermore they are of interest for the development and confirmation of statistical models, introduced for these channels. For reasons of precision as well as flexibility, we apply a digital correlation method to determine the time-variant impulse response of mobil radio channels. We choose a two level excitation sequence, which offers maximum likelihood channel estimation with optimal processing gain. Sampled versions of impulse response snapshots, i.e. discrete-time substitutes, are received. The throughput of modern signal processors is high enough to allow on-line processing. In the frequency range of 900 MHz, measurements with bandwidth up to 400 kHz can be done. Echo path delay times up to 127 μs are observable. Depending on the desired accuracy of measured data, bounds for the standard deviation of phase noise in carrier generation are given. With a high resolution parametric estimation procedure single echo paths can be identified as well as a low word-count description of the channel may be aquired.**

1 Introduction

Land mobile communication systems constitute a market with rapidly increasing volume. The digital D-NET is expected to further reduce end-user equipment cost, which will in turn account for wide-spread use.
For device development resp. improvement and net planning, accurate information on the properties of the mobile radio channel are fundamental. In addition measured data on the channel are required to validate theoretical assumptions. We describe a system capable to measure impulse responses as an appropriate characterization of the time varying mobile radio channels, e.g. as specified for GSM communication systems.

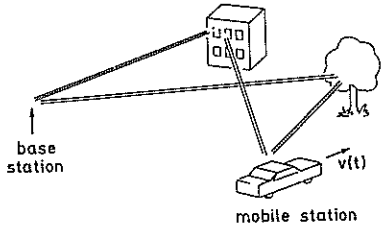


Figure 1-1: Model of the mobile radio channel

In a mobile radio environment, the transmitted signal is propagated to the receiver along a variety of paths. The impulse response $h(t)$ of such *multipath propagation channels* (fig.1-1) is frequently modeled as a sum of attenuated and delayed unit impulses

$$h(t) = \sum_{\nu=1}^{p} a_\nu \, \delta_0(t - \tau_\nu), \qquad a_\nu(\tau) \in \mathbb{C} \qquad (1.1)$$

with expected delay times τ_ν of up to 100 μs. This corresponds to a difference in path lengths of approx. 30 km.
Additionally, the properties of the channel may rapidly vary in time: Both transmitter and receiver are potentially moving, but also other vehicles may influence the propagation characteristics. In conclusion, the impulse response depends on the time instance τ of observation.
A more detailed model will thus account for this *time variance*:

$$h(t, \tau) = \sum_{\nu=1}^{p(\tau)} a_\nu(\tau) \, \delta_0(t - \tau_\nu(\tau)) \qquad (1.2)$$

The Doppler frequency $f_D = f_m \cdot v/c$, depending on the carrier frequency f_m, the vehicle velocity v, and the speed of light c, serves as an upper bound for the time variance of the channel. For $v < 100$km/h we get $f_D < 100$Hz. It should be noted that the above model is useful for device development and post-processing of measured data as explained in section 5, but it is not a prerequisite for the measurement itself.
The measuring system is capable of repeatedly determining snapshots of the time-variant impulse response in the 900 MHz frequency range at a bandwidth of up to 400 kHz. Delay times τ of up to 127 μs may be observed. As 127 samples of the measured complex valued impulse responses are stored on hard disk immediately after each snapshot, the maximum repetition rate is 22.86 ms. The widely digital implementation as described in section 3 yields high accuracy.

2 Correlation measurement

Large S/N-ratios in the measured impulse responses require a high energy content of the transmission signal. The direct method to determine an impulse response by impulse excitation has some disadvantages in this context, because the analogue components of a measuring device delimit the usable signal amplitude range. Impulse compression techniques, which e.g. are based on two level pseudo-noise sequences, allow a higher energy content of the transmitted signal for a given amplitude range.
The method to be used was investigated in some detail in [Mar89.2]. In principle it works with a periodic excitation of the object under test by a modified PN-sequence $v(k)$. In the receiver a correlation is done by means of linear filtering (fig.2-1). Both the excitation and the correlation sequences

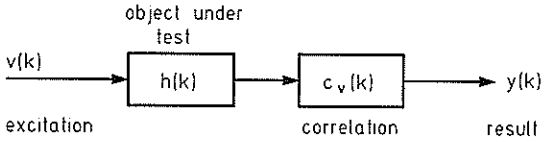


Figure 2-1: Measurement of discrete-time impulse responses via PN-sequences

can be expressed in terms of the L-periodic PN-sequence $pn(k_{\mathrm{mod}L})$:

$$v(k) = A + pn(k_{\mathrm{mod}L}) \qquad \text{and} \qquad (2.1)$$

$$c_v(k) = \begin{cases} \frac{1+A}{A\cdot L-1} + pn([-k]_{\mathrm{mod}L}) & \text{if } 0 \le k < L\,, \\ 0 & \text{else.} \end{cases}$$

If the impulse response $h(k)$ is causal and at least approximately zero for $k \ge L$, the result becomes

$$y(k) = (L+1)\cdot h(k) \qquad \text{if } 0 \le k < L\,. \qquad (2.2)$$

In [Rup89] it was shown that this method offers maximum likelihood channel estimation, if the received signal is perturbed by additive white gaussian noise. For the additive constant A the optimal choice is

$$A_o = (1 - \sqrt{L+1})/L\,. \qquad (2.3)$$

Choosing this optimum, the correlation filter degenerates to the matched filter for the used excitation and offers the best possible S/N-ratio in the measured impulse response, if there is additive white noise at the correlator input.
The digital correlation has to be adapted for the determination of time variant continuous-time impulse responses $h(t,\tau)$. Obviously only bandlimited discrete-time snapshots

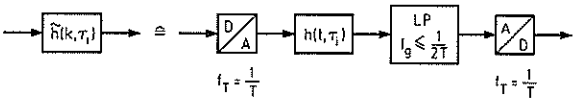


Figure 2-2: Result of measuring a continuous-time impulse response using a digital method

$\tilde{h}(k,\tau_i)$ (fig.2-2) can be taken from the object under test by a digital method. This will not result in quality deteriorations, if the bandwidth of the measuring device is larger than the bandwidth of the object under test. This bandwidth is mainly determined by the chip frequency f_T. We use $f_T = 1\mathrm{MHz}$ and a PN-period $L = 127$. The channels of 400kHz bandwidth can be measured without loss of information, if the maximum relative propagation time is less than $L/f_T = 127\mu s$.

3 Realization

As outlined in the block diagrams of transmitter (fig. 3-1) and receiver (fig. 3-2), the measuring system is composed of

1. generation of the discrete-time base-band signal
2. D/A conversion
3. transmitter modulation and amplification stages, antenna
4. receiver amplification and demodulation stages
5. A/D conversion
6. digital processing stage
7. oscilloscopic display of the impulse response (in dB)
8. storage of measured, complex valued impulse responses on hard disk

During the measurement, the transmitter is stationary whereas the receiver is operated in a small truck driving at velocities of up to 80 km/h (50 mph).

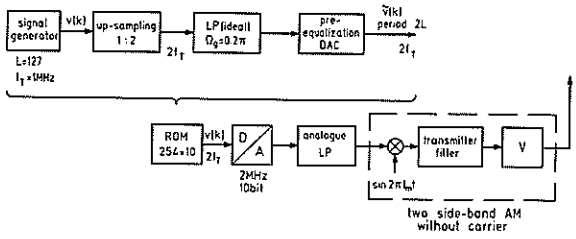


Figure 3-1: Block diagram of transmitter

One period of the discrete-time base-band signal composed of an interpolated, band-limited, and pre-equalized pseudo noise sequence is periodically read out of a ROM. After D/A conversion, low-pass filtering, modulation, and amplification in the transmitter, the signal is fed to the antenna. The magnitude spectrum of the transmitted signal is almost constant in the interesting frequency range.
In general, the channel will not show any symmetry with respect to the RF band center or carrier frequencies. Therefore, the base-band equivalent impulse response to be measured will be complex valued and a two-channel quadrature demodulation is required in the receiver. The high precision possible with digital signal processing hardware is best maintained if the jitter due to A/D converters is minimized and alignment problems with analogue filters are eliminated. This suggests to carry out the A/D conversion in an IF range of 500 - 900 kHz followed by a digital quadrature demodulation into the actual base-band instead of an analogue quadrature demodulation with separate A/D conversion of both channels. The digital quadrature signal is then modified by a complex low-pass and down-sampling by a factor

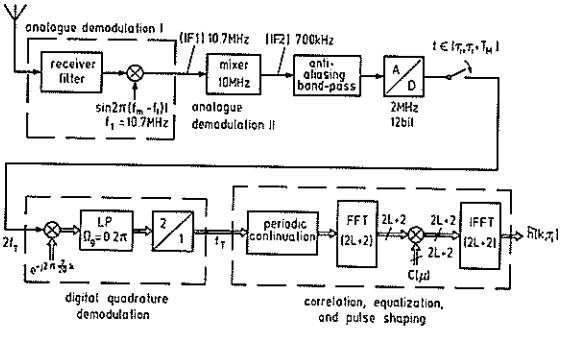


Figure 3-2: Block diagram of receiver

of 2. With one step of a fast convolution similar to the overlap-save method the digital correlation, an equalization of both the magnitude and phase frequency response of the device itself, and additional pulse shaping are performed. This convolution requires only two FFTs of length 256 and 256 complex multiplications per impulse response. Together with forming the logarithmic magnitude of the result for oscillographic display, these computations amount to a total of about 8 ms on a DSP56001 fixed-point signal processor. Storing the complex valued results on a hard disk takes up to 15 ms per block, thus every 22 ms a new impulse response can be computed and stored.
For later post-processing, additional data are required: With a car navigation system and measuring wheels mounted to the car, the truck position is gained. Also, the instantaneous time of each snap-shot is recorded. Besides taking snap-shots of the channel's impulse response in equal units of time, also the distance driven can be used as a trigger

criterion.
In summary, the characteristics of the device are as follows:

bandwidth	400 kHz, subject to a software based reduction, if desired
time resolution	5 μs, using cosine channel pulse shaping (see [Mar89.2])
max. delay time	127 μs
max. trigger rate	44 complex impulse responses per second
accuracy	$\approx$ 48 dB impulse-peak to noise level if one echo is present
user interface	menu driven
data storage	either magnitude or complex valued impulse response, augmented by time and positional data

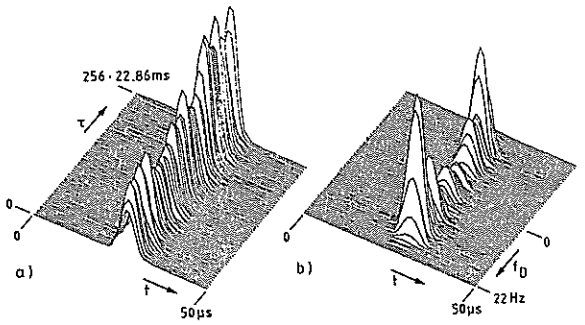


Figure 3-3: Impulse response envelopes and Doppler spectra

At low vehicle velocities, even Doppler spectra may be acquired. This can be seen from results of propagation measurements carried out in the area of Darmstadt to test the device. Fig. 3-3a) and 3-3b) show impulse response envelopes and the Doppler spectrum when the receiver drove at approx. 15 km/h along a straight line from the transmitter towards a close-by large building. Due to the small distance, the direct path and the path reflected at the building cannot be distinguished (fig. 3-3a), but the characteristic two-sided Doppler spectrum is still gained (fig. 3-3b). These figures result out of a post processing of data, done by Dr. Lorenz and his colleagues (Forschungsinstitut of the Deutsche Bundespost).

4 Effects of Phase Noise

In order to check the accuracy of the measuring device, some nonideal properties of the physical environment have to be taken into account. Besides the noisy perturbations of the channel itself, which are mainly determined by the transmitter power, noisy inaccuracies unavoidable at carrier generation in the analogous modulator stages are dominant disturbers.

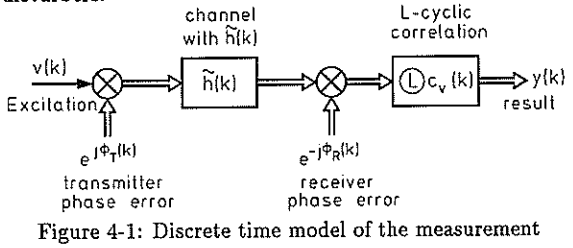


Figure 4-1: Discrete time model of the measurement

Based on perfect synchronization of receiver and transmitter, which is achieved by using rubidium frequency standards, a simple discrete-time equivalent base-band model particularizes the measurement (fig. 4-1). The excitation and correlation sequences are known from sect.2. The sequences $\Phi_T(k)$ and $\Phi_R(k)$ describe the noisy phase uncertainties in the transmitter and the receiver part of the measuring device. They are assumed to be wide sense stationary zero-mean white noise with variance σ_T^2 and σ_R^2 respectively.

A straight forward analysis leads to the following equation for the result of the measurement:

$$y(k) = \sum_{\kappa=0}^{L-1} \tilde{h}(\kappa) \cdot \left[v(k_{\mathrm{mod}L}) \cdot e^{j\Phi(k_{\mathrm{mod}L},\kappa)} \right] * c_v(k) \,, \quad \text{where } \Phi(k,\kappa) = \Phi_T(k) - \Phi_R(k-\kappa) \,. \tag{4.1}$$

Obviously there are no statistical bonds between the random processes in the transmitter and receiver. Thus the "overall" phase noise $\Phi(k,\kappa)$ is a sample sequence of a zero-mean random process with the variance $\sigma^2 = \sigma_T^2 + \sigma_R^2$.
Using the linear approximation $e^{j\Phi(k,\kappa)} \approx 1 + j\Phi(k,\kappa)$, which is valid if $|\Phi(k,\kappa)| \ll 1$ i.e. if $\sigma \ll 1$, $y(k)$ can be split up into a signal component, which identifies the result of ideal, noise free measurement, and into a noise component $n(k)$:

$$y(k) = (L+1) \cdot \tilde{h}(k) + n(k) \,, \text{ where} \tag{4.2}$$

$$n(k) = j \sum_{\kappa=0}^{L-1} \tilde{h}(\kappa) \overbrace{\sum_{\mu=0}^{L-1} \Phi(\mu,\kappa) v(\mu) c_v([k-\kappa-\mu]_{\mathrm{mod}L})}^{w(k,\kappa)} \,.$$

Equ.(4.2) holds in the range $0 \le k < L$. The noise component is essentially determined by a sequence $w(k,\kappa)$ consisting of the superposition of statistical independent events $\Phi(\mu,\kappa)$. It is a sample sequence of a zero-mean random process, the variance of which depends on the time instant $k-\kappa$:

$$\sigma_w^2(k-\kappa) = \sigma^2 \sum_{\mu=0}^{L-1} |v(\mu) \cdot c_v([k-\kappa-\mu]_{\mathrm{mod}L})| \tag{4.3}$$

$$= \sigma^2 \sum_{\mu=0}^{L-1} |A_o + pn(\mu)| \cdot |A_o + pn([\mu-k+\kappa]_{\mathrm{mod}L})| .$$

Taking into account the properties of PN-sequences [Wil76], equ.(4.3) is reduced by combinatorial considerations:

$$\sigma_w^2(k-\kappa) = \begin{cases} (L+1)\cdot\sigma^2 & , \text{if } [k-\kappa]_{\mathrm{mod}L} = 0 \\ (L - A_o^2 - 2A_o)\cdot\sigma^2 & \text{else.} \end{cases} \tag{4.4}$$

After these preparations we are able to focus on the quality of the output sequence $y(k)$. An informative measure for this quality is the ratio between the signal and the noise power:

$$\frac{S}{N} = \frac{(L+1)^2 \sum_{k=0}^{L-1} |\tilde{h}(k)|^2}{\sum_{k=0}^{L-1} |\sum_{\kappa=0}^{L-1} \tilde{h}(\kappa) \cdot w(k,\kappa)|^2} \tag{4.5}$$

$$\ge \frac{(L+1)^2 \sum_{k=0}^{L-1} |\tilde{h}(k)|^2}{\sum_{\kappa=0}^{L-1} |\tilde{h}(\kappa)|^2 \cdot [\underbrace{|w(\kappa,\kappa)|^2}_{<9\sigma_w^2(0)} + \underbrace{\sum_{k=0,k\neq\kappa}^{L-1} |w(k,\kappa)|^2}_{\approx (L+1)\cdot\sigma_w^2(k_{\mathrm{mod}L}\neq 0)}]} \,.$$

The random variable $w(\kappa,\kappa)$ is a superposition of L statistically independent events and therefore approximately

gaussian distributed. Thus $|w(\kappa,\kappa)|$ lies in the $3\sigma_w$ neighbourhood of its mean zero with a probability of 99.7%. Inequality (4.5) offers a lower S/N-bound, which depends on the variance of the overall phase noise:

$$\frac{S}{N} \geq \frac{1}{1.05 \cdot \sigma^2}\,, \qquad (L = 127)\,. \tag{4.6}$$

Using high quality frequency synthesizers for carrier generation both in the transmitter and in the receiver, a standard deviation $\sigma < 0.5^o$ is a realistic value. Therefore, taking into account only the effects of phase noise, a worst case S/N-ratio of better than 41dB in the measured impulse response can be achieved.

5 Echo Estimation

Physical measuring devices always cause linear distortions of low- or bandpass type. Therefore, time resolution of continuous-time impulse response measurements is principally bounded. The loss of all information related to frequency ranges outside the band of measurement is especially critical, if a system of large bandwidth like a multipath channel has to be measured. However, if some structural properties of the object under test are known a priori, these can be used to enhance the quality of the measurement results in a post processing step.
Multipath propagation in its idealized form is described by an impulse response of form (1.1). The exponential structure of the corresponding transfer function

$$H(j\omega) = \sum_{\nu=1}^{p} a_\nu e^{-j\omega\tau_\nu}\,. \tag{5.1}$$

is known a priori, whereas the number p of echo paths, their delay times τ_ν and their damping coefficients a_ν have to be determined from the measured data. To solve this frequency domain problem, all methods can be adapted, which were developed for the estimation of the frequencies and amplitudes of expontential time functions. For more details see [Mar89.1] and [Mar89.2].
Here we present a typical estimation result based on an impulse response measured in a rural area near Darmstadt. The impulse response consists of $L = 127$ complex valued, equally spaced samples at $t = kT$ ($T = 1\mu s$) formed by a direct path and by some reflections, which were caused by outlines of far distant villages (fig.5-1). The echo estimation

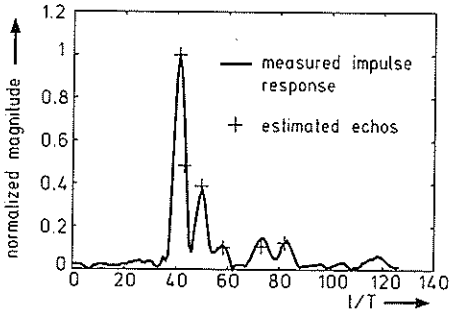


Figure 5-1: A measured impulse response and the corresponding result of echo estimation

was done using a total least squares modification [Rah86] of an eigenanalysis oriented linear prediction approach suggested in [Tuf82]. For the input $N = 49$ spectral samples were taken, calculated from the measured impulse response by means of DFT. The underlying model order was chosen to $n = 16$. By an eigenvalue analysis the echo number was estimated to be $p = 6$. The estimated delay times and amplitudes can be depicted from fig.5-1.
The estimation result offers an informative description of the propagation properties. Especially an echo is resolved, which was hidden in the measured impulse response by the direct path. Besides this quality improvement, the result possess a welcome feature. Compared with the measured impulse response consisting of 127 complex samples, it allows an approximative describtion of the channel behavior by 6 echo delays and damping factors only. This may be especially useful in connection with channel simulation or data bases, which intend to combine topographical arrangements and propagation characteristics. A comparison of magni-

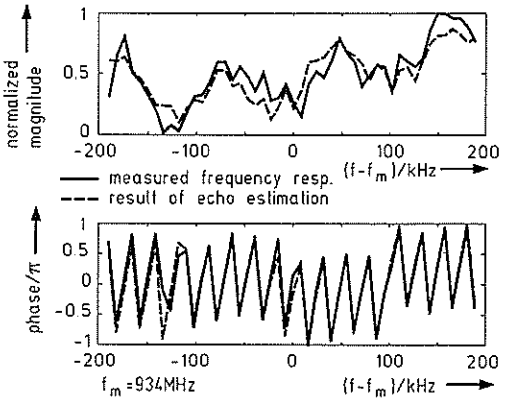


Figure 5-2: Magnitude resp. phase of the measured frequency response and of its estimated equivalent

tude resp. phase of the measured frequency response and of its estimated low parameter count equivalent illustrates the approximative character in the band of measurement (fig.5-2). The differences are mainly due to a mismatch of the underlying multipath model and due to noisy perturbations in the measured data, which tend to be reduced by the estimation algorithm employed.
Finally it should be noted, that the presented post processing cannot be performed on-line due to its high computation load.

Acknowledgement

This work was done based on a research contract with the Forschungsinstitut of the Deutsche Bundespost. The authors wish to acknowledge helpful discussions with Dr. R.W. Lorenz and his colleagues, as well as the support they got.

References

[Mar89.1] U.Martin, H.W.Schüßler: *"High resolution echo estimation - An application of Prony's method in the frequency domain"*, AEÜ, vol.43 (1989) 3.
[Mar89.2] U.Martin, R.Reng, H.W.Schüßler, K.Schwarz: *"Determination of Wide Band Impulse Responses"*, URSI ISSSE, Erlangen, September 1989, pp.393-396.
[Rah82] M.A.Rahman, Kai-Bor Yu: *"Improved Frequency Estimation Using Total Least Squares Approach"*, Proc. IEEE ICASSP, Tokyo, April 1986, pp.1397-1400.
[Tuf82] D.W.Tufts, R.Kumaresan: *"Estimation of Multiple Sinusoids"*, Proc. IEEE, Vol.70-9, 1982, pp.975-989.
[Rup89] J.Ruprecht: *"Maximum-Likelihood Estimation of Multipath Channels"*, ETH Zürich, diss. 1989.
[Wil76] F.J.MacWilliams, N.J.A.Sloane: *"Pseudo-Random Sequences and Arrays"*, Proc. IEEE, Vol.64, 1976-12, pp.1715-1729.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A NEW QUASI-ANALYTICAL SIMULATION METHOD FOR THE ESTIMATION OF ERROR RATE IN SATELLITE COMMUNICATION SYSTEMS

R. BAUDIN and F. CASTANIE, Member Eurasip.

GAPSE-ENSEEIHT
2 rue Camichel, 31071 TOULOUSE Cx, FRANCE

Abstract - In this paper, we deal with a new quasi-analytical method for bit error rate (BER) estimation in digital satellite communication systems. First we discuss the performances of the classical simulation methods and show their inefficiency when low bit error rate values have to be estimated. Then, we expound the principles of a modified quasi-analytical BER estimator and apply it to a typical nonlinear satellite link. Results obtained for realistic examples are presented.

I. INTRODUCTION

Computer simulation is a powerful tool for both analysis and design of digital communication systems. It is particularly the case for nonlinear satellite links because of the difficulty of analyzing in closed form such band-limited nonlinear systems. In fact, for these systems, no complete analytical treatment of the performance evaluation problem has yet been obtained.

Thus, a number of simulation techniques are available for estimating the symbol error probability, which is the most important parameter in the performance analysis of digital communication systems.

Applied to the typical satellite channel we describe in Section II of this paper, these techniques appear to be impractical or inaccurate when low error rate values have to be estimated (Section III). In order to overcome this problem we use a modified quasi-analytical estimator (Section IV) which allows accurate and fast evaluation of low bit error rate. The results we found for different channels are also presented in Section IV.

This work is a part of R. Baudin's thesis and has been supported by the French Space Agency (CNES) under contract N° 832 CNES 602400.

II. DESCRIPTION OF THE CHANNEL MODEL

The communication system we investigate is a two-link satellite channel shown in fig. 1. In such a system, the uplink channel is only critical because ground signal reception can be obtained with large gain antennas. Thus we only consider perturbations due to uplink white gaussian noise. The digital link is completely described using the block diagram shown in fig. 2.

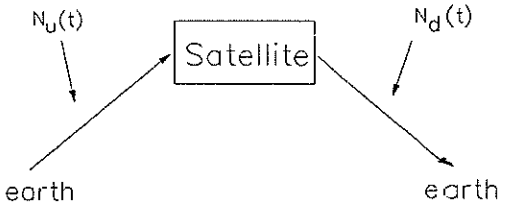


Fig. 1. The typical two-link satellite system

The system is characterized by quadrature phase shift keying (QPSK) modulation with a baud rate of $60 Mbit/s$, nonlinear amplification in the repeater by means of a travelling wave tube (TWT) operating near saturation, and bandpass filters. The noise is assumed to be zero mean white gaussian with variance σ^2 on each subchannel.

Numerical simulation of the system is performed using COMLIB [1], a software simulation package developed and tested at the French Space Agency (CNES). This package, which consists of a library of

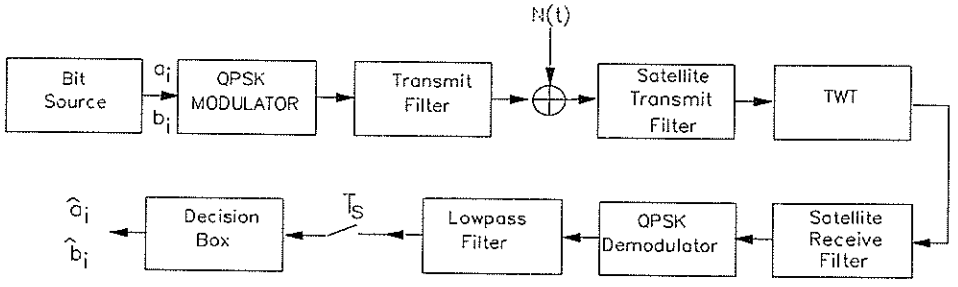


Fig. 2. Model for a QPSK satellite communication system

Fortran subprograms, uses a frequency domain approach and allows fast simulations of a large number of communication systems.

Applied to the case of our satellite link, COMLIB provides Butterworth and Chebyschev subprograms for filter simulation and different TWT subprograms based on experiment measurements. Furthermore, many classical techniques for bit error rate estimation (as for example, error counting or quasi-analytical estimation) are also available on this package.

In the following section we briefly discuss the performances of these methods.

III. CLASSICAL SIMULATION TECHNIQUES FOR BIT ERROR RATE ESTIMATION

The traditional Monte Carlo method, which basically consists of error counting, can be applied to nonlinear channels without any restrictive assumption [2] but it is impractical on account of the excessively large sample size required when low bit error rate values are desired. As an example, direct Monte Carlo simulation requires at least 10^7 bits to correctly estimate a typical error probability of 10^{-6}. Each bit requiring 10 samples, this means a computational burden of 10^8 samples per simulation run.

Several techniques are available for reducing the sample size requirement. Two of these are based on the asymptotic approximation of the probability density function tail [3]-[4] and allow a sample size reduction by a factor of about 10 .

Another well-known method called Importance Sampling [5] leads to a significant reduction of the sample size, but only in the case of short memory length channels. For common systems with memory length equal to four or five bits, the sample size reduction is of the same order as with the other techniques.

The quasi-analytical method is by far the fastest one because the noise is not simulated; it is just taken into account by means of the following error probability formula :

$$P_e = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}erfc\left[\frac{x_n}{\sigma\sqrt{2}}\right] \qquad (1)$$

where P_e is the error probability evaluated on the in-phase (or quadrature) channel and $\{x_n\}_{n=1..N}$ are the noiseless samples at the channel output.

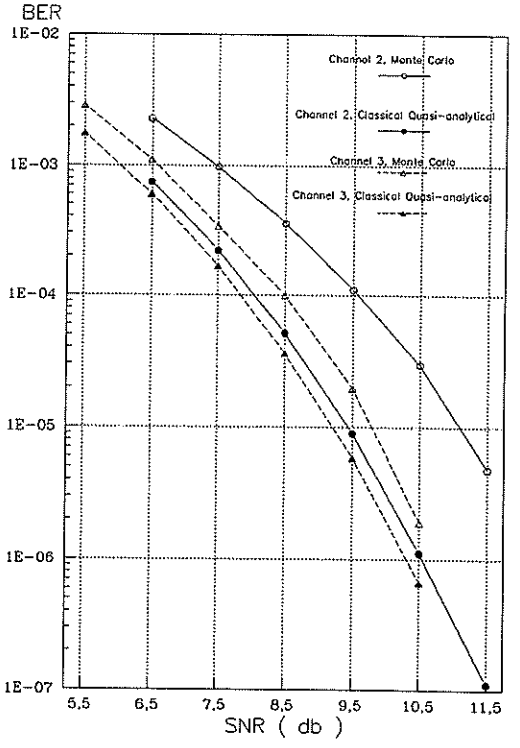


Fig. 3. Comparison of the classical quasi-analytical method with the Monte Carlo method

Since the noise is not simulated, only a few bits are required to obtain the probability of error [2]. However, equation (1) implies that the noise is assumed to be **gaussian at the output** of the channel. This assumption is not true for nonlinear channels with memory, therefore the BER estimations are not accurate.This inaccuracy can easily be seen on fig. 3. where, for comparison, Monte Carlo BER curves and quasi-analytical curves are plotted for two different channels (these are described below). Compared to Monte Carlo results, classical quasi-analytical estimations can lead to an error of about one power of ten for a BER value of 10^{-6}.

In order to improve the accuracy of the method, a modified quasi-analytical bit error rate estimator, which is convenient for nonlinear channels with memory, is dealt with in the next section.

IV. PERFORMANCES OF THE MODIFIED QUASI-ANALYTICAL TECHNIQUE

We only give here the principles and results concerning the modified quasi-analytical estimator. All theoretical justifications are given in [6].

The problem of evaluating the global BER on a QPSK system can be reduced [2] to the computation of the error probability on each (in-phase and quadrature) subchannel. Thus we only derive formulas for in-phase channel and then compute the associated bit error rate.

Let P_e be the in-phase error probability and assuming the "zeroes" and "ones" have equal a priori probabilities, then :

$$P_e = \frac{1}{2}P_0 + \frac{1}{2}P_1 \qquad (2)$$

where P_0 and P_1 are the conditional error probabilities given that a "zero" or a "one" has been sent.

Now let Z_n be the signal sample at the in-phase channel output and let X_n be the sample which should be obtained at the output **if there were no noise.** Then we define a sample N_n that for convenience we call "noise sample" and whose value is :

$$N_n = Z_n - X_n \qquad (3)$$

Using simulation we can easily obtain X_n and Z_n, and consequently the N_n samples.

It is shown in [6] that efficient estimators of the error probabilities P_0 and P_1 are given by :

$$P_0 = \frac{1}{N}\sum_{n=1}^{N} P_{0n}(X_n) \qquad (4a)$$

$$P_1 = \frac{1}{N}\sum_{n=1}^{N} P_{1n}(X_n) \qquad (4b)$$

where :

$$P_{0n}(X_n) = \int_{-\infty}^{V_T - X_n} p(N_n/X_n)\,dN_n \qquad (5a)$$

$$P_{1n}(X_n) = \int_{V_T - X_n}^{+\infty} p(N_n/X_n)\,dN_n \qquad (5b)$$

$p(N_n/X_n)$ is the conditional probability density function (p.d.f) of N_n for a given value of the noiseless sample X_n and V_T is the decision threshold.

As in traditional quasi-analytical method, a small number of bits N is sufficient to correctly evaluate P_0 and P_1.

In order to compute the probabilities $P_{0n}(X_n)$ and $P_{1n}(X_n)$ we may have to estimate N p.d.f $\{p(N_n/X_n)\}_{n=1..N}$. This can be achieved using a Gram Charlier expansion of these p.d.f and computing their conditional moments by means of a second-order development of the two variable function associated with the TWT.

The final result is [6] :

$$P_{0n} \approx \frac{1}{2}erfc\left[\frac{X_n + m_n - V_T}{\sigma_n\sqrt{2}}\right] - c_{3n}\phi^{(2)}\left[\frac{V_T - X_n - m_n}{\sigma_n}\right] + c_{4n}\phi^{(3)}\left[\frac{V_T - X_n - m_n}{\sigma_n}\right] - \ldots \qquad (6a)$$

$$P_{1n} \approx \frac{1}{2}erfc\left[\frac{V_T - X_n - m_n}{\sigma_n\sqrt{2}}\right] + c_{3n}\phi^{(2)}\left[\frac{V_T - X_n - m_n}{\sigma_n}\right] - c_{4n}\phi^{(3)}\left[\frac{V_T - X_n - m_n}{\sigma_n}\right] + \ldots \qquad (6b)$$

where :

$$\phi(x) = \frac{1}{\sqrt{2\pi}}e^{-x^2/2}$$

$$\phi^{(k)}(x) = (-1)^k H_k(x)\phi(x)$$

$$c_{3n} = \frac{M_{3n}}{3!\,\sigma_n^3}$$

$$c_{4n} = \frac{1}{4!}\left(\frac{M_{4n}}{\sigma_n^4} - 3\right)$$

.....

$H_k(x)$ is the Hermite polynomial of degree k and m_n,

σ_n^2, M_{3n}, M_{4n}, ... are the conditional central moments of N_n.

As is shown in [6], these results are only available under the assumption of **low noise** i.e. **low error rate**, which is the case that interests us.

Since the computational complexity increases exponentially with moment order, the expansion has to be limited to three terms. Fig. 4 shows the results obtained for three different channels.

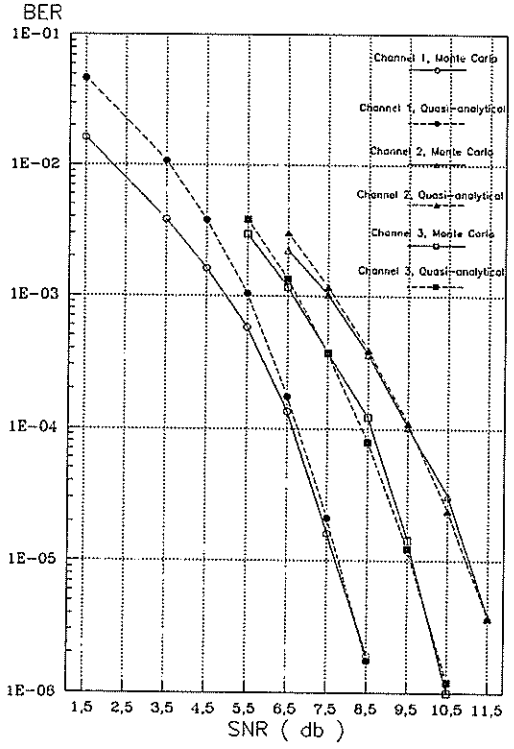


Fig. 4. Comparison of the modified quasi-analytical method with Monte Carlo method

For each channel, the baud rate was set to $60\,Mbit/s$. Transmit filter bandwith was $100\,Mhz$ for Channel 1 and $80\,Mhz$ for Channels 2 and 3 and two different kinds of TWT were used for Channels 1,2 and 3. The Monte Carlo curves were plotted using **one million bits** for error counting and for the quasi-analytical curves, only **127 bits** were used.

In fig. 4 the accuracy of our method can be seen to be quite good for BER values below 10^{-4} although only three terms of the Gram Charlier expansion are used.

It is worthy of note that the additional computational load required by equation (6) is independant of the target BER, which is not the case for methods based on error counting. For a BER of 10^{-6} we have found our method to be more than 100 times faster than the traditional Monte Carlo method. The simulation programs are embedded on a CYBER calculator and the computer time does not exceed 100 seconds CPU for each estimation of the BER.

V. CONCLUSION

We have described and applied a new useful method for the estimation of low bit error rate in digital satellite links. This method is based on a modified quasi-analytical estimator of the BER and appears to be much more accurate than the classical one in the case of nonlinear communication channels.

The results we obtained are found to have a good accuracy for BER values below 10^{-4}. Furthermore the method appears to be over 100 times faster than the Monte Carlo method for a typical bit error rate of 10^{-6}.

Acknowledgement

The authors would like to thank Mr J. Sombrin of the CNES for his advice and help in carrying out the present work.

References

[1] J. Sombrin, H. Donat, "Documentation du logiciel COMLIB," CNES, 1981.

[2] M.C. Jeruchim, "Techniques for estimating the bit error rate in the simulation of digital communication systems," *IEEE Trans.*, Vol. SAC-2, Jan. 1984, pp. 153-170.

[3] S.B. Weinstein, "Estimation of small probabilities by linearization of the tail of a probability distribution function," *IEEE Trans.*, Vol. COM-19, Dec. 1971, pp. 1149-1155.

[4] S.B. Weinstein, "Theory and application of some classical and generalized asymptotic distributions of extreme values," *IEEE Trans.*, Vol. IT-19, March 1973, pp. 148-154.

[5] K.S. Shanmugam, P. Balaban, "A modified Monte Carlo simulation technique for the evaluation of error rate in digital communication systems," *IEEE Trans.*, Vol. COM-28, Nov. 1980, pp. 1916-1924.

[6] R. Baudin, F. Castanié, "A quasi-analytical simulation method for bit error rate estimation in nonlinear digital satellite links," Internal Report, submitted for publication.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Techniques for the Efficient Simulation of Communication Systems

Letizia Lo Presti and Marina Mondin
Dipartimento di Elettronica, Politecnico di Torino
C.so Duca degli Abruzzi, 24, 10129 Torino (ITALY)

In this paper, techniques for the efficient time-domain simulation of communication systems are proposed and analyzed. Multirate sampling techniques are described, that can allow every system block or section to be simulated using the more appropriate sampling rate, avoiding oversampling. The use of decimated filters is introduced, as well. In order to speed-up the filtering operations, the use of block processing techniques is described, that can exploit the very powerful capabilities of FFT algorithms. The algorithm performances are evaluated with the package for the time-domain simulation of communication systems TOPSIM IV [5].

1 Introduction

Computer simulation is a very useful tool for both analysis and design of communication systems [6]; CPU time is often a critical point of this approach. This paper deals with some of the techniques that can be used to minimize the required CPU time in a time domain simulation environment. The oversampling of the low-rate sections can be avoided by means of multirate simulation and/or the use of decimated filters. For filter simulation, although time-domain simulation is considered, efficient filtering operations can be performed in the frequency domain. For this reason, the use of block processing is introduced, which is equivalent to perform frequency domain filtering into a time-domain environment.

All these signal processing techniques have been implemented within the framework of TOPSIM IV [5] simulation package.

2 Multirate simulation

In time-domain simulation the required sampling rate depends on the bandwidth of the system under test, and, in particular, the sampling rate must satisfy the sampling theorem for the section with the largest bandwidth. This choice can result in a large oversampling of the low-frequency sections; this problem can be avoided using different sampling frequencies within the same simulation program.

Two types of signals are present in telecommunication systems: baseband and bandpass. Baseband signals are represented by real samples, while the analytic signal representation [9] is used for bandpass signals, that are therefore represented by complex samples.

Two main problems must be solved in multirate simulation: the first is the interpolation of bandlimited signal when sampling rate must be increased, the second is the conversion of narrowband signals from real to analytic representation, when the sampling rate must be decreased.

2.1 Increasing sampling rate

Let us denote with T_c the sampling time corresponding to the high sampling rate signal $x(t)$, and with $T_c' = KT_c$ the sampling time corresponding to the low sampling rate signal $x'(t)$, where K is the expansion ratio, assumed as integer in the range $[10 \div 1000]$. We want to derive the samples $x(nT_c)$ of $x(t)$ from the available samples $x'(nT_c')$ of $x'(t)$.

The first algorithm that was tested was an interpolation based on the linear combination of a limited number of past low-rate samples [10]. This tecnique was very robust, but also very expensive in terms of memory occupation, and has therefore been discarded.

Given a samplig interval T_c', the simulation bandwidth (referred in the following as "baseband") is equal to $1/2T_c'$. If the signal must flow through a section with higher sampling rate, the simulation baseband in this section will be larger, and some of the replicas may fall within the new baseband, giving rise to some distortion. A suitable low-pass filter can cut off the unwanted replicas, reducing such distortion. A possible conversion algorithm can therefore be performed creating an intermediate high-speed sequence of samples by placing $K-1$ null samples between the samples of the input low-speed sequence and then filtering it by a low-pass filter, in order to reduce the unwanted spectral replicas. The second algorithm that has been examined uses in fact a suitably designed elliptic interpolation filter. The choice of the elliptic filter family is based on the necessity of controlling the in-band ripple A_{max}, the stop-band attenuation A_{min}, the lower frequency in the transition bandwidth, f_{PB}, and the upper frequency in the transition bandwidth, f_{SB}. In our case, a filter identified by $A_{max} = 0.1$ dB, $A_{min} = 40$ dB, $f_{PB}T_c' = 0.55$, $f_{SB}T_c' = 0.7875$ has been selected as interpolator. The filter order is 7, and therefore the number of operations (additions and sums) that must be performed at every conversion step is reasonably low. For further details about this method, refer to [10].

2.2 Decreasing samplig rate

In this case, when both the high-rate and the low-rate samples are real, or they are both complex, the output low-rate samples are obtained decimating the original high-rate input sequence. A different approach should be used when the input signal is a bandpass one and its spectrum is inside the high-rate baseband and outside the low-rate baseband, so that a change in the signal representation, from the baseband to the analytic one, is required. In order to perform this conversion, the evaluation of the Hilbert transform of the input signal is required.

Two different structures, IIR and FIR, have been tested. The IIR structure approximates a narrowband Hilbert trans-

former. The narrowband characteristic is not a limitation because all the signals described by means of the analytic signal representation are bandlimited. As the transfer function $H_h(f)$ of the Hilbert transformer is equal to $j\mathrm{sgn}(f)$, it corresponds to a system that performs a phase shift equal to $-\pi/2$ on all the signal components. To perform this kind of phase-shift, two second order digital unit amplitude transfer functions $H_i(z)$ have been designed

$$H_i(z) = \frac{\alpha_i + \beta_i z^{-1} + z^{-2}}{1 + \beta_i z^{-1} + \alpha_i z^{-2}} \quad i = 1, 2,$$

where α_i and β_i, $i = 1, 2$, are chosen in such a way that $H_1(z)$ and $H_2(z)$ present a phase difference of $\pi/2$ inside the signal bandwidth. The signals at the output of these two filters are then linearly combined to get the high-rate complex envelope representation of the incoming signal [10]. To obtain the required low-rate complex envelope representation, decimation is performed. This procedure is computationally very simple, and it requires 16 products and 8 addition for every conversion step. The drawback of this approach is that, as the considered structure contains IIR filters, the conversion procedure must be executed at high-rate. This drawback can be avoided using a FIR structure, because non-recursive filters allow automatic decimation.

For this reason, a second algorithm, based on the truncation of the impulse response, has been tested. It consists in the approximation of a Hilbert transformer by means of a FIR structure.

The ideal impulse response $h_h(n)$ of a digital Hilbert transformer is

$$h_h(n) = \begin{cases} \frac{2\sin^2(\frac{\pi n}{2})}{n\pi} & n \neq 0 \\ 0 & n = 0 \end{cases} \qquad (1)$$

In order to ensure the filter causality, the impulse response $h_h(n)$ must be simmetrically truncated from $n = -M$ to $n = M$ and delayed of M samples. In this way, a new impulse response $h_d(n)$ will be obtained, whose samples are the required values of the FIR filter taps. A value of $M = K$ has been chosen, so that the approximated Hilbert transformer introduces a delay $T_c' = KT_c$ on the converted signal, equal to one low-rate sampling interval. The obtained $h_d(n)$ function has very desirable properties: it has odd simmetry, and half of the coefficients are zero. For this reason, every converted sample requires $\lfloor \frac{N+1}{2} \rfloor$ products and $\lfloor \frac{K+1}{2} \rfloor$ addition, for a total of $\lfloor K+1 \rfloor$ products and $\lfloor K+1 \rfloor$ addition for every conversion step. Although for high values of K the considered FIR filter requires a large number of operations, all the sum and multiplications must be performed at low-rate, i. e. once every K simulation steps, while during all the other steps the filter memory register is updated, but no other operations are performed. This procedure makes the FIR structure from 10% to 35% faster than the IIR narrowband structure.

The FIR Hilbert transformer described in the previous section is an example of decimated filter. In a decimated filter, the output samples are evaluated only when these samples are effectively used by the simulation program. In all the other cases, the filter memory is updated, but all the additions and products required by the evaluation of the output sample are avoided.

2.3 Simulation tests

The described multirate sampling techniques have been tested, using filtered Gaussian noise as test-signal, and varying its spectral position in order to evaluate the system performances in all the possible situations [10]. The algorithm performances are evaluated by means of the fidelity criterion based on the measure of the correlation coefficient between the original and the converted signal (see [10] and the references therein), which provides a performance measure expressed as a signal-to-noise ratio S/N. In the case of the Hilbert transformer, the complex envelopes of the original and of the converted signal have been correlated.

The signal to noise ratios obtained in these tests for the different conversion algorithms, are summarized in Tab. 1. The value f_x/B_x represents the ratio between the center frequency and the signal bandwidth.

a

K	(S/N)
10	40.3
20	39.1
40	37.6
50	36.6
60	35.9
80	35.0
100	35.4
300	35.4
900	34.3

b

K	f_x/B_x	FIR	IIR
10	10	40.0	37.3
	12.5	37.6	30.2
50	50	43.4	40.2
	62.5	43.0	40.0
100	100	41.9	35.5
	125	40.4	37.3
500	500	33.9	32.9
	625	33.2	31.6
1000	1000	25.3	25.2
	1250	35.5	33.0

Table 1: (a) Performances of the low-rate to high-rate conversion by means of an elliptic interpolating filter.
(b) Performances of the high-rate to low-rate conversion by means of approximated Hilber transformers.

2.4 CPU-time consumption

The saving in CPU time obtained with multirate sampling techniques depends on the simulated system structure and on the value of the compression ratio. Let's imagine to simulate the cascade of three different blocks, denoted as (1), (2) and (3). The generic i-th block performs N_i elementary operations on every signal sample, and each operation requires an amount of time equal to T_i. Without multirate simulation, the three blocks are simulated with rate f_c, where $f_c = 1/T_c$. If multirate simulation is used, blocks (1) and (3) are simulated at low sampling rate f_c/K, and block (2) is simulated at high sampling rate f_c. If N_t denotes the total number of samples, the total time required for the simulation without multirate sampling techniques can be expressed as

$$T_{totA} = N_t[N_1T_1 + N_2T_2 + N_3T_3].$$

If T_k denotes the conversion time required for one sample, the total time required for the simulation with multirate sampling techniques can be expressed as

$$T_{totB} = N_t[N_1T_1/K + N_2T_2 + N_3T_3/K + T_k].$$

Multirate sampling techniques can save CPU time if the quantity T_{totB} is less than T_{totB}, that is

$$K > \frac{N_1T_1 + N_3T_3}{N_1T_1 + N_3T_3 + T_k}$$

$$T_k < \frac{K-1}{K}(N_1T_1 + N_3T_3).$$

The last condition shows that multirate sampling tecniques can led to a meaningful time saving if blocks (1) and (3) perform time consuming operations, like, for instance, filtering.

Say T_L the CPU time required by the low sampling rate section without multirate sampling, and T_H the CPU time

required by that at high sampling rate, the maximum reduction factor in computation time is $1 + T_L/T_H$, which is significant only if $T_L > T_H$. The reduction factors obtained in the simulation of pratical systems, including coding and spread-spectrum, are in the range of 3 to 5.

3 Filter simulation

The simulation of filters is based on the *simulation theorem* described in [1]. This theorem states that a linear analog system $H_a(f)$ with input $x(t)$ and output $y(t)$ can be simulated by means of a digital filter (called *digital simulator* or *digital model*), with a proper transfer function $H(z)$ and input signal $x[n] = x(nT_c)$. It can be shown that the output $y[n]$ of $H(z)$ equals the sample values $y(nT_c)$ of $y(t)$, if $x(t)$ is bandlimited, i.e., if its spectrum $X(f) = 0$ for $|f| > B_x$, and $H(z)$ is such that

$$H(e^{j2\pi fT_c}) = H_a(f) \quad \text{for} \quad |f| < B_x. \tag{2}$$

From (2), that represents the *simulation condition* for $H_a(f)$, we have that the simulation of filters requires the design of a digital system with a transfer function $H(z)$ assigned in the unit circle. In practical cases, $H(z)$ is a rational function, which can be represented, in the discrete-time domain, by means of a recursive equation of the type

$$\begin{aligned} y[n] \;=\; & a_0x[n] + a_1x[n-1] + ... + x_M[n-M] \\ & + b_1y[n] + ... + b_My[n-M] \end{aligned} \tag{3}$$

relating $x[n]$ and $y[n]$. Equation (3) implements a time-domain simulation of filters, as it allows the evaluation of a single output sample at each simulation step.

Two types of digital models can be used: *a*)IIR structures, easily evaluated only when the transfer function of the analog filter is known in the complex plane s; *b*) IFIR structures (Interpolated FIR, as described in [2,3]), used when only $H_a(f)$, or equivalently $H_a(\omega)$, is known, and $H_a(s)$ is unknown.

These methods are widely used in many simulation packages ([5,6]) and give very accurate results in most cases. Fast results are obtained with IIR models, while IFIR filters are generally more time-consuming. In this last case, another possible approach is to perform the filtering process in the frequency domain, so exploiting the very powerful capabilities of FFT algorithms.

This filtering method, based on FFT, processes a block of N input samples simultaneously, and therefore it can not produce one output sample at each simulation step, as recursive equation (3) does. For this reason we call this type of filter simulation *block processing method*. Section 4 is devoted to a detailed description of this method.

4 Block processing simulation

The evaluation of the output of a filter with transfer function $H_a(f)$ and input $x(t)$ can be digitally performed in the frequency domain, by using DFT (*Discrete Fourier Transform*) techniques as follows:

1. a sequence of N samples $x(nT_c)$ of $x(t)$ is transformed by FFT, producing a sequence of frequency samples $\overline{X}(nf_0)$, being $f_0 = 1/NT_c$. Notice that $\overline{X}(nf_0)$ is not exactly the sequence of the samples of the input spectrum $X(f)$, because of the aliasing.
2. N samples of $H_a(f)$ are evaluated in the Nyquist bandwidth $(-1/2T_c, 1/2T_c)$, obtaining a sequence $H_a(nf_0)$, which must be properly ordered as required by FFT. The ordered sequence, denoted by H_n, is multiplied by $\overline{X}(nf_0)$, producing a sequence of N frequency samples $\overline{Y}(nf_0)$.
3. A sequence of N output samples $\overline{y}[n]$ is obtained by an inverse FFT of $\overline{Y}(nf_0)$

We call this filtering technique *N-FFT method*. In practical cases, this method requires some modifications due to the following reasons.

- The described N-FFT method corresponds to perform a circular convolution of $x[n]$ with $h[n]$, which is the inverse discrete Fourier transform (IDFT) of H_n. Filtering must be performed by means of a *linear convolution*, then $x[n]$ and $h[n]$ must be zero padded before performing DFT techniques. This operation is well-known and its description can be found, for example, in [4]. The result of the zero padding is the use of a $2N$-FFT technique.
- In simulation the sequence $x[n]$ is generally very long and then the $2N$-FFT method would require a large amount of memory. This can be avoided, as $x[n]$ is generally much longer than $h[n]$ and then the *overlap-add* technique (see, for example, [4]) can be used. With this method the value of N depends only on the length of $h[n]$.

We mean by *block processing* the simulation of filters based on a $2N$-FFT method with zero padding and overlap-add. In the following, we will use the term *block length* to indicate the number N of samples simultaneously processed.

4.1 The simulation condition in block processing mode

Now we have to answer the following question: *if $y(t)$ is the analog output of $H_a(f)$, which is the condition to have $\overline{y}[n] = y(nT_c)$?*

We observe that the block processing method corresponds to a linear convolution of $x[n]$ with $h[n]$, and then to a digital filtering with a FIR filter with N coefficients equal to $h[n]$. For the simulation theorem, to obtain $\overline{y}[n] = y(nT_c)$, the frequency response of this FIR filter must be equal to the analog transfer function $H_a(f)$ for $|f| \leq B_x$.

Therefore we have that the definition of an accurate digital model is the main problem also in the frequency domain simulation of filters. The quality of the results depends on the capability of designing a proper FIR model. We could use this model in a time-domain simulation by filtering with a linear convolution and we would find exactly the same results obtained with block processing. Of course block processing is computationally more efficient.

4.2 The choice of the block length

Both the model quality and the CPU time depend on the block length N and on the sampling frequency $f_c = 1/T_c$. Here we suppose to have chosen a proper value of f_c and we only consider the parameter N. To improve the model, it is generally necessary to increase N, until condition (2) is adequately satisfied. A further increase of N would be useless.

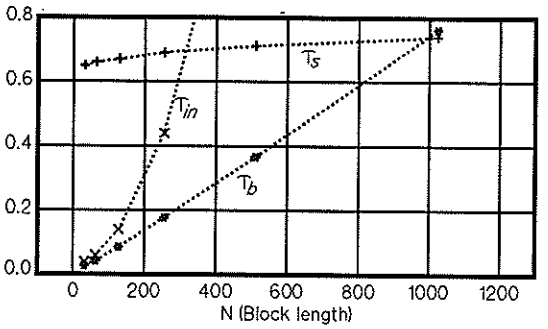


Figure 1: CPU time in block processing mode

The relationship between N and CPU time is not so simple. Let us denote with τ_{in} the CPU time required to evaluate the filter model and to perform some checks on the model quality and with τ_L the CPU time necessary to process L samples. We can reasonably foresee that τ_L mostly depends on the time spent, every N simulation steps, to perform all the FFT operations, while the time consumed to store the N samples of each block is negligible. We denote with τ_b the CPU time spent for processing one block and with N_b the number of blocks to be processed in L simulation steps. We have $N_b = L/N$, and

$$\tau_L = N_b \tau_b = \frac{L\tau_b}{N}. \tag{4}$$

By observing that τ_b is proportional to $N \log_2 N$, from (4) we find that τ_L is weackly dependent on N. In Fig. 1 we have an example run on VAX 8800, with $L = 70,000$; the curves represent the values of τ_b, $\tau_s = \tau_L/70$, and τ_{in}; the filter has a raised cosine transfer function, [7]. As expected, τ_s is quite constant, while τ_{in} increases as N increases; therefore we see that it is better to choose short blocks. The best choice for N will be the minimum value giving an adequate model.

5 Comparison of digital filtering methods

Time domain filter simulation and block processing are both implemented in TOPSIM IV.

The two methods have been tested and very accurate results have been found in both cases. Notice that in most cases the problem is not the quality of the model, but the CPU time required for producing the filter output.

To compare CPU times, let us consider a lowpass Chebyshev filter, with bandpass edge normalized to 1Hz, simulated with the following three methods: *a*) time domain simulation with an IIR model (obtained by bilinear transformation, as described in [8]); *b*) time domain simulation using an IFIR model, obtained as described in [3]; *c*) block processing. Figure 2 shows τ_L ($L = 100,000$) versus f_c. We find constant values for the case *a* and *c*; in case *b* we have variables values, as the complexity of the IFIR structure depends on f_c. We observe that the fastest simulation is obtained with an IIR model.

Many other examples have been considered, which allows us to conclude that IIR models are the fastest ones. Unfortunely IIR models are easily found only when the filter is known in the Laplace plane. Moreover they can not simulate linear phase filters. When an IIR model can not be found, block processing mode is generally the best solution. Accurate results can be obtained in both cases by a proper choice of the digital model.

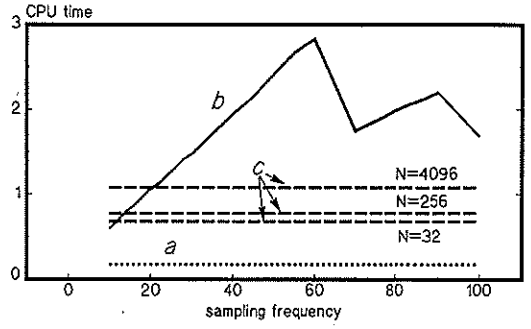


Figure 2: CPU time for a Chebyshev filter

6 Conclusions

Analysis and design of complex communication systems require, very often, digital computer simulation, that can be very CPU-intensive. In the package TOPSIM IV, some techniques have been implemented, that can dramatically reduce the CPU time requirements of simulation. In this paper we described, in particular: a) multirate sampling techniques, that allow the use of different sampling frequencies, within the same simulation program; b) block processing techniques, that allow frequency domain filtering operations (exploiting the very powerful capabilities of FFT algorithms), into a time domain simulation environment.

References

[1] **A. Papoulis,** *Signal Analysis,* McGraw-Hill Book Company, New York, 1977

[2] **Yrjö Neuvo, Domg Cheng-Yu, Sanjit K. Mitra,** *Interpolated Finite Impulse Response Filters,* IEEE Trans. on Acous., Speech, and Sign. Proc., June 1984

[3] **L. Lo Presti,** *Time-Domain Simulation of filters defined by a set of measured points,* MELECON '85

[4] **E. O. Brigham,** *The Fast Fourier Transform and its Applications,* Prentice-Hall International, 1988

[5] **M. Ajmone Marsan, et al.,** *Digital Simulation of Communication System with TOPSIM III,* IEEE Jour. on Select. Areas Comm., Vol. SAC-2, Jan. 1984

[6] Special Issue on Computer-Aided Modeling, Analysis and Design of Communications Systems, IEEE J. Select. Area Comm., Vol. SAC-2, Jan. 1984

[7] **S. Benedetto, E. Biglieri, V. Castellani,** *Digital Transmission Theory,* Prentice-Hall, Englewood Cliffs, NJ, 1987

[8] **L. Lo Presti,** *Prewarping Techniques in Filter Simulation by Bilinear transformation,* Signal Processing Vol. 5, No. 6, Nov. 1983

[9] **S. O. Rice,** *Envelopes of narrowband signals.* Proc. IEEE. vol.70, pp. 692-699, July 1982.

[10] **M. Pent, L. Lo Presti, M. Mondin and L. Zaccagnini,** *Multirate sampling techniques for simulation of communication systems,* IASTED Int. Symp. on Modelling, Ident. and Control, Grindelwald, Switzerland, Feb. 1987

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

DATA COMMUNICATION RECEIVERS BASED ON NEURAL NETS

L. Diez del Rio, S. Martínez Contreras[1] and J. Gómez Mena
ETSI de Telecomunicación Ciudad Universitaria
28040 Madrid

Neural Nets have been aplied in many fields. That is because his verastility. In this paper we show that neural nets could be used at the receiver of a data communication system, to compensate the disturbance caused by linear and non-linear channels. First we consider a model for a non-linear channel with memory. Then we describe the architecture of a receiver ussing time delay neural nets (TDNN). After that we present some experimental results.

1. INTRODUCTION

Intersymbol interference, ISI, in data communications is one of the main factors which limitate the data throughput, another one is the noise. Usually receivers deal with the first impairment by mean of adaptive filters which cancel ISI trying at the same time to maintain the noise at enough low levels. These filter can operate at symbol rate, the T equilizers in both feed-forward or feed-back configuration [1], or frationally spaced operating at a fraction of the symbol rate [2].

All of these receivers are based on the important fact that the channel behavior is linear. When non-linear effects are present another kind of receivers have to be used, for instance those based on Volterra series,[3]

In this paper we are going to consider the applicability of neural nets to the design of equilizers for both linear and non-linear channels. Many common equalizers can be considered as special cases of neural nets, as have been shown in [4]. Inputs to the neural nets will be the received signals and delayed version of it. We call to this nets, following [5], time-delay-neural nets, TDNN. Particullarly we analyze the base-band transmission in one dimenssion.

Let a(n) be the sequence of transmitted symbols, a(n) in the unidimensional case can be defined as belonging to the following set:

$$a(n) \in \{\pm 1, \pm 2, \pm 3 ..... \pm M\} \qquad (1)$$

We concentrate in the case of M=1 Additionally we assume that the receiver signal is sampled at baud rate with a fixed phase.

In the next paragraph we defined the channel model that we study. The following one includes a descripcion of the nets to be ussed and the training algorithm. We finalized the paper with a descripcion of the experimental results obtained.

2. CHANNEL MODEL:

Let us to consider the transmission of a pulse amplitude modulated signal defined as:

$$x(t) = \sum_{k=-\infty}^{\infty} a(k)p(t-kT) \qquad (2)$$

where *a(n)* is the coordinated of transmited symbols as in (1) and p(t) is the basic pulse, for instance a raised cosine one. When we transmit this signal through a time invariant linear channel we obtain:

$$y(t) = \sum_{k=-\infty}^{\infty} a(k)q(t-kT) \qquad (3)$$

[1] Now with PAGE IBERICA S.A; Av. Industria 26 Polígono Tres Cantos. 28760 Madrid.

This work was supported by PLANYCIT under project: Tratamiento Avanzado de la Información

where q(t) is the convolution of p(t) with the impulse response of the time-invariant channel. Additionally we include in the model a non-linear filter without memory. We assume that its output signal w(t) has a polynomic relationship with y(t):

$$w(t) = \alpha_0 y(t) + \alpha_1 y^2(t) + \alpha_2 y^3(t) \qquad (4)$$

This analogycal channel model is represented in Figure 1.

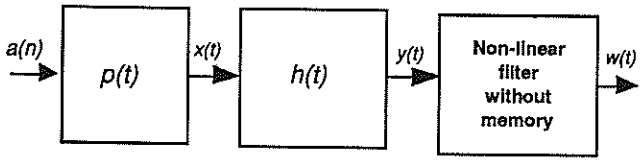


Figure 1: Continous-time model of the communication channel

If we sample this signal at time instants $t_n = nT + a$ whe obtain the sequence:

$$z(n) = w(nT + a) \qquad (5)$$

Then $z(n)$ can b modelled by mean of a linear time invariant filter and a discrete-time without memory non-linearity as it is shown in Figure 2.

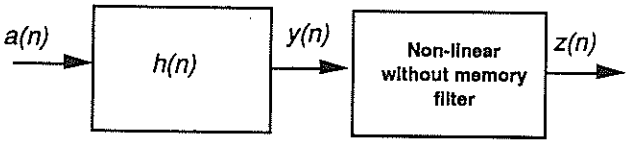


Figure 2:Discrete -time model of the channel

3.- ARCHITECTURE OF TDNN BASED RECEIVER

From the received sequence z(n) we try to decide which symbol has been transmitted. The signal z(n) and its delayed samples are processed by the neural net. Then, the input to this net is the set of samples: z(n-L), z(n-L-1).....z(n+L) when the symbol n has to be estimated. L is selected in accordance with the channel memory. Detected symbols can also be ussed as additional inputs to the TDNN . In this paper we consider the problem of deciding the symbols in a feed-forward way, without decission feedback. This decission scheme is represented in Figure 3.

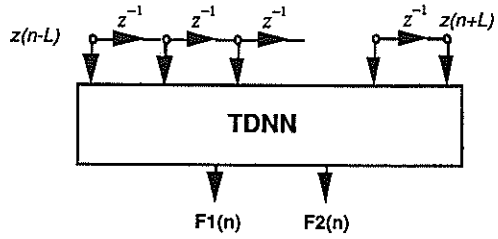


Figure 3: A TDNN based receiver

This net is trained in order to obtain:
- When a(n)=1 then F1(n)=1 and F2(n)=0
-When a(n)=-1 then F1(n)=0 a nd F2(n)=1

We use a multilayer net [ ] . Each neuron in the input layer is connected to all neurons in the first layer. The outputs of this layer is connected to the following layer, each neuron output with all the neurons in the next layer. In the same way the additional layers except the output layer that has only two neurons. Each with F1 or F2 output.

Each neuron has the structure shown in Figure 4 and the network architecture is shown in Figure 5. Observe that we use a neurons with a non-linearity that has derivatives. In this way gradient type algorithms can be used for training. An interesting alternative is to have hard non-linearities, such as in [5].

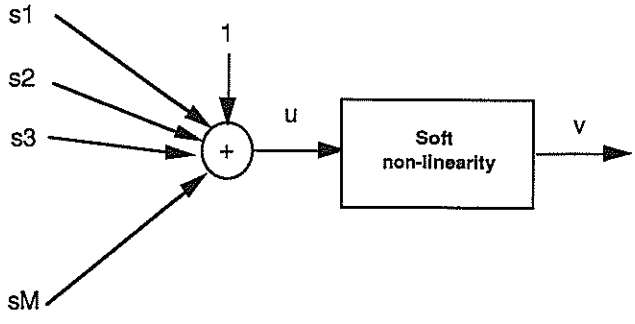


Figure 4: Neuron structure.

The soft or sigmoid non-linearity has the following input-output relationship:

$$v = \frac{1}{1+e^{-u}} \qquad (6)$$

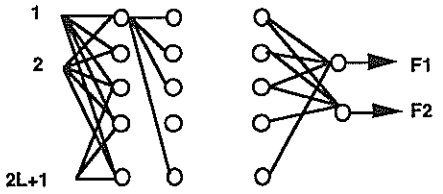


Figure 5: Neural network architecture

The training algorithm is the modified back-propagation algorithm [6] which formulation is as follows:

$$w_{ij}(n+1) = w_{ij}(n) + \Delta w_{ij}(n) \qquad (7)$$

$$\Delta w_{ij}(n) = -\eta \frac{\partial E}{\partial w_{ij}} + \alpha \Delta w_{ij}(n-1) \qquad (8)$$

$$0.05 \le \eta \le 0.75$$

$$0 \le \alpha \le 0.9$$

$\eta \rightarrow$ learning coefficient

$\alpha \rightarrow$ smoothing coefficient

where w_{ij} are the coefficients which conect the neuron j of a layer with the neuron i of the previous layer and E is the distance between the desired output and the real output of the network (mean square error).

This algorithm try to minimize the mean square error between the real output and the desired output of thisenetwork.

4.- EXPERIMENTAL RESULTS

At this point we will show some results obtained using the receiver proposed previously and the model channel introduced in figure 2.

In order to make some comments about the results we show some of the FIR filters that we have used in the simulations:

1.- $H(z) = 1 - 0.5z^{-1} + 0.25z^{-2} - 0.125z^{-3} + 0.075z^{-4}$

2.- $H(z) = -0.3 + 0.5z^{-1} + z^{-2} - 0.2z^{-3}$

3.- $H(z) = 0.6 - 0.3z^{-1} + z^{-2} - 0.3z^{-3}$

We used as non-linearity of the channel model the polynomial:

$$w(y) = y + 0.5y^2 + 0.7y^3$$

Once the channel parameters are defined, we must specified the architecture of the receiver and the coefficients of the training algorithm. The neural net is composed of two layers with three neurons in the first layer and two neurons in the second one. The parameters of the training algorithm are [5]:

$$\eta = 0.15 \qquad \alpha = 0.9$$

The length of the input to the neural net is equal to the number of samples of the filter used.

We test the receiver behavior at the same time that we make the training of the receiver. In this way we can view the evolution of the error and convergence rate.

When *FILTER 1* and *FILTER 2* are used the neural net converge after train it with two hundred symbols. This convergence is showed in figure 6.

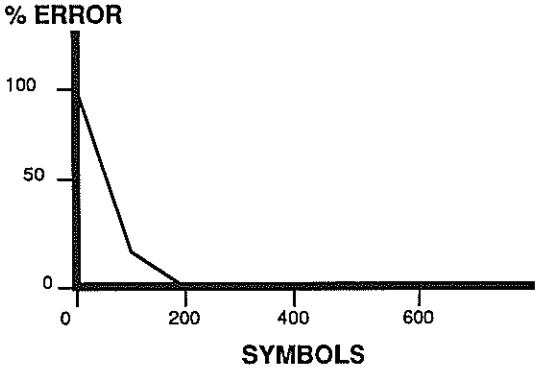


Figure 6: Representation of the convergence of error rate

When we used the *FILTER 3* we obtained an unsucessful convergence of the receiver. The reason of this behavior is that the neural net decribed above has not enough freedom degrees to compensate the disturbance of this channel. It is known that the inverse of a FIR filter is an IIR filter with infinite memory. So it is supposed that if we make the input to the neural net longer, the behavior of the receiver would be better. In this way we made the same simulation for the *FILTER 3*, using five

inputs to the neural net and the result was the convergence of the receiver after train the neural net with seven hundred symbols.

5.- CONCLUSIONS

We have shown that a TDNN based receiver can compensate the efects of a non linear chanel. Then, this receiver is an alternative for a linear equalizer when non linear efects are present at the received signal.

6. REFERENCES:

[1].- S.U.H. QURESHI. Adaptive Equalization. In Advanced Digital Communications System and Signal Processing Techniques.Ed.K. FEHER Englewood Cliffs. New Jersey. 1987

[2].- G. UNGERBOECK: Fractionally Tap-Spacing and Consequences for Clock Recovery in Data Modems.IEEE Trans. on Commm. August 1976.

[3].- E. BIGLIERE, A. GERSHO, R.D. GITLIN, TL.L. LIM: Adaptive Cancellation of Nonlinear Intersymbol Interference for Voiceband Data Transmission. IEEE Journal on Selected Areas in Communications. Sept. 1984.

[4].- R. GARCIA, J. GOMEZ MENA, L. DIEZ DEL RIO. Proc. Internat. Conference on Acoustics, Speech and Signal Processing. Glasgow. United Kingdom. May 1989.

[5].- B.WIDROW, Neural nets for adaptative filtering and pattern recognition. Conputer IEEE. March 1988.

[6].- L.W.CHAN, F.FALLSIDE, An adaptative training algorithm for back propagation networks. Computer Speech and Language, Academic Press,1987.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A MICROCOMPUTER–BASED GENERAL ARCHITECTURE FOR RADIOCOMMUNICATION SIGNAL CLASSIFICATION AND DIGITAL DEMODULATION

J. I. Portillo–García, J. P. Sancho–Marco, L. Vergara–Domínguez, J. M. Páez–Borrallo, B. Ruiz–Mezcua

Dpto. Señales, Sistemas y Radiocomunicaciones
Escuela Técnica Superior de Ingenieros de Telecomunicación
Universidad Politécnica de Madrid
Ciudad Universitaria s/n, 28040 MADRID

We propose in this paper some methods for carrying out modulation type recognition and digital demodulation in a micro–computer based general architecture. A new simplified method for modulation signal classification is developed. The method is based upon the recognition of the shape of some feature vectors obtained from the signal. Afterwards, we face the problem of digital demodulation of ASK and FSK signals. Some schemes are tested and compared from the point of view of the feasibility of implementation on a microcomputer. Finally, we propose a general architecture for performing the above–mentioned tasks, based on the standard IEEE–488 bus.

1. INTRODUCTION

Knowing the type of modulation of a given communication signal is very important both in civil and military applications [2] [3]. Digital signal processing methods in conjunction with pattern recognition techniques allow the design of systems that automatically perform the above tasks. On this line, some previous results on a general approach to a radiocommunication signal classifier have been reported by the authors [4] [5] [6].

On the other hand, it seems necessary to design easy–to–implement modulation classification systems, which could be based on commercial available equipments.

This work completes and extends the work on previous references [4] [5] [6] in three ways:

(a) Firstly, the proposed algorithms have been modified and simplified in order to allow the implementation of the classifier system in a general purpose microcomputer. An example of the performance of the new algorithm is given below.

(b) Secondly, the ability to demodulate a signal classified as ASK or FSK has been added to the system. Six digital demodulators have been implemented and compared, and we present results for their capability for a real–time response in a microcomputer–based classification architecture.

(c) A general purpose microcomputer–based architecture for a general radiocommunication signal classifier is proposed. Besides, the simplifications introduced are useful in a dedicated hardware implementation of the classifier.

2. CLASSIFICATION PROCEDURE IMPROVEMENTS

The classification algorithm and modulation parameter extraction procedure described in [4] [5] [6], based upon numeric pattern recognition, are modified and simplified.

First, we obtain the feature vector (envelope, frequency and phase histograms) [4] [5] [6]. Then the feature vector is classified using a decision tree (see Figure 1). The information contained in each histogram is coded into two main parameters: the number of regions in the histogram and the type of each region (peak or wide region).

The algorithm takes an histogram, eliminates the values lower than a threshold, and isolates a zone with a maximum value and all the non–zero adjacent values. To decide if a zone corresponds to a wide or a narrow peak, we perform the following comparison:

$$\frac{S_3}{S_T} \lessgtr U_1 \qquad (1)$$

where

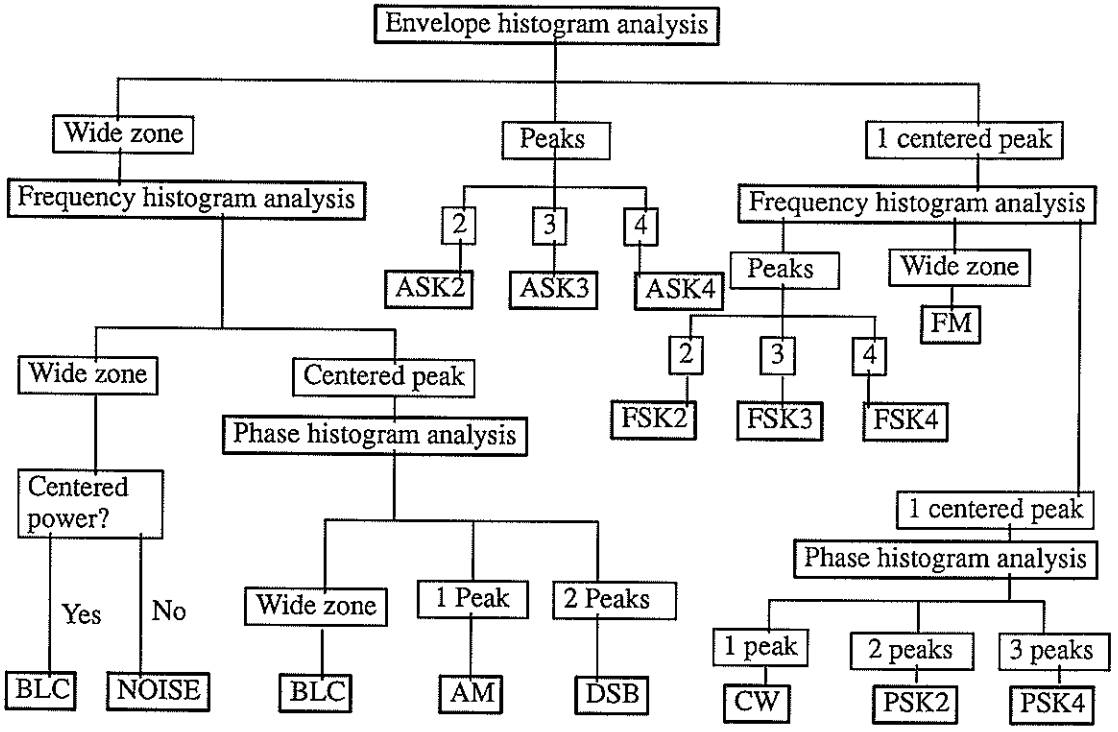


Figure 1. Modulation classification tree

$$S_T = \sum_{i=k}^{i=k+l} h_i \qquad (2a)$$

$$S_3 = h_{k_m-1} + h_{k_m} + h_{k_m+1} \qquad (2b)$$

are the sum of the histogram values in a zone and the sum of the maximum value and the two adjacent ones. U_1 is an heuristic threshold, which depends on the noise and the environment.

The new algorithm presents the following advantages over the one presented in [4] [5] [6]:

– It avoids the numeric pattern recognition procedure, and the matrix products. It saves memory and processing time, and it is faster and more appropriate for a real–time implementation.

– It avoids the classifier training procedure, since it does not use numeric pattern recognition methods. Instead, it realizes the matching with predefined patterns.

– With the new classification procedure, the classification process is completely under control. Therefore, the actions for correcting classification errors are more direct than with the numeric pattern recognition classifier. As an example, the new algorithm allows the easy inclusion of two new "hypothetical classes" called ASK3 and FSK3 (see Figure 2) to easily face the problem of missing one level, in the four–level digital modulations ASK4 and FSK4.

Figure 2 shows the confusion matrix of the results of the new classification algorithm. The simulation conditions are the following:

– SNR: varies randomly from 15 to 35 dB (uniform distribution).
– Number of trials: 100.
– SSB carrier reduction: varies randomly from 10 to 30 dB (uniform distribution).
– AM modulation index: varies randomly from 0.5 to 0.9 (uniform distribution).
– FM frequency deviation: varies uniformly from .1 to .9 (uniform distribution).

We see from this example the satisfactory results for all modulation types.

3. DIGITAL DEMODULATION

We consider here the problem of real–time digital demodulation of ASK and FSK signals, as an additional classifier improvement.

ASK demodulators

The three schemes tested are:

	AM	DSB	SSB	FM	ASK2	ASK4	FSK2	FSK4	PSK2	PSK4	CW	NOI.	ASK3	FSK3	NC
AM	100														
DSB	1	99													
SSB			98									2			
FM				78			2							4	16
ASK2					99										1
ASK4	1					90							1		8
FSK2							97								3
FSK4	1						5	42		1	5			28	18
PSK2									100						
PSK4										100					
CW											100				
NOI.												100			

Figure 2. Confusion Matrix.

(a) Instantaneous envelope extraction.

(b) Signal energy extraction.

(c) A new fast scheme based on applying an instantaneous frequency dependent function to two consecutive samples. The estimate of the instantaneous frequency can be obtained by a zero–crossing algorithm. Let us define:

$$F = \begin{cases} f_i & f_i \le 0.25 \\ 0.5 - f_i & f_i > 0.25 \end{cases} \qquad (3)$$

where f_i is the value of the instantaneous frequency. We obtain an estimate of the instantaneous envelope as:

$$e[i] = x[i] + x[i-1] + f|x[i] - x[i-1]| \quad (4)$$

with $x[i]$ and $x[i-1]$ two consecutive signal values and f a correction factor given by:

$$f = 2\frac{\sin(\pi F)}{\sin(2\pi F)} - 1 \qquad (5)$$

we compute (4) using two consecutive selected points in a baud time.The points must be of equal sign if $f_i \le 0.25$, and of equal sign if $f_i > 0.25$. This has the additional advantage of using only a little set of samples.

FSK demodulators

The three schemes tested are:

(a) Instantaneous frequency extraction.

(b) Zero–crossing rate estimation.

(c) Griffiths' algorithm [1]. The results are not included in the examples below, since it is too slow compared with the other two methods. This is due to the fact that it must perform an FFT for each new signal sample.

Results

We have compared the different demodulation schemes from a point of view of speed, since it is a very important feature in a real–time implementation. We have simulated the signal points in a file, using a PC–AT (8 Mhz). We have not taken into account the file access time. The band is normalized to 0.–0.5 Hz.

Example 1: ASK, wide band (0.05 – 0.4). SNR = 15 dB. The speeds obtained (in bauds) are:

	ASK2	ASK4
Scheme (a)	19	18
Scheme (b)	125	110
Scheme (c)	220	205

Example 2: ASK, narrow band (0.35 – 0.45). SNR = 25 dB. The speeds obtained (in bauds) are:

	ASK2	ASK4
Scheme (a)	5.5	5.7
Scheme (b)	41	37
Scheme (c)	213	197

We see from these results that scheme (c) is the most adequate for real–time operation, since it is the fastest one and its speed does not depend on the bandwidth. On the other hand, it is the least reliable for narrow band signals and low signal to noise ratio.

Example 3: FSK, wide band (0.1 – 0.4). SNR = 15 dB. The speeds obtained (in bauds) are:

	FSK2	FSK4
Scheme (a)	2.6	1.2
Scheme (b)	112	64

Example 4: FSK, narrow band (0.1 – 0.2). SNR = 15 dB. The speeds obtained (in bauds) are:

	FSK2	FSK4
Scheme (a)	0.07	0.39
Scheme (b)	107	60

Scheme (b) is the most adequate, with constant speed, which does not depend on the bandwidth. In general, the performance of the FSK–oriented schemes is more sensitive to noise and bandwidth than the performance of the ASK schemes.

4. A MICROCOMPUTER–BASED SCHEME

A general scheme for a microcomputer–based radiocommunication signal demodulation is proposed in Figure 3. The scheme is built around a standard bus like IEEE–488 bus, and it uses an RF receptor, an A/D converter and a microcomputer system. The microcomputer controls the RF receptor parameters (tuning, IF filter type and bandwidth, frequency offset to center the band, etc.), and the A/D converter parameters (gain, sample rate, filter type, etc.) Once we have sampled the signal, we process it with the fast algorithm explained. Detailed explanations on the flow control and type of components used will be matter of a next paper.

5. CONCLUSIONS

A new algorithm based on the histogram shape recognition that improves the performance of an existent scheme has been implemented. We have tested some schemes for digital demodulation. Two of those schemes are appropriate for a real–time implementation of the system. We have also proposed a general architecture for real–time radiocommunication signal classification and digital demodulation.

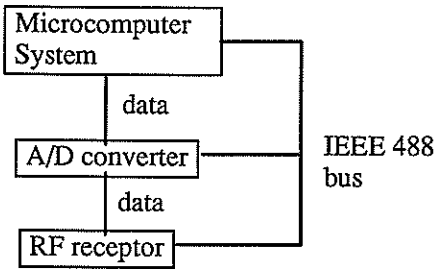


Figure 3. General system architecture.

6. REFERENCES

[1] Griffiths, J., "Rapid Measurement of Digital Instantaneous Frequency", IEEE Trans. Acoust., Speech and Signal Processing, vol. ASSP–23, pp. 207–222, Apr. 1975.

[2] Jondral, F., "Automatic Classification of High Frequency Signals", Signal Processing, vol. 9, pp. 177–190, Oct. 1985.

[3] Liedtke, F., F., "Computer Simulation of an Automatic Classification Procedure for Digitally Modulated Communication Signals with Unknown Parameters", Signal Processing, vol. 6, pp. 311–323, Aug. 1984.

[4] Portillo–García, J., Vergara–Domínguez, L., Páez–Borrallo, J., M., Ruiz–Mezcua, B., "Modulated Signal Classification: A General Approach", Proc IAESTED–87, pp. 127–129.

[5] Vergara–Domínguez, L., Páez–Borrallo, J., M., Portillo–García, J., Ruiz–Mezcua, B., "A Radiocommunication Signal Classifier", Proc. EUSIPCO–88, pp. 1361, 1364.

[6] Vergara–Domínguez, L., Páez–Borrallo, J., M., Portillo–García, J., Ruiz–Mezcua, B., "A General Approach to the Automatic Classification of Radiocommunication Signals", submitted for acceptance to Signal Processing.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

RECOGNITION OF LOW MODULATION INDEX AM SIGNALS IN ADDITIVE GAUSSIAN NOISE

Slobodan D. Jovanović, Miloš I. Doroslovački, Marina V. Dragošević

"Boris Kidrič" Institute of Nuclear Sciences-Vinča

Computer System Design Lab.P.O.Box 522, 11001 Belgrade, Yugoslavia

Abstract: An efficient discrimination feature for distinguishing between an AM signal and an unmodulated carrier in a noisy environment is proposed. The classification threshold corresponding to a predetermined false AM classification probability is derived and is independent of the noise level. The probability of correct AM classification is expressed in terms of modulating signal to noise ratio. A suitable signal processing scheme based on adaptive algorithms is proposed.

1. INTRODUCTION

As the importance of radio broadcasting surveillance increases there appears to be more and more interest in the automatization of the radio monitoring process and particularly in the automatic recognition of the applied modulation type. A global solution to this problem, based on anvelope and instantaneous frequency tracking, feature extraction and classification, is described in literature [1,2,3]. Accepting the general ideas which have appeared in literature a number of specific problems have to be considered regarding feature selection, the design of the feature extractor and classifier and the performance analysis.

This paper focuses on the problem of distinguishing between a slightly AM modulated signal and a pure carrier in a noisy environment. The need to resolve such a situation may well appear in practice, due to the nonstationarity of the modulating signals. The record length of the received signal subjected to the identification procedure is finite and it is quite possible that the modulated signal dynamic in the recorded segment is of the order of the noise power. Such a signal can be considered as an AM signal with a low modulation index. It is difficult to distinguish such a signal from a noisy unmodulated carrier using only features based on signal anvelope parameters. Considerable problems exist even if the modulation index is not too low since the noise power level is usually unknown.

Starting from the phasor representation of the signal and from very few additional assumptions an efficient discrimination feature is proposed and investigated in the paper. A suitable signal processing scheme is also presented.

2. STATEMENT OF THE PROBLEM

The additive noise is assumed to be a zero-mean narrow-band Gaussian process with uncorrelated and independent in-phase and quadrature components having equal variances σ^2. No assumptions are made concerning the modulating signal, except that it is uncorrelated with the additive noise; it may be random and even nonstationary, with an unknown distribution. Thus the AM signal in additive noise is assumed to be

$$r(t)=A\left(1+m(t)\right)\cos(\omega t+\theta)+n(t) \qquad (1)$$
$$=\left(A\left(1+m(t)\right)\cos\theta+n_p(t)\right)\cos\omega t-\left(A\left(1+m(t)\right)\sin\theta+n_q(t)\right)\sin\omega t$$

with the corresponding phasor representation as in Fig. 1. Here $m(t)$ is the modulating signal with the following properties $|m(t)|<1$, $\mathbb{E}\left(m(t)\right)=\mu(t)$, $\mathrm{Var}\left(m(t)\right)=\sigma_m^2(t)$, while $n(t)=n_p(t)\cos\omega t-n_q(t)\sin\omega t$ is the narrow band additive noise and $m(t)$, $n_p(t)$, $n_q(t)$ are assumed to be statistically independent. Without loss of generality we shall assume that $A=1$ and also $\theta=0$. Then the quadrature projections of $r(t)$, namely $p(t)$ and $q(t)$ have the following expectations and variances

$$\mathbb{E}\left(p(t)\right)=1+\mu(t) \qquad \mathrm{Var}\left(p(t)\right)=\sigma_m^2(t)+\sigma^2 \qquad (2)$$

$$\mathbb{E}\left(q(t)\right)=0 \qquad \mathrm{Var}\left(q(t)\right)=\sigma^2 \qquad (3)$$

The quadrature component $q(t)$ is normally distributed, i.e. $q(t)\in N(0,\sigma^2)$. If the carrier is unmodulated then the in-phase component $p(t)\in N(1,\sigma^2)$, but if some modulation is present then the distribution of $p(t)$ is unknown with an unstationary mean value and variance which is greater then the variance of $q(t)$.

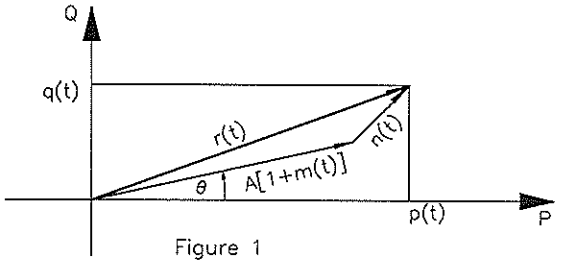


Figure 1

We have to choose between the two hypothesis:

H_0: Noisy sinusoidal signal is present

H_1: Noisy AM signal is present

Basicly this is the problem of distinguishing between two random processes with different stationarity properties and different variances. According to the foregoing discussion a natural approach is to estimate and compare the variances of the quadrature components.

3. DISCRIMINATION FEATURE AND CLASSIFICATION

We shall assume that L quadrature component samples $p(t)$ and $q(t)$ are available for $t=1,2,\ldots,L$. The mean value and the variance estimates of the quadrature components $p(t)$ and $q(t)$ are computed according to the formulae

$$\hat{\mu}_p=\frac{1}{L}\sum_{i=1}^{L}p(i) \qquad \hat{\mu}_q=\frac{1}{L}\sum_{i=1}^{L}q(i) \qquad (4)$$

$$\hat{\sigma}_p^2=\frac{1}{L-1}\sum_{t=1}^{L}\left(p(t)-\hat{\mu}_p\right)^2 \qquad \hat{\sigma}_q^2=\frac{1}{L-1}\sum_{t=1}^{L}\left(q(t)-\hat{\mu}_q\right)^2 \qquad (5)$$

Assuming that both $n_p(t)$ and $n_q(t)$ are discrete white processes the conditional expectations of these variance estimates may be calculated,

$$\mathbb{E}\left(\hat{\sigma}_p^2\middle|m(1),\ldots,m(L)\right)=\sigma^2+\frac{1}{L-1}\sum_{t=1}^{L}\left(m(t)-\frac{1}{L}\sum_{i=1}^{L}m(i)\right)^2\geq\sigma^2 \qquad (6)$$

$$\mathbb{E}\left(\hat{\sigma}_q^2\right)=\sigma^2 \qquad (7)$$

Therefore the following discrimination feature is introduced

$$f=\hat{\sigma}_p^2/\hat{\sigma}_q^2 \qquad (8)$$

and the decision rule is

$f\geq K \rightarrow$ Choose H_1

$f<K \rightarrow$ Choose H_0

where $K>1$ is the decision threshold.

Next we shall determine the distribution of $\hat{\sigma}_p^2$, $\hat{\sigma}_q^2$ and f under both hypotheses and we shall investigate the relationship between K and the probability of false AM signal presence decision $P(H_1|H_0)$, as well as the probability of correct AM signal presence decision $P(H_1|H_1,m(1),\ldots,m(L))$.

4. PROBABILITY OF FALSE AND CORRECT CLASSIFICATION

Introducing $p_0(t)=p(t)-\hat{\mu}_p$ and including $p_0(L)=-\sum_{t=1}^{L-1}p_0(t)$ in the expression for $\hat{\sigma}_p^2$ one easily obtains

$$\hat{\sigma}_p^2=\frac{1}{L-1}\sum_{t=1}^{L}p_0(t)^2=\frac{1}{L-1}\,p_0^T P p_0 \qquad (9)$$

where $p_0^T=[p_0(1)\ldots p_0(L-1)]$ and P is a $(L-1)\times(L-1)$ circulant matrix

$$P=\begin{bmatrix}2&1&1\ldots1\\1&2&1\ldots1\\.&.&.&.\\1&1&1\ldots2\end{bmatrix} \qquad (10)$$

In a similar manner we shall introduce

$$m_0(t)=m(t)-\frac{1}{L}\sum_{i=1}^{L}m(i) \qquad (11)$$

and $m_0^T=[m_0(1)\ldots m_0(L-1)]$ and the $(L-1)\times(L-1)$ covariance matrix

$$R=\mathbb{E}\left((p_0-m_0)(p_0-m_0)^T\right)=\frac{\sigma^2}{L}\begin{bmatrix}L-1&-1&-1\ldots&-1\\-1&L-1&-1\ldots&-1\\.&.&.&.\\-1&-1&-1\ldots&L-1\end{bmatrix} \qquad (12)$$

We may notice that $RP=\sigma^2 I$. R is also a circulant matrix which may be diagonalized by a Fourier matrix. We shall further introduce

$$z=R^{-1/2}(p_0-m_0) \qquad (13)$$

Putting $p_0=R^{1/2}z+m_0$ and $P=\sigma^2R^{-1}$ in the expression for $\hat{\sigma}_p^2$ one easily obtains

$$\hat{\sigma}_p^2=\frac{\sigma^2}{L-1}\left(R^{1/2}z+m_0\right)^T R^{-1}\left(R^{1/2}z+m_0\right) \qquad (14)$$

$$= \frac{\sigma^2}{L-1}\left(z+R^{-1/2}m_0\right)^T\left(z+R^{-1/2}m_0\right) \qquad (15)$$

Since $\mathbb{E}z=0$ and $\mathbb{E}zz^T=I$ and $z\in N(0,1)$ we find out that the expression

$$(L-1)\hat{\sigma}_p^2/\sigma^2=\left(z+R^{-1/2}m_0\right)^T\left(z+R^{-1/2}m_0\right) \qquad (16)$$

has the χ'^2 distribution, i.e. the noncentral chi-squared distribution with $L-1$ degrees of freedom and the expectation $L-1+a$ and the variance $2(L-1+2a)$, where

$$a=\left(R^{-1/2}m_0\right)^T\left(R^{-1/2}m_0\right)=m_0^TR^{-1}m_0=\sigma^{-2}m_0^TPm_0 \qquad (17)$$

$$= \sigma^{-2}\sum_{t=1}^{L} m_0(t)^2=(L-1)\tilde{\rho} \qquad (18)$$

and $\tilde{\rho}$ can be viewed as an instantaneous modulating signal to additive noise ratio. Therefore

$$\mathbb{E}\left(\hat{\sigma}_p^2\middle|m(1),\ldots,m(L)\right)=\mathbb{E}\left(\hat{\sigma}_p^2\middle|\tilde{\rho}\right)=\sigma^2(1+\tilde{\rho}) \qquad (19)$$

$$\mathrm{var}\left(\hat{\sigma}_p^2\middle|m(1),\ldots,m(L)\right)=\mathrm{var}\left(\hat{\sigma}_p^2\middle|\tilde{\rho}\right)=2\frac{\sigma^4}{L-1}(1+2\tilde{\rho}) \qquad (20)$$

Similarly $(L-1)\hat{\sigma}_q^2/\sigma^2$ has the central chi-squared, χ^2 distribution with L-1 degrees of freedom and

$$\mathbb{E}\left(\hat{\sigma}_q^2\right)=\sigma^2 \qquad (21)$$

$$\mathrm{var}\left(\hat{\sigma}_q^2\right)=2\frac{\sigma^4}{L-1} \qquad (22)$$

Hence the ratio of the two variance estimates

$$f=\frac{\hat{\sigma}_p^2}{\hat{\sigma}_q^2}=\frac{\left(z+R^{-1/2}m_0\right)^T\left(z+R^{-1/2}m_0\right)}{z^Tz} \qquad (23)$$

under hypothesis H_1 has a noncentral $F_{L-1,L-1}$ distribution, $F'_{L-1,L-1}$ and it is dependent on the modulating signal only through $\tilde{\rho}$, while, under hypothesis H_0 it has a central $F_{L-1,L-1}$ distribution.

Due to the well known properties of the F distribution the decision threshold providing an a priori required probability of false AM signal presence decision can be determined regardless of the noise and carrier power levels. The threshold is merely dependent on the number of samples and the required false AM signal classification probability. This is a very convenient property of the proposed discrimination feature.

By virtue of the central limit theorem for $L\gg1$ the estimates $\hat{\sigma}_p^2$ and $\hat{\sigma}_q^2$ as well as $\hat{\sigma}_p^2-K\hat{\sigma}_q^2$ converge in distribution to the normal distribution $N\left(\mu_e(\tilde{\rho}),\sigma_e(\tilde{\rho})\right)$ where

$$\mu_e(\tilde{\rho})=\sigma^2(1+\tilde{\rho}-K) \qquad \sigma_e(\tilde{\rho})=\frac{2\sigma^4}{L-1}(1+2\tilde{\rho}+K^2) \qquad (24)$$

Therefore the probabilities of false and correct AM classification may be approximated by

$$P(H_1|H_0)=P(f\geq K|H_0)=P(\hat{\sigma}^2-K\hat{\sigma}^2\geq0|H_0)$$

$$\simeq\frac{1}{2}\,\mathrm{erfc}\left(-\frac{\mu_e(0)}{\sqrt{2}\,\sigma_e(0)}\right)\simeq\frac{1}{2}\,\mathrm{erfc}\left(\frac{\sqrt{L}}{2}\frac{K-1}{\sqrt{K^2+1}}\right) \qquad (25)$$

$$P(H_1|H_1,\tilde{\rho})=P(f\geq K|H_1,\tilde{\rho})=P(\hat{\sigma}_p^2-K\hat{\sigma}_q^2\geq0|H_1,\tilde{\rho})$$

$$\simeq\frac{1}{2}\,\mathrm{erfc}\left(-\frac{\mu_e(\tilde{\rho})}{\sqrt{2}\,\sigma_e(\tilde{\rho})}\right)\simeq\frac{1}{2}\,\mathrm{erfc}\left(\frac{\sqrt{L}}{2}\frac{K-1-\tilde{\rho}}{\sqrt{K^2+1+2\tilde{\rho}}}\right) \qquad (26)$$

5. QUADRATURE COMPONENT EXTRACTION

Next a suitable signal processing scheme is proposed as in Fig. 2.

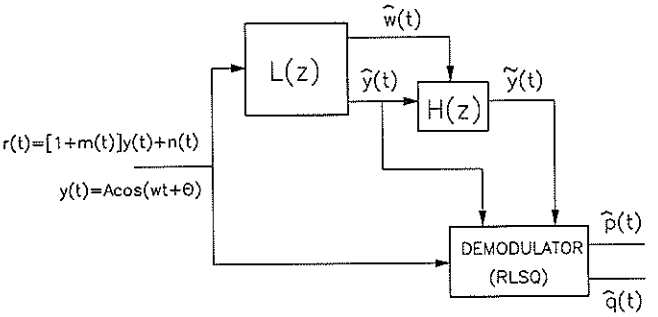


Figure 2

To extract the quadrature components from the incoming signal it is necessary that the local oscillator be matched to the carrier both in frequency and in initial phase. The lack of the a priori information on these values can be overcome by applying an adaptive algorithm. A recursive maximum likelihood or a recursive generalized least squares algorithm can be successfully used for adaptive line enhancement end frequency estimation. The following form of the adaptive line enhancer is considered:

$$L(z)=1-\left(1+\hat{a}(t)z^{-1}+z^{-2}\right)/\left(1+\alpha_t\hat{a}(t)z^{-1}+\alpha_t^2z^{-2}\right)$$

where $\hat{a}(t)$ is a recursively obtained estimate of the parameter $a=-2\cos\omega$, ω is the a priori unknown carrier frequency and α_t is the time-varying pole contraction factor which controls the convergence

rate, the asymptotic accuracy of the frequency estimate and the sharpness of the line enhancer. We let $\alpha_t \to \alpha$ as $t \to \infty$, $0<1-\alpha \ll 1$ in order to obtain a sufficient reduction of both noise and modulation at the ALE output. Thus the ALE output, $\hat{y}(t)$ can serve as the reference signal for the in-phase component extraction. A simple FIR type filter is derived to introduce the $\pi/2$ phase shift in the ALE output sinusoid,

$$H(z)=b_0+b_1 z^{-1} \tag{27}$$

$$b_0=\frac{-\cos(\hat{\omega}(t))}{\sin(\hat{\omega}(t))}=\frac{\hat{a}(t)}{\sqrt{4-\hat{a}(t)^2}} \tag{28}$$

$$b_1=\frac{-\cos(\hat{\omega}(t))}{\sin(\hat{\omega}(t))}=\frac{2}{\sqrt{4-\hat{a}(t)^2}} \tag{29}$$

The FIR filter output, $y(t)$ signal is then fed as the reference signal into the quadrature component extractor.

Once the reference signals are available the extraction of the quadrature components can be performed by two product demodulators followed by adequate decimators. The subsequent samples in the sequences which are generated by product demodulators are correlated due to NF filtering which is incorporated in the demodulators in order to eliminate the double frequency components. The decimators are introduced to avoid this correlation effect, i.e. to pick the samples corresponding to zero correlation and disregard the rest. For example if an ideal NF filter with cut-off frequency π/M is required in the product demodulator then by picking every M^{th} output sample an uncorrelated sequence is obtained as assumed in designing the classifier.

Alternatively a recursive least squares algorithm, RLSQ, [4] with a low forgetting factor may be used to estimate the quadrature components. A low forgetting factor is necessary for good tracking capabilities and for fast exponential decay of $\hat{p}(t)$ sequence correlation and $\hat{q}(t)$ sequence correlation. The algorithm should be slightly modified by substituting the projection vector $Z(t)^T=[\cos\omega t \ \sin\omega t]$ with $Z'(t)^T=[\hat{y}(t) \ \tilde{y}(t)]$.

A qualitative illustration of the proposed approach is given by phase diagrams on Fig. 3, corresponding to an unmodulated noisy carrier and a

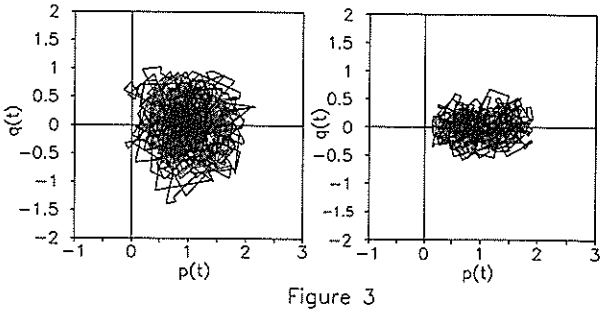


Figure 3

noisy AM signal with modulating signal to noise ratio of 0dB.

6. CONCLUSION

A simple measurement of the received signal anvelope variance is a straight forward approach to distinguish AM modulated signals from unmodulated carriers. However the decision threshold can only be set on the basis of training samples and the performance may get poor in a noisy environment. Both noise and modulation may contribute to the anvelope variance. The principal uncertainty is: Which amount of the measured variance should be attributed to noise? The proposed method resolves the ambiguity by decomposing the received signal into the in phase and in quadrature components. Effectively the noise level is measured in the q channel to serve as a reference when deciding wether the measured p channel variance is due to modulation and noise or due to noise only.

Simulation experiments verify the theoretical analysis and show that the proposed scheme is a highly reliable tool for separating AM signals with a low modulation index from an unmodulated carrier even if the signal to noise ratio is poor.

LITERATURE

[1] F.F. Liedtke, "Computer simulation of an automatic classification procedure for digitally modulated communication signals with unknown parameters", Signal Processing, Vol.6. No4, Aug. 1984.

[2] F. Jondral, "Automatic classification of high frequency signals", Signal Processing, Vol.9. No3, Oct. 1985.

[3] Y.T. Chan, L.G. Gadbois, "Identification of the modulation type of a signal", Signal Processing, Vol.16. No2, Feb. 1989.

[4] M.V. Dragošević, S.D. Jovanović, "RLS type amplitude and phase estimator in modulation mode recognition applications", Proc. of the 5th EUSIPCO, Barcelona, 1990.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

RLS TYPE AMPLITUDE AND PHASE ESTIMATOR IN MODULATION MODE RECOGNITION APPLICATIONS

Marina V. Dragošević and Slobodan D. Jovanović

"Boris Kidrič" Institute of Nuclear Sciences-Vinča
Computer System Design Lab.
P.O.Box 522, 11001 Belgrade, Yugoslavia

Abstract: New solutions to the problem of distinguishing and classifying multiple digital signals under rather severe conditions are proposed using the RLS type amplitude and phase estimator. The steady state and transient performance of the estimator is analyzed theoretically.

1. INTRODUCTION

The recursive least squares algorithm is very convenient for estimating amplitudes and phases of multiple superimposed sinusoidal and other narrow-band signals in broad-band noise. This is achieved via quadrature component estimation which is a linear fitting problem. The quadrature estimates $\hat{P}_k(t)$ and $\hat{Q}_k(t)$ are obtained by minimizing the weighted cumulative squared error

$$J(t)=\sum_{i=0}^{t} \lambda^{t-i}\left(y(i)-\sum_{k=1}^{n}\left(\hat{P}_k(t)\cos\omega_k i+\hat{Q}_k(t)\sin\omega_k i\right)\right)^2 \qquad (1)$$

where $y(i)$ is the noisy signal, ω_k , $k=1,\ldots,n$ are the known component frequencies and λ is the forgetting factor satisfying $0<\lambda\leq 1$. Then the amplitudes $\hat{A}_k(t)$ and the phases $\hat{\varphi}_k(t)$ are defined by $\hat{A}_k(t)^2=\hat{P}_k(t)^2+\hat{Q}_k(t)^2$, $tg\hat{\varphi}_k(t)=-\hat{Q}_k(t)/\hat{P}_k(t)$.

Several important applications of the RLS amplitude and phase algorithm are suggested in the paper. The proposed applications strongly motivate the investigation of the steady-state performance and the tracking behavior of the RLS amplitude and phase estimation algorithm.

A theoretical analysis of the RLS type AR parameter estimator has been presented in literature, [1]. The analysis of the RLS type quadrature component estimator can be based on the same methodology. It is even simpler and can be performed more thoroughly. Yet it seems that such results have not been reported so far.

2. POSSIBLE APPLICATIONS OF THE AMPLITUDE AND PHASE ESTIMATOR

An application of the RLS algorithm for amplitude and phase estimation has recently been mentioned in literature in connection with the adaptive harmonic signal retrieval, [2]. If the model order is overestimated the algorithms for frequency estimation of superimposed multiple sinusoids may yield false frequencies as well as the true ones. Then the true presence of a sinusoid at a discovered frequency is verified by means of its amplitude estimation. In order to set the detection threshold properly it is necessary to know the RLS estimate distribution. An additional possibility which is proposed in this paper is related to the fact that many popular frequency estimation algorithms are biased. A slight biasedness of the frequency estimation algorithm results in a linear trend in the estimated phase. The slope of the trend could be used to correct the frequency estimate. Both detection and accurate frequency estimation problems are important in radio signal surveillance.

An other topic is amplitude and phase tracking for modulated signals in a noisy environment. Specifically, the convenience of the RLS type amplitude and phase estimator and tracker in modulation mode recognition applications is considered. The ideas described bellow work well even under severe conditions, such that the analyzed signal segment contains more than one emission with selective fading effects etc.

Although common in practice, these problems are not addressed in literature.

Several solutions to the problem of automatic modulation mode recognition which have recently been proposed in literature are largely based on the "universal demodulator" concept, [3]. Obviously a RLS type algorithm could be used for this purpose, i.e. to extract quadrature components, amplitudes and phases. In this way several spectral peaks can be processed without an extra filter. This is an advantage over classical demodulators. These spectral peaks may belong to a single emission (e.g. an FSK4 signal) or to more than one emission (e.g. two FSK2 or four ASK or an FSK and two ASK signals). Even though radio location information may not be available, the ambiguity can be resolved by analyzing signal components corresponding to different spectral peaks separately and by combining the results afterwards. First of all adequate discriminating features should be selected so that signal components with digitally modulated amplitudes (e.g. ASK, or parts of FSK signals) can be distinguished from the rest. The active interval of a digitally modulated amplitude is not too representative since it is often subjected to ripples and mild fading. Thus basic properties of the corresponding amplitude wave shapes are that they contain a great deal of very low, close to zero values and sharp edges. Actually the squared amplitude estimate is analyzed, thus avoiding square rooting and still preserving the basic wave shape properties. Extensive experiments show that very few quantitative features describing these properties allow for the proper digital amplitude classification even if considerable parts of the segment are effected by fading. However, the two basic properties of digitally modulated amplitude estimates are related to the transient duration and the estimate variance during zero intervals. Therefore these aspects of the RLS algorithm will be analyzed. The next step in the newly proposed procedure is to perform binary quantization of the amplitude estimates which have been classified as digital. In effect this is a detection problem. Both empirical and analytical results are welcome in order to set the decision threshold. The purpose of binary quantization is to eliminate possible ripples and reduce fading effects in the amplitude estimate, as well as to make further analysis easier. Then binary sequences are compared to check for mutual complementarity leading to the FSK decision. If two spectral peaks belonging to a single FSK signal are closely spaced in frequency they may be undistinguishable by the preprocessing, peak discovering part of the whole procedure. Then a single frequency is estimated and no digital amplitudes are discovered at this frequency, but a piece-wise constant and abruptly changing slope of the estimated phase leads to the correct conclusion. Such a behavior of the RLS algorithm is intuitively clear and will be analytically proved.

3. THE RLS ALGORITHM

The minimization of (1) with respect to $\hat{P}_k(t)$ and $\hat{Q}_k(t)$ leads to the following algorithm

$$\hat{\theta}(t)=\Big(\sum_{i=1}^{t}\lambda^{t-i}Z(i)Z(i)^T\Big)^{-1}\Big(\sum_{i=1}^{t}\lambda^{t-i}Z(i)y(i)\Big) \qquad (2)$$

where $\hat{\theta}(t)^T=[\hat{P}_1(t)\ldots\hat{P}_n(t)\ \hat{Q}_1(t)\ldots\hat{Q}_n(t)]$ and $Z(i)^T=[\cos\omega_1 i\ldots\cos\omega_n i\ \sin\omega_1 i\ldots\sin\omega_n i]$. The estimate (2) can be expressed in the well known recursive prediction error (RPE) form,

$$\hat{\theta}(t+1)=\hat{\theta}(t)+\gamma_t R(t)^{-1}Z(t)e(t) \qquad (3)$$

$$R(t)=R(t-1)+\gamma_t\big(Z(t)Z(t)^T-R(t-1)\big) \qquad (4)$$

$$e(t)=y(t)-\hat{\theta}^T(t)Z(t) \qquad (5)$$

where $\lambda=\gamma_{t-1}(1-\gamma_t)/\gamma_t$ or, equivalently, $\gamma_t=\gamma_{t-1}/(\lambda+\gamma_{t-1})$. Two cases are of special interest. The case $\lambda=1$, i.e. $\gamma_t=1/t$ is suitable for constant amplitude estimation, while the case $\lambda<1$, i.e. $\gamma_t=1-\lambda$ is appropriate for time-varying amplitude tracking. In the algorithm (3), (4) the Kalman gain $\gamma_t R(t)^{-1}Z(t)$ is a deterministic quantity and the inversion of R(t) can be avoided by virtue of the matrix inversion lemma. The algorithm (3), (4), (5) for $\gamma_t=1/t$ can be interpreted as the Gauss-Newton RPE algorithm, [4], related to the criterion function

$$J(\hat{\theta})=\lim_{t\to\infty}\frac{1}{t}\,\mathbb{E}\,|e(t)|^2 \qquad (6)$$

4. ANALYSIS

4.1. *The estimation of constant amplitudes at known frequencies*

When $\lambda=1$, i.e. $\gamma_t=1/t$, then

$$R(t)=\frac{1}{t}\sum_{i=1}^{t} Z(i)Z(i)^T \qquad (7)$$

If the model order is correct and if additive noise is white and normally distributed, then the algorithm (3), (4), (5) represents the recursive maximum likelihood (RML) method, the quadrature component estimates are normally distributed and tend with probability one to the true values and $\sigma_\xi^2 R(t)^{-1}/t \approx 2\sigma_\xi^2 I/t$ is the covariance matrix of the estimation error. If the adopted model order is not correct or if the assumed frequencies do not match the true ones, then the analysis of the criterion function extrema in (6) shows that the quadrature component estimates tend to the true values at the assumed frequencies, even if these values are zero and even if there are other sinusoidal components in the signal.

4.2. *The estimation of on-off keying modulated amplitudes at known frequencies*

When $\lambda<1$, i.e. $\gamma_t=1-\lambda$, then

$$R(t)=(1-\lambda)\sum_{i=1}^{t} \lambda^{t-i} Z(i)Z(i)^T \qquad (8)$$

or

$$R(t)=\lambda R(t-1)+(1-\lambda)Z(t)Z(t)^T \qquad (9)$$

Combining (3), (5) and (9) with $\gamma_t=1-\lambda$, one gets

$$R(t)\theta(t+1)=\lambda R(t-1)\theta(t)+(1-\lambda)Z(t)y(t) \qquad (10)$$

Using (9) and introducing definitions $\Delta\hat{\theta}(t)=\hat{\theta}(t)-\theta$ and $e_{opt}(t)=y(t)-\theta^T Z(t)$ where θ is the piece-wise constant true value, the expression (10) becomes

$$R(t)\Delta\hat{\theta}(t+1)=\lambda R(t-1)\Delta\hat{\theta}(t)+(1-\lambda)Z(t)e_{opt}(t) \qquad (11)$$

Iterating backwards until the initial value $\Delta\hat{\theta}(1)$ one obtains

$$R(t)\Delta\hat{\theta}(t+1)=\lambda^t R(0)\Delta\hat{\theta}(1)+(1-\lambda)\sum_{i=1}^{t}\lambda^{t-i}Z(i)e^*_{opt}(i) \qquad (12)$$

If a full order model is used and if $e_{opt}(i)=\xi(i)$ is normally distributed then the quadrature component estimates are also normally distributed. The expectation of the difference between the estimated and the true value is

$$\mathbb{E}\left(\Delta\hat{\theta}(t+1)\right)=\lambda^t R(t)^{-1}R(0)\Delta\hat{\theta}(1) \qquad (13)$$

It tends to zero in an exponential manner with a time constant proportional to $1/(1-\lambda)$. The expression (13) is valid for initial convergence as well as after an abrupt change in the amplitude value. In the latter case $\Delta\hat{\theta}_K(1)$ represents the magnitude of the jump and the time scale is reinitialized. If a reduced order model is used , i.e. if some of the amplitudes are not being estimated although they exist in the signal, then the expectation of the last term in (12) is not zero and it causes oscillations in the expected value of the estimated amplitudes. The oscillations will not decay as if λ were 1.

The covariance matrix of the estimation error can be determined on the basis of (12). Far enough from a transition one may put $\lambda^i \ll 1$ and thus one easily obtains

$$\mathbb{E}\left(\Delta\hat{\theta}(t+1)\Delta\hat{\theta}(t+1)^T\right)=\frac{1-\lambda}{1+\lambda}R(t)^{-1}R(t)^{(2)}R(t)^{-1}\sigma_\xi^2 \qquad (14)$$

and

$$\mathbb{E}\left(\Delta\hat{\theta}(t+1)\Delta\hat{\theta}(t+1+\tau)^T\right)=\lambda^\tau\frac{1-\lambda}{1+\lambda}R(t)^{-1}R(t)^{(2)}R(t+\tau)^{-1}\sigma_\xi^2 \qquad (15)$$

where $R(t)^{(2)}$ represents the value which would be obtained if the forgetting factor λ were substituted by λ^2 in (8). Especially, if $0<1-\lambda\ll 1$, and $t>1/(1-\lambda)$, then

$$\mathbb{E}\left(\Delta\hat{\theta}(t+1)\Delta\hat{\theta}(t+1)^T\right)\approx(1-\lambda)I\sigma_\xi^2 \qquad (16)$$

An other special case which is of practical importance is the single sinusoid estimator with $\omega=\pi/2$, for which one may easily derive the following expression valid for any λ

$$\mathbb{E}\left(\Delta\hat{\theta}(t+1)\Delta\hat{\theta}(t+1)^T\right)=(1-\lambda^2)/(1+\lambda^2)I\sigma_\xi^2 \qquad (17)$$

4.3. *The estimation of amplitudes at estimated frequencies*

The estimated frequencies may differ from the true ones ω_ν by the values $\Delta\omega_\nu$, which are usually very small and $\nu=1,2,\ldots,n$,

$$Z(i)^T=[\cos(\omega_1+\Delta\omega_1)i..\cos(\omega_n+\Delta\omega_n)i\ \sin(\omega_1+\Delta\omega_1)i..\sin(\omega_n+\Delta\omega_n)i].$$

If the true values at ω_ν are, for example, $P_\nu=A_\nu$ and $Q_\nu=0$, then the optimal values with respect to the adopted criterion are $P_\nu(i)=A_\nu\cos(\Delta\omega_\nu i)$ and $Q_\nu(i)=A_\nu\sin(\Delta\omega_\nu i)$. Thus the optimal estimated value of the amplitude would be equal to the true amplitude A_ν , whereas the estimated phase would be linearly time-varying, the slope being equal to the mismatch between the estimated and the true

frequency. If the model is of a sufficient order then $e_{opt}(t)=\xi(t)$, while

$$e(t)=e_{opt}(t)+\left(\underline{\theta}^{T}(t)-\hat{\theta}(t)^{T}\right)Z(t) \qquad (18)$$

so that the recursion for the estimation of the quadrature components can be written as follows

$$\hat{\theta}(t+1)=\hat{\theta}(t)+(1-\lambda)R(t)^{-1}Z(t)Z(t)^{H}\left(\underline{\theta}(t)-\hat{\theta}(t)\right)$$
$$+(1-\lambda)R(t)^{-1}Z(t)e_{opt}(t) \qquad (19)$$

Due to the orthogonality of $Z(t)$ and $e_{opt}(t)$ the expectation of the last term is zero, therefore

$$\mathbb{E}\hat{\theta}(t+1)=\left(I-R(t)^{-1}Z(t)Z(t)^{T}+\lambda R(t)^{-1}Z(t)Z(t)^{T}\right)\mathbb{E}\hat{\theta}(t)$$
$$+(1-\lambda)R(t)^{-1}Z(t)Z(t)^{T}\underline{\theta}(t) \qquad (20)$$

This expression defines the relationship between the expectation and the optimal value of the estimated amplitudes when there is a discrepancy between the true frequencies and the assumed, estimated ones. In the case of a single sinusoid the recursion for the amplitude estimates acts approximately as a first order, single pole, filter to the optimal value. The expectation of the estimated amplitude is a little bit smaller then the optimal value. The decrease is proportional to $\left(\Delta\omega/(1-\lambda)\right)^{2}$ assuming that $\Delta\omega$ is much smaller then $1-\lambda$, which is easily obtained when estimating frequency by a sophisticated algorithm, such as the generalized least squares, maximum likelihood or stochastic Gauss-Newton and making the corresponding ARMA model contraction factor α, [5], less then λ. The frequency estimation error which is proportional to $(1-\alpha)^{3}$ is negligible from the point of vie of the corresponding amplitude estimation accuracy. The phase trend is the same as in the optimal estimate, since both quadrature components are subjected to the same filter.

5. DISCUSSION AND CONCLUSION

The algorithm is shown to be consistent and asymptotically efficient in the case of steady state sinusoids and stationary noise. If the additive noise is normally distributed so is the quadrature component estimation error. Under mild conditions the normalized squared amplitude estimate is χ^{2}-distributed. The normalization is with respect to the additive noise variance, which can be estimated, e.g. by frequency estimator residual measurements, [6]. Roughly speaking the transition period is proportional to $1/(1-\lambda)$. The expression for the covariance matrix is derived for steady state intervals. and it can be much simplified for some important cases, like $t\gg 1/(1-\lambda)$ or $0<1-\lambda\ll 1$ or $n=1$, $\omega=0.5\pi$ (Some times the sampling frequency is chosen in such way that the spectral peak appears in the middle of the analyzed frequency band). Correct digital amplitude recognition calls for a short transition and a low variance during zero intervals. Concerning λ these requirements turn to be opposite and an intermediate value has to be picked carefully for a good compromise. The correlation between the estimated quadrature component lags decays approximately exponentially. The algorithm is shown to behave like a translator from a given frequency and a first order low pass filter. If there is a mismatch between the true frequency of the sinusoid and the algorithm frequency then both quadrature components oscillate, they are equally displaced from the zero frequency and are submitted to the same amount of delay and attenuation resulting in the correct phase trend.

Experimental investigations based both on simulated and real signals confirm the applicability of the described ideas even for rather low signal to noise ratios, interleaved multiple source signals and in the presence of selective fading which is not too severe.

LITERATURE

[1] E. Eleftheriou, D.D. Falconer, "Tracking properties and steady-state performance of RLS adaptive filter algorithms", IEEE Trans. ASSP-34, pp.1097-1109, 1986.

[2] A. Nehorai, "Adaptive comb filtering for harmonic signal enhancement", IEEE Trans. ASSP-34, 1986.

[3] f. Jondral, "Automatic classification of high frequency signals", Signal processing 9, pp. 177-190, 1985.

[4] L. Ljung, T. Söderström, "Theory and practice of of recursive estimation", MIT Press, 1983.

[5] M.V. Dragošević, "An improved IIR-ALE and frequency tracker", Proc. EUSIPCO 88, pp. 583-586, Grenoble, 1988.

[6] M.V. Dragošević, S.D. Jovanović, S.S. Stanković, "Reduced order modeling of of harmonic signals", Proc. 12éme GRETSI, pp. 157-160, Juan-les-pins, 1989.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

IMPLEMENTATION OF A VOR/ILS PRECISION DETECTOR USING THE TMS32010 DIGITAL SIGNAL PROCESSORS

Matti Isohookana, Pentti Leppänen

University of Oulu, Telecommunication Laboratory
Linnanmaa X2, SF-90570 Oulu, Finland

In this paper the implementation of a digital VOR/ILS precision detector for a calibration receiver of the navigation system in civil aviation has been considered. The implementation of the VOR/ILS precision detector has been done with five TI TMS32010 digital signal processors. Sampling is done using the bandpass sampling theorem directly from the IF of 500 kHz. The VOR/ILS precision detector includes among other things non-coherent AM and FM detectors. A phase difference detector and a modulation depth detector have been developed specially for this application. The VOR/ILS precision detector has been found to work well even in situations where there exists a frequency inaccuracy caused e.g. by Doppler effect.

1. INTRODUCTION

VOR (VHF Omnidirectional Radio Range) and ILS (Instrument Landing System) are standard navigation aids in civil aviation [1]. The VOR, which operates in the band 111.975 MHz to 117.975 MHz, is used in short distance route navigation and the ILS, which operates in the bands 108 MHz to 111.975 MHz and 328.6 MHz to 335.4 MHz, is used in approach and landing.

The VOR beacon radiates a VOR signal which can be represented, from an aircraft point of view, in the form [2]

$$v(t) = \{1 + m_1\cos(2\pi ft - \theta) + m_1\cos[2\pi f_{SC}t + \beta \int_0^t (\cos 2\pi f\tau)d\tau]\} A\cos 2\pi f_C t \quad (1)$$

where $A\cos 2\pi f_C t$ is an RF carrier signal, m_1 an AM modulation depth of 0.3, $\cos(2\pi ft - \theta)$ a 30 Hz variable tone, $\cos 2\pi ft$ a 30 Hz reference tone, β the FM modulation index of 16±1 and f_{SC} the subcarrier frequency of 9960 Hz. θ is the phase difference between the variable and reference tones which depends on the aircraft direction from the beacon compared to magnetic north.

The ILS consists of a localizer and a glide path system [1]. The principles of both systems are the same but they work in different frequency bands. The localizer system guides the aircraft laterally on a path along the runway, and the glide path system guides the aircraft along a path that meets the runway at a shallow angle of approximately 3°. The amplitude modulated ILS localizer signal and the ILS glide path signal can be represented from an aircraft point of view in the form [2]

$$l(t) = (1 + m_1\cos 2\pi f_1 t + m_2\cos 2\pi f_2 t)\, A\cos 2\pi f_C t \quad (2)$$

where $\cos 2\pi f_1 t$ is a 90 Hz tone, $\cos 2\pi f_2 t$ a 150 Hz tone and $A\cos 2\pi f_C t$ an RF carrier signal. m_1 and m_2 are modulation depths which depend on the deviation from the front course line and the deviation from the glide path. On the front course line the modulation depths are $m_1 = m_2 = 0.2$ and on the glide path $m_1 = m_2 = 0.4$.

The detectors, which are used to detect in route navigation the phase difference θ and in approach and landing the DDM (Difference in Depth of Modulation) of the ILS localizer signal and the ILS glide path signal, respectively, are based on analog signal processing. Though they work pretty well, they have, however, some week points such as a drift with temperature, humidity and age, in addition analog filters have a nonlinear phase response and high quality precision detectors are expensive. By using digital signal processing (DSP) these problems can be largely avoided.

2. VOR AND ILS DETECTION USING DSP

The block diagrams of the digital VOR and ILS localizer or ILS glide path detector are shown in Figs. 1 and 2, respectively [2]. The VOR detector consists of a quadrature envelope detector, a quadrature FM detector, a phase difference detector and filters. All filters are linear phase FIR filters to ensure that no phase distortion is caused in the detection. The ILS detectors consist of a quadrature envelope detector, a modulation depth detector and linear phase FIR filters which are maximally flat in the passband.

2.1. Sampling

The VOR and ILS signals are mixed to the intermediate frequency of 500 kHz in a VOR/ILS receiver. The IF signals, which may have the frequency inaccuracy of ±1.5 kHz caused by the Doppler effect and the frequency inaccuracy of the transmitter and the receiver, are sampled using the bandpass sampling theorem. The sampling frequency f_S is 51.282 kHz, and it is based on the bandwidth of 23.88 kHz of the VOR signal and a main clock of 20 MHz used in the hardware implementation.

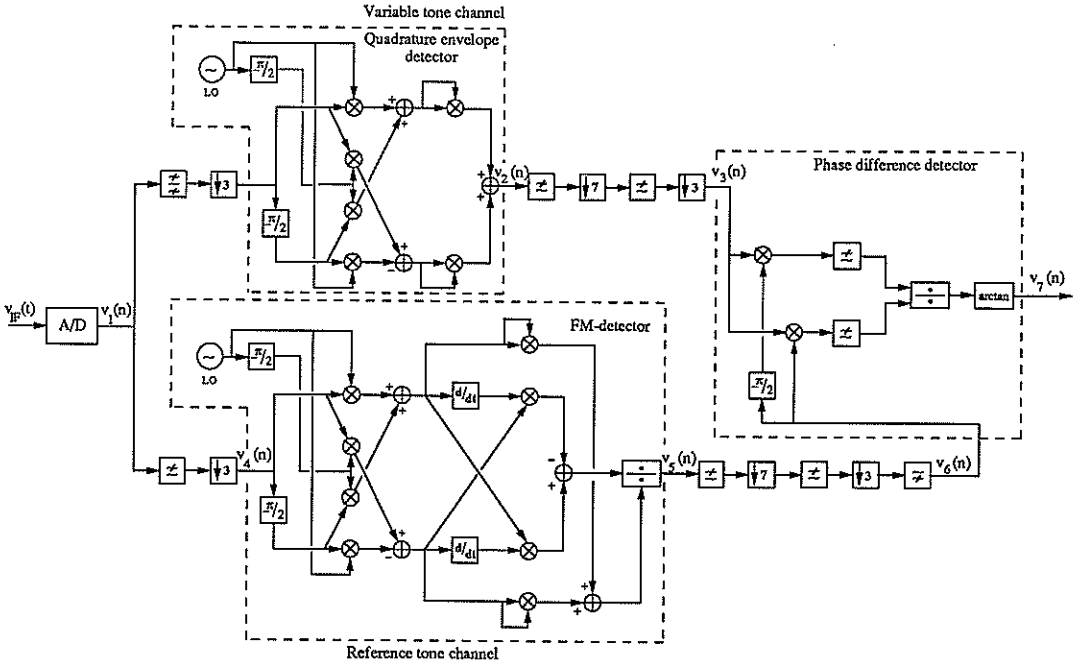


Fig. 1. Block diagram of the VOR detector.

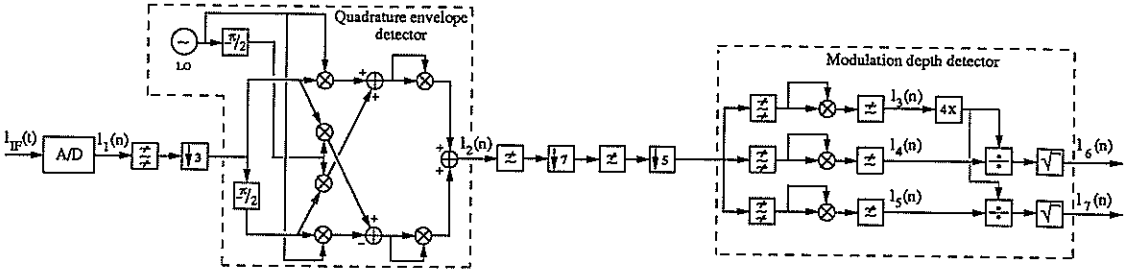


Fig. 2. Block diagram of the ILS localizer or glide path detector.

2.2. VOR detection

In the VOR detector the output of the A/D converter becomes

$$v_1(n) = \{1 + m_1\cos(2\pi fnT_1 - \theta) + m_1\cos[(2\pi f_{sc}nT_1 + \varphi(nT_1)]\} \cdot A(nT_1)\cos 2\pi f_{c1}nT_1 \quad (3)$$

where $v_1(n) = v_{IF}(nT_1)$, T_1 is the sampling period, nT_1 $\varphi(nT_1) = \beta\int_0(\cos 2\pi f\tau)d\tau$ and f_{c1} is the new carrierfrequency of 12.82 kHz. $A(nT_1)$ is the amplitude of the carrier, which may vary slowly because of receiver's AGC action. The spectrum of the sampled signal is shown in Fig. 3.

In the variable tone channel of the VOR detector the variable tone is detected. First the carrier component and the sidebands of the variable tone are bandpass filtered. The bandwidth of the filter is $f_s/6$ and the center frequency $f_s/4$, the filtered signal can be decimated by a factor of three. The variable tone is detected using a quadrature envelope detector where the square rooth operation has been omited. The detector does not need any lowpass filter as detectors based on complex demodulation do. Instead the Hilbert transformer, which is very simple to realized as a linear phase FIR filter, is used. The output of the quadrature envelope detector becomes

$$v_2(n) = A^2(nT_2)[1 + m_1\cos(2\pi fnT_2 - \theta)]^2 \quad (4)$$

where $T_2 = 3T_1$. The second harmonic of the variable tone

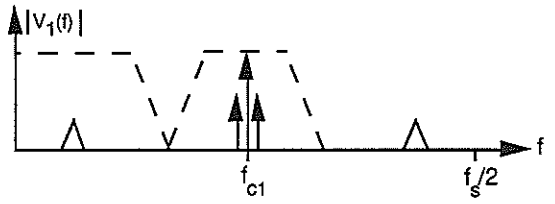


Fig. 3. Spectrum of $v_1(n)$

is lowpass filtered off and the filtered signal is decimated by a factor of 21. The filtering and decimation are implemented as a computationally efficient two stage structure [3]. The output of the variable tone channel becomes

$$v_3(n) = A^2(nT_3)[1 + \tfrac{1}{2}m_1^2 + 2m_1\cos(2\pi fnT_3 - \theta)] \quad (5)$$

where $T_3 = 21T_2 = 63T_1$.

In the reference tone channel the reference tone is detected. The lower sideband of the FM modulated subcarrier signal, which is a mirror image of the upper sideband of the original subcarrier, is first lowpass filtered and then decimated by a factor of three. The output of the decimator becomes

$$v_4(n) = \tfrac{1}{2}A(nT_2)\, m_1 \cos[2\pi(f_{c1} - f_{sc})nT_2 - \varphi(nT_2)] \quad (6)$$

This kind of filtering method has the weakness that only half of the subcarrier signal power is utilized in FM detection. However, the advantage is that an AM detection of the wideband signal is avoided. The FM detector used to detect the reference tone is an "arctan type" quadrature FM detector which eliminates the envelope variation in $v_5(n)$. The inphase component and the quadrature component of $v_5(n)$ are generated using a Hilbert transformer, implemented by an FIR filter. The differentiators are seven "tap" FIR filters.

The output of the FM detector becomes

$$v_5(n) = \varphi'(nT_2) - \gamma'(nT_2) \quad (7)$$

where $\varphi'(nT_2) = \beta\cos(2\pi fnT_2)$ and $\gamma'(nT_2)$ is the dc or low frequency component caused by the frequency difference between the subcarrier signal and the local oscillator signals. After lowpass filtering, decimation by a factor of 21 and highpass filtering the output of the reference tone channel becomes

$$v_6(n) = \varphi'(nT_2) = \beta\cos(2\pi fnT_3) \quad (8)$$

The phase difference θ between the variable and reference tones is detected by a phase difference detector. The upper path of the detector detects the sine of the phase difference θ and the lower path the cosine of the phase difference θ, respectively. By dividing the sine term with the cosine term the influence of the variable tone level and the reference tone level on the end result are excluded. Finally the arctan operation gives the wanted result

$$v_7(n) = \theta \quad (9)$$

2.3. ILS detection

In the ILS localizer or ILS glide path detector (Fig. 2) the output of the A/D converter becomes

$$l_1(n) = [1 + m_1\cos2\pi f_1nT_1 + m_2\cos2\pi f_2nT_1]\, A(nT_1)\cos2\pi f_{c1}nT_1 \quad (10)$$

where $l_1(n) = l_{IF}(nT_1)$. The spectrum of the sampled signal (Fig. 4) consists of a carrier component at the frequency of 12.82 kHz, as well as sidebands of the 90 Hz tone and sidebands of the 150 Hz tone.

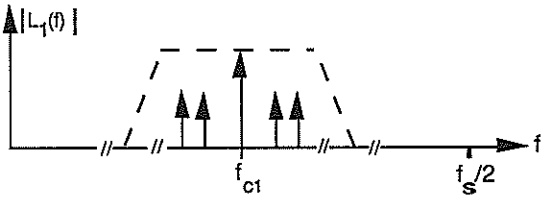


Fig. 4. Spectrum of $l_1(n)$

The modulating tones and the dc component caused by the carrier are quadrature detected using the same detector structure that was used to detect the variable tone. Only the cutoff frequencies of the bandpass filter and the Hilbert transformer are changed. The output of the quadrature detector becomes

$$l_2(n) = A^2(nT_2)[1 + m_1\cos2\pi f_1nT_2 + m_2\cos2\pi f_2nT_2]^2 \quad (11)$$

Unfortunately, the quadrature envelope detector produces also an extra dc component, caused by the modulating tones, as well as second harmonics of the modulating tones and sum and difference frequencies of the modulating tones, preventing the determination of the modulation depths using the dc component and the detected modulating tones. Under these circumstances the modulation dephts m_1 and m_2 have to be determined from the 60 Hz, 90 Hz and 150 Hz tones using the modulation depth detector, illustrated in Fig. 2. Before the modulation depth detection the sampling rate is, however, decreased by a factor of 35.

In the modulation depth detector the 60 Hz, 90 Hz and 150 Hz tones are first bandpass filtered, and then the powers of these filtered tones are measured. The powers become

$$l_3(n) = \tfrac{1}{2}A^4(nT_3)m_1^2\, m_2^2,$$

$$l_4(n) = 2A^4(nT_3)m_1^2 \quad \text{and} \quad (12)$$

$$l_5(n) = 2A^4(nT_3)m_2^2.$$

After a few steps of calculation the outputs of the modulation depth detector become

$$l_6(n) = m_2 \quad \text{and} \quad l_7(n) = m_1 \qquad (13)$$

The DDM can be defined now as

$$DDM = m_1 - m_2 \qquad (14)$$

3. PERFORMANCE OF THE DETECTOR

The performance of the VOR/ILS detector (DET) has been measured in the laboratory with IF test signals generated by a VOR/ILS generator (GEN). Some results of the measurements are shown in Figs. 5 - 7. Fig. 5 shows the difference $\theta_{GEN} - \theta_{DET}$ as a function of f_c, θ_{GEN} being 5^o in the VOR detection. The phase difference θ can be detected with an accuracy better than $\pm$ 0.4° in the frequency band 495 kHz to 503 kHz.

Fig. 6 shows the difference $DDM_{GEN} - DDM_{DET}$ as a function of f_c, DDM_{GEN} being ±0.000 DDM in the ILS localizer detection. The corresponding results of the ILS glide path detection are shown in Fig. 7. These results mean that when the deviation of the aircraft from front course line is small, the difference of the modulation depths m_1 and m_2 of the ILS localizer signal can be detected with an accuracy better than 0.0005 DDM in the frequency band 497.5 kHz to 502.5 kHz. Respectively, the difference of the modulation depths m_1 and m_2 of the ILS glide path signal can be detected with an accuracy better than 0.001 DDM in the same frequency band.

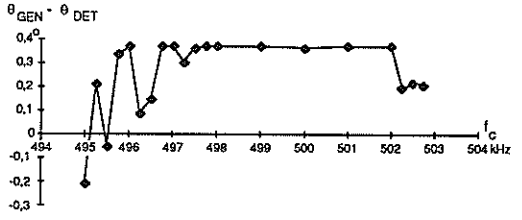


Fig. 5. The difference $\theta_{GEN} - \theta_{DET}$ as a function of f_c in the VOR detection.

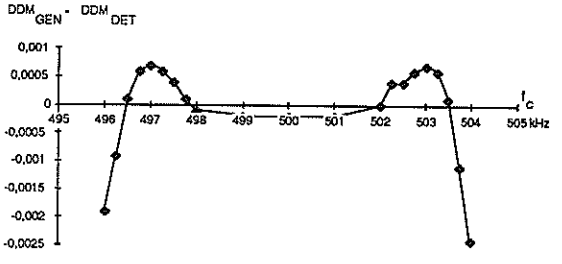


Fig. 6. The difference $DDM_{GEN} - DDM_{DET}$ as a function of f_c in the ILS localizer detection.

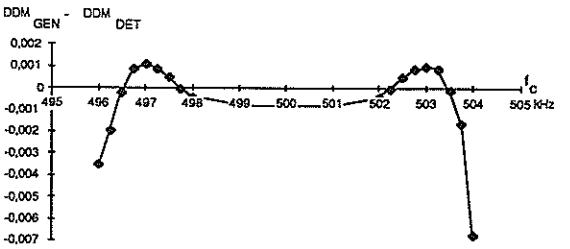


Fig. 7. The difference $DDM_{GEN} - DDM_{DET}$ as a function of f_c in the ILS glide path detection.

As the accuracy of the older VIR-108 VOR/ILS Receiver [4] is ± 1° in the VOR detection and about ± 0.003 DDM in both ILS detections when the aircraft is on the front course line and on the glide path, it can be seen that the performance of the new VOR/ILS detector is considerably better than that of the older one.

4. IMPLEMENTATION

The implementation of the VOR/ILS detector has been done with two 12 bit A/D converters and five TMS32010 signal processors. To each processor about 3/4 kwords assembly code have been written. The processors which are connected partly in parallel form, partly in pipeline form (Fig. 8), are working either in the VOR detection mode or in the ILS detection mode. Before implementation the VOR/ILS detector was simulated with the ILS (Interactive Laboratory System) simulation program.

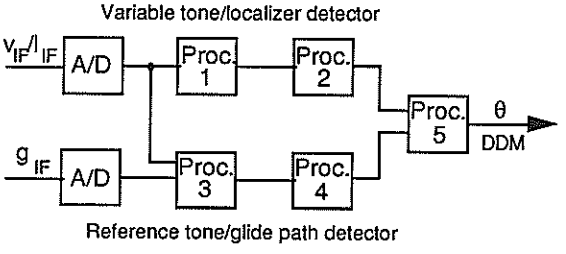


Fig. 8. The high level hardware architecture of the VOR/ILS detector.

REFERENCES

[1] International Standards and Recommended Practice, Aeronautical Telecommunications, Annex 10, Vol. I.

[2] Isohookana M., Leppänen P., A VOR/ILS Precision Detector for a Calibration Receiver of the Navigation System in Civil Aviation, Report No. 35, University of Oulu, Telecommunication Laboratory, 1989, (in Finnish).

[3] Crochiere R. E., Rabiner L. R., Multirate Digital Signal Processing, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1983.

[4] Collins Avionics Instruction Book, Collins Avionics Group, Rockwell International, 1977.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

EFFICIENT GENERATION OF PASSBAND DIGITALLY MODULATED SIGNALS

Krzysztof WESOŁOWSKI*

Institute of Electronics and Communications, Technical University of Poznan, 60-965 Poznan, Poland

A method of generation of digitally modulated signals is presented which is particularly efficient in the implementation of data modems by the standard digital signal processors. It allows for generation of passband signals with arbitrary spectral characteristics including nonsymmetry relative to the carrier frequency.

1. INTRODUCTION

Rapid development of DSPs, microprocessors specially designed for intensive calculations required in digital signal processing, allowed for the all-digital implementation of the medium- and high-speed voiceband data modems. Number crunching capabilities of the DSPs are due to the high-speed hardware multiplier placed on the chip, pipelining and parallelizing of the operations and internal devices. The new features of the DSPs changed the objectives of the modern design of digital transmitters and receivers. The minimization of the number of multiplications is no longer the main criterion of the design optimization. Instead, the minimization of the required memory resources and the final speed of the device have become the main goal to achieve. A typical DSP has a limited number of RAM cells on the chip and larger data files have to be stored in the external, expensive and power consuming RAM chips. On the other hand, some DSPs feature ROM or EPROM on the chip aimed to store the program and constant data. The compact design which leads to avoiding the external memory results in significant decreasing of the cost of the device. This paper presents the transmitter design method which takes into account the above mentioned objectives and limitations.

2. GENERATION OF THE PASSBAND DIGITAL SIGNALS

Digital medium- and high-speed modems apply DPSK or QAM modulation. The signal samples generated by the all-digital transmitter can usually be described by the formula

$$s(nT_s) = \sum_{k=-\infty}^{+\infty} a_k \cdot p(nT_s - kT) \cdot \cos(2\pi f_c nT_s) - b_k \cdot p(nT_s - kT) \cdot \sin(2\pi f_c nT_s)] \qquad (1)$$

Symbols a_k, b_k denote the data symbols which carry digital messages in the time interval $kT<t<(k+1)T$. T is the signalling interval, whereas T_s is the sampling interval. Usually $T/T_s=m$, where m is integer. Symbol f_c denotes the carrier frequency. For the CCITT V.26, V.27 and V.32 modems the carrier frequency equals 1800 Hz and is simply related to the signalling rate equal to 1200, 1600 and 2400 Bd, respectively. Thus, the following equality holds

$$P \cdot T = Q \cdot (1/f_c) \qquad (2)$$

where P and Q are small integers. One can easily find that for the V.26 modem P=2, Q=3, for the V.27 modem P=8, Q=9 and for the V.32 modem P=4, Q=3. Similar relations hold for the full-duplex V.22 and V.22bis modems which have the carrier frequencies f_{c1}=1200 Hz, f_{c0}=2400 Hz and the signalling rate equal to 600 Bd. The function p(t) describes the shape of the baseband pulse. The desired spectral properties of the passband signal $s(nT_s)$ are achieved by careful design of the pulse p(t). For spectrally efficient signals the function p(t) spans more than one signalling period,

---

* on leave to University of Kaiserslautern, Institute of Communications Engineering, P.O. Box 3049, D-6750 Kaiserslautern, BRD

This work was sponsored by Polish Ministery of National Education (CPBP 02.16) and the Alexander von Humboldt Foundation

however, its duration is in practice limited. Thus, formula (1) transforms into the expression

$$s(nT_s) = \sum_{k=i}^{i+L-1} \left(a_k \cdot p(nT_s - kT) \cdot \cos(2\pi f_c nT_s) - \right.$$

$$\left. - b_k \cdot p(nT_s - kT) \cdot \sin(2\pi f_c nT_s) \right) \qquad (3)$$

where $iT < nT_s < (i+1)T$.

There are several possibilities of generation of signal (3). The first is the direct implementation of formula (3) which was described by P.J.Van Gerwen *et al* [1]. Echo modulation [2] can also be applied to generate this signal. However, because of spectral shaping realized in the baseband these methods allow the generation of the signals having a symmetrical spectrum with respect to the carrier frequency f_c.

There are situations where the unsymmetrical spectrum of the digital signal is highly recommended. The example of such a signal is that, transmitted by the CCITT V.22 or V.22bis modem. As the result of the FDM method applied in full-duplex transmission over two wire links, the locally transmitted signal should generate as small a portion as possible of the overall power in the band occupied by the received signal. Let us assume that the received signal arrives at the level of -48 dBm and the locally generated signal is on the level of 0 dBm and leaks to the local receiver through the hybrid which attenuates by 10 dB. Therefore the power density spectrum of the transmitted signal in the received signal band should be at the level of -70 dBm in order to ensure the received signal to the interfering signal power ratio to be in the order of 30 dB. In the range of frequencies outside the received signal band, the power density spectrum is satisfactory at the level of -50 dBm.

As opposed to the baseband shaping, the passband spectral shaping of the digitally modulated signals allows the completion of the above mentioned spectral requirements. The characteristics $H_A(f)$ of the analog filter placed on the output of the digital-to-analog converter can be also taken into account in the passband filter synthesis.This can be done by synthesis of the digital filter which has the characteristics $H(f) = H_T(f)/H_A(f)$ in the passband. $H_T(f)$ is the transmitter passband filter characteristics which would be synthesized if the lowpass analog filter on the output of the D/A converter were ideal. Thus, the output signal samples are described by the formula

$$s(nT_s) =$$

$$= h(nT_s) * \sum_{k=i}^{i+L-1} \left(a_k \mathrm{rect}\left(\frac{nT_s - kT - T/2}{T}\right) \cos(2\pi f_c nTs) - \right.$$

$$\left. - b_k \cdot \mathrm{rect}\left(\frac{nT_s - kT - T/2}{T}\right) \cdot \sin(2\pi f_c nT_s) \right) \qquad (4)$$

where $h(nT_s)$ is the impulse response of the passband shaping filter which, as well as its baseband predecessor, spans L signalling intervals. The asterisk denotes discrete convolution. As previously, the direct implementation of formula (4) is straightforward. However, to generate the output sample $s(nT_s)$ mL+2 multiplications and mL+1 additions are necessary. This number of operations results from the implementation of the passband FIR filter $h(nT_s)$. Also mL coefficients and mL data RAM storage locations are required. For the V22.bis example the sampling frequency $f_s = 1/T_s$ = 9600 Hz can be assumed. Thus m=16 samples are generated within one modulation interval T. If we assume that L=4 then the FIR filter length mL is equal to 64. It has been found that this filter length, in conjunction with the minimax criterion for the filter synthesis, is sufficient to satisfy the described above spectral requirements for passband and stopband attenuation. Fig.1 presents the amplitude characteristic of this filter for transmission in the low channel. The filter for the high channel has analogous characteristics. The filter coeficients have been achieved using the Remez exchange optimization algorithm. It is worth noting once more that such a filter requires 64 data RAM storage locations. Thanks to high "number crunching" capabilities of the DSPs, the number of multiply-and-add operations is not so critical. However, the fast on-board RAM storage locations have to be carefully disposed because of their very limited number, in particular in the older versions of DSP chips such as TMS 32010. For this microprocessor 64 RAM cells are almost a half of their total number.

The number of operations and required RAM storage locations can be considerably decreased by the application of the method proposed in this paper. Formula (4) can be rewritten as follows

$$s(nT_s) =$$

$$= \sum_{k=i}^{i+L-1} a_k \cdot \left(h(nT_s) * \mathrm{rect}\left(\frac{nT_s - kT - T/2}{T}\right) \cos(2\pi f_c nT_s) \right) -$$

$$- b_k \cdot \left[h(nT_s) * \mathrm{rect}\left(\frac{nT_s - kT - T/2}{T}\right) \sin(2\pi f_c nT_s) \right] \quad (5)$$

It can be easily shown that if n=mk+j and j=1,...,m then

$$s((mk+j)T_s) =$$

$$= \sum_{i=1}^{L+1} \left[a_{k+1-i} \cdot p_{i,j,k} - b_{k+1-i} \cdot q_{i,j,k} \right] \quad (6)$$

where

$$p_{i,j,k} = \sum_{n=n_{min}}^{n_{max}} h(nT_s) \cos\left(2\pi f_c (j-n) T_s + k \bmod(P) \frac{2\pi Q}{P}\right)$$

$$q_{i,j,k} = \sum_{n=n_{min}}^{n_{max}} h(nT_s) \sin\left(2\pi f_c (j-n) T_s + k \bmod(P) \frac{2\pi Q}{P}\right) \quad (7)$$

and

$$n_{min} = \max[1, ((i-2)m+j+1)],$$

$$n_{max} = \min[mL, ((i-1)m+j)]$$

Let us note, that for the V.22bis transmitter the coefficients $p_{i,j,k}$ and $q_{i,j,k}$ do not depend on k because P=1; Q=2 for the low channel and Q=4 for the high channel. In this particular case the multiple number of full carrier periods is contained in the modulation period. As we see, the modulation process reduces to simple, short FIR filtration (6) with cyclicly changing filter coefficients. For the V.22bis example, when L=4 and m=16 the generation of one output sample requires only 10 multiply-and-add operations. Also only 10 RAM storage locations are required for retaining 2(L+1) data symbols used in the generation of the output sample. We see that for this example the number of operations and storage locations is reduced six times as compared with the straightforward implementation of equation (4). The price paid for this achievement is the enlarged size of the filter coefficient array placed in ROM. In the considered example the array size is equal to 180. On the other hand the proposed method results in much shorter machine code than the direct implementation of equation (4).

The method presented above has been applied in the V.22bis transmitter implemented by the TMS 32010 microprocessor. The TMS 32010 assembler subroutine required about 150 cycles for generation of the first sample in the signalling period. Besides the realization of equation (6) the procedure contains quadbit-to-data symbol mapping and differential encoding. Each of the remaining 15 samples in the signalling interval required about 100 microprocessor cycles. The filter coefficients $p_{i,j,k}$ and $q_{i,j,k}$ were stored as 16-bit words in the program ROM. The output transmitter sample should be 12-bit long to ensure sufficiently high signal-to-quantization noise ratio. Fig.2 presents the energy density spectrum of the output 12-bit sample sequence which is the response of the low channel transmitter to the single data symbol pair $(a_1,b_1)=(3,3)$ followed by the sequence of zeroes. Assuming the statistical independence of the data symbols and their equal probability this spectrum is proportional to the power density spectrum. As we see in Fig.2 the quantized signal satisfy the spectral requirements.

Similar results are presented for the V.32 transmitter. In this case the spectral requirements are not so strong. When we assume m=4 samples in each modulation period (f_s=9600 Hz) then the filter order mL=32 is sufficient to ensure the signal level below -40 dB in the stopband (see Fig.3). Fig.4 shows the energy spectrum of the response to the signal pair $(a_1,b_1)=(3,3)$ followed by the sequence of zeroes. Because of the lower spectral requirements we do not observe deterioration of the spectrum of the 12-bit quantized signal as it was for the V.22bis transmitter.

The above examples indicate that, the method presented in this paper is general and can be applied in any QAM or DPSK transmitter with a symmetrical or unsymmetrical spectrum. Its main advantage is the low computational and data RAM requirements. Any method of the transmitter filter synthesis can be applied for the calculation of the filter coefficients $h(nT_s)$. The method presented by P. Chevillat and G. Ungerboeck [3] seems to be particularly suitable for this purpose.

REFERENCES

[1] Van Gerwen, P.J, Verhoeckx, N.A.M, Van Essen, H.A., Snijders, F.A.M., Microprocessor Implementation of High-Speed Data Modems, *IEEE Trans. Commun.*, Vol.COM-26, No.5, 1978, pp.493-498

[2] Choquet, M.F., Nussbaumer, H.J., Microcoded Modem Transmitter, *IBM J. Res. Develop.*, July 1974, pp.338-351

[3] Chevillat, P.R., Ungerboeck, G., Optimum FIR Transmitter and Receiver Filters for Data Transmission over Band-Limited Channels, *IEEE Trans. Commun.*, Vol.COM-30, No.8, 1982, pp. 1909-1915

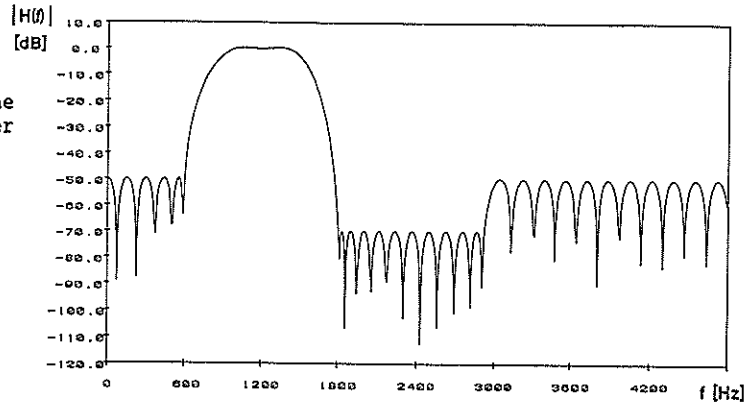


Fig.1. Amplitude characteristics of the low channel transmitter filter for V.22bis modem

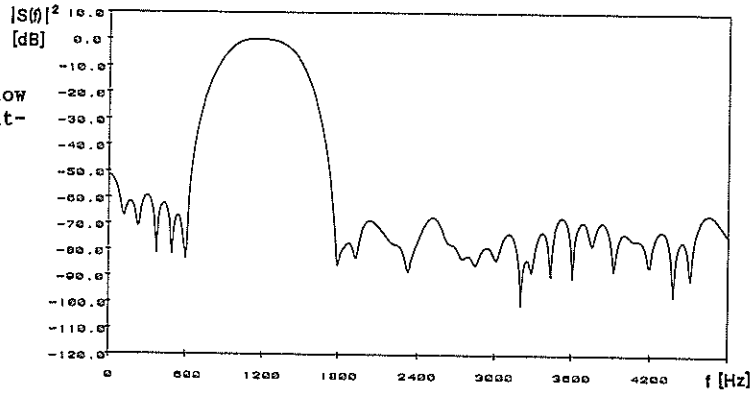


Fig.2. Energy spectrum of the 12-bit low channel V.22bis transmitter output signal

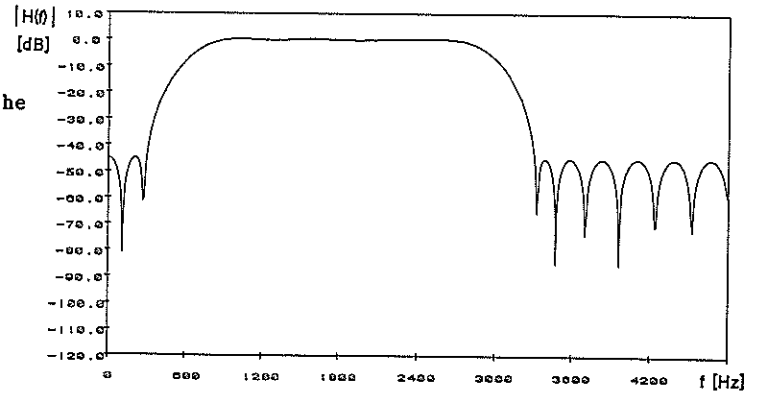


Fig.3. Amplitude characteristics of the V.32 transmitter filter

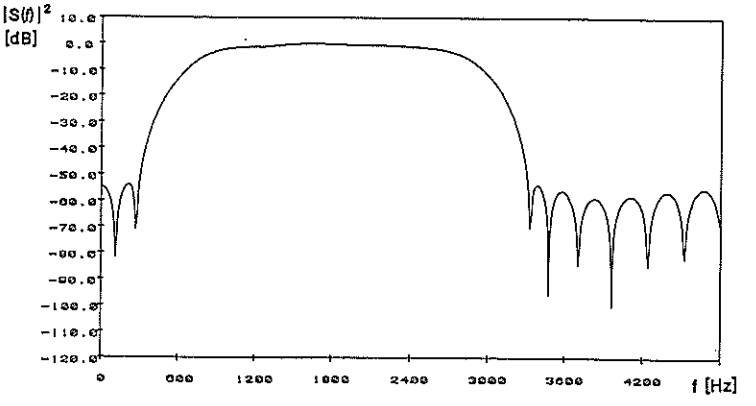


Fig.4. Energy spectrum of the 12-bit V.32 transmitter output signal

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

PARALLEL DECODING OF GENERALIZED CONCATENATED CODES

Ezio Biglieri

Dipartimento di Elettronica, Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129 Torino (Italy) (e-mail: biglieri@itopoli.bitnet) [1]

In this paper we study soft decoding of generalized concatenated codes. The complexity problem is solved by using algorithms based on highly parallel structures that are suitable for VLSI implementation. This solution offers a number of advantages, viz., (a) The codes of this family are very powerful (in fact, many of them have been found to outperform the best codes previously known). (b) The encoder has an architecture which is intrinsically parallel. (c) A decoder can be designed which performs soft decoding in parallel, with a "systolic" architecture easily amenable to VLSI implementation. The loss of optimality with respect to maximum-likelihood decoding is very small. (d) The encoder-decoder structure is modular and very flexible, so that it can be used for a different code if the need arises. (e) Unequal protection can be achieved quite naturally.

1 Introduction

While the potential effectiveness of error-control coding in digital communications is undisputed, a major obstacle to its use, particularly if highly powerful (and hence complex) codes are used, is the complexity of the decoder. The complexity barrier has motivated many suboptimum detection algorithms with the usual price of performance degradation, or the choice of coding schemes that do not exploit in full the potential of coding theory. For example, it is known that if hard-decision (algebraic) decoding is used instead of soft decoding, a loss of about 2 dB in efficiency can be achieved. However, soft decoding may entail an unacceptable complexity. Furthermore, the requirement of real-time operation constrains the coding processing time not to exceed a certain fixed amount.

This paper deals with a class of highly powerful codes called "generalized concatenated codes" and their decoding algorithms. In particular, we study parallel soft-decoding algorithms whereby the complexity problem is solved by using highly parallel algorithms that are suitable for VLSI implementation. An added attractive feature of the schemes we propose is that the structure of the decoder is modular and flexible, which accounts for easy reconfiguration if a new coding scheme is introduced to accommodate for changing transmission requirements.

2 Generalized concatenated codes

The concept of a generalized concatenated code was developed by Blokh and Zyablov [3] and by Zinov'ev[10]. This concept allows the construction of an exceedingly large class of powerful codes with excellent distance properties and a flexible choice of parameters. As observed in [10], the parameters afforded by these codes improve upon more than 300 codes known when the paper was published and tabulated in [8]. More recently, Imai and Hirakawa [6] and Ginzburg [5] have described constructions of signal sets based on the same concept. Powerful modulation schemes can be generated with an arbitrary signal distance and with a regular structure (see also [1]).

Fig. 1 shows the construction of an L-stage generalized concatenated code (GCC). L block encoders $C_1, \cdots, C_L$ over the Galois fields $GF(q_1), \cdots, GF(q_L)$, with the same block length n, Hamming distances $d_1, \cdots, d_L$, and cardinalities $M_1, \cdots, M_L$, accept source symbols and output the L blocks $(a_{i1}, a_{i2}, \cdots, a_{in})$, $i = 1, \cdots, L$, of n symbols each. The output codeword $\mathrm{x}_1, \cdots, \mathrm{x}_n$ depends on these L n-tuples, i.e., on the matrix

$$\mathbf{A} = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ & & \cdots & \\ a_{L1} & a_{L2} & \cdots & a_{Ln} \end{bmatrix}$$

so that we can write $\mathrm{x}_1, \cdots, \mathrm{x}_n$ in the form

$$\mathrm{x}_j = f(\mathrm{a}_j) \qquad (1)$$

where a_j, $j = 1, \cdots, n$, are the colums of the matrix $\mathbf{A}$.

Now, consider the code $B^{(L)}$ over $GF(q)$ with block length ν, cardinality $\prod_{i=1}^{L} q_i$, and minimum Hamming distance δ_L. Next, consider a partition of the code $B^{(L)}$ into q_L subcodes $B_{i_L}^{(L-1)}$, $i_L = 0, 1, \cdots, q_L - 1$, with minimum Hamming distance δ_{L-1}. Then, each

[1] Formerly with the Electrical Engineering Department, UCLA, Los Angeles, CA (USA). This research was supported by NSF under Grant NCR-8814407.

of the $B_{i_L}^{(L-1)}$ is partitioned into the q_{L-1} sub-subcodes $B_{i_{L-1}i_L}^{(L-2)}$, $i_{L-1} = 0, 1, \cdots, q_{L-1} - 1$, with minimum Hamming distance δ_{L-2}. Each of the sub-subcodes $B_{i_{L-1}i_L}^{(L-2)}$ is partitioned into the q_{L-3} sub-sub-subcodes $B_{i_{L-2}i_{L-1}i_L}^{(L-3)}$, $i_{L-2} = 0, 1, \cdots, q_{L-2} - 1$, and so forth, until we are left with codes with only one codeword (an element of $B^{(L)}$). These are the words $B_{i_1,i_2,\cdots,i_L}^{(0)}$, and will be identified through their subscript by denoting them by $\mathrm{b}(i_1, i_2, \cdots, i_L)$. More specifically, eq. (1) becomes explicitly

$$\mathrm{x}_j = \mathrm{b}(a_{1j}, a_{2j}, \cdots, a_{Lj}) \in B^{(L)}.$$

Finally, observe that the L-tuple $i_1, i_2, \cdots, i_L$ can take on $\prod_{i=1}^{L} q_i$ values, exactly as many as each one of the vectors a_j defined above. The function $f(\cdot)$ maps a_j into the codeword $\mathrm{b}(i_1, i_2, \cdots, i_L) \in B^{(L)}$ whose indices $i_1, \cdots, i_L$ are the values of the components of a_j. By doing this, we have constructed a *generalized concatenated code*. This code has block length $n\nu$, cardinality $\prod_{i=1}^{L} M_i$, and minimum Hamming distance

$$d_{\min} \geq \min\left(\delta_1 d_1^H, \delta_2 d_2^H, \cdots, \delta_L d_L^H\right) \qquad (2)$$

3 Staged decoding

In this paper we consider soft decoding algorithms based on *parallel structures*. A massive amount of parallelism in the decoder is the key that breaks the complexity bottleneck by reducing both the processing time and the latency time, and hence allows real-time operation.

Staged decoding of generalized concatenated codes is based on an idea proposed by Imai and Hirakawa in [6] for demodulation of certain multidimensional modulation schemes. With this procedure the information bits protected by the most powerful code are decoded first by using maximum-likelihood (soft) decoding. Then the bits protected by the second most powerful code are decoded, and so on. This procedure is suboptimum, but reduced decoding complexity will result. In particular, staged decoding is amenable to an architecture with pipelined parallelism.

The idea underlying staged decoding is the following. An L-level code $[\mathcal{C}_1, \cdots, \mathcal{C}_L]$ is decoded by decoding the component codes in sequence. $\mathcal{C}_L$, the most powerful code, is decoded first. Then $\mathcal{C}_{L-1}$ is decoded by assuming that $\mathcal{C}_L$ was decoded correctly. Further, $\mathcal{C}_{L-2}$ is decoded by assuming that the two previous codes were decoded correctly, and so forth.

The block diagram of a staged decoder is shown in Fig. 2 for $L = 3$. Let the received signal vector be

$$\mathrm{r} = f(\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3) + \mathrm{n},$$

where n denotes the noise vector. The receiver must produce an estimate of the three codewords $\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3$ in order to demodulate r. In principle, the Euclidean distances from r of all the possible vectors $f(\mathrm{c}_1, \mathrm{c}_2, \mathrm{c}_3)$ should be computed, and the minimum such distance determined. This is an impractical procedure for large signal constellations. In staged decoding, decoder $\mathcal{D}_3$ produces an estimate of the codeword c_3 for all the possible vectors $\mathrm{c}_1, \mathrm{c}_2$ ($\mathrm{c}_1, \mathrm{c}_2$ are not constrained to be codewords of $\mathcal{C}_1, \mathcal{C}_2$ here). Next, decoder $\mathcal{D}_2$ produces an estimate of the the codeword c_2 for all the possible choices of the vector c_1, *by assuming that the choice of* c_3 *was correct.*(c_1 is not constrained to be a codeword of $\mathcal{C}_1$ here.) Finally, decoder $\mathcal{D}_1$ produces an estimate of the codeword c_1 *by assuming that the choice of* c_2 *and of* c_3 *was correct.* Observe that decoding of $\mathcal{D}_\ell$ will provide us with the estimate of k_ℓ source symbols. Thus, at each stage of the procedure we obtain a chunk of source symbols, that are sent to the parallel-to-serial converter.

3.1 Suboptimality of staged decoding

While it has the advantage of creating pipelined parallelism in the decoding process, the staged-decoding algorithm is suboptimum.

Thus, the question is, how much loss of optimality (i.e., how much loss of potential coding gain) shall we incur if we use parallel architectures at intermediate signal to noise ratios? Preliminary simulation results [2] show that for codes of reasonable complexity this loss is marginal, and in some cases even zero. The latter statement may actually seem surprising at first: however, a closer examination of the staged decoding procedure will show that in some cases staged decoding and maximum-likelihood decoding are actually the same algorithm. This occurs whenever the code has *two* stages, and code $\mathcal{C}_1$ is non-redundant. In particular, we can prove the rather unexpected result that the (15,11,3) Hamming code, interpreted as a generalized concatenated code, can be decoded in parallel without any loss of optimality.

3.2 Soft decoding

Soft decoding of a block code is based on the maximization of the scalar product $\mathrm{y} \cdot \mathrm{x}^i$ of two vectors, y being the vector observed at the channel output, and x^i any of the possible codewords. A word-level systolic processor for the search of the index i that maximizes the scalar product $\mathrm{y} \cdot \mathrm{x}^i$ was proposed in [4], and is shown in Fig. 3 for a code having 6 codewords of block length $n = 4$. This linear array is composed of n *Inner Product Processors* (IPP) and one *Comparator Processor* (CP). The computation performed by the array is given by

$$\hat{i} = \max_{1 \leq i \leq M}{}^{-1} \sum_{\ell=1}^{n} y_\ell x_\ell^i$$

where the inverse means "output the index $\hat{i}$ which achieves the maximum." The components y_ℓ of the vector y remain stationary in the array for the duration of one symbol period, while the n components of x^i enter the array skewed in time by one clock cycle. IPP's operate on their two inputs x and y to produce the output $d_{\text{out}} = d_{\text{in}} + xy$. The CP compares its input

to the content of a register R, and saves the largest of these words in R on the subsequent clock cycle. The index of the signal vector that yields the larges scalar product is also saved by the CP after each comparison.

In general, the number of processors (including the CP) required to implement this systolic structure is $n+1$, and it can be seen that the time required for decoding is reduced by a factor of n with respect to a single-processor architecture.

3.3 Structure of the decoders $\mathcal{D}_\ell$

We are now in a position to describe the structure of the decoders $\mathcal{D}_\ell$, $\ell = 1, \cdots, L$ [9]. We assume here *binary codes* for simplicity, and we start by describing $\mathcal{D}_L$. Its block diagram is shown in Fig. 4. The components $r_1, \cdots, r_n$ of the received vector $\mathbf{r}$ are presented in parallel at the decoder input. Decoding of $\mathcal{C}_L$ corresponds to the choice of a sequence of binary symbols $a_{L1}, \cdots, a_{Ln}$ (a codeword of $\mathcal{C}_L$), which in turn corresponds to a sequence of subcodes $B_{i_L}^{(L-1)}$, with $i_L \in \{0,1\}$. Each component r_i of $\mathbf{r}$ is sent to two systolic circuits of the type described before (see Fig. 4, where I denotes the set of inner-product processors and C the comparator processors). These circuits compute the minimum distance of r_i from each one of the codewords in $B_0^{(L-1)}$ and $B_1^{(L-1)}$, respectively. The results of this computation are then sent to another linear systolic processor, as shown in Fig. 4. All the possible codewords of $\mathcal{D}_L$ are shifted sequentially in this circuit. The i-th circuit element, $i = 1, \cdots, n$, adds to its input the quantity $\min \| r_i - B_0^{(L-1)} \|^2$ (if $a_{Li}^m = 0$) or the quantity $\min \| r_i - B_1^{(L-1)} \|^2$ (if $a_{Li}^m = 1$). The processor denoted C outputs the index of the codeword that provided the maximum sum at the output of processor n. This index corresponds to the decoded codeword in $\mathcal{C}_L$, and consequently to the estimate of k_L source symbols. At this time, $\mathcal{D}_L$ has ended its task, and is ready to accept the next received vector $\mathbf{r}$ for pipelined operation. $\mathcal{D}_{L-1}$ enters now into action. Its structure and operation are essentially the same as for $\mathcal{D}_L$, with two differences. The first difference involves the codewords entering the systolic circuit: they are now the codewords of $\mathcal{C}_{L-1}$. The second difference involves the inner-product processors, that now must choose a sequence of elementary subconstellations at level $L-2$. If $\hat{a}_{L1}, \cdots, \hat{a}_{Ln}$ denotes the codeword chosen by $\mathcal{D}_L$, then the two outputs of the i-th processor will be $\min \| r_i - B_{0\hat{a}_{Li}}^{(L-1)} \|^2$ and $\min \| r_i - B_{1\hat{a}_{Li}}^{(L-1)} \|^2$. The output of $\mathcal{D}_{L-1}$ will be a block of k_{L-1} source symbols, as well as the estimated codeword of $\mathcal{C}_{L-1}$. The latter is sent to $\mathcal{D}_{L-2}$, and so forth with straightforward changes.

We observe here that the operation of this circuit does not require any special property (for example, linearity) of the codes $\mathcal{C}_1, \cdots, \mathcal{C}_L$. On the other hand, the time required for a full cycle of operations will increase linearly with the number of signals in each subconstellation, as well as on the number of codewords in each code. Thus, for efficient use of this architecture these two numbers must be kept reasonably small.

Finally, observe that the first chunk of k_L source symbols is "protected" by code $\mathcal{C}_L$, the second chunk k_{L-1} is "protected" by $\mathcal{C}_{L-1}$, and so forth. We may achieve *unequal protection* of different chunks of source symbols by increasing the power of some of the codes $\mathcal{C}_i$. This would allow to add extra protection to some particularly sensible source symbols, those carrying more perceptual significance.

References

[1] E. Biglieri and M. Elia, "Multidimensional modulation and coding for band-limited digital channels," *IEEE Trans. Inform. Theory*, Vol. 34, No. 4, pp. 803-809, July 1988.

[2] E. Biglieri and F. Pollara, *Manuscript in preparation.*

[3] É. Blokh and V. V. Zyablov, "Coding of generalized concatenated codes, *Problemy Peredachi Informatsii*, Vol. 10, No. 3, pp. 45-50, July-September 1974.

[4] G. A. Davidson, P. R. Cappello, and A. Gersho, "Systolic architectures for vector quantization," *IEEE Trans. Acoustics, Speech, and Signal Processing*, Vol. 36, No. 10, October 1988, pp. 1651-1664.

[5] V. V. Ginzburg, "Mnogomernije signaly dlya neprerivnogo kanala," *Probl. Peredachi Informacii*, no. 1, pp. 28-46, Jan.-Mar. 1984 (in Russian). English translation: "Multidimensional signals for a continuous channel," *Probl. Inform. Transmission*, pp. 20-34, 1984.

[6] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Trans. Inform. Theory*, vol. IT-23, pp. 371-377, 1977.

[7] S. Y. Kung, *VLSI Array Processors*. Englewood Cliffs, NJ: Prentice Hall, 1988.

[8] N. J. A. Sloane, "A survey of constructive coding theory and a table of binary codes of highest known rate," *Discrete Math.*, Vol. 3, Nos. 1-3, pp. 265-294, 1972.

[9] R. M. Tanner, "Algebraic construction of large Euclidean distance coding/modulation systems," University of Santa Cruz, Tech. Rep. UCSC-CRL-87-7, 1987.

[10] V. A. Zinov'ev, "Generalized cascade codes," *Problemy Peredachi Informatsii*, Vol. 12, pp. 5-15, January-March 1976.

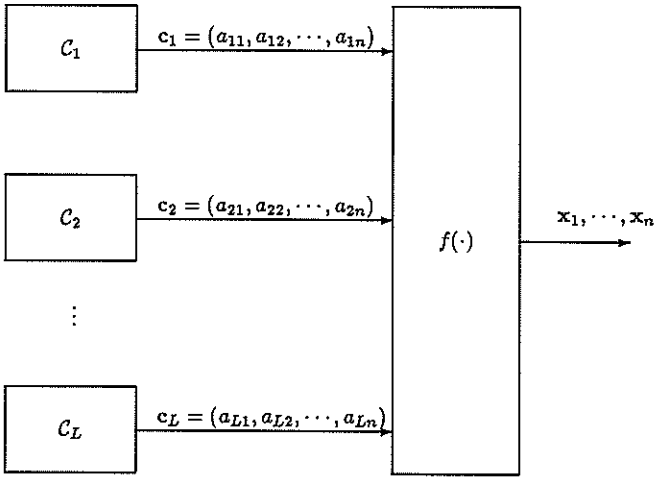


Figure 1: The construction of an L-level generalized concatenated code.

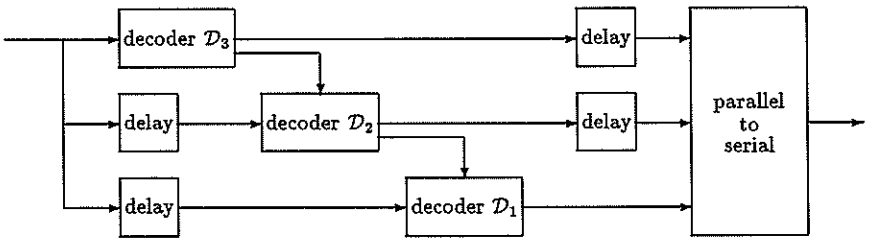


Figure 2: Staged decoder for a 3-level code.

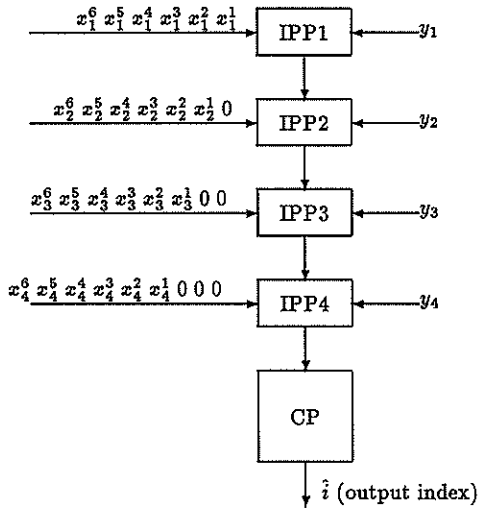


Figure 3: Linear systolic architecture for soft decoding.

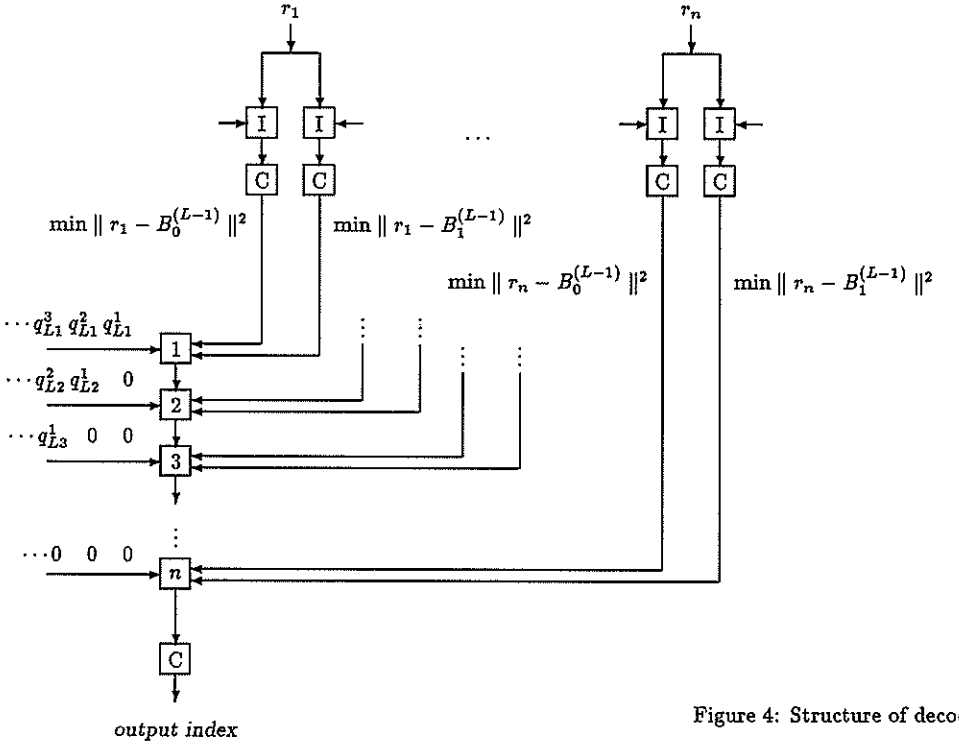


Figure 4: Structure of decoder $\mathcal{D}_L$ for binary codes.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

SPEECH SIGNAL INTERPOLATION UNDER LOSSES IN A TRANSMISSION CHANNEL

Roman NEMIROVSKY, Vilnis LIEPIŅŠ

Institute of Electronics & Computer Science, Latvian Academy of Sciences, Riga

This paper considers the wafeform recovery of a speech signal in transmission over packet-switched computer networks. To decrease the impact of losses of speech packets the samples of fragment are mixed,divided into standart packets and transmitted. If a packet is lost during transmission, then, after a reverse permutation, several samples of initial signal appier between the lost samples. The lost samples are restored due to the first and second-order interpolation using the correlation properties of the signal.

1. INTRODUCTION

It is known that during speech transmission over packet-switched networks some fragments of speech signals are lost. Admissible percentage of losses is limited on the level about 1-3% [1] to ensure acceptable speech quality. As a result, the average network load is limited, and the efficiency of the data network with combined data and speech transmission markendly decreases.

A lot of approaches are used to admit more losses without decreasing of speech quality. The authors of [2] propose to use a variable speech packet encoding rate to enable the smoothing of the effect of network overloads on the received speech quality. Paper [3] proposes the classification of speech segments in accordance with their structure. The packets belonging to different classes are assigned different priorities of delivery. When network overloads appear,the packets with a lower priority are discarded first. At the receiving end regeneration of lost packets is made.

The mentioned works deal mainly with information transfer based on encoding of speech signal parametrs. The present paper deals with the transfer based on waveform encoding using pulse-code modulation.

Section 2 considers the principle of signal samples permutation in waveform recovery. Section 3 analyses first-order interpolation methods. Section 4 analyses the possibilities of the second-order interpolation. Section 5 discusses some experimental results of the recovery of phrases.

2. PERMUTATION OF SIGNAL SAMPLES IN WAVEFORM RECOVERY.

For the channels wits error bursts the sequence is often subjected to a permutation prior to transmission and is recovered at the receiving end. In this case the errors are distributed in a more uniform manner [4]. We will use this principle for packet speech transmission. The source sequence of samples of a signal segment is memorised, some permutation is made, wich is followed by separation into packets and transfer. At the receiving end a reverse procedure is performed. If a packet is lost during transfer, lost samples are separated by one or several samples of the source signal. Such a procedure enables the recovery of losses as a result of correlation.

Interpolation of samples was applied in [5]. A signal is divided into two groups, the even and the odd samples. Each group is formed into a packet. Following the reverse permutation, in the case of the loss of one packet, one source sample appears between the missing samples. This allows to apply extrapolation and first-order interpolation. The "odd-even" alternation allows to use only the correlation of the neighbouring samples for the recovery. Using the information on a greater number of samples, it is possible to recover the value of the lost samples more accurately. To do this the samples of the source signal have to be interchanged on a segment containing more than two packets. For example, using a block encoder [4], the sequence of samples is written into a **nxm** matrix columnwise and read rowwise. Having defined the length of the row **n** as being equal to the packet length, after reading we will obtain **m**

packets. If one of **m** packets is lost, then after the reverse permutation the samples will be separated by **(m-1)** samples of the source signal. In this case interpolation procedures ranging from the zero order and up to the **(m-1)**-th order may be applied for the recovery.

3. RECOVERY BASED ON THE FIRST-ORDER INTERPOLATION

We will assume that not more than one packet is randomly lost from the sequence of signal samples consisting of **m** packets. The sequence X(i) (i=1,2,...) of the centered signal X(t) is transmitted. Following the reception and reverse permutation, in the case of the loss of one packet, the received sequence Y differs from X only in the points i=j where the samples are missing. In this case the probability is P(i=j)=1/m. The transmission error is e(i)=X(i)-Y(i). The mean square error of the transmission of a sequence consisting of **m** packets is

$$E[e^2(i)] = \sum_i \{X(i)-Y(i)\}P(i=j) = E[\{X(i)-Y(i)\}^2]/m. \quad (1)$$

If the estimates of the values of the missing samples are equated to zero, then

$$E[e^2(i)]=E[X^2(i)]/m=R(0)/m, \quad (2)$$

where R(0) is the value of the autocorrelation function for a zero shift.
For the first-order interpolation:

$$Y(i)=\alpha(-1)X(i-1) + \alpha(+1)X(i+1) \quad (3)$$

It is usually assumed that $\alpha(-1)=\alpha(+1)=\alpha$, and for the most commonly used procedure $\alpha=0.5$ (linear interpolation). This procedure was refered to in [5] as non-adaptive.

For $Y=0.5[X(i-1)+X(i+1)]$ the mean square error will be:

$$E[e^2(i)=[1.5R(0)-2R(1)+0.5R(2)]/m, \quad (4)$$

where R(k) is the value of the autocorrelation function of the signal with the shift equal to k sampling intervals.

The error can be minimized by applying the adaptive approach. As it was demonstrated in [5], by calculating the error for interpolation with the use of (3) and by determining the minimal error depending on the coefficients $\alpha(-1)$ and $\alpha(+1)$, we obtain:

$$\alpha=\alpha(-1)=\alpha(+1)=r(1)/[1+r(2)], \quad (5)$$

where $r(k)=R(k)/R(0)$ is the autocorrelation coefficient.

Figure 1 shows the charts of the errors of recovering a signal which corresponds to the sounds "a", "d", "c", "sh". On the vertical axis are laid off, in percent, the values of the normalised mean square error $q=E[e^2]/R(0)$. On the horizontal axis are laid off the ordinal numbers of the appropriate signal segment with length n=128 samples. A real signal was used in the experiments. A situation was simulated in which every fourth packet is lost (m=4). When no recovery is applied, the error at the receiving end equals 25%, since from (2) we have q=1/m.

It follows from the figure 1 that practically for all signal fragments $q_{1.2}<q_{1.1}$.

4. RECOVERY BASED ON THE SECOND-ORDER INTERPOLATION

The estimate Y(i) of the value of the lost samples X(i) using the known samples with numbers (i+1),(i-1); (i+2),(i-2) will be derived using the expression:

$$Y(i)=b1[X(i-1)+X(i+1) + b2[X(i-2)+X(i+2)] \quad (6)$$

The normalised mean square error of the second-order interpolation will be:

$$q_2=[(1+2b1^2+2b2^2) - 4b1(1-b2)r(1) + 2(b1^2-2b2)r(2) + 4b1b2r(3) + 2b2^2r(4)]/m \quad (7)$$

Like for the first-order interpolation, the procedure for determining the coefficients b1, b2, may be non-adaptive and adaptive. For a non- adaptive procedure it is possible to use some known family of polynomials, for example, the family of the second-order Chebyshev's polynomials. It is possible to show that b1=0.667 and b2=-0.167. Then (7) is converted to the following form:

$$q_{2.1}=[1.94-3.11r(1)+1.56r(2)-$$
$$-0.44r(3)+0.056r(4)]/m \qquad (8)$$

Comparing (8) and (4) for the normalised errors we obtain the approximate equation

$$q_{1.1}-q_{2.1}=[\{r(1)-0.44\}-$$
$$-\{r(2)-0.44r(3)\}]/m \qquad (9)$$

Analysing (9), it is easy to see that the efficiency of interpolation is determinined by the type of autocorrelation function of the signal. Specifically, $q2.1<q1.1$ for the signals for which the correlation between the adjacent samples is high and afterwards it rapidly decreases. This is true for many speech sounds [6]. However, for the hushing sounds and fricatives the value of r(1) may not be high. Then it is possible that $q_{2.1}>q_{1.1}$. In selecting coefficients in (6) the second-order interpolation allows to take account of the effect of all the values of correlation function. The formulae for the optimum values of coefficients b1 and b2 will be obtained by equating the partial derivatives dq2/db1 and dq2/db2 to zero in (7). This is illustrated by Figure 1 which shows the experimental results of signal recovery for 25% losses for the first and second-order non-adaptive and adaptive interpolation. For the sound "a" and "d" the second-order interpolation provides better results than the first-order interpolation. For the sounds "c" and "sh" this is true for adaptive interpolation, while with the use of the second-order Chebyshev's interpolation the recovery error increases. Figure 2 (a-voiced sound,b-unvoiced sound) shows the waveforms of the fragments of speech signal at the transmitting and receiving ends.

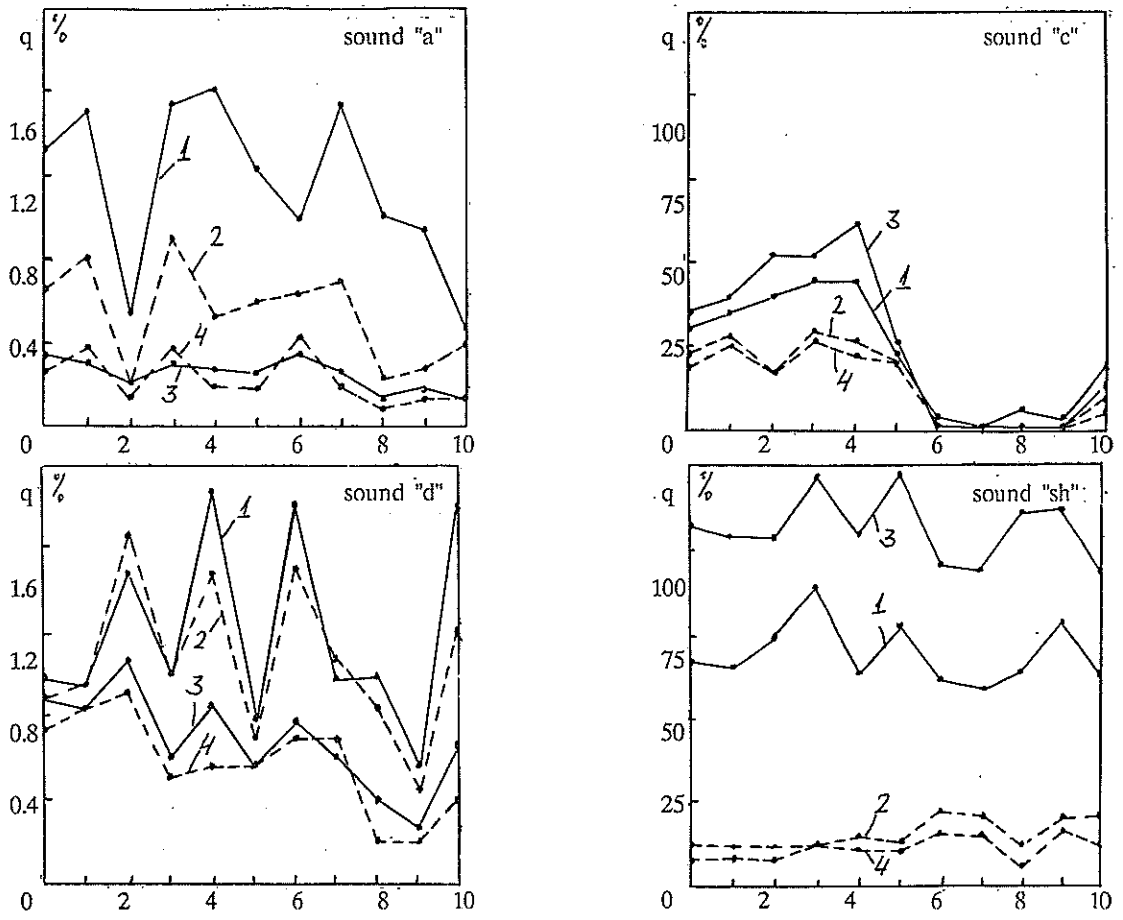


Figure 1. Errors of recovering a signal which corresponds to the sounds "a", "d", "c", "sh".
1 - linear interpolation ($q_{1.1}$)
2 - 1-st order adaptive interpolation ($q_{1.2}$)
3 - 2-nd order Chebyshev's interpolation ($q_{2.1}$)
4 - 2-nd order adaptive interpolation ($q_{2.2}$)

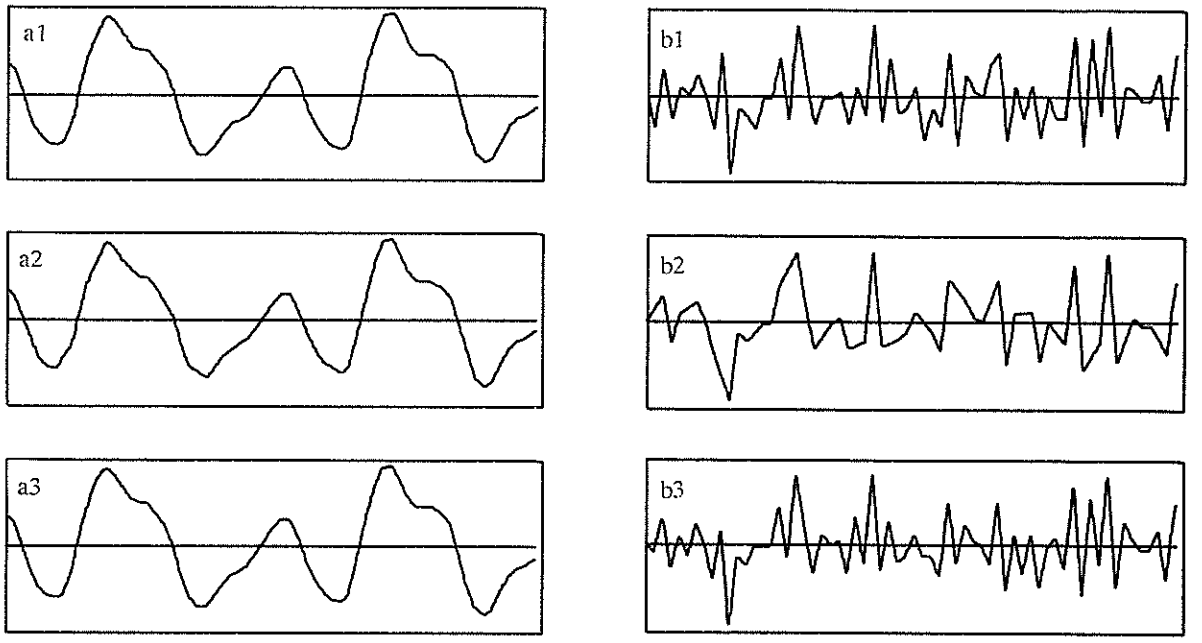


Figure 2. Waveforms of inital and recovered speech signals.
1 - the source signal
2 - the signal after losses of 25% packets and following recovery using the linear interpolation
3 - the signal after recovery using second-order adaptive interpolation

5. EXPERIMENTAL RESULTS AND CONCLUSIONS

Experiments have been made on the recovery of losses in different speech phrases. Signal samples are divided into packets, 128 samples each. Packets are combined into the groups of packets. Within one group permutations are made so as to ensure that all 4 packets are interrelated. Each fourth packet is discarded, following which a reverse permutation and recovery are performd. The use was made of the first- and second-order interpolation - non-adaptive and adaptive. The table shows the integral error estimates for all four recovery procedures for different speech phrases up to 5 sec. in length.

Table. Speech recovery errors

Type of interpolation	Error distrib. range
Linear interpolation	5-12 %
1-st ord. adapt. interp.	3-5 %
2-nd ord. Chebyshev's interpolation	5-15 %
2-nd ord. adapt. interp.	2-4 %

The investigations made testify to the efficiency of the approaches that involve waveform recovery. The first-order adaptive interpolation yields the results, acceptable in terms of sound quality, for the loss of 25% packets. The second-order adaptive interpolation yields better results both in terms of sound quality and mean square error.

REFERENCES

[1]. Chlamtac I. Computer Networks and ISDN Systems (vol.10, 2,1985, pp.81-96).
[2]. Bially T., Crold B., Seneff S. IEEE Trans. on Communic. (vol. Com-28, 3, March 1980, pp. 325-333).
[3]. Petr P.W., DaSilva L.A., Frost V.S. IEEE Journal on Selected Area in Communic., (vol.7, 5,June 1989, pp 644-656).
[4]. Clark G.G., Cain J.B. Error Correction Coding for Digital Communication (Plenum Press, New York, 1982, pp,352).
[5]. Jayant N.S., Christensen S.W. IEEE Trans. and Communic.,(vol.Com-29, 11, 1981, pp.101-109).
[6]. Rabiner L.R., ShafferR.W. Digital Processing of Speech Signals, (Prentice Hall, New Jersey 1978, pp.436).

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

ADAPTIVE LMA ECHO CANCELLER IN BASEBAND DATA TRANSMISSION WITH "IMPROVED" ERROR REFERENCE

José M. Páez-Borrallo, Francisco Lorenzo-Speranzini, Juan J. Marí y Marí
Dpto. SSR, ETSI Telecomunicación, UPM,
Ciudad Universitaria s/n, 28040 Madrid, SPAIN

This work presents a new idea on how to improve the reference error for an adaptive echo canceller working with binary baseband signals. The procedure exploits the binary nature of the involved signal to selfcancellate the masking effect of the incoming signal in a full-duplex transmission link. The adaptive scheme uses a LMA algorithm to maintain low the total computational burden. An introductory convergence analysis for the overall scheme and some simulation results are also given.

I. INTRODUCTION

Presently, one of the biggest drawbacks in conventional adaptive schemes for local echo cancellation on a two-wire full-duplex communication datalink, resides in the interfering and undesired additive effect caused by the incoming signal (and intersymbol interference) on the residual echo (see Figure I.1). This mainly takes place in advanced stages of convergence where the residual echo is substantially smaller than the total incoming signal $y(n)+i(n)$ (usually <-15 dB).

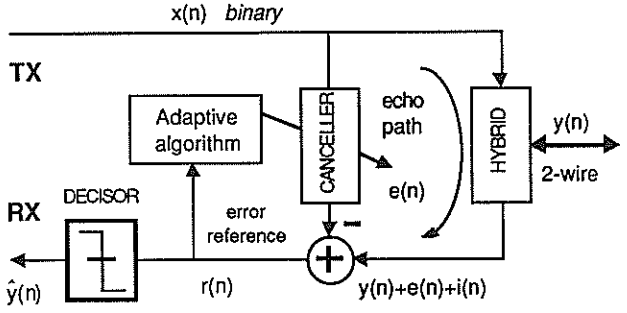


Figure I.1

The incoming signal masks the adaptive "natural" error reference of the algorithm, the residual echo, thus forcing a slow convergence towards the selected steady state value. To avoid this undesired behaviour, some new schemes based on the improvement of the error reference are found in the literature [1, 2, 3, 4, 5]; namely: a) adding an uncorrelated dithering signal to the error reference, b) removing (part of) the incoming signal (or estimations) from the error reference (decision-directed schemes) [4,6] c) implementation of a double-talk detector, and d) exploiting some statistical characteristics of the incoming signal to alleviate its presence in the error reference.

In this work, we present a new adaptive scheme for baseband data transmission derived from method d). The technique tries to gain advantage of the 2-level nature of the incoming signal to improve the error reference signal which guides the adaptive algorithm. The idea, first introduced in [5], is supported by the fact that the received signal, sampled at the symbol rate, takes only two possible and opposite values with relative high probabilities (usually 50%-50% depending on the line code and samplig phase). Therefore, a delayed (not too retarded) sample might be a good replica of the current sample level (at least statistically speaking), thus offering the possibility of cancelling, by simply addition (or substraction), the masking effect in the *old* error reference (see Figure I.2) and providing a *new* and *cleaner* error reference.

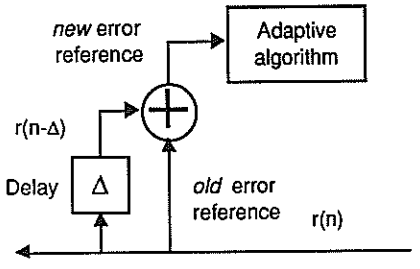


Figure I.2

This procedure is not, of course, totally free of risk, since the delayed sample plus (or minus) the *old* error reference may, even deteriorate the *new* one in, at least, half of the overall working time. Nevertheless, accepting the fact that the masking effect is only significant when the residual echo is small enough, the benefit obtained by its total cancellation counterbalances the loss of doubling the masking disturbance in such advanced stages of convergence.

In the present work, we analyze an adaptive echo canceller with some modifications to that described in [5] (see Figure I.3).

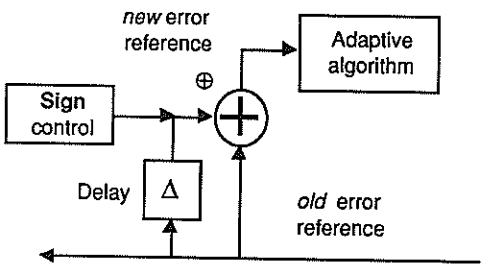


Figure I.3

The main modification presented here introduces a *sign control* which commands the delayed sample sign. This is necessary, since, to obtain an effective improvement in the convergence, as we prove in the work, the choice of a fix sign value (plus or minus in [5,7]) depends on the unknown crosscorrelation sign of the current and delayed residual echos. Thus, since there is not an *a priori* knowlegde of such a correlation (depends on the initial tap settings and other factors), a conservative sign criterium has to be adopted. Here, we propose two of them for the sign control: 1) alternating the sign every sample and 2) generating a stochastic sign for each sample. Both procedures do not fix the sign and, therefore, avoid the last pointed drawback offering an improvement of the overall convergence.

The analysis assumes bipolar-binary data contaminated with a residual and independent Gaussian intersymbol interference in a full-duplex communication link. It introduces a perspective on the choice of a constant adaption step which guarantees the convergence for any given initial tap settings (initial crosscorrelation) and power of intersymbol interference. The usual hypotheses of joint stationarity and stochastic independence among the involved processes and random variables are assumed.

II. THE ALGORITHM

The adaptive algorithm is based on the minimization of the *absolute* error reference by means of following the inverse of the stochastic gradient direction (**L M A** algorithm). Due to the linear character of the error surface, the resulting adaptive gradient algorithm simplifies its implementation since only a sign function is necessary to update the echo canceller taps. Its expresion is:

$$\mathbf{c}(n+1) = \mathbf{c}(n) - \mu\,\mathrm{sgn}[r(n) \oplus r(n-\Delta)]\mathbf{x}(n) \quad \text{(ii.1)}$$

where: **c**(.) are the canceller taps, **r**(.) the total incoming signal plus the residual echo, μ the adaption step, Δ the number of delayed samples and **x**(.) the outgoing signal. The symbol $\oplus$ denotes the current sign, plus or minus, depending on the selected sign logic.

The operation inside brackets in (ii.1) yields:

$$[y(n) \oplus y(n-\Delta)] + [i(n) \oplus i(n-\Delta)] + [\varepsilon(n) \oplus \varepsilon(n-\Delta)] =$$

$$= u(n) + z(n) + [\varepsilon(n) \oplus \varepsilon(n-\Delta)] \quad \text{(ii.2)}$$

where $u(n)$ is now a ternary signal of levels $-2\sigma_y$, 0 and $2\sigma_y$, $z(n)$ remains Gaussian and $\varepsilon(n)$ is the residual echo at time n. It means that the *differential* residual echo $\varepsilon(n) \oplus \varepsilon(n-\Delta)$ is now masked with a centered trimodal interference $u(n)+z(n)$, thus allowing sometimes (when $u(n)=0$) a cleaner reference, and therefore, implying that the sign gradient in (ii.1) be a more realistic replica of the true sign of ε(n).

III. CONVERGENCE ANALYSIS

Denoting $\mathbf{c}_h$ as the hybrid taps, the residual echo can be computed as $\varepsilon(n)=[\mathbf{c}_h-\mathbf{c}(n)]^t\mathbf{x}(n)$. Therefore, considering (ii.1) and the assumed hypotheses of joint independence, we can write the following recursive equation for the residual echo variance [8]:

$$\sigma_\varepsilon^2(n+1) = \sigma_\varepsilon^2(n)\left[1 - \mu S_1(n) + \mu^2 S_2(n)\right] \quad \text{(iii.1)}$$

where:

$$S_1(n) = 2\sigma_x^2 \frac{E\{\varepsilon(n)\,\mathrm{sgn}[u(n)+z(n)+\varepsilon(n) \oplus \varepsilon(n-\Delta)]\}}{\sigma_\varepsilon^2(n)}$$

$$S_2(n) = \frac{N\sigma_x^4}{\sigma_\varepsilon^2(n)} \quad \text{(iii.3)}$$

and N is the number of coefficients to update.

Assumig that $\varepsilon(n)$ and $\varepsilon(n-\Delta)$ is a bivariate Gaussian random variable with correlation coefficient $\rho(n)$, $u(n)$ a ternary signal with probability levels of 1/4, 1/2 and 1/4 respectively and $z(n)$ a zero-mean, independent Gaussian random variable of variance $2\sigma_i^2$, $S_1(n)$ becomes, before taking expectations on $\oplus$:

$$S_1(n) = E_\oplus\left\{\frac{\sigma_x^2[1 \oplus \rho(n)]}{\sigma_\varepsilon(n)\sqrt{\pi\left\{\sigma_i^2 + \sigma_\varepsilon^2[1 \oplus \rho(n)]\right\}}} \times \right.$$

$$\left. \times\left[1 + \exp\left(-\frac{\sigma_y^2}{\sigma_i^2 + \sigma_\varepsilon^2(n)[1 \oplus \rho(n)]^2}\right)\right]\right\} \quad \text{(iii.4)}$$

Then, the final expression of $S_1(n)$ for the sotcastich sign criterium will be:

$$S_1(n) = \frac{1}{2}E_\oplus\{\bullet / \oplus = -1\} + \frac{1}{2}E_\oplus\{\bullet / \oplus = +1\} \quad \text{(iii.5)}$$

that is, it consists in averaging (iii.4) with the substitutions of the $\pm$ signs. The corresponding average expressión for the alternate sign criterium is equivalent to that of (iii.5).

In (iii.4) the correlation coefficient $\rho(n)$ appears to be a key factor for the overall behaviour. Depending on its proximity to ±1 and the current sign adopted by $\oplus$, the resulting value of $S_1(n)$ may vary notably. In some cases, it will strongly improve the overall convergence, but in others, it will make it slower. This is the reason because a conservative choice of an alternate sign (or stochastic) in (iii.4) may alleviate the undesired effect of not compatible delayed reference sign and crosscorrelation sign.

The temporal behaviour of $\rho(n)$, although here not shown, is quasi-constant during the convergence time and zero at the steady state (at least its average values) and therefore highly predictable. This fact allows a more simplified convergence analysis just using $\rho(n)\approx\rho$ and it also would lead to a less consevative sign criterium with the use of a crosscorrelation sign predictor which would determine the right sign of the delayed error reference to be used in the algorithm.

IV. ADAPTION STEP DESIGN

For brevity (iii.1) can be expressed as:

$$\sigma_\varepsilon^2(n+1)=\sigma_\varepsilon^2(n)P(\mu,n) \qquad \text{(iv.1)}$$

where $P(\mu,n)$ is a time-dependent quadratic polynomial in μ responsible of the algorithm's convergence. The convergence condition requires $P(\mu,n)<1$, $\forall n\in[0,\infty)$. It implies for the adaption step μ:

$$(0)<\mu<\min\left[\frac{S_1(n)}{S_2(n)}\right],\ \forall n\in[0,\infty) \qquad \text{(iv.2)}$$

Since for this algorithm the ratio in (iv.2) is an increasing monotonic function of the residual echo variance $\sigma_\varepsilon^2(n)$ and the adaption step must be bounded by its minimum value, it obviously results:

$$\mu=\frac{S_1(\infty)}{S_2(\infty)} \qquad \text{(iv.3)}$$

where assuming the usual topics in the steady state, i.e., $\sigma_\varepsilon^2(\infty)<<1$, $\sigma_\varepsilon^2(\infty)<<\sigma_y^2$, $\sigma_\varepsilon^2(\infty)<<\sigma_i^2$ and $\rho(\infty)=0$ (total decorrelation), we can give an easy equation for the adaption step design which depends on the desired steady state value.

$$\mu\approx\frac{\sigma_\varepsilon(\infty)}{N\sigma_x^2\sqrt{\pi}} \qquad \text{(iv.4)}$$

This expression shows that a smaller $\sigma_\varepsilon^2(\infty)$ value requires a smaller μ, thus making slower the overall convergence (the habitual trade-off in the stochastic gradient algorithms).

V. COMPUTER SIMULATIONS

In order to verify the above assertions and analysis we have simulated the proposed technique in the computer with the following parameters:

* Canceller length, N=10; transmission levels, $\sigma_x=\pm1$; reception levels, $\sigma_y=\pm1$; without intersymbol interference and background noise ; hybrid taps, $c_h(i)=(-1)^i\,1/\sqrt{2}$ (i=0,N-1); initial tap settings, $\mathbf{c}(0)=\mathbf{0}$ (initial cross correlation $\hat{\rho}(0)\approx-0.9$) and finally, a steady state value, $\sigma_\varepsilon^2(\infty)/\sigma_y^2=-30$dB ($\mu$= 0.00178).

We run three different simulations, the first one incorporates the stochastic sign control, and the other two use fix signs (+ and -). Figure V.1 shows the results for the stochastic sign choice. The solid line corresponds to the theoretical result obtained from the equations of the above analysis. The dotted line is the corresponding simulated result. Figure V.2 shows the results for the fix sign algorithms (in all cases five realizations were averaged). Here we can observe the great difference in the speed of convergence between these two algorithms when the chosen fix sign (+ or -) does not coincide with that of the cross correlation. It is also of interest the speed differences between the results corresponding to the best case (+ sign) and the stochastic (conservative) sign algorithm.

For comparison purposes, we also show the theoretical result (dashed line) of the classical LMS algorithm for a identical steady state value of -30 dB (μ=0.0002).

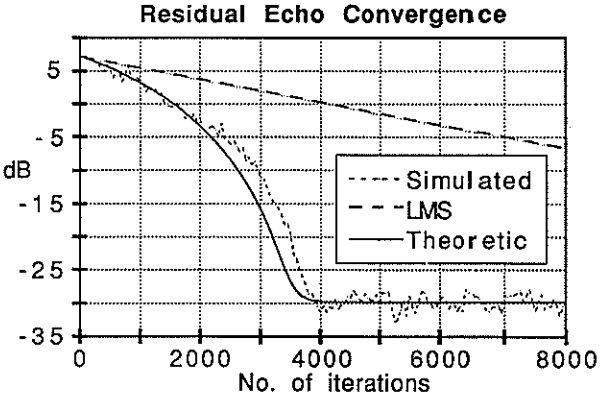


Figure V.1

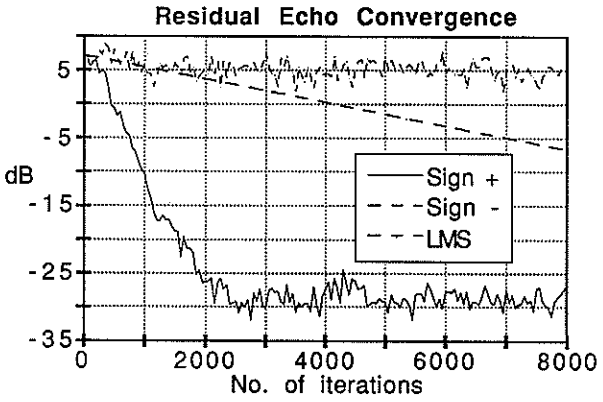


Figure V.2

In these figures you can also observe that the algorithms converge towards the selected steady state value of -30 dB, thus validating the adaption step design formula.

VI. FINAL SUMMARY

In this paper, we present and analyze a new adaptive algorithm to be used as echo canceller in a two-wire baseband data communication link. This technique exploits the statistical binary nature of the involved signals to alleviate the presence of the incoming signal in the error reference. The technique consists basically in adding to the current reference a delayed (not too retarded) sample of itself. This action provides, in at least half of the time for independent binary signals (more for other line codes), a cleaner error reference for the stochastic gradient **LMA** algorithm, thus improving the overall speed of convergence. Nevertheless, the same simple action originates a big dependence of the convergence on the crosscorrelation sign of the current and delayed residual echoes. The study and simulations here carried out shows that this sign forces the choice of the delayed reference sign to be opposite to the crosscorrelation sign. To avoid this previous knowlegde, two conservative sign criteria are here adopted: using a stochastic and independent sign or using an alternate sign. Both criteria perform identically from an average point of view and provide a fairly good convergence behaviour.

The simplified study and results here presented prove the latter assertions. They incorporate this new algorithm, or a modification of it, to the list of alternative schemes to be used in the future broadband **ISDN**, since, as it is shown in the paper, it can improve the **LMS** behaviour (and also others) with a very reduced computational burden (it only needs N+2 additions and a scaling operation per iteration).

REFERENCES

[1] Falconer, D.D. et al.: *"Adaptive echo cancellation. AGC structures for two-wire full-duplex data transmission"*; Bell Sys. Tech. Journal, vol. 58, pp. 1593-1616, Sep. 1979.

[2] Mueller, K.H.: *"A new digital echo canceller for two-wire full-duplex data transmission"*; IEEE Trans. on Communications, vol. COM-24, pp. 956-962, Sep. 1973.

[3] Falconer, D.D.: *Adaptive reference echo cancellation"*; IEEE Trans. on Communications, vol. COM-30, pp. 2083-2094, Sep. 1982.

[4] Mazo, J.E.: *"Analysis of decision-directed equalizer convergence"*; Bell Sys. Tech. Journal, pp. 1957-1876, Dec. 1980.

[5] Kanenasa, A.et al.: *"An echo canceller adaption methof for digital subscriber loop transmission"*; in Proc. ICASSP-86.

[6] Macchi, O. et al.: *'Convergence analysis of self-adaptive equalizers"*; IEEE Trans. on Inf. Th., vol. IT-30, pp. 161-176, Mar. 1984.

[7] Kanemasa, A. et al: *"A study on the subscriber loop transmission system for ISDN based on the echo cancellation technique"*, in Proc. ICC-85.

[8] Páez-Borrallo, J.M. et al.: *"Convergence analysis of L1 and L2 decision-directed adaptive echo cancellers for baseband data transmission"*, submitted to Signal Processing for publication.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

THE COMPARISON OF THREE IMPLEMENTATION METHODS OF AN ECHO CANCELLER FOR 2400 BIT/S FULL-DUPLEX MODEM BASED ON A SIGNAL PROCESSOR

Hanna BOGUCKA

Technical University of Poznań,
The Institute of Electronics and Telecommunication,
Ul. Piotrowo 3A, 60-965 Poznań, Poland.

In this paper the possibility of applying three echo canceller structures for V.26.ter modem is considered. These are as follows: the voice-type, the data-driven and the data-driven cascade structure. It is proposed to develop the last mentioned structure. The implementation is based on a signal processor. Some interesting, new ways to achieve time-optimized implementation programs are described. Finally, computer simulation results are provided.

1. INTRODUCTION

From all known realization methods of the full-duplex, two-wire data transmission the idea of an echo cancellation warrants the longest range. The structure of a transmission system based on this idea has been presented in many publications: [1]-[6]. It applies hybrids, which due to the impedance mismatch cause the energy leakage of the transmitted signal from the transmitter to the local receiver. This leakage known as the echo signal has two components: the near-end echo, coming from the local hybrid and the far-end echo, caused by reflections in a line or imperfect matching of the far-end hybrid. Both are eliminated by the echo canceller in the following way: a) an echo replica is calculated on the basis of the transmitted and received signals (e.g. by the means of a transversal filter or look up table), b) then, the echo replica is subtracted from the far-end signal. The result of this subtraction is known as the error signal.
The echo estimation requires numerous multiplications and additions if a transversal filter is applied in a canceller. The number of multiplications, on which so much attention has been paid till now ([1]), is not so important, when the implementation is based on a signal processor.
In this paper research tending towards designing the echo canceller for V.26. ter modem is shown. Three implementation methods of the canceller based on the case of the TMS32010 signal processor are presented.
In section 2 the requirements of the CCITT V.26.ter Recommendation and all the parameters of an echo signal taken into account are shortly described. A possibility of applying the voice-type canceller for V.26.ter modem is discussed in section 3. Section 4 contains description of the data-driven echo canceller structures based on the idea of subcancellers. One of them applies cascade-connected cancellers, which reduce echo, first, roughly and then, precisely. The time optimization of the programs implementing all presented structures is described and simulation results are provided.

2. SYSTEM REQUIREMENTS

The CCITT V.26.ter Recommendation refers to the data transmission of 2400 bit/s. The symbol rate is: 1/T= 1200 Bd. Two types of 4-QAM modulation indicated as "A" and "B" are realizable. In this work version "B" is taken into account.
For our consideration the sampling frequency has been choosen six times higher than the symbol rate, that is: 1/T'= 7200 Hz.
It has been also assumed that the echo impulse response extends over 20 symbol intervals of the transmitted data, so it has 120 samples.
The near-end echo level is about 34 dB higher than the level of the far-end signal. A signal-to-noise ratio is required to be at least 20 dB, so an echo canceller has to reduce the echo level by at least 54 dB.
The canceller applies the stochastic gradient algorithm in two versions: the least- mean-square (LMS) or the block- update algorithm.
The CCITT V.26.ter Recommendation describes the test data sequence required for initial adaptation of the canceller. This sequence should be transmitted during the start-up procedure for at least 650 ms. This time has been also assumed for simulation.

3. THE VOICE-TYPE ECHO CANCELLER

In traditional realization known as the voice-type the transmitted signal is an input signal. It's structure is widely known ([4], [6]) and will not be presented here. The length of the delay line in a canceller relates to the number of samples of the echo impulse response (120).
The assembler program implementing this

structure has been time-optimized. To obtain the shortest realization time, an inverse order of the echo estimate calculation and coefficient updating has been applied. First, a coefficient is updated using the input signal sample and the value of the error signal from the previous sampling moment according to the formula 1:

$$c_i^n = c_{i-1}^n + \beta \cdot e_{i-1} \cdot d_{i-1}^n \qquad (1)$$

where:

c_i^n is the value of the n-th coefficient at the moment: $i \cdot T'$,
β is the adaptation step size,
e_i is the value of the error signal at the moment: $i \cdot T'$,
d_i^n is the value of the n-th input data bit from the delay line at the moment: $i \cdot T'$.

Then, it is multiplied by the corresponding bit of the input signal and added to the value of the estimated echo at the present moment according to the formula 2:

$$\varepsilon_i = \sum_{n=1}^{N} c_i^n \cdot d_i^n \qquad (2)$$

where:

ε_i is the value of the echo estimate calculated at the moment: $i \cdot T'$,
N is the number of samples of the echo impulse response: N=120.

After the echo estimate is calculated, it is subtracted from the actual value of the far-end signal sample. As we see, the value of the error signal from the previous sampling moment: e_{i-1} and the value from the last position of the delay line: d_{i-1}^N have to be stored, in order to be used for coefficients updating at the given moment (the other values d_{i-1}^n need not to be stored, because they are still in the delay line shifted to new positions). This way the necessity to read the canceller coefficients twice from the data memory is avoided and the execution time is shortened by 5%.

Simulation results show, that this type of a canceller is suitable only for the transmission of a signal which has good stochastic properties (samples should have the uniform distribution and there should be no correlation between them). In this case, when the optimum step size β_{opt} is used in the gradient algorithm (see formula 3:

$$\beta_{opt} = \frac{1}{2 \cdot N \cdot D} \qquad (3)$$

where D is the power of the input signal d(t)), the echo attenuation of more than 100 dB after 650 ms is achieved (see fig.1.).

The transmitted signal samples in V.26.ter modem do not comply with previously specified requirements, since this signal can be written according to the formula 4.

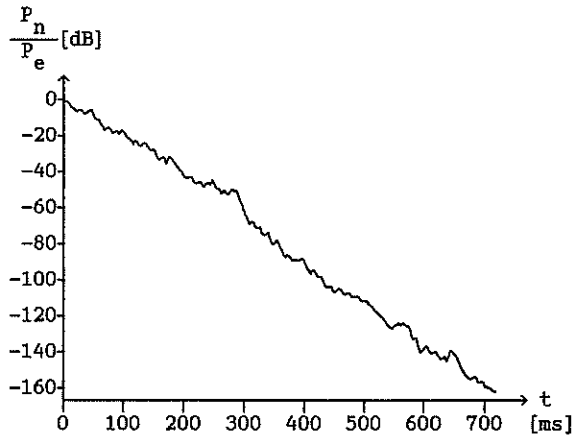


Figure 1. Echo attenuation in the voice-type canceller (uniform distribution of the input samples); P_n - the power of the un-cancelled echo signal at the output of the canceller; P_e - the power of the echo signal at the input of the canceller.

$$d(t) = \sum_{j=1}^{K} a_j \cdot h(t-j \cdot T) \cdot \cos(2 \cdot \Pi \cdot f_c \cdot t) + \sum_{j=1}^{K} b_j \cdot h(t-j \cdot T) \cdot \sin(2 \cdot \Pi \cdot f_c \cdot t) \qquad (4)$$

where:

d(t) is the value of the transmitted signal at the moment: t,
a_j is the value of the in-phase component,
b_j is the value of the quadrature component,
h(t) is the transmit filter impulse response,
f_c is the currier frequency: 1800 Hz,
K is the length of the digital transmit filter.

As the values of signals a_j and b_j have the uniform distribution, the summation of these values causes that some summation results are more probable than the others. On the other hand, the value of d(t) at each sampling moment depends on the buffer contens of the transmit filter, which does not change during six succeeding sampling periods. Thus, it is better to apply the block-update algorithm with the block size M=6 and update coefficients only once for six sampling periods. Unfortunately, this tends to slow convergence. Moreover, in such an adaptive filter analog input samples have to be stored very accurately.

These two disadvantages cause that the required dynamic range is not achieved after 650 ms. Consequently, the traditional voice-type structure is not a very good solution for the V.26.ter modem.

4. DATA-DRIVEN ECHO CANCELLER

4.1. Basic structure

During last few years data-driven echo canceller structure has been popularized, where coded and scrambled data are the input signal of a canceller. If it is a transversal filter, it's coefficients must relate to the convolution of the transmit filter and the echo channel impulse responses. As the sampling frequency (1/T') is higher than the symbol rate (1/T), the canceller's delay line contains L-1 zeros for each data symbol, where L=T/T'. Thus, the number of multiplications required for the echo estimation is reduced, as compared with a standard voice-type canceller. This notice gave the idea of a new structure based on subcancellers, which has been presented in many publications (e.g.: [1],[2],[3]). This structure is shown in figure 2.

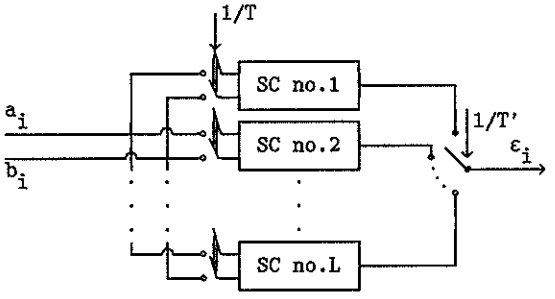


Figure 2. The DD echo canceller structure based on subcancellers; SC- subcanceller

Since L=T/T'=6, six subcancellers are applied. In order to obtain full dynamic range (echo should be attenuated at least 54 dB), the step size requires 12-bit representation, 12-bit A/D and D/A converters should be applied and finally, the coefficients of the transversal filter should be represented by 24-bit words. Thus, each coefficient needs two microprocessor words, what has a great influence on the way, in which the signal processor multiplies it by data from the delay line. After a data symbol is multiplied by the "less-significant" (LS) word of the corresponding coefficient, the sign extension is imposed on the result. This sign extension should be suppressed somehow before adding the previously mentioned result to the result of multiplication by the "more-significant" (MS) word of the coefficient.
The problem to realize double precision multiplication was solved by introducing non-conventional way of coding. The coefficient double word representation is not regarded as LS and MS words, but as the sum of two components. The first one is the LS word and the second is the result of subtraction the LS word with sign extension from the whole double word conventional representation. The idea of such coding is shown in figure 3.

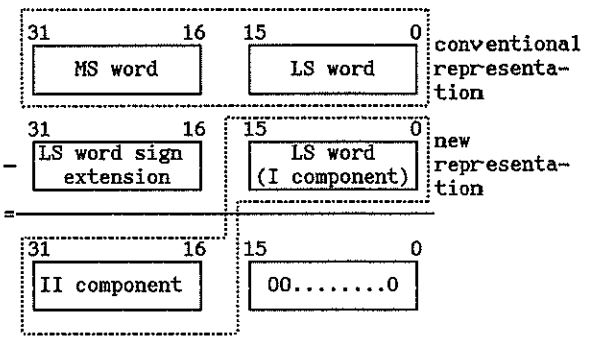


Figure 3. The idea of the filter coefficients coding.

For such represented value the double precision multiplication can be easily realized. We multiply the data symbol by the first component and then, by the second one. We add both results and obtain full multiplication result, coded in conventional way. Obviously, this double word representation must be changed, what can be easily done according to fig.3. Thanks to previously described coding the realization time is considerably shortened.
Also in case of the DD echo canceller an inverse order of the echo estimation and coefficients updating has been applied.
After summation of all clock cycles needed for one subcanceller realization it proved, that it takes 2489 cycles. If the clock frequency is 5 MHz, the program execution time is:
T_e=0.4987 ms. The value of T_e is mostly determined by the fact, that TMS32010 signal processor has only a 144 word data memory. Consequently, all the double word coefficients have to be stored in the program memory. The read (TBLR) and write (TBLW) instructions take most of the time (3 clock cycles), what has a significant influence on the value of T_e.

As T=0.8333 and

$$\mathrm{int}\left[\frac{T}{T_r}\right] = \mathrm{int}\left[\frac{0.8333}{0.4978}\right] = 1$$

(where int(·) is the integer function of ·), one signal processor is able to perform operations of only one subcanceller.
Simulation results show, that if the initial values of the coefficients are zeros, the DD canceller structure warrants the echo attenuation of more than 60 dB after 650 ms. It can be seen in fig.4. This confirms the correctness of such implementation.

4.2. The DD echo canceller cascade structure

This type of a canceller was originally introduced by E.Arnon, W.Chomik and M.Elder for ISDN applications [7], however it can be also considered as the potential solution for V.26.ter echo cancellation. In this canceller structure each subcanceller is split into two cascade-connected units, what is shown in fig.5.

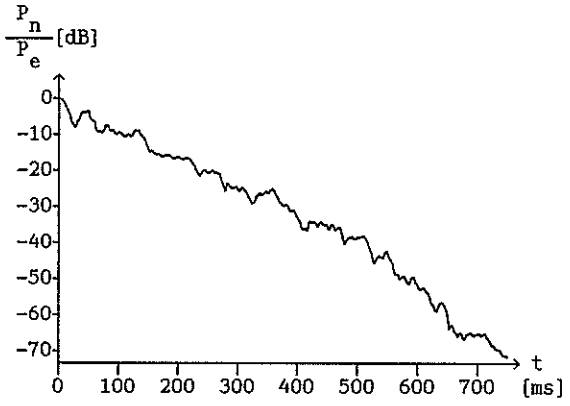


Figure 4. Echo attenuation in the DD canceller

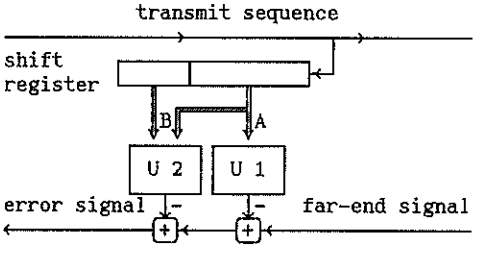


Figure 5. The DD echo canceller cascade structure; A - first bits of the input signal covering large amplitudes of the echo, B - full bit-range of the input signal, U1 - first unit, U2 - second unit.

Each unit applies the gradient algorithm. The first one has the smaller number of coefficients than the other. It reduces large amplitudes of the echo, which cover first 3 ÷ 4 symbol intervals. After this rough pre-cancellation an input far-end signal undergoes precise cancellation in the next unit with the full number of coefficients. Each component of the cascade has a smaller dynamic range than the whole subcanceller, so, each of it's coefficients can be coded using only one 16-bit memory word. Thus, because double precision operations are not required, the time of cancellation is considerably shortened and the use of 8-bit A/D and D/A converters is possible. Futhermore, the bigger number of coefficients can be stored in data memory, what reduces the use of TBLR and TBLW instructions. Basing on the idea of the previously described cascade structure it is possible to split each subcanceller into more than two units. This operation does not shorten the execution time, when the TMS32010 processor is applied for the implementation, because the TMS32010 operates on 16-bit words. However it increases the echo estimation accuracy and the signal-to-noise ratio at the output of the canceller. In table 1 basic parameters of the cascade structures are summarized.

Table 1.

nu	nb	cl	t_e[ms]	ns
1	24	12	0.4978	1
2	16	8	0.3606	2
3	8	4	0.3901	2

nu- the number of units in the cascade,
nb- the number of bits of the coefficients representation,
cl- the A/D and D/A converters length,
t_e - the assembler program execution time,
ns- the number of subcancellers implemented by one signal processor.

5. Conclusions

From all implementation methods presented here the last one (the DD echo canceller cascade structure) seems to be the best suited. As the echo estimate accuracy rises with the number of units in the cascade, it is possible that we obtain the good performance with the cascade-connected voice-type cancellers. The inferior accuracy typical for voice-type cancellers could be eliminated this way. However no research work has been done on this suggestion. All three methods presented here are exactly described to give the opportunity of a good choice to the designer.

Acknowledgments

The author wishes to thank Dr. K. Wesołowski for his valuable advices and comments.

References:

[1] Guidoux, L., Peuch, B. Binary Passband Echo Canceller in a 4800 Bit/s Two-Wire Duplex Modem, IEEE J. Select. Areas. Commun., pp. 711-721, Sep. 1984.
[2] Werner, J.J., An Echo-Cancellation-Based 4800 Bit/s Full-Duplex DDD Modem, IEEE J. Select. Areas. Commun., pp.722-730, Sep.1984
[3] Verhoeckx, N.A.M., van den Elzen, H.C., Snijders, F.A.M., van Gerwen, P.J., Digital Echo Cancellation for Baseband Data Transmission, IEEE Transactions on Acoustics Speach and Signal Processing, vol. ASSP-27, No.6, pp. 768-781, Dec. 1979.
[4] Digital Signal Processing Applications with the TMS320 Family, Theory, Algorithms and Implementations, Texas Instruments, 1986.
[5] Arnon, E., Chomik, W., Elder, M., A Transmission System for ISDN Loops, IEEE Bell-Northern Research, Ottawa, Canada, pp. 192-200, 1986.
[6] Cowan, C.F.N., Grant, P.M., Adaptive filters (USA, New Jersey, 1985).
[7] "CCITT Red Book", Rec.V.26.ter.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

DECONVOLUTION OF A MIXED PHASE SEQUENCE BY TIME DOMAIN CEPSTRAL TRANSFORMATIONS

Radomir T. Sokolov, James C. Rogers

Electrical Engineering Department, Michigan Technological University,
Houghton, Michigan, 49931, USA

The deconvolution of composite signals is approached using a new technique, the Time Domain Cepstral Transformations (TDCT). The technique is based entirely on time domain calculations in which a truncated time sequence is directly transformed to the cepstral domain where deconvolution is achieved using the usual frequency invariant filters. An inverse transform is used to obtain the desired signal. Examples presented show improved results of the new method over conventional techniques.

1. INTRODUCTION

Homomorphic deconvolution of a signal obtained by convolving a mixed phase sequence and an impulse train having nonuniform impulse spacing is an important signal processing problem; seismic signals are an example. We apply a new technique, the TDCT method introduced by Sokolov [1,2]. This method of complex cepstrum system realization is entirely based on time domain calculations. It thus avoids or minimizes problems associated with the common method based upon the Fourier Transform (FT), Oppenheim and Schafer [3]. A window is often applied to non-stationary data when the overall convolutional model holds only on a short-time basis. However, application of a window affects the convolutional model [4]. The TDCT method avoids the need for special time windows, the analyzed signal is simply gated (rectangular window). Also, there is no need for unwrapped phase calculations.

2. DERIVATION OF THE TDCT METHOD

Homomorphic deconvolution introduced by Oppenheim and Schafer [5,6] is a widely used nonlinear technique, Childers et al [7].
The complex cepstrum is defined as the inverse Z-transform of the log of the Z-transform of an ordinary time sequence $x(n)$. The complex cepstrum exists if the complex logarithm $\log(X(z))$ is analytic. This is achieved in the FT method by calculating the unwrapped phase and finding the number of sample periods by which the discrete-time (DT) sequence $x(n)$ should be shifted to obtain the aligned DT sequence $y(n)$ [6]. In the TDCT method we propose another sequence alignment method.
A nonlinear difference equation relates an ordinary (aligned) sequence y and its complex cepstrum sequence, c [6].

In some special cases (minimum or maximum phase) this implicit relation in terms of y and c can be reduced to implicit recurrence expressions [6]. However, the TDCT method

$$n\, y(n) = \sum_{k=-\infty}^{k=+\infty} k\, c(k)\, y(n-k) \qquad (1)$$

provides an *explicit* and *unique* transformation between an ordinary time mixed phase sequence y and c and vice versa. With notation $c=\{c(n)\}$ and $y=\{y(n)\}$ we have

$$c = T_{yc}\, y \quad , \qquad y = T_{cy}\, c \qquad (2)$$

where T_{yc} is the TDCT from $y \longrightarrow c$ and T_{cy} is the TDCT from $c \longrightarrow y$. Since only a single sequence, x, is available, we do not build a convolutional vector space that contains x. Instead we use (1) to establish matrix relations between y and c and find the TDC transformations [1,2].

$$y = \left[\, y_{-M}\; y_{-M+1} \cdots y_0\; y_{-1} \cdots y_M \,\right]^T$$
$$c = \left[\, c_{-M}\; c_{-M+1} \cdots c_0\; c_{-1} \cdots c_M \,\right]^T \qquad (3)$$

The definitions of y and c in (3) restrict the infinite-dimensional convolutional and cepstral spaces to (2M+1) dimensions. Equation (1) becomes:

$$n\, y_n = \sum_{k=-M}^{k=M} k\, c_k\, y_{n-k} \quad , \quad n = -M, \ldots, M \qquad (4),$$

In matrix form equation (4) is:

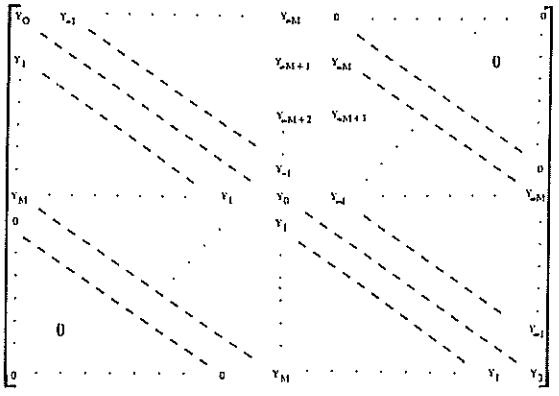
$$\mathbb{D}\, y = A\, \mathbb{D}\, c \qquad (5)$$

$\mathbb{D}$ is a diagonal matrix with main diagonal $[-M,-M+1,\ldots,-1,0,1,\ldots,M]$. Matrix A (7) is banded Toeplitz and the entries are samples of y. Since sequences y and c are restricted to 2M+1 samples, (5) is an approximation of (6).

$$A\, \mathbb{D}\, c - \mathbb{D}\, y = \epsilon \qquad (6)$$

where ϵ is some error vector that depends on the (2M+1) finite-dimensional approximation of the infinite-dimensional spaces. We call ϵ

the residual error. It is necessary to construct the generalized inverse $\mathbb{G}_{yc}$ of the



(7)

singular matrix $\mathbb{A}\,\mathbb{D}$, restricted by (6) so:

$$c = \mathbb{G}_{yc}\,\mathbb{D}\,y = \mathbb{T}_{yc}\,y \qquad (8)$$

The investigation of these restrictions is covered in detail in [1,2]. Here we assume that M is large enough so that $\epsilon(0) \cong 0$. The characteristics of the important items in the above equations can be summarized:

1. The system of equations (5) is inconsistent.
2. Both vectors y and c belong to the orthogonal complement of the vector $\mathbb{D}\,\mathbb{J}\,y$ that is completely situated inside the row space of $\mathbb{A}\,\mathbb{D}$.
3. y and c are orthogonal to each other and with weighted matrix $\mathbb{J}\,\mathbb{D}$.
4. The vector $\mathbb{D}\,y$ is completely inside the row space of matrix $\mathbb{A}\,\mathbb{D}$.

Equation 5 can be manipulated and rewritten:

$$-n\,c_n\,y_0 = -n\,y_n + \sum_{\substack{K=-M \\ n \neq K}}^{K=M} K\,c_K\,y_{n-K} \qquad (9)$$

where $n \in \{-M, \ldots, M\}$. In matrix form (9) is:

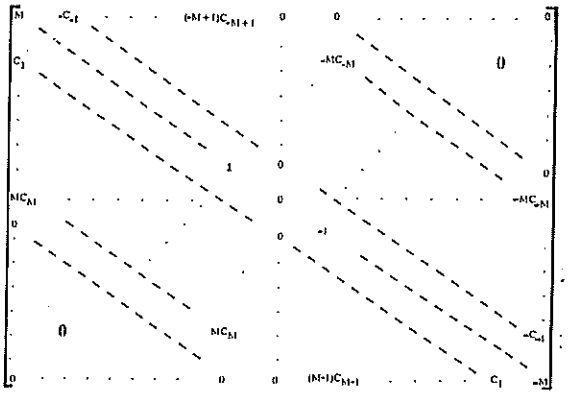
$$y_0\,\mathbb{D}\,c = -\mathbb{B}\,y \qquad (10)$$

The $(2M+1) \times (2M+1)$ matrix $\mathbb{B}$ is shown in (11). Entries of $\mathbb{B}$ are the samples of the cepstral sequence. In summary:

1. Both vectors, y and c belong to the orthogonal complement of the vector $\mathbb{J}\mathbb{D}c$ which belongs to the row space of $\mathbb{B}$.
2. y and c are orthogonal to each other and with weighted matrix $\mathbb{J}\,\mathbb{D}$.
3. The vector $\mathbb{D}\,c$ is completely inside the row space of matrix $\mathbb{B}$.

If A is a nonsingular matrix the row space of $\mathbb{A}\mathbb{D}$ has dimension 2M. Thus the transformation of y into the vector c should have domain and range equal to the 2M-dimensional row space of $\mathbb{A}\,\mathbb{D}$. The calculation of $c(0)$, is a one-dimensional problem and can be done separately [1,2]. With such a setting we actually restrict the consideration to a transformation having range and domain equal to the row space of the singular matrix $\mathbb{A}\mathbb{D}$.

In [1,2] a theorem is stated (partly due to Rao and Mitra ; Ben-Israel and Greville) which gives the necessary and sufficient conditions for construction of the



(11)

generalized inverse with specified column and row spaces. Also it is shown that conditions of the theorem are met in the case of the inversion of the singular matrix $\mathbb{A}\mathbb{D}$.

The transformation $\mathbb{T}_{yc}$ which uniquely transforms y into c is given by:

$$\left[I - \frac{(\mathbb{D}\mathbb{J}y)(\mathbb{D}\mathbb{J}y)^T}{(\mathbb{D}\mathbb{J}y)^T(\mathbb{D}\mathbb{J}y)}\right](\mathbb{A}\,\mathbb{D})^T\left[(\mathbb{A}\,\mathbb{D})^2\,(\mathbb{A}\,\mathbb{D})^T\right]^{(1)}\mathbb{A}\,\mathbb{D}^2 \qquad (12)$$

The (1)generalized inverse can be calculated as a Moore-Penrose pseudoinverse since it is also a (1) generalized inverse. It is also possible to consider the Moore-Penrose pseudoinverse of $\mathbb{A}\mathbb{D}$ in order to get the $\mathbb{T}^{+}_{yc}$ transformation [1,2].

The TDC transformation $\mathbb{T}_{yc}$ (12) does not minimize the error in (6). However, the residual error obtained, $\epsilon = \mathbb{A}\mathbb{D}c - \mathbb{D}y$, will be of minimum norm $\|\epsilon\|$ (over the set of different shift positions) when the ordinary time sequence x is properly aligned. We have tested this assertion in a number of synthetic examples and verified that it is true in these cases. Also, it was observed that $\|\epsilon\|$ is very sensitive to the shift position of y. This lends confidence to the procedure for time alignment which does not require an unwrapped phase calculation. In addition, a sequence which is rectangle windowed (simply truncated) need not be modified by some specific window in order to achieve precise alignment. Hence, the TDCT method can treat cases when the poles and zeros of the time sequences's Z-transform are situated on the unit circle, provided that the finite dimensional approximating spaces are of the sufficient dimension. Recall the second summary above which gives geometric

relations of interest for the unique $c \longrightarrow y$ TDC transformation. It should have domain and range equal to the 2M-dimensional row space $\mathbb{B}$. Therefore the generalized inverse $\mathbb{G}_{cy}$ of the singular matrix $\mathbb{B}$ satisfies

$$y = -x(0)\,\mathbb{G}_{cy}\,\mathbb{D}\,c = \mathbb{T}_{cy}\,c \qquad (13)$$

Thus we want $\mathbb{T}_{yc}\,\mathbb{T}_{cy}$ to be an identity transformation. From the summaries one can see there is a "duality" between the transformation $\mathbb{T}_{yc}$ and $\mathbb{T}_{cy}$. A similar approach produces transformation $\mathbb{T}_{cy}$ given by:

$$-y(0)\left[\, I - \frac{(\mathbb{J}\,\mathbb{D}\,c)(\mathbb{J}\,\mathbb{D}\,c)^T}{(\mathbb{J}\,\mathbb{D}\,c)^T(\mathbb{J}\,\mathbb{D}\,c)} \,\right] \mathbb{B}^T(\mathbb{B}^2\mathbb{B}^T)^{(1)}\mathbb{B}\mathbb{D} \qquad (14)$$

3. Example One

Figure 1(a) gives the pole/zero locations of a system with the impulse response of Figure 2(a). Convolving this with the pulse train in Figure 2(b) produces the signal in Figure 3. We analyze a 64 point sequence from Figure 3 starting at sample 40 which does not coincide with an impulse of the composite signal.

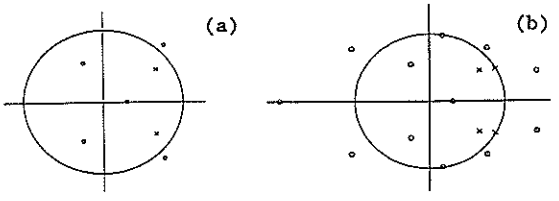


Fig. 1 Pole/Zero locations (a) Example One (b) Example Two

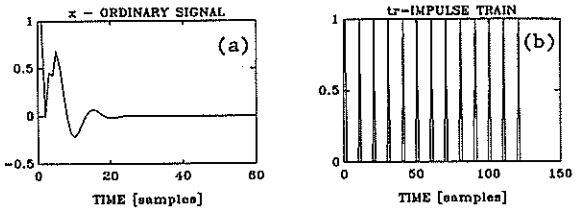


Fig. 2 Ordinary Time Signal and Impulse Train

The TDCT complex cepstrum of x is shown in Figure 4. The theoretical cepstrum (TH) for the signal very closely matches that Figure. The TDCT complex cepstrum (64 sample long) for the composite signal is shown in Figure 5. Compare Figures 4 and 5 and note that the impulse train positions are precisely located in the cepstral domain. Also, the low cepstral time part of the TDCT complex cepstrum closely approximates the TDCT (or TH) complex cepstrum of Figure 4.

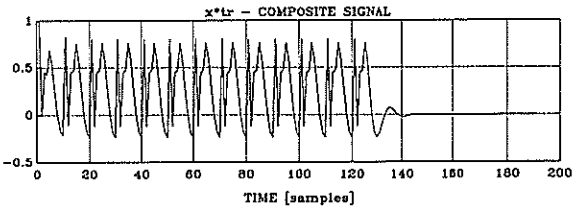


Figure 3 Composite Signal

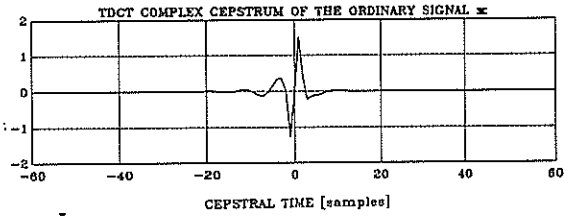


Fig. 4 TDCT Complex Cepstrum of the Signal x

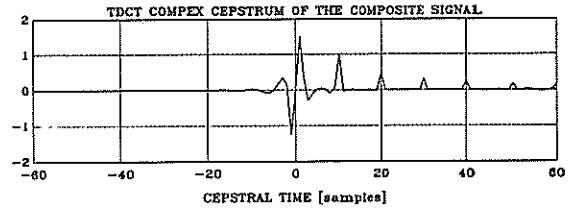


Figure 5 TDCT Complex Cepstrum of Rectangle Windowed Composite Signal

Thus a rather precise deconvolution of the impulse train and the basic mixed phase signal can be obtained. The FT complex cepstrum (Fig. 6) calculated after appending zeros and exponential weighting gives large dissimilarities with Figure 4 in low cepstral time. Ref. [11] was used to calculate the FT cepstrum. Also the FT high cepstral portion is missing the last impulse (see Figures 5 and 6). Thus deconvolution of the composite signal into the basic mixed phase sequence and impulse train, would be inaccurate by the FT method.

The rectangle windowed composite signal was aligned and the minimum error $\|\epsilon\| = 2.182\ 10^{-5}$ was obtained by a left shift of 3 sample places. Other shifts gave larger errors: x shifted 2 places gave 50.08 and 4 places gave $9.7\ 10^{3}$. Other rectangle window widths and starting points were tried. The longer the sequence the better the results obtained.

Proper alignment is necessary. For example: a

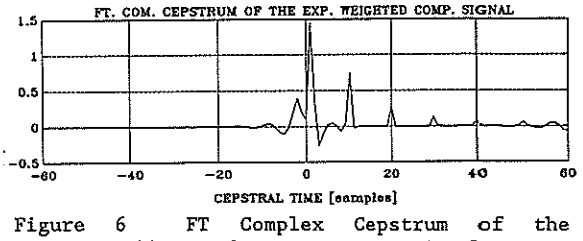


Figure 6 FT Complex Cepstrum of the Exponentially Weighted Composite Signal

window 64 samples long with onset at the 46-th sample of the composite signal (Figure 3) is

shifted k=5 samples to the left. The $\|\epsilon\|$ for k=3 is 1.077 10^3, for k=4 it is 0.790 but for k=5 it is 0.0033. For larger values of k the error is still over 0.0033.

4. Example Two

The complex cepstrum (TDCT) for the non uniform 3-impulse train of Figure 7(a) is shown in Figure 7(b). The impulse train is the same as that analyzed by Stoffa et al [8]. However, we don't apply any exponential window to precondition the signal. The TDCT cepstrum has the exact numerical values as those calculated by equation (17) [8] with a=1.

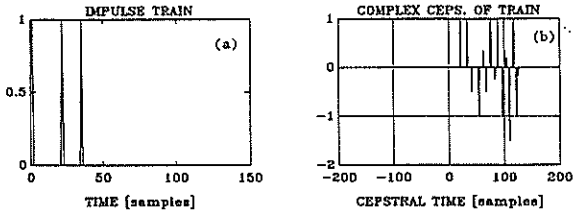


Figure 7 (a) Impulse Train (b) Complex Cepstrum (TDCT) of the Train

A mixed phase sequence (Figure 8(a)) was produced by gating (rectangular window) the impulse response of a system with the zero/pole distributions of Figure 1(b). Note that some zeros and poles are situated almost on the unit circle (r=0.99). The TDCT complex cepstrum in Figure 8(b) has numerical values very close to those calculated by formula 10.45 of [3]. Convolving signals from Figures 7(a) and 8(a) produces Figure 9.

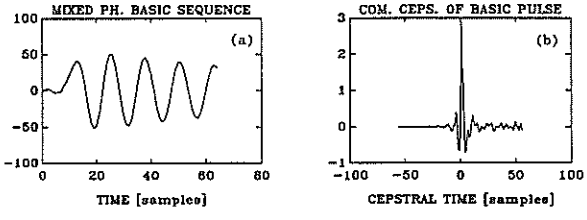


Figure 8 (a) Mixed Phase Basic Sequence
(b) TDCT Cepstrum of Basic Sequence

The TDCT cepstrum of Figure 9 and that of the basic pulse compare favorably in Figure 10 which promises precise deconvolution.

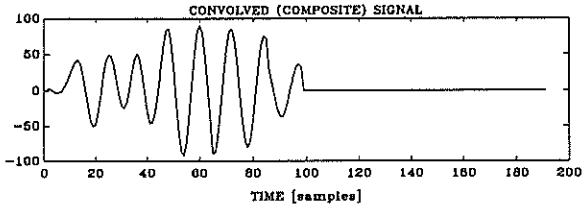


Figure 9 Composite Signal

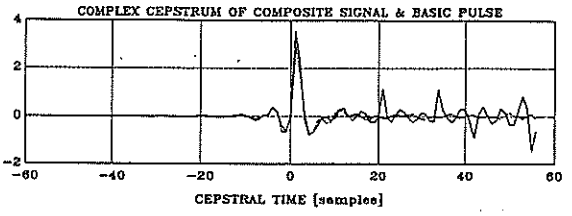


Figure 10 Cepstrum of Composite Signal and Basic Pulse

5. Conclusion

The recovery of the basic pulse depends on the length of the composite signal under consideration (the longer the better). We emphasize that no specific window other than the implicit - rectangular window from gating the time sequence was used. Also, no phase unwrapping calculations are necessary yet the TDCT method provides superior results. The method is robust in determining impulse trains, which is a promising feature for seismic signal analysis and for time delay estimation. The improved performance is obtained at the expense of a higher computational load compared to the FT method.

References

[1] R. T. Sokolov, 1989, Time Domain Cepstral Transformations", Ph.D. dissertation, Elect. Eng. Dept., Michigan Technological University.
[2] R. T. Sokolov, 1989, "Time Domain Cepstral Transformations", sub. to IEEE Trans., ASSP
[3] A.V. Oppenheim and R.W. Schafer, 1975, "Digital Signal Processing", Prentice-Hall.
[4] J. M. Tribolet, T. F. Quatieri, A. V. Oppenheim, 1977, "Short-Time Homomorphic Analysis", Proc. ICASSP, pp. 716-722.
[5] A. V. Oppenheim, 1965, "Superposition in a Class of Nonlinear Systems", Technical Report 432, Research Lab. of Electronics, M. I. T., Cambridge, MA.
[6] R. W. Schafer, 1968, "Echo Removal by Discrete Generalized Linear Filtering", Ph.D Thesis, Dep. of EE, M. I. T., Cambridge, MA.
[7] D G. Childers, D. P. Skinner, R. C. Kemerait, 1977, "The Cepstrum: A Guide to Processing", Proc. IEEE, vol. 65, No. 10, 1428-1443.
[8] P.L. Stoffa, P. Buhl, and G. Bryan, 1974, "The Application of Homomorphic Deconvolution to the Shallow-Water Marine Seismology--Part I: Models", Geophys., v 39, no. 4, pp. 406
[9] J. M. Tribolet and T. F. Quatieri, 1979, Programs for Digital Signal Processing, IEEE Press.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A BLIND DECONVOLUTION METHOD

Ryszard MAKOWSKI

Institute of Telecommunication and Acoustics, Technical University of Wrocław,
Wybrz. Wyspiańskiego 27, 50-370 Wrocław, Poland

A new method of blind deconvolution, applicable in the analysis of output of the system (being a parallel connection of elementary vibrating systems), excited with a short-term signal has been proposed. Some results of the deconvolution of test-signals, proving the effectiveness of the method, have been presented. The method is to be implemented in the deconvolution of seismic signals gathered in copper mines for which the use of predictive and homomorphic deconvolution fails. The results of the seismic signals deconvolution are very prospective.

1. INTRODUCTION

Let the output y(t) of the system be given as

$$y(t) = x(t) * h(t) + n(t) \qquad (1)$$

where x(t) is the system input, * denotes convolution, h(t) is the system impulse response and n(t) is a noise. The task to be solved is separation of the excitation x(t) and the system impulse response h(t). If n(t) = 0, and y(t) as well as h(t) are both known, it is possible to determine x(t) using inverse filtering. Sometimes, however, neither x(t) nor h(t) are known. In these cases we are dealing with a blind deconvolution problem. Well-known methods of blind deconvolution require an assumption about the system impulse response or/and the input signal. Deconvolution subjects are seismic signals gathered in the copper mines. Unfortunately, predictive and homomorphic deconvolution methods performed to these signals have not provided satisfactory results. It has turned out that signals observed in the copper mines, differ essentially from those occurring in seismic exploration or in earthquakes, because of differences in conditions of seismic waves propagation.

If:

- the input is a short-term impulse,
- the system is constituted of parallel connection of elementary systems, i.e.

$$h(t) = \sum_{n=1}^{N} h_n(t) \qquad (2)$$

- the elementary systems are of vibrating type,
- the noise is weak,

then the deconvolution can be performed using the method proposed below. The method consists of two basic stages: the first includes estimation of the system impulse response parameters (Sec. 2), the second - the inverse filtering implementing the estimators of the system parameters (Sec. 3), associated with an optimization (Sec. 4).

2. ESTIMATION OF THE SYSTEM IMPULSE RESPONSE PARAMETERS

If $x(t) \approx a_x \delta(t-t_x)$ and $H_n(f) = FT\{h_n(t)\} \approx a_n \delta(f-f_n)$, where $\delta(\circ)$ is an impulse, then following (2) we get

$$y(t) \approx a_x h(t-t_x) + n(t) = \sum_{n=1}^{N} a_x h_n(t-t_x) + n(t) \approx \sum_{l=1}^{L} \alpha_l w_l(t) \qquad (3)$$

where the set of coefficients $\{\alpha_l\}$ denotes the representation of a signal y(t) in a given basis $\{w_l(t)\}$. Let the impulse responses be given as

$$h_n(t) = b_n 1(t-t_n)(t-t_n) e^{-a_n(t-t_n)} \sin(2\Pi f_n(t-t_n)) \qquad (4)$$

where b_n is a multiplier, 1(t) denotes Heaviside step, t_n is a delay, a_n describes vibration dumping and f_n is the vibration frequency. The set of functions, containing all possible impulse responses of the elementary systems (up to a constant multiplier) can be written as

$$\{v_{ijk} = c_{ijk}\, 1(t-t_i)\,(t-t_i)\, e^{-a_j(t-t_i)} \sin(2\pi f_k(t-t_i))\} \quad (5)$$

where c_{ijk} denotes the normalizing factor (i.e. $\|v_{ijk}\| = 1$). We will denote the set (5) as Θ.

Since, in practice, the gathered signals are discrete, finite-energy signals, the problem should be considered in the unitary space l_2. The set Θ will contain linearly-dependent elements, so only some members of Θ should be used in approximation of a given signal. Having stated the problem in this way, the next question to be answered is which elements of Θ should one employ in order to minimize the approximation error, associated with a given number of the elements. The problem has been solved recursively, making a choice step by step [1]. Thus, the algorithm of choice of a single element consist of two stages. v_{ijk} can be written as

$$v_{ijk}(t) = c_{ijk}\, E_{ij}(t) \sin(2\Pi f_k t+\varphi) = c_{ijk}\, E_{ij}(t) \sin(2\Pi f_k(t-t_f)) \quad (6)$$

where $E_{ij}(t)$ is the element envelope. In the first stage, one calculates the parameters f_k and φ or t_f, in the second - the envelope parameters. The element is determined if the three parameters: the frequency f_k, the exponent a_j and the delay t_i are known.

The complete algorithm of choice of the elements of Θ consists of the following steps:

a). Determination of the frequency f_k of the elementary vibrating system,

b). Determination of the phase-shift φ or, equivalently, of the quantity t_f and then - of the set of quantities $\{t_i\}$ being a set of possible delays of the approximating element. t_f is computed as the inner-product of the approximated signal and the two elements of basis : $z_1 = \sin(2\pi f_k t)$ and $z_2 = \cos(2\pi f_k t)$. Having determined t_f one determines the set $\{t_i\}$ via zero crossing of $\sin(2\pi f_k(t-t_f))$,

c). Determination of the envelope parameters of the impulse response, i.e. the quantities t_i and a_j. The algorithm of choice of the element results from the theorem stating that if the element $\underline{v}_{ijk}$ yields the smallest mean-square error then the inner-product $(y,\underline{v}_{ijk})$ is the biggest [3].

The steps a) - c) result in the optimum choice of the one element of the set Θ. It is usually necessary to employ several approximating elements. Further steps of the proposed algorithm are:

d). Determination of the multipliers α_i by the orthogonal projection of the signal on the subspace spanned by the chosen elements of the basis,

e). Subtraction of the projection from the approximated signal,

f). Repetition of the steps a) - e) assumed or dependent on the results number of times.

Following the algorithm, the chosen subset is a set of linearly-independent function [3]. Consequently, it can constitute a basis $\{w_i\}$ of a subspace of l_2. Thus, the estimators of the parameters $\{\alpha_n, f_n, a_n, t_n : n = 1,\ldots,N\}$ describing the system impulse response are computed. Now, it is possible to perform inverse filtering for input signal estimation.

3. INVERSE FILTERING OF THE SIGNAL

The system transfer function H(z) is a rational function

$$H(z) = \frac{A(z)}{B(z)} \quad (7)$$

where

$$A(z) = \gamma_1 z^{-t_1} A_1 \prod_{n=2}^{N} B_n + \ldots + \gamma_r z^{-t_r} A_r \prod_{\substack{n=1 \\ n\neq r}}^{N} B_n + \ldots + \gamma_N z^{-t_N} A_N \prod_{n=1}^{N-1} B_n \quad (8)$$

and

$$B(z) = \prod_{n=1}^{N} B_n(z) \quad (9)$$

Given H(z), it is possible to perform the inverse filtering, i.e.

$$x(t) \approx y(t) * {}^{-1}h(t) \quad (10)$$

where ${}^{-1}h(t)$ is given by

$$H(z)\ {}^{-1}H(z) = 1 \quad (11)$$

The transfer function H(z) of the system considered is, in general, a non-minimum phase. Consequently, the inverse filter may be unstable. To avoid this inconvenience, one can force compliance of H(z) to a minimum phase function and then apply a non-minimum phase, all-pass compensator [4].

In the presented method of deconvolution merely estimator $\tilde{H}(z)$ is known. Form of $\tilde{H}(z)$ is the same like H(z), besides that the exact value of parameter v is replaced by its estimator $\tilde{v}$. Consequently, the inverse filtering results in the estimator of the input, i.e.

$$\tilde{X}(z) = Y(z)\,\tilde{H}^{-1}(z) \approx X(z)\,H(z)\,\tilde{H}^{-1}(z) \qquad (12)$$

If $H(z)\tilde{H}^{-1}(z) = 1$ (which holds if the parameters are known exactly), the inverse filtering results in the exact input. Without any additional information, e.g. about the x(t) energy, a possibility of estimation the x(t) up to a constant multiplication is usually satisfactory. This means that the result is also satisfactory if $H(z)\tilde{H}^{-1}(z) = \text{const}$. The use of test-signals in the evaluation of the deconvolution quality shows that the parameters estimation errors, especially the multipliers α_n and the time delays t_n, essentially disturb the deconvolution results. The essential questions to be asked now are: How far estimation errors of the system parameters disturb the result of the inverse filtering, which should be the excitation? Is it possible to perform effectively deconvolution under such conditions?

4. OPTIMIZATION

It has been proved in [2] that if the estimation errors of multipliers γ_n and delays t_n are not equal to zero then the inverse filtering besides of the desired input results in some additional terms. Those terms are actually the echoes of the input with different amplitudes and delays. (see Figs. 1c and 2a). In order to minimize the estimation errors of the parameters describing the system, one can apply an optimization. Searching for a minimum of a function whose independent variables are the value of the coefficients α_n and the delays t_n, while the value of the cost function is a measure of the appearance and the amplitude of the input echoes in the output of the inverse filter, one will be faced with the situation when the estimators $\tilde{\alpha}_n$ and $\tilde{t}_n$ will be close to the multiplier values α_n and to t_n (up to a multiplicative constant and delay respectively), i.e. $\tilde{\alpha}_n = G\,\alpha_n$ and $\tilde{t}_n = t_n + t_0$, $n = 1,..,N$. Thus, performing optimization one can get

$$\tilde{x}(t) = G\,x(t-t_0) \qquad (13)$$

Results presented above as well as the simulations have shown that the cost function can be based on the inverse filter output. Several different cost functions and their effectiveness have been examined using test-signals. The best results have been obtained for the cost function defined on the base of the power cepstrum. In the optimization step Powel method has been applied. Its usefulness and effectiveness of the proposed blind deconvolution method have been confirmed by achieved results.

5. THE DECONVOLUTION RESULTS

Graphs presented in Fig. 1 show influence of the optimization on effectiveness of test-signal deconvolution. The method of blind deconvolution, presented in this paper, is implemented in copper mines to analysis of the observed seismic signals in order to estimate system impulse response and excitation. The signals are first sampled (sampling frequency - 100Hz), preprocessed (low-pass filtering and exponential windowing), and then - deconvolved. In Fig. 2 we show the results of the input estimation and corresponding power cepstrum.

6. CONCLUSION

For the class of systems and inputs specified in Introduction, the new method of blind deconvolution has been proposed. The method has been implemented in the deconvolution of seismic signals gathered in copper mines for which the use of predictive and homomorphic deconvolution

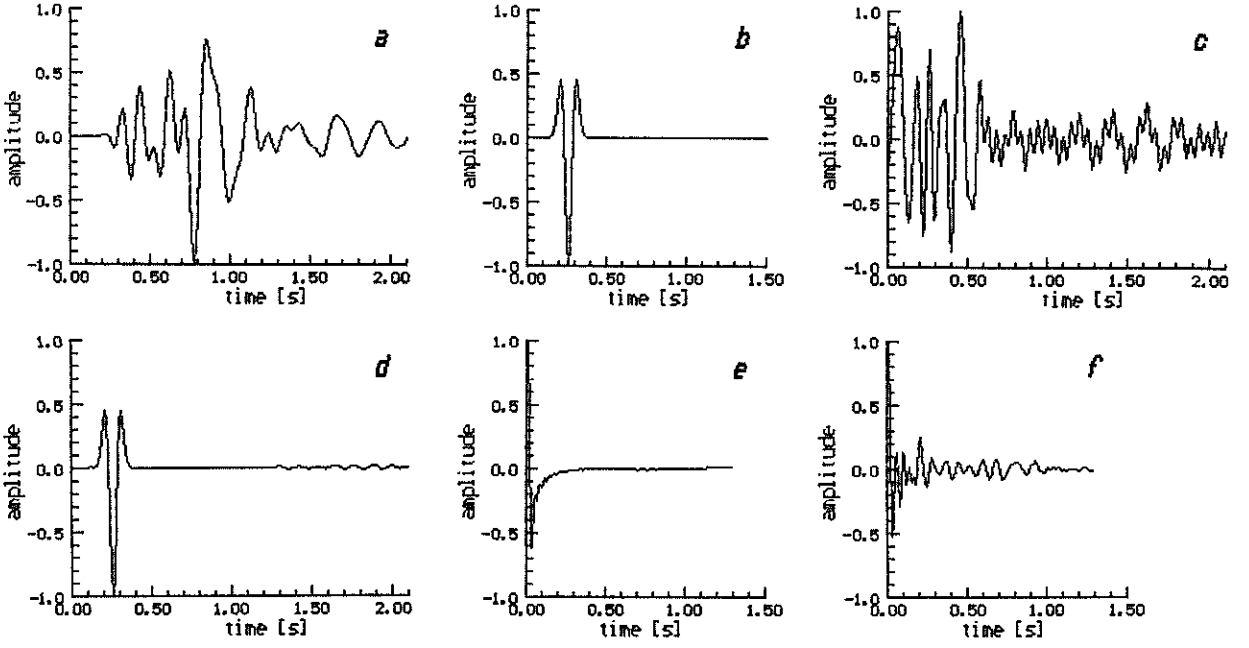


Fig. 1. The output test-signal of the system - a, input of the system - b, the inverse filter output without the optimization - c, the inverse filter output with optimization - d, power cepstrum of the signal of Fig. 1a -e, power cepstrum of the signal of Fig. 1b - f.

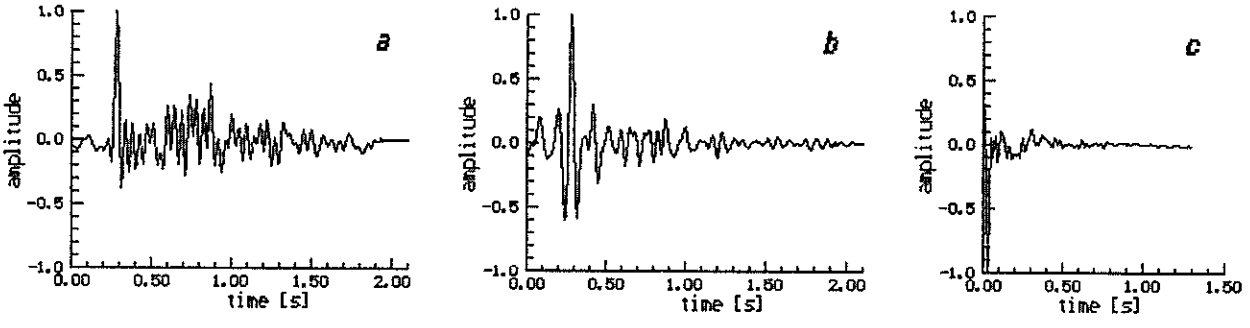


Fig. 2. Results of the inverse filtering of a seismic signal: the deconvolution result obtained without optimization - a, the deconvolution result with optimization - b, power cepstrum of the signal of Fig. 2b - c.

failed. The results of the seismic signals deconvolution are very prospective.

ACKNOWLEDGEMENTS

I wish to thank J.Gronowski and J.Zarzycki for their comments and assistance at various stages of this work.

REFERENCES

[1] Makowski R., Parameters Estimation by Signal Projection on Non-orthogonal Set, Proceedings Inter. AMSE Confer. Modeling and Simulation, Istanbul 1988, AMSE Press, Vol. 1C.

[2] Makowski R., Inverse Filtering in the Case of Some Incompleteness in System Description, Proceedings - XI National Conference on Circuit Theory and Electronic Circuits, Rytro, 1988.

[3] Makowski R., Fitting of Non-orthogonal Basis Matching a Given Signal, Reports Inst. Telecommunication and Acoustics, Technical University of Wrocław, 1990.

[4] Sengbush R.L., Hu S.T., Wiener-Levinson Deconvolution of Nonminimum Phase Seismic Data, Proceedings - Offshore Technology Conference; XVIII Annual Meeting, Houston, 1986.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

THEORETICAL COMPARISON OF TWO NOISE REDUCTION METHODS

Gérard FAUCON, Saïd TAZI MEZALEK

Laboratoire Traitement du Signal-IRISA, Université de Rennes I, Campus de Beaulieu, 35042 Rennes Cedex, France

This paper reports a study of two methods to estimate a signal when two observations signal + noise are available. Both methods assume that speech signals, as also disturbing noises, are strongly correlated. The aimed application is the enhancement of the noisy speech for radio-mobile applications. Both methods are studied theoretically with hypotheses close to those encountered in practice : the coherence between speech signals is equal to unity in module, and the coherence between noises is variable. This study, using optimal filters, allows us to evaluate the influence of the coherence function between noises.

1. INTRODUCTION

Our concern is the estimation of a signal when two observations signal + noise are available. Signals and noises are independent. The aim is the noise reduction on speech signals recorded in a car for radio-mobile applications. Wherever the microphones are placed in the vehicle, they always pick up some speech signal. So, every method based on adaptive noise cancelling with a reference noise available must be discarded. We write the observations :

$$x_1 = s_1 + b_1 \quad \text{and} \quad x_2 = s_2 + b_2$$

and we assume the signal to estimate is s_1. For our application, the module of the coherence between signals is close to 1, $\forall f$, and the coherence between noises depends principally upon the distance between the microphones. In this paper, the structures we elaborate are based upon a strong coherence between noises. To carry out our theoretical study, we consider a quasi-realistic situation : we compute a noise reduction factor in terms of a variable coherence function between noises and by assuming the coherence function between signals is equal to 1.

2. PRESENTATION OF BOTH METHODS

Both methods are dual and built on two stages [1,2]. The role of the first stage is to create, in the ideal case, either a signal reference or a noise reference as input of the second stage.

For the first one called S.I.N.C. (Signal Identification + Noise Cancelling) and given Figure 1, the transfer function between speech signals is learned by the filter F_1 in absence of noise, in the stopped car. This transfer function is assumed to be stationary in the moving car. In a second step, the noises are present, the filter F_1 is locked and at the output of the first stage, we obtain $r_{1,t} = s_{1,t} + b_{1,t} - F_1(s_{2,t} + b_{2,t})$ which reduces to $b_{1,t} - F_1(b_{2,t})$ if the transfer function between $s_{1,t}$ and $s_{2,t}$ is well identified. This previous quantity is used as the reference input of a second stage, whose primary input $p_{1,t}$ is either $x_{1,t}$, or $[x_{1,t} + F_1(s_{2,t} + b_{2,t})]/2$. The second stage acts a noise canceller and must estimate the disturbing noise present in $p_{1,t}$ from the reference $r_{1,t}$.

The second structure, given Figure 2, called N.I.D.R. (Noise Identification + Distortion Reduction) is the replica of the first one. We profit of the intermittence of the speech signal to learn the transfer function between noises by a filter F_1'. When signal arrives, F_1' is locked and its output becomes $r_{2,t} = s_{1,t} + b_{1,t} - F_1'(s_{2,t} + b_{2,t})$ which reduces to $s_{1,t} - F_1'(s_{2,t})$ if the transfer function between noises is well identified.

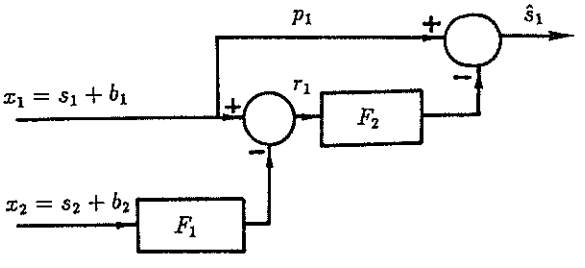


Figure 1 : S.I.N.C. Structure

$r_{2,t}$ is used as a signal reference for the second stage, whose primary channel is $p_{2,t} = x_{1,t} = s_{1,t} + b_{1,t}$. $r_{2,t}$ is filtered by F_2' to give, at its output, an estimate of the signal $s_{1,t}$.

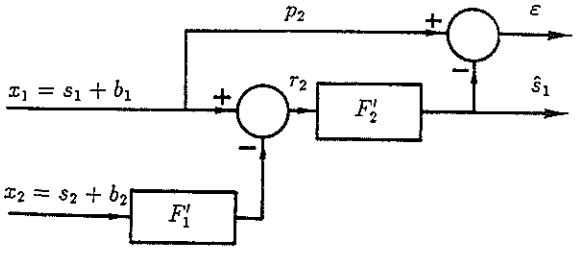


Figure 2 : N.I.D.R. Structure

3. PERFORMANCE ANALYSIS

Practical implementation of both methods employs digitally implemented discrete-time causal linear filters. The attainable performances are limited by the filtering lengths, digital word lengths, and the speed of convergence of the adjustement filters. However, useful bounds can be obtained by assuming non causal optimal filters with infinite memory, and stationary signal and noise statistics.

To make an analysis with hypotheses near to those encountered in our problem, we assume the coherence between signals is 1 in module, as for the coherence between noises, it's variable and may be written $\rho_{b_1 b_2} = \rho_b\, e^{j\theta_b}$, with $\rho_b \leqslant 1$. $|\rho_{s_1 s_2}| = 1$ means the signal s_1 and s_2 are derived from the same signal by linear filtering ; we write : $s_1 = s$, $s_2 = H\,s$, then $\rho_{s_1 s_2} = e^{-j\theta_s}$ where $\theta_s = \arg H$. For each method, we compute the power spectral density (p.s.d.) of the output noise b_{out}. We deduce a noise reduction factor, defined for each frequency by :

$$R(f) = \frac{\gamma_{b_1}(f) - \gamma_{b_{out}}(f)}{\gamma_{b_1}(f)} \qquad (1)$$

When $\gamma_{b_{out}} = 0$, $R = 1$, the noise reduction is complete and when $\gamma_{b_{out}} = \gamma_{b_1}$, $R = 0$, there is no noise reduction. If R becomes negative, the output noise b_{out} is greater than b_1.

a) S.I.N.C. Structure

In a first step, the signals are coming alone : $x_1 = s$, $x_2 = H\,s$ and thus $F_1 = 1/H$. In the second step, the noises are present and the reference channel of the second stage is $r_{1,t} = b_{1,t} - F_1(b_{2,t})$. This noise reference is ideal since $|\rho_{s_1 s_2}| = 1$. Let the primary channel of the second stage be : $p_{1,t} = s_{1,t} + b_{1,t}$. Then we have :

$$\gamma_{b_{out}} = \gamma_{b_1}\left(1 - |\rho_{b_1 r_1}|^2\right) \qquad (2)$$

We have to compute $\rho_{b_1 r_1}$ in terms of $\rho_{b_1 b_2}$.

$$\rho_{b_1 r_1} = \frac{\gamma_{b_1 r_1}}{\gamma_{b_1}^{1/2}\,\gamma_{r_1}^{1/2}}$$

with $\gamma_{b_1 r_1} = \gamma_{b_1} - F_1^*\,\gamma_{b_1 b_2}$

$$\gamma_{r_1} = \gamma_{b_1} + |F_1|^2\,\gamma_{b_2} - F_1^*\,\gamma_{b_1 b_2} - F_1\,\gamma_{b_1 b_2}^* \qquad (3)$$

After some transformations and with $\alpha_b = \sqrt{\gamma_{b_1}/\gamma_{b_2}}$ we obtain finally :

$$\gamma_{b_{out}} = \gamma_{b_1}\,\frac{|H^{-1}|^2\,[1 - |\rho_{b_1 b_2}|^2]}{|H^{-1} - \alpha_b \rho_{b_1 b_2}|^2 + [1 - |\rho_{b_1 b_2}|^2]\,\alpha_b^2} \qquad (4)$$

Consider now the second possibility, by choosing as primary channel for the second stage, the half-sum of the observation x_1 and of the ouput of the filter F_1 : $p_{1,t} = s_{1,t} + b'_{1,t}$ with $b'_{1,t} = \frac{1}{2}\,[b_{1,t} + F_1(b_{2,t})]$.

The p.s.d. of the output noise is given by :

$$\gamma_{b_{out}} = \gamma_{b'_1}\,\left[1 - |\rho_{b'_1\, r_1}|^2\right] .$$

Express $\rho_{b'_1 r_1}$: $\rho_{b'_1 r_1} = \dfrac{\gamma_{b'_1 r_1}}{\gamma_{b'_1}^{1/2}\,\gamma_{r_1}^{1/2}}$

with

$$\gamma_{b'_1 r_1} = \frac{1}{2}\left[\gamma_{b_1} - |F_1|^2\gamma_{b_2} - F_1^*\gamma_{b_1 b_2} + F_1\gamma_{b_1 b_2}^*\right]$$

$$\gamma_{b'_1} = \frac{1}{4}\left[\gamma_{b_1} + |F_1|^2\gamma_{b_2} + F_1^*\gamma_{b_1 b_2} + F_1\gamma_{b_1 b_2}^*\right] .$$

γ_{r_1} is given by (3). After some transformations, we find that $\gamma_{b_{out}}$ is given by the same expression as previously (4). We conclude that results are identical whatever the chosen primary channel of the second stage is, when the filters used are optimum. Now, we can compute R for this S.I.N.C. structure. From (1) and (2), we can write $R = |\rho_{b_1 r_1}|^2$. We obtain finally :

$$R = \frac{|\alpha_b - F_1^*\,\rho_{b_1 b_2}|^2}{|F_1 - \alpha_b\,\rho_{b_1 b_2}|^2 + \alpha_b^2\,(1 - |\rho_{b_1 b_2}|^2)} \qquad (5)$$

with $F_1 = 1/H$.

b) N.I.D.R. Structure

The filter F_1' of the first stage is learned in presence of noises alone, $F_1' = \gamma_{b_1b_2}/\gamma_{b_2}$. Let $\varepsilon_{b,t}$ be the output noise $b_{1,t} - F_1'(b_2,t)$. Its p.s.d. is $\gamma_{\varepsilon_b} = \gamma_{b_1}(1 - |\rho_{b_1b_2}|^2)$. The filter F_2' of the second stage is updated in presence of signal and noise. The primary channel and the reference channel of this stage are respectively :

$$p_{2,t} = s_{1,t} + b_{1,t}$$

$$r_{2,t} = s_{1,t} - F_1'(s_{2,t}) - \varepsilon_{b,t}$$

The filter F_2' is given by $\gamma_{p_2r_2}/\gamma_{r_2}$. The output noise added to the estimated signal will be $|F_2'|^2 \gamma_{\varepsilon_b}$. Let us give the expressions necessary to the computation of F_2' :

$$\gamma_{p_2r_2} = \gamma_{s_1} - F_1'^* \gamma_{s_1s_2} + \gamma_{b_1\varepsilon_b}$$

$$\gamma_{r_2} = \gamma_{\varepsilon_b} + \gamma_{s_1}(1-|\rho_{s_1s_2}|^2) + \gamma_{s_2}\left|F_1' - \frac{\gamma_{s_1s_2}}{\gamma_{s_2}}\right|^2$$

with $\gamma_{\varepsilon_b} = \gamma_{b_1} - F_1'^* \gamma_{b_1b_2}$

Let $\alpha_s = \sqrt{\gamma_{s_1}/\gamma_{s_2}}$, $\delta_1 = \gamma_{s_1}/\gamma_{b_1}$ (signal-to-noise ratio relative to x_1). We obtain :

$$F_2' = \frac{1 + \frac{1}{\delta_1}\left(1 - |\rho_{b_1b_2}|^2\right) - \frac{\alpha_b}{\alpha_s}\delta_1^*}{\frac{1}{\delta_1}\left(1 - |\rho_{b_1b_2}|^2\right) + \left|1 - \frac{\alpha_b}{\alpha_s}\rho_{b_1b_2}\right|^2} \quad (6)$$

R will be computed so :

$$R = 1 - |F_2'|^2\left(1 - |\rho_{b_1b_2}|^2\right) . \quad (7)$$

4. EVALUATION OF PERFORMANCES

We wish to compare both structures, with regard to $\rho_{b_1b_2}$. Other parameters occur : H, α_s, α_b, δ_1 . H is the transfer function relating the signals s_1 and s_2, we have chosen H = 1 ; that is equivalent to restore the signals identical, (it's theoretically possible since $|\rho_{s_1s_2}| = 1$) and to have two noises b_1 and $H^{-1}(b_2)$ additive on each channel to the same signal s. The figures 3 to 6 show the factor R as a function of $\rho_{b_1b_2}$ chosen real, for various sets of parameters α_s, α_b, δ_1. The situations are given in the legend of each figure. For some situations, the N.I.D.R. structure is the more performant. In the other hand, for both structures, R doesn't vary monotically with the coherence $\rho_{b_1b_2}$. Nevertheless, these results are obtained for only one frequency.

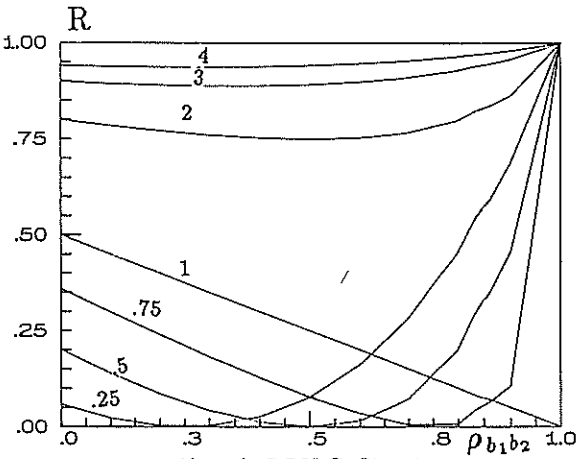


Figure 3 : $R = f(\rho_{b_1b_2})$, S.I.N.C. Structure
α_b parameter

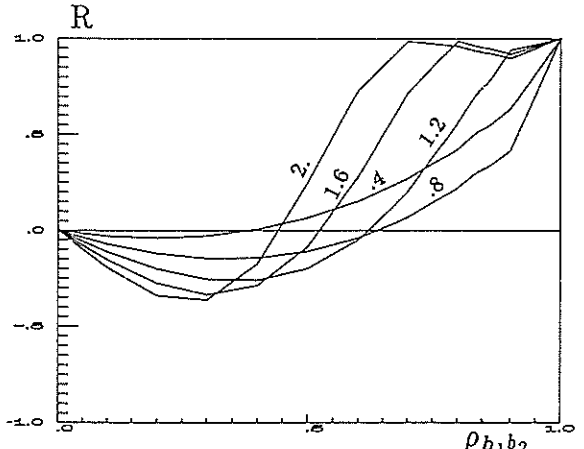


Figure 4 : $R = f(\rho_{b_1b_2})$, N.I.D.R. Structure
$\delta_1 = 0\ dB$; α_b parameter

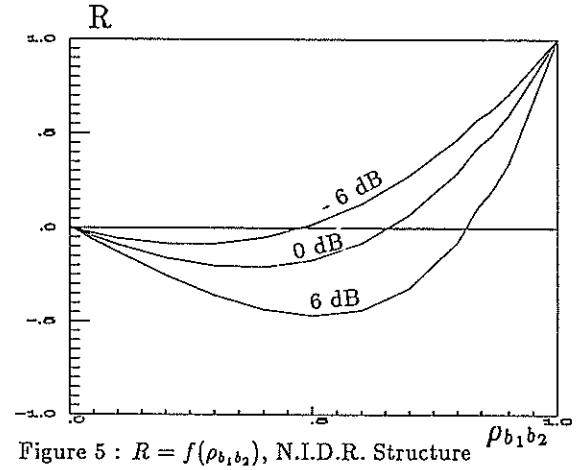


Figure 5 : $R = f(\rho_{b_1b_2})$, N.I.D.R. Structure
$\alpha_b = 1$; δ_1 parameter

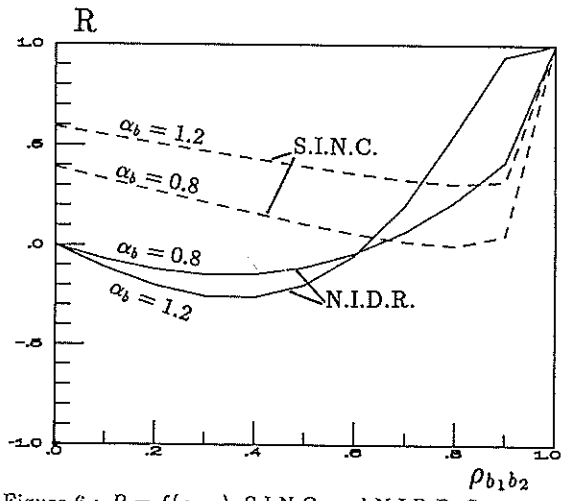


Figure 6 : $R = f(\rho_{b_1 b_2})$, S.I.N.C. and N.I.D.R. Structures

Therefore,it's interesting to obtain theoretical results on the power of the output noise $E\left[b_{out}^2\right]$ and to compare them against experimental results obtained with adaptive filters.

The models of the signals and noises are :

$$s_t = e_t + 1.4\ s_{t-1} - 0.9\ s_{t-2}$$

$$n_{1,t} = e'_t - \alpha_1\ n_{1,t-1} \qquad (8\text{-a})$$

$$b_{1,t} = n_{1,t} + \beta_1\ m_t \qquad (8\text{-b})$$

$$n_{2,t} = e'_t - \alpha_2\ n_{2,t-1} \qquad (8\text{-c})$$

$$b_{2,t} = n_{2,t} + \beta_2\ m'_t \qquad (8\text{-d})$$

where e_t, e'_t, m_t, m'_t are white gaussian noises ; β_1 and β_2 allow to modify the coherence between noises b_1 and b_2.

The theoretical value of $E\left[b_{out}^2\right]$ is obtained in the following manner : for each frequency, we compute $\rho_{b_1 b_2}$ and other parameters α_s, α_b, δ_1 from models given in (8), then we deduce R, $\gamma_{b_{out}}$ and by integration over all frequencies, we compute $\varepsilon = E\left[b_{out}^2\right]$. The simulation results are obtained by using adaptive filters with a forgetting factor weak and by respecting the processing in two steps, for each structure. The adaptive filters are least-squares lattice structures. To evaluate the error ε of the N.I.D.R. structure, we must send the noises alone, after convergence of both filters F'_1 and F'_2. The results are presented in table 1. The forgetting factor of the lattice structures is $\lambda = 0.9999$, the length of the filtering is $N = 5$ and the primary channel of the second stage is, in each structure, delayed of a sample to obtain the optimum filter of the second stage causal.

		$\beta_1=\beta_2=0.1$	$\beta_1=\beta_2=0.15$	$\beta_1=\beta_2=0.2$
S.I.N.C.	Theo.	0.091	0.163	0.233
	Exp.	0.103	0.195	0.292
N.I.D.R.	Theo.	0.115	0.268	0.5
	Exp.	0.126	0.308	0.604

Table 1 : Value of $\varepsilon = E\left[b_{out}^2\right]$
$\alpha_1 = 0.4$; $\alpha_2 = 0.9$.

For the studied cases, when β_1 or β_2 increases, the coherence between noises reduces and the error ε is growing. The S.I.N.C. structure is more performant than the N.I.D.R. structure. The obtained results are justifiable.

5. CONCLUSION

For the problem of the estimation of a signal when two observations signal + noise are available, two structures are proposed. These assume that signals, as also disturbing noises, are well correlated. In our application of noise cancelling for radio-mobile applications, the coherence between signals is close to 1 while the coherence between noises is variable according to the distance between microphones. A study is made on each structure to evaluate a noise reduction factor, when filters used are optimal. Experiments on simulated signals are carried out to confirm theoretical results.

Acknowledgements : The authors wish to thank the C.N.E.T. LANNION A (France) for their support in this work.

REFERENCES

[1] Faucon, G., Tazi Mezalek, S., Le Bouquin, R., Study and Comparison of three structures for enhancement of noisy speech, ICASSP 89, Glasgow.

[2] Gardner, W.A., Agee, B.G., Two-Stage Adaptive noise Cancellation for Intermittent-Signal Applications, IEEE on IT, vol. 26, n° 6, Nov. 1980.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

ROBUST PREDICTIVE DECONVOLUTION USING MEDIAN TYPE FILTERS

Lin Yin, Jaakko Astola, and Yrjö Neuvo

Department of Electrical Engineering
Tampere University of Technology
P. O. Box 527, SF–33101 Tampere, Finland

Abstract-In this paper, we propose a new robust predictive deconvolution algorithm which is based on lattice structure. Under weighted l_1 norm error criterion, the reflection coefficients of the lattice filter are estimated by choosing median from reflection coefficient sequences defined by the forward and backward error residuals. Computer simulation results demonstrate that the new method is much more robust than the methods based on the l_2 and l_1 norm criterions.

I. Introduction

Deconvolution is a problem of fundamental importance in many signal processing applications such as, seismic, speech and mechanical engineering. One of the deconvolution techniques is the simple predictive one which is traditionally based on the least squares error criterion (l_2 norm criterion) [1]. It is well known that, if measured data is contaminated by aberrant noise, deconvolution techniques based on l_2 norm criterion give poor performance. Robust statistics as discussed by Tukey [2] and Huber [3] has the desirable goal of finding ways to process the data which are essentially insensitive to aberrant noise.

In predictive deconvolution, insensitivity to aberrant data values can be achieved by designing filters which minimize a measure of the prediction errors, different from the traditional least squares or l_2 norm. It has been shown that in some applications the robustness of the least absolute error criterion or l_1 norm criterion can be particularly well adopted [4,8,9]. The robustness of the least absolute error criterion can be explained as follows: the least squares criterion heavily weights large errors while the l_1 norm criterion ensures an equal weight on the predictable and unpredictable points of the signals, which makes the l_1 solution tend to ignore outliers. On the other hand, deemphasizing larger errors results in l_1 norm residuals which contain more zeros than the l_2 norm residuals. In the context of deconvolution, the l_1 norm residuals tend to have the connotation of being a " sparse spike train ". For seismic applications, this is a desirable quality.

Because the l_1 norm filter has so many good properties, several l_1 norm algorithms are proposed, including linear programming l_1 algorithm [4,5,6,7] and lattice structure based algorithm [9,10]. But most of l_1 norm algorithms can not guarantee the stability of the inverse filter. On the other hand, the performance of the l_1 norm algorithms is indeed deteriorated by the aberrant noise. However, the degradation is not so serious as in the case of the l_2 norm filter. In this paper we propose a new robust predictive deconvolution algorithm which is based on Burg's lattice structure but reflection coefficients (RC's) are estimated by using median filter. The new algorithm is derived under weighted l_1 norm error criterion. The weights emphasize small errors and deemphasize the aberrant noise, which is appropriate to the problem of deconvolution. Indeed, in deconvolution, each impulse corresponding to a large prediction error should be preserved. Two kinds of robust RC estimation algorithms are discussed in this paper according to the different RC sequences, one of which can guarantee the stability of the inverse filter.

II. Lattice Formulations

Predictive deconvolution assumes that the discrete signal $s(n)$ can be approximated by a linear combination of p past values:

$$s(n) = \sum_{i=1}^{p} a_i s(n-i) + e(n) \qquad (1)$$

where $e(n)$ is the prediction error (i.e., deconvolution result). In other words, sequence $s(n)$ can be expressed as the output of a linear filter driven by sequence $e(n)$. The transfer function of this filter is defined by:

$$H(z) = \frac{1}{A(z)} \qquad (2)$$

where

$$A(z) = \sum_{k=0}^{p} a_k z^{-k}, \quad a_0 = 1 \tag{3}$$

is known as the inverse filter. If $H(z)$ is stable and the inverse filter, $A(z)$, is driven by the signal $s(n)$, its output will be the prediction error $e(n)$. $A(z)$ can be implemented as a lattice filter. Equations of the lattice filter are

$$f_0(n) = b_0(n) = s(n) \tag{4}$$

$$f_i(n) = f_{i-1}(n) + K_i b_{i-1}(n-1), \quad 1 \le i \le p \tag{5}$$

$$b_i(n) = K_i f_{i-1}(n) + b_{i-1}(n-1), \quad 1 \le i \le p \tag{6}$$

$$e(n) = f_p(n) \tag{7}$$

where $f_i(n)$ and $b_i(n)$ are the forward and backward residuals, respectively, K_i is the reflection coefficient. The K_i's in the lattice filter are uniquely related to the predictor coefficients. For a stable $H(z)$, one must have

$$|K_i| < 1, \quad 1 \le i \le p \tag{8}$$

In the lattice formulation, the RC's can be computed by minimizing some norm of the forward residual $f_i(n)$, or the backward residual $b_i(n)$, or a combination of the two. Several methods of estimating the RC's were summaried by Makhoul [11]. In practice, we tend to prefer the use of Burg algorithm because it minimizes a reasonable and well-defined error criterion. In Burg algorithm, the RC of the *ith* stage K_i is obtained by minimizing the sum of the square values of the forward and backward prediction errors at the output of the stage

$$K_i^B = -\frac{2\sum_{n=i+1}^{N} f_{i-1}(n) b_{i-1}(n-1)}{\sum_{n=i+1}^{N} b_{i-1}^2(n-1) + f_{i-1}^2(n)} \tag{9}$$

III. New Robust Deconvolution Methods

According to the discussion above, the performance of the lattice filter depends entirely on the estimation of the RC's. The robust estimation of the RC's can guarantee the robustness of deconvolution. In this section we will derive a new class of robust estimation algorithms of RC's under weighted l_1 norm criterion.

A. Weighted l_1 Norm Criterion

It is quite easy to prove that under the following minimization criterion:

$$\min_{K_i} \sum_{n=i+1}^{N} \frac{1}{|b_{i-1}(n-1)|}|f_i(n)| + \frac{1}{|f_{i-1}(n)|}|b_i(n)| \tag{10}$$

we can derive the estimate of RC

$$K_i^B = MED[k_i^f(i+1), ..., k_i^f(N), k_i^b(i+1), ..., k_i^b(N)] \tag{11}$$

where $MED[.]$ indicates median operation and

$$k_i^f(n) = -\frac{f_{i-1}(n)}{b_{i-1}(n-1)}, \quad i+1 \le n \le N \tag{12}$$

$$k_i^b(n) = -\frac{b_{i-1}(n-1)}{f_{i-1}(n)}, \quad i+1 \le n \le N \tag{13}$$

are defined as the forward and backward reflection coefficient sequence.

The proof is straightforward. Replacing $f_i(n)$ and $b_i(n)$ with Eq.(5) and (6), we have

$$\min_{K_i} \sum_{n=i+1}^{N} |K_i + \frac{f_{i-1}(n)}{b_{i-1}(n-1)}| + |K_i + \frac{b_{i-1}(n-1)}{f_{i-1}(n)}| \tag{14}$$

It is easy to see that the optimal solution of K_i is Eq.(11).

The heuristic explanation about robustness of the median operation is given as follows. In l_2 norm criterion, heavily weighting the larger prediction errors (or residuals) will make the final estimates of RC's better if there is no observation error. This is because the larger residuals are signal components and are not whiten at the corresponding stage. But when the outliers occur, the weights are mainly emphasizing the noise, which makes l_2 norm criterion quite sensitive to aberrant noise. Although l_1 norm criterion ensures an equal weight on all residuals, large errors still dominate the forward and backward residuals when outliers exist. Further deemphasizing the larger error in the weighted l_1 norm criterion will certainly alleviate the effect of outliers.

The RC sequences have many interesting properties which can be used to analyze the performance of the lattice filter. According to the different distribution of RC sequences, the l_2 and l_1 norm methods can be rederived [12] and some useful conclusions were obtained. When the exciting noise is distributed as Gaussian, the optimal estimators of RC's are Eq.(9). But if the exciting noise is Laplacian, then the optimal estimations of RC's can only be obtained under l_1 norm criterion [9]. This is the reason why l_1 norm method is particularly suitable for predictive deconvolution [4].

Because every element of the RC sequences is not guaranteed to be less than one, RC estimates from Eq.(11) can not guarantee the stability of the lattice filter.

B. New Stable RC Sequence

In order to overcome the problem of unstability of lattice filters resulting from the above RC sequences, another RC sequence is defined, which also has the same performance as the above but the stability of lattice filter is guaranteed naturally since every element of RC sequence is less than one in magnitude. The new RC sequence has the following expression:

$$k_i^B(n) = -\frac{2f_{i-1}(n)b_{i-1}(n-1)}{b_{i-1}^2(n-1) + f_{i-1}^2(n)}, \quad i+1 \le n \le N \tag{15}$$

It can be proven that:

If $k_i^B(n)$ of this sequence obeys Gaussian distribution $N(K_i^B, \frac{1}{f_{i-1}^2(n)+b_{i-1}^2(n-1)})$, the ML estimation of K_i^B is the same as the Burg's solution Eq.(9).

However, when observations are contaminated by outliers, the following equation gives better results.

$$K_i^B = MED[k_i^B(i+1), ..., k_i^B(N)] \qquad (16)$$

This solution can give even better results than Eq.(11) although it cannot be derived directly from minimizing some error criterion. Because all the elements of the above RC sequence are less than or equal to one, the stability of the filter is guaranteed.

So we have gotten two kinds of robust predictive deconvolution algorithms which have stronger resistance to outliers than l_2 and l_1 norm algorithm. But when the observations are contaminated by Gaussian noise not by outliers, the performance of the new algorithms is not better than l_2 and l_1 norm results.

IV. Simulation Results

This section will demonstrate some performance of l_2, l_1 norm and the new robust deconvolution methods. In particular, the similarity of these three deconvolution methods when applied to good data (i.e., no outliers) will be illustrated. Also, the insensitivity of the new methods fit to occasional wild data values will be examined.

Fig.1(a) is a plot of a minimum wavelet. Fig.1(b) contains a random spike sequence. The convolution of this wavelet with random spike train produces the time series which is displayed in Fig.1(c). This trace will be used to represent the good data situation. Fig.1(d) shows the same convolution as Fig.1(c) augmented by three outliers.

Fig.2(a) to Fig.2(d) show the results obtained by l_2, l_1 norm and new robust methods when the observation is from Fig.1(c). The order of all the predictive filters is 3. Recall that the goal of the predictive deconvolution is to remove the effects of the wavelet and recover the convolved spike series. The filter which has been calculated here is not long enough to completely undo the convolution process, but it has managed to make most of the original spikes visible. As might have been expected, when the data is very regular, l_1 norm algorithm gives the best result.

In order to demonstrate the robustness of the new methods, we use the observation in Fig.1(d). Fig.3(a) to Fig.3(d) show the results obtained by the above four kinds of algorithms. To see the results clearly, we cut all the figures to values ± 0.5. By contrast, when there are aberrant values in time series, there are remarkable difference in the filters which are calculated via four techniques. In these four methods, l_2 norm algorithm is most sensitive to the outliers, and new robust algorithms are most insensitive. The sensitivity of l_1 norm method to outliers lays between these two.

V. Conclusions

Based on lattice structure, a new robust predictive deconvolution algorithm has been proposed. The lattice RC's are estimated by choosing median from the RC sequences defined by the forward and backward residuals. If the observation is "good" (no outliers), l_2, l_1 norm and the new methods all give good performance. But when observation data is contaminated by outliers, a good estimate of RC is only obtained under weighted l_1 norm criterion.

References

[1] E. A. Robinson, " Statistical pulse compression , "*Proc. of the IEEE*, vol.72, no.10, pp.1276-1289, Oct. 1984.

[2] J. W. Tukey, " Nonlinear (nonsuperposable) methods for smoothing data," *in Conf. Rec., 1974 EASCON*, p.673.

[3] D. J. Huber, *Robust Statistics*, New York: Wiley, 1981.

[4] R. Yarlagadda, J. B. Bednar and T. L. Watt, " Fast algorithms of l_p deconvolution, " *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol.ASSP-33, no.1, pp.174-182, Feb. 1985.

[5] R. D. Armstrong, E. Frome, and D. S. Kung, " A revised simplex algorithm for the absolute deviation curve fitting problem," *Commun. Statist. - Simulat. Comput.*, vol.B8, no.2, pp.175-190, 1979.

[6] I. Barrodale and F. D. K. Roberts, "An improved algorithm for discrete l_1 approximation," *SIAM J. Numer. Anal.*, vol.10 no.5, 1973.

[7] - , " Algorithm 478: solution of an overdetermined system of equations in l_1 norm," *Commun. ACM*, vol.17, no.6, 1974.

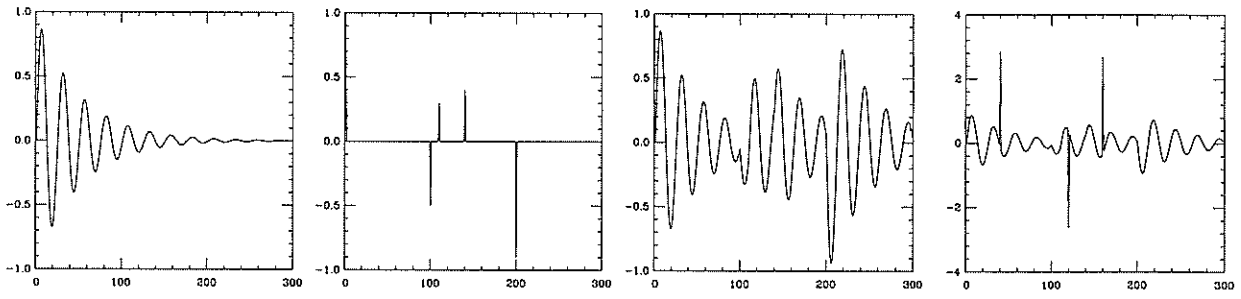
[8] J. F. Claerbout and F. Muir, " Robust modeling with erratic data, " *Geophysics*, vol.38, no.5, pp.826-844, 1973

[9] E. Denoel and J-P. Solvay, " Linear prediction of speech with a least absolute error criterion," *IEEE Trans. Acoust.,Speech,Signal Processing*, vol.ASSP -33, pp.1397-1403, Dec. 1985

[10] J. Alcazar-Fernandez and F.J. Fraile-Pelaez, " A lattice algorithm for l_p deconvolution," *Proc. ICASSP - 88*, pp.1746-1749, 1988.

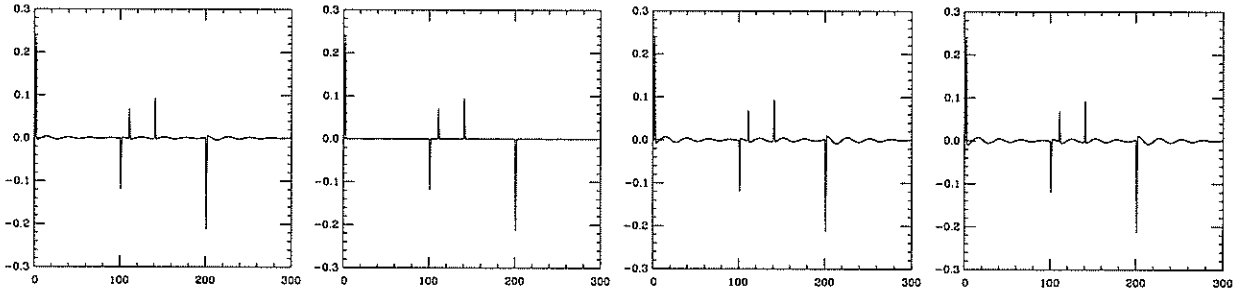
[11] J. Makhoul, " A class of all-zero lattice digital filters: properties and applications," *IEEE Trans. Acoustics, Speech, Signal Processing*, vol. ASSP-26, pp. 304 - 314, Aug. 1978

[12] L. Yin, J. Astola, and Y. Neuvo,"Application of median operation in robust estimation of AR model," *Proc. ICASSP - 90*, 41.E5.15, 1990.



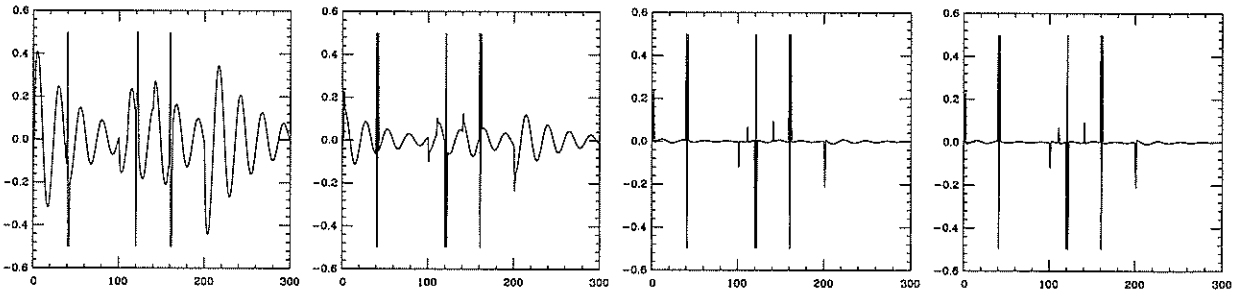
(a) (b) (c) (d)

Fig. 1. *(a) A minimum delay wavelet. (b) A random spike sequence. (c) The convolution of wavelet (a) with spike train (b). (d) The same convolution as (c) augmented by three outliers.*



(a) (b) (c) (d)

Fig. 2. *Deconvolution results when the observation is "good" (without outliers). (a) The ordinary Burg deconvolution. (b) L_1 norm deconvolution. (c) New method 1 (Eq.(11)) deconvolution. (d) New method 2 (Eq.(16)) deconvolution.*



(a) (b) (c) (d)

Fig. 3. *Deconvolution results when observation contaminated by outlier noise. Since the outliers have a very large amplititude, the following figures are cut to values ± 0.5 in order to see the desired signal properly. (a) The ordinary Burg deconvolution. (b) L_1 norm deconvolution. (c) New method 1 (Eq.(11)) deconvolution. (d) New method 2 (Eq.(16)) deconvolution.*

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

COMPARISON OF LMS AND STABILIZED FTF ALGORITHMS FOR MODEM ECHO CANCELLATION

R. Atay, Ph. Artaud, P. Baylou, B. Joseph and M. Najim
ENSERB - 351, cours de la Liberation, 33405, Talence, France.
D. Aboutajdine, LEESA - Faculté des Sciences. BP 1014, Rabat. Morocco.

The echo is a part of the transmitted signal which is added to the received one when using two-wire cable or satellite transmission. The problem is generaly solved using an adaptive transversal filter. The adaptation of the filter parameters is performed through the well known Least Mean Squares (LMS) algorithm. This paper deals with the use of the stabilized versions of the fast transversal filter (FTF) presented by Slock and Kailath (1988) as we have used for modem echo cancellation. We propose to evaluate and compare these algorithms taking into account perturbations encoutered in a real situation. To this end the evaluation will be performed by simulating a V26ter modem on a fixed point DSP.

1. INTRODUCTION

When designing modems for full duplex data transmission with superposed transmetter and receiver spectra , an adaptive echo canceller is necessary to eliminate the echo signal due to the imperfection of the differential line transformer and impedance mismatching. Depending on the chosen communication channel, the echo can be 50 to 130 milliseconds long. The number of coefficients used to estimate the echo impulse response is always greater than 100.
Because of its simplicity the Least Mean Squares (LMS) algorithm [3],[11] is very popular and widely used to identify echo impulse response. This class of algorithms presents some drawbacks however a slow convergence rate. The Recursive Least Squares algorithm (RLS) offers the alternative, but the complexity of its classic version is such that it does not easily allow an implementation on a digital signal processor (DSP).
D.T.M.Slock & T.Kailath [2] and A.Gilloire & A.Benallal [9] have developped numerically stable versions of fast RLS : Fast Transversal Filters (FTF) [5]. They analyse the time error propagation and show that the FTF (7N) is exponentially instable. They suggest computing some variables in two methods and use the difference between the two computation methods as an error to drive the algorithm numerical behaviour. The algorithm obtained is stable and keep the number of operations proportional to N(8N or 9N). In this paper we propose to adapt the obtained FTF algorithm to estimate the impulse response of the echo signal.
To this end the evaluation of the stabilized FTF performances will be performed by simulating a V26ter modem [1].
In the second part, we will present the echo canceller in bilateral transmission. In the third one, we present the derivation of RLS and fast RLS, the error analysis and modelling method, summarize the stabilization technique and finally give the resulting stabilized algorithm(9N). The fourth part is dedicated to the presentation of experimental results .

2. ECHO CANCELLATION

For a full duplex transmission goal, one has to use four wires telecommunication line, time multiplexing or frequency multiplexing. Later another way was developed: transmitting in the same band but using an echo canceller to separate the signal received from the echo of the transmitted one. This technique allows the data flow to be doubled while maintaining the same performances.
The echo signal can be attenuated using the adaptive echo cancelling technique [6] . It is based on the hypothesis stating that the echo signal is linearly related to the transmitted one. This is equivalent to designing a filter W with an impulse response. Applying the transmitted signal as an excitation to the filter we will obtain at the output an estimation of the echo signal which can be substracted from the signal received (fig1). This task is accomplished in the initialization period where the modem is simply transmitting. It enables the echo impulse response to be estimated. However since the echo is subject to time variation this impulse response is updated along the transmission period.

3. ADAPTATION OF THE FTF ALGORITHM TO ECHO CANCELLATION

3.1. Adaptive filtering and Recursive Least Squares approach.

Let d(T) and y(T) be two correlated signals. Our objective is to build a linear estimation $\hat{d}(T)$ of d(T) from the samples of y(T) such that (fig 2):

$$\hat{d}(T) = W^{*}(T/N).Y(T/N)$$

Where : $Y^{*}(T/N) = [y(T),y(T-1), ...,y(T-N+1)]$

* is the transpose operator and W(T/N) the parameter vector used to estimate $\hat{d}(T)$. The best estimation $\hat{d}(T)$ of d(T) is a solution of the minimisation problem according to the criterion :

$$\xi(T/N) = \sum_{t=0}^{T} \lambda^{T-t} [d(T)-W^{*}(T/N).Y(T/N)]^{2}$$ (λ is a forgetting factor)

Godard[8] shows that W(T/N) can be optimally estimated as :

$$W(T/N) = \Phi^{-1}(T/N).p(T/N)$$

where :

$$\Phi(T/N) = \sum_{t=0}^{T} \lambda^{T-t}.Y(t/N).Y^{*}(t/N)$$

is the autocorrelation matrix and

$$p(T/N) = \sum_{t=0}^{T} \lambda^{T-t} .d(t).Y(t/N)$$ is the intercorrelation vector

The computing complexity is proportional to N^2 which means that this filter is not feasible for real time application.

Godard has proposed a recursive relation to compute :

$$W(T/N) = W(T/N-1) + \varepsilon^{p}(T/N).\Phi^{-1}(T/N).Y(T/N)$$

where $\varepsilon^{p}(T/N) = d(T) - W^*(T-1/N).Y(T/N)$ is the a priori estimation error at time T .

Morf, Ljung and Falconer [6] developed an algorithm called "Fast Kalman" to update the gain defined by the relation: $c(T/N) = \Phi^{-1}(T/N).Y(T/N)$

For this algorithm, the number of operations is 10N.

Later J.M. Cioffi and T. Kailath [5] proposed a 7N version. These two algorithms use forward and backward prediction filters which we denote a(T/N) and b(T/N). They are excited by the signal y(T) and give as an output the forward e(T/N) and backward r(T/N) errors . The powers of this errors are respectively $\alpha(T/N)$ and $\beta(T/N)$. with :

$$e(T/N) = y(T) - a^*(T/N).Y(T-1/N)$$
$$r(T/N) = y(T-N) - b^*(T/N).Y(T-1/N)$$

3.2. Stabilization of the FTF algorithm:

Many authors [2],[9] and [10] show that the precedent algorithms are numerically instable and have proposed a stabilized version. They consider the error propagation phenomena as a non-linear dynamic system described by the following state space equation:

$$\Theta(T) = f[\Theta(T-1),d(T),Y(T/N)] \quad (2)$$

$\Theta(T)$ is a vector whose elements are the recursively computed variables.

They observe[2] that the implementation of this algorithm using a DSP decreases significantly the precision of the recursively computed variables. The algorithm will then propagate approximated variables represented in the vector $\Theta(T)$ whose behaviour is described by the modified equation :

$\hat{\Theta}(T) = f[\hat{\Theta}(T-1),d(T),Y(T/N)] + V(T)$ where:

V(T) is the equivalent noise to the roundoff errors.

Let $\Delta\Theta(T) = \hat{\Theta}(T) - \Theta(T)$:

A linearization of the equation (2) near the real trajectory of $\Theta(T)$ gives the system (3) :

$\Delta\Theta(T) = \Delta\Theta(T-1) . F(T) + V(T)$

$F(T) = \nabla_{\Theta(T)} f[\Theta,d(T),Y(T/N)]$ (3)

The convergence is directly related to the stability of the system (3); therefore stabilizing this system is equivalent to ensuring algorithm convergence. This is equivalent to studying the stability of the free excitation system (4). This can be if F(T) has all its eigenvalues modulus less than 1. $\Delta\Theta(T) = \Delta\Theta(T-1) . F(T)$ (4)

The vector $\Theta(T)$ is chosen such that :

$\Theta(T) = [-a(T/N)\ \alpha(T/N)\ -b(T/N)\ \beta(T/N)\ c(T/N)\ \gamma^{-1}(T/N)]$

$\gamma^{-1}(T/N)$ is defined using the Kalman gain and the vector of observations.

$$r(T/N) = B^*(T/N).Y(T/N+1)$$
$$\gamma^{-1} = 1 - c^*(T/N).Y(T/N)$$

A complete analysis of the propagation error system (4) can be found in [2].

The basic idea of the FTF stabilization is to compute some variables using two different but equivalent equations. If the computer precision is infinite we should obtain the same result. Otherwise in finite precision implementation, the two resulting values are different because of troncatures and roundoff errors. This difference is used to handle the numerical propagation error through a feedback control loop. Slock & Kailath [2] compute other variables using this difference.

The backward error can be computed in two ways:

as a convolution product of B(T-1/N) and Y(T/N+1) :

$$r_c^p(T/N) = B^*(T-1/N).Y(T/N+1)$$

or in a recursive way as in the instable version of FTF:

$$r_s^p(T/N) = -\lambda.\beta(T-1/N).[c(T/N+1)]_{N+1}$$

The Kalman gain itself can be computed in two ways:

as a function of the backward error :

$$[c_c(T/N+1)]_{N+1} = -\lambda.\beta^{-1}(T-1/N).r_c^p(T/N)$$

or using a recursive way to update this vector

$$[c_s(T/N+1)]_{N+1} = [c(T-1/N)]_N + [c(T/N+1)]_1.[A(T-1/N)]_N$$

The same operation can be applied to the inverse of the likelihood variable :

as a convolution product $\gamma^{-1}(T/N) = 1 - c^*(T/N).Y(T/N)$

or in a recursive way :

$$\gamma^{-1}(T/N+1) = \gamma^{-1}(T-1/N) - [c(T/N+1)]_1.e^p(T/N) \quad \text{and}$$
$$\gamma_s^{-1}(T/N) = \gamma^{-1}(T/N+1) + [c(T/N+1)]_{N+1}.r^p(T/N)$$

We can note that two supplementary convolution products which correspond to 2N additionnal operations has been introduced.

The final variables are computed as a combination of both results such that :

$$r^p(T/N) = r_s^p(T/N) + K.[r_c^p(T/N) - r_s^p(T/N)]$$

Where K is the feedback factor.

The same computing technique is applied to

$\gamma^{-1}(T/N)$ and $[c(T/N+1)]_{N+1}$

The resulting value of $r^p(T/N)$ is introduced in computing other variables with different factors (Ki). Slock & Kailath use 5 coefficients [2]. The choice Ki = 0 corresponds to the FTF (7N).

The choice of these factors is crucial for the numerical stability of the algorithm. Slock & Kailath have also shown that the optimal choice is function of the forgetting factor λ, the filter order N and the the input signal characteristics. For a white noise excitation they have developed a routine minimising the next criterion over an horizon $T1 = 10^5$:

$$\sum_{T=0}^{T1} \{ (r_s^p(T/N) - r_c^p(T/N))^2 + (\gamma_s^{-1}(T/N) - \gamma_c^{-1}(T/N))^2 \}$$

3.3. Adaptation of the stabilized FTF to echo cancellation:

where the near-end echo and the far-end one are 16ms long and 1200ms from each other.
If the sampling frequency is 7200 Hz, the echo filter length must be 8000 coefficients. This filter is not feasible. In the sequel we will propose another filter structure in order to simplify the preceding one taking into account the special echo structure. The proposed filter will include a first one whose length is equal to the near-end echo , a delay line whose length is equal to the period d separating the two echos and a second filter with the same length as the first one. In this structure the maximum number of coefficients is 230. In general the time delay d and the length d1 are unknown but can be estimated at the beginning of the transmission period.
In this part we study the effects of this structure on the robustness and feasibility of the echo canceller. We have proposed two structures.
For the first structure, the estimation of the echo is given by:

$$\hat{e}(n) = \sum_{i=0}^{8749} w_i y(n-i)$$

where w_i are the components of the parameters vector W.
The second structure (figure 3) is made of two parallel and equal length filters. They identify the near-end and the far-end echo respectively. Their length is 115 coefficients and they are estimated using:

$$\hat{e}_{pr}(n) = \sum_{i=0}^{114} w_i y(n-i) \quad \text{and} \quad \hat{e}_{loin}(n) = \sum_{i=8645}^{8760} w_i y(n-i)$$

In order to estimate the filter parameters, we need to compute the two error signals concerning near-end and far-end echos, but we only know the signal corresponding to their sum and the received one. However in general the near-end and far-end echos are uncorrelated since they are far from each other and because of the different scramblers in the receiver and the transmitter.
Therefore the estimated error is given by :

$$\varepsilon(T/N) = [r(T) + e(T)] - \hat{e}_{pr}(T/N) - \hat{e}_{loin}(T/N)$$

$$\varepsilon(T/N) = [r(T) + e_{pr}(T) - \hat{e}_{pr}(T/N)] + e_{loin}(T) - \hat{e}_{loin}(T/N)$$

$$= r_1(T) + e_{pr}(T) - \hat{e}_{pr}(T/N)$$

where $r_1(T)$ is uncorrelated with $e_{pr}(T)$
And :

$$\varepsilon(T/N) = [r(T) + e_{pr}(T) - \hat{e}_{pr}(T/N)] + e_{loin}(T) - \hat{e}_{loin}(T/N)$$

$$= r_2(T) + e_{loin}(T) - \hat{e}_{loin}(T/N)$$

where $r_2(T)$ is uncorrelated with $e_{loin}(T)$

We can then use the error signal $\varepsilon(T/N)$ to update the filter parameters for estimating simultaneously the near-end and far-end echos. It is important to note that when starting the estimation of the near-end echo, it is adapted alone because the far-end echo has not yet arrived. Then when the latter reaches the receiver, the adaptive filter begins its adaptation.

4. EXPERIMENTAL RESULTS

In this part we will present some experimental results concerning the evaluation of the proposed structures. In particular, the numerical stable behaviour of the FTF, its better performances when compared to LMS, and the simulations are accomplished in a simulated modem.
The noiseless case (r(t) = 0) corresponds to the initialisation period in a modem: in this period the modem is only operating in the transmitting mode. The two filters are 39 coefficient length. The Forgetting factor is 0.997.
We use the a priori error signal as a criterion for performance comparison between the two algorithms.

$$\varepsilon^P(T/N) = d(T) - W^*_{pr}(T-1/N).Y(T/N) - W^*_{loin}(T-1).Y(T-ret/N)$$

where the delay "ret" is equal to 1000 samples.

The vectors $W_{pr}(T-1/N)$ and $W_{loin}(T-1/N)$ denote respectively the near-end and far-end echo estimators.
In fact we use the following ratio :

$$10.\log_{10}\frac{E[\varepsilon^P(T)^2]}{E[d(T)^2]} \qquad \text{with} \quad d(t) = e(t) .$$

which represents the energy of the echo error estimation normalized by the energy of the echo. Figure 5 presents the noiseless case results . These results will be compared to those resulting from the use of LMS algorithm in the same conditions.
In the situation of bilateral transmission ($r(t) \neq 0$) where the received signal or any other signal (quantification noise, far-end echo, near-end echo, ...) is considered as perturbating signal. The signals d(T) and r(T) levels are fixed as in the real situation[7] such that the signal to noise ratio is equal to 34 dB:

$$10.\log_{10}\frac{E[d(T)^2]}{E[r(T)^2]} = 34 \text{ dB} \qquad \text{with} \quad d(t) = r(t) + e(t)$$

The performance criterion is defined in this situation by the following relation :

$$10.\log_{10}\frac{E[(r(T) - \hat{r}(T))^2]}{E[(d(T) - r(T))^2]}$$

We can easily show that this criterion is more general than the previous one and then include it as a special case. Figure 6 presents the results in the noisy situation.
In order to verify the performances of FTF in terms of convergence speed and precision of the estimation, we use the LMS in the same structure as for the FTF.
First we suppose that we are in an initialisation phase : r(t) = 0. As for FTF, N equals 39.

$$W_{pr}(T/N) = W_{pr}(T-1/N) + \beta . \varepsilon^P(T/N).Y(T/N)$$

$$W_{loin}(T/N) = W_{loin}(T-1/N) + \beta . \varepsilon^P(T/N).Y(T-ret/N)$$

where $W_{pr}(T/N)$ and $W_{loin}(T/N)$ are respectively the near-end and far-end echo estimators.

Since it is well known that β, the step size, creates a compromise between the convergence speed and the precision of the estimation, different values of this parameter have been used.

The best result is obtained for β=0.1. When comparing this result to FTF , we note that this latter is more performant. The difference between LMS and FTF algorithms after the convergence has been established is 90dB.
When using FTF,in bilateral transmission case ($r(t) \neq 0$), the signal to noise ratio is fixed to 34 dB for the simulation. The results are reported in figure 6 with those corresponding to FTF. We observe a decrease in the superiority of FTF compared to LMS, but it is nevertheless at least 20 dB more performant.

We have observed that the simulated echo canceller using the stabilized version of FTF has better performances than using the LMS algorithm. We have also noted that for the special characteristics input signal (modem signal), the FFT algorithms keep their numerical stable behaviour even when implemented on 16 bits DSP [3]. After this feasability study we show that the echo canceller can be implemented using two DSP TS 68930, with FTF (9N) as the adaptive algorithm.

Références:

[1]. Recueil des recommandations du CCITT, tome VIII, "Transmission de données", *CCITT, Place des Nations, CH 1211 Genève 20.*

[2]. D.T.M.Slock and T.Kailath : "Numerically Stable Fast Recursive Least Squares Transversal Filters", Proc. of IEEE-ICASSP-88, pp 1365-1368, New York.

[3]. B.Joseph, P.Baylou, M.Najim, and D.Aboutajdine "Mise en oeuvre de l'algorithm FTF stabilisé sur un processeur spécialisé 16 bits à virgule fixe", Douzième Colloque GRETSI pp 877-880 June 89, Juan les Pins .

[4].A.M.Alvarez:"Echo canceller design and implementation of a CCITT V32 Modem: Software solution" ICASSP-89, pp 1384-1387, Glasgow,UK.

[5]. J.M.Cioffi and T.Kailath : "Fast, recursive least-squares transversal filters for adaptive filtering ". IEEE. Trans. on ASSP, vol - 32, n°2, pp 304-337, 1984.

[6]. L.Ljung, M.Morf, and D.Falconer, " Fast calculation of gain matrices for recursive estimation schemes", Int. J. Control, vol - 27, n°1, pp 1-19, 1978.

[7]. J.M.Stein, "Les modems pour transmission de données". Masson. Paris 1987. (Published in French)

[8]. D.Godard, "Channel equalization using Kalman filter for fast data transmission," IBM J. of Res. and Dev., pp 267-273 , May 1974.

[9]. A.Benallal and A.Gilloire : "A new method to stabilize fast RLS algorithms based on a first-order model of the propagation of numerical errors " Proc. of IEEE-ICASSP-88, pp 1373-1376, New York.

[10]. J. L.Botto and G.V.Moustakides : "Stabilizing Fast Kalman Algorithms". Trans. on. ASSP. vol. -,37, n° 9. Sept 1989. pp 1342-1348.

[11]. B.Widrow and M. Stearns "Adaptive Signal Processing", Prentice Hall, New York, 1985.

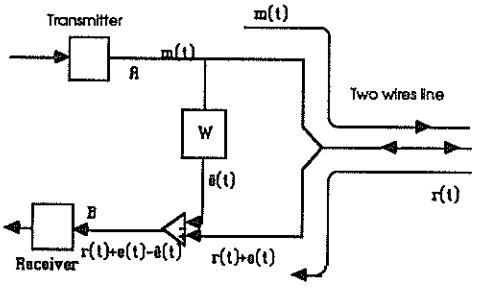


Figure 1 - Adaptive echo canceller filter

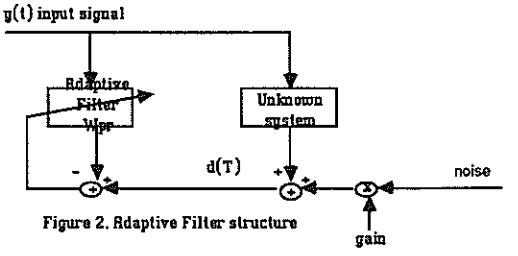


Figure 2. Adaptive Filter structure

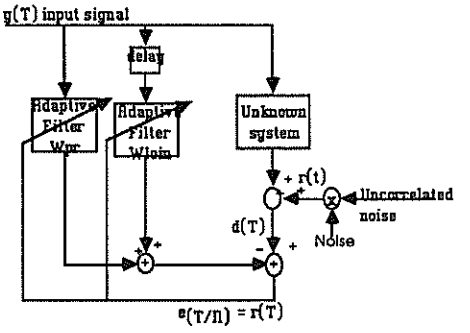


Figure 3.:Adaptive Echo Canceller structure

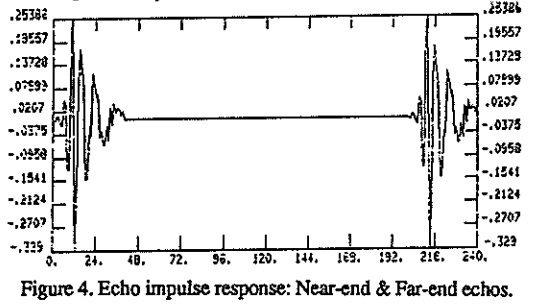


Figure 4. Echo impulse response: Near-end & Far-end echos.

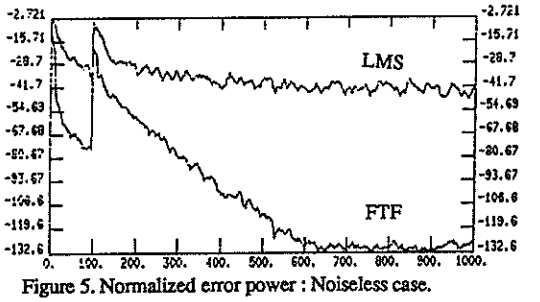


Figure 5. Normalized error power : Noiseless case.

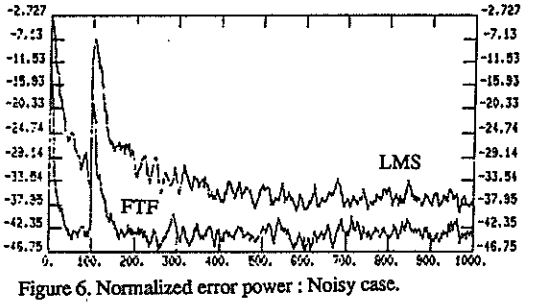


Figure 6. Normalized error power : Noisy case.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A New Sub-band Two-model IIR Structure for Acoustic Noise Cancellation

Sen M. Kuo and Bob H. Lee*

Department of Electrical Engineering, Northern Illinois University, DeKalb, Illinois 60115, U.S.A.

This paper presents a new sub-band adaptive IIR noise cancellation scheme whose structure enables the cancellation of repetitive noise in a received speech signal. Comparing with the classical adaptive transversal noise cancellation method, this new structure has two major differences. First, the quadrature mirror filter (QMF) has been used to decompose signals into sub-bands in the time domain. Second, a new constrained adaptive noise canceller was developed using two adaptive IIR processes, *foreground notch filtering and background line enhancing*, to remove the multiple-frequency, or harmonic, noise from a received signal. Computer emulation was done using a set of actual speech signals corrupted by car engine noise. Experiments demonstrate that more than 20 dB signal-to-noise ratio improvements can be achieved.

1. INTRODUCTION

In our daily life we hear a variety of sounds, plenty of which are discordant to our ear and we often wish such sounds to be eliminated. Examples are the sound created by the rotating machines, vehicles, and transformer noise. Many adaptive noise control systems have been developed recently to cancel such noises. There are two major categories of adaptive filters, adaptive finite impulse response filters and infinite impulse response filters, which may be used as an adaptive noise canceller (ANC). Traditionally, ANC has been realized with adaptive finite impulse response (FIR) filters[1], [2]. The advantages of FIR adaptive noise cancelers are inherently stable and easy implementation. Also, they usually will converge to a global optimum solution.

The study of car engine noise shows that car engine noise consists of not only fundamental frequency but also its harmonics. The major noise components are most likely to occupy the lower frequency band. For example, a spectrum of a received signal, a male speech corrupted by an engine noise running at 4000 rpm, is shown in figure 1. In such a case, common adaptive noise cancellations using LMS algorithm lack power to cancel all of the harmonic components. The reason is probably due to the multi-frequency phenomenon. Usually, the LMS-based adaptive noise canceller will have a different convergence speed at different frequencies. Since car engine noise includes many harmonics, the LMS adaptive noise canceller will drive most of its weights to cancel only part of the noise components. The performance of a classical LMS noise canceller is also illustrated in figure 1. Only one of the major noise components has been greatly reduced in this case.

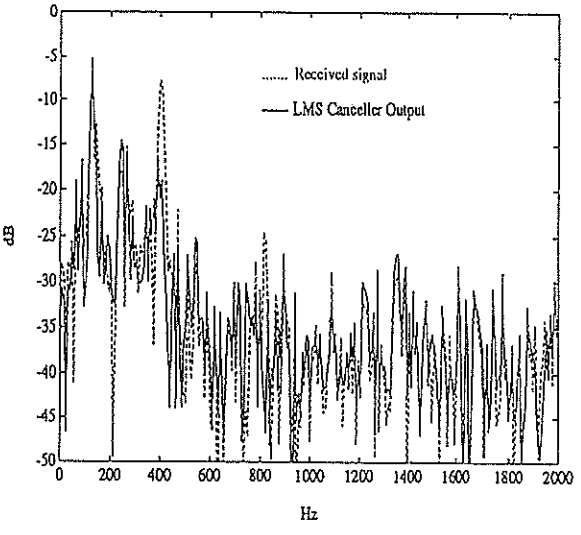


Figure 1 Power spectra of the received signal and LMS adaptive noise cancellation result.
*Received signal = Speech + Engine noise (4000 rpm).

* Bob H. Lee is now with U.S. Robotics, Inc., 8100 North McCormick Blvd., Illinois, 60076, U. S. A.

Unlike their counterparts, adaptive infinite impulse response (IIR) filters can usually match physical systems much better and require lower orders especially if the system has poles. However, the mean squared error function for adaptive IIR filters may not always be quadratic. Stability is another problem which makes the adaptive IIR filter application more difficult to implement[3]. Recently, a number of constrained adaptive IIR filters[4]-[6] have been introduced as adaptive line enhancers. These IIR structures have good performances and are efficient for the enhancement of sinusoids when suitable initial conditions have been selected. However, these structures do not have good performances when modified as adaptive noise cancellers. Furthermore, these IIR filters may miss some of noise frequencies if the adjacent interferences are located very close.

In this paper, a new two-model sub-band adaptive noise cancellation method is introduced for the purpose of canceling engine noise and its harmonics. The quadrature mirror filter banks are employed to split signals into sub-bands. In each sub-band two-model noise canceller is realized as a cascade structure of time-varying notch filters which is used to remove the engine noise from the received signal, and a parallel structure of adaptive bandpass filters which is used to estimate the fundamental frequencies and harmonics of the engine noise in received signal. The time varying notch filters and adaptive bandpass filters are all constrained second order IIR filters. The poles of the IIR filter are forced to be located inside the unit circle to ensure the stability. The variables of the IIR filter are the function of the frequency of noise components. After the system converges the notch frequencies will line up the frequencies of the noise components such that the noise in received the signal will be removed or reduced. The experiment results show that more than 20 dB signal to noise ratio improvement has been achieved.

2. QUADRATURE MIRROR FILTER BANKS

The quadrature mirror filter technique has been used to split signals into a number of sub-bands in time and frequency domain for years[7], [8]. The two-channel QMF bank, in figure 2, is one of the easiest and most commonly employed structures. The analysis bank is composed of a lowpass filter $H_l(z)$ and a highpass filter $H_h(z)$, which splits the incoming signal x(n) into two sub-bands. The lowpass signal $x_l(n)$ and highpass signal $x_h(n)$ are then decimated by a factor of two. After the noise cancellation process, signals are passed through interpolators and reconstructed to form the output signal. Aliasing will be caused by down-sampling. The lowpass filter $F_l(z)$ and highpass filter $F_h(z)$ are needed at synthesis end to eliminate the aliasing components. The requirement to remove the aliasing is to design the filter coefficients that satisfied the following conditions

$$H_h(z) = H_l(-z),\ F_l(z) = H_l(z),\ F_h(z) = -H_h(z).$$

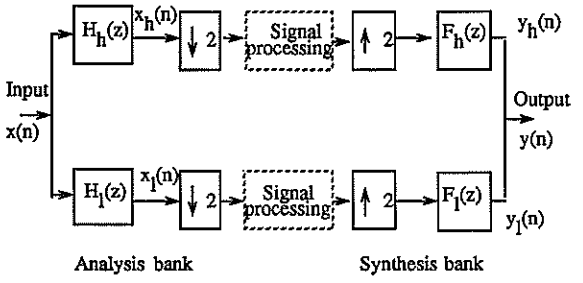


Figure 2 Two-channel quadrature mirror filter bank.

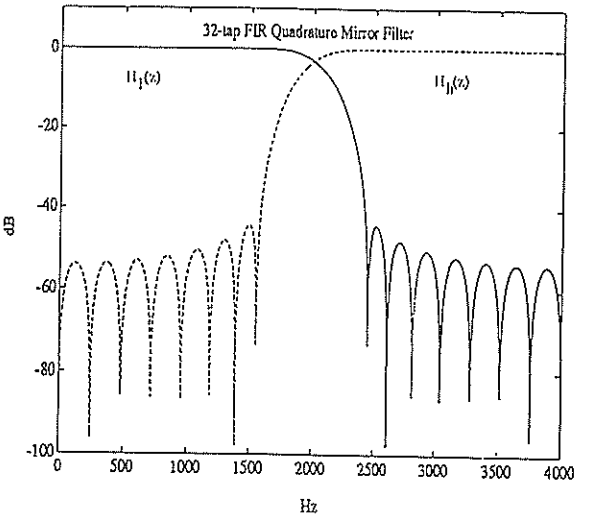


Figure 3 (a) Frequency response of 32-tap symmetric FIR quadrature mirror filter.
(b) Frequency response of 8-tap asymmetric IIR quadrature mirror filter.

The analysis and synthesis filters can be realized as FIR or IIR filters depending upon the application. The FIR QMF has the properties of linear phase, good stability, but needs longer filter length to achieve a sharp

cut-off response. While the IIR QMF needs many less coefficients for the same cut-off frequency response requirement. However, IIR filter suffers the non-linear phase characteristic. The detailed QMF design techniques were given in references [7],[8]. In figure 3 we present the frequency response of the quadrature mirror filters which we used in our simulations.

3. SUB-BAND TWO-MODEL ADAPTIVE IIR NOISE CANCELLER

Figure 4 illustrates a brief block diagram of the new sub-band adaptive IIR noise canceller. The binary tree structure of the quadrature mirror filter banks were used to split adjacent frequencies into different sub-bands in the time domain. At each lower frequency sub-bands, the new adaptive noise cancellation technique was employed. Since very little car engine interference exists in the high frequency range, a simple delay line was used in those sub-bands. After the accomplishment of the noise cancellation, the sub-band signals were interpolated by a quadrature mirror filter and recombined as the output of the noise canceller.

Each sub-band noise canceller is the combination of a parallel structure of adaptive IIR bandpass filters and a cascade structure of time-varying IIR notch filters as shown in figure 5. The purpose of using parallel bandpass filter structure is to trace the multiple sinusoidal components buried in the reference signal. Then, the error signal of each bandpass filter was used for updating the adaptive weights. After the canceller converges, the cascade time varying notch filters in foreground process will greatly reduce the multi-rate interference in primary channel.

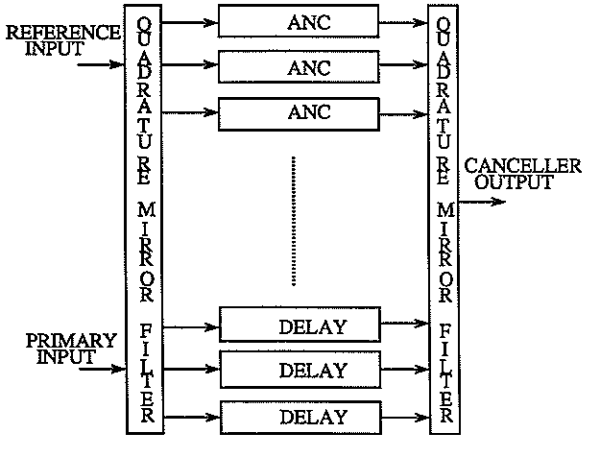


Figure 4 Two-model sub-band adaptive IIR noise canceller.

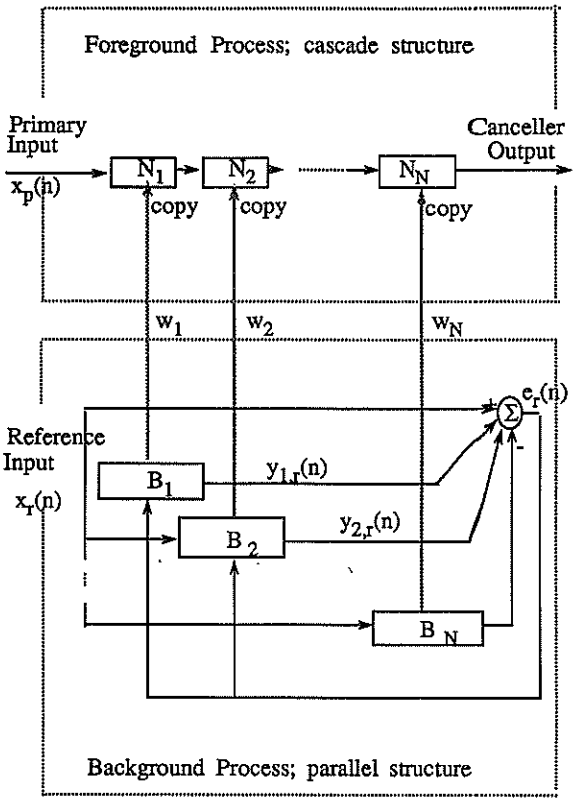


Figure 5 Parallel-cascade adaptive IIR two-model noise cancellation.

We decomposed the parallel-cascade two-model adaptive IIR structure as a single stage structure. The following derivation is based on a single stage two-model structure. However, the parallel and cascade structure can be derived using the same way. The basic structure of the two-model adaptive IIR noise cancellation scheme is illustrated in figure 6. The second order IIR adaptive bandpass filter we used has a predetermined radius($0 << r < 1$) to ensure the poles of the adaptive IIR bandpass filter located inside the unit circle.

The output of the adaptive bandpass filter, $y_r(n)$, is described by

$$y_r(n) = G_b x_r(n) - G_b r x_r(n-2) + w(n) y_r(n-1) - r^2 y_r(n-2)$$

where $x_r(n)$ is the reference input, $w(n)$ is the weight of the adaptive bandpass filter, the constants $G_b = 1 - r$.

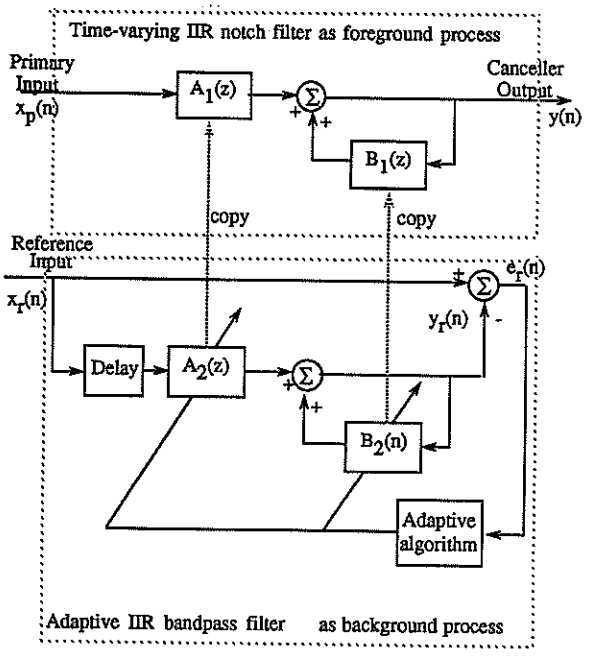


Figure 6 Two-model IIR adaptive noise canceller.

We have also developed a new time-varying IIR notch filter used as the noise canceller, with a different structure than other IIR notch filters[9], [10]. The similar constrain on the radius($0 << r < 1$) of the poles is also applied to the IIR notch filter. The complex-conjugate pole-zero pairs of the IIR notch filter are constrained as (1) the poles are placed inside the unit circle, and (2) the corresponding zeros are located right on the unit circle with the same radians to ensure the stability. The weight of the time-varying notch filter is copied from the adaptive IIR bandpass filter during the process. When the adaptive filter converges, the central frequency of the time-varying IIR notch filter will be the same as the sinusoidal frequency in primary signal and then, the harmonic, or multi-frequency, noise will be greatly attenuated by notch filters. The output of the single stage adaptive two-model IIR noise canceller, y(n), is written as

$$y(n) = G_n[x_p(n) - G_n w(n) x_p(n-1) + x_p(n-2)] + w(n)y(n-1) - r^2 y(n-2)$$

where $x_p(n)$ is the primary input(signal plus noise), $G_n = 1/r$. The bandwidth of the adaptive filter depends upon the radius of the poles. The larger the radius, the narrower the bandwidth will be formed. A sharp bandwidth can provide good estimation of the noise frequency. However, it may result slow tracking speed if the bandwidth is too narrow. The same consideration is also true to the time-varying notch filter. A large radius of the poles of the notch filter will form a sharp null. Considering the speech signal will also be attenuated around the notch, a sharp null is preferred.

4. ADAPTIVE ALGORITHM

The error of the adaptive IIR bandpass filter is obtained by subtracting $y_r(n)$ from reference signal $x_r(n)$,

$$e_r(n) = x_r(n) - y_r(n).$$

This error signal is used for weight updating. The central frequency of the adaptive bandpass filter will line up with an incoming sinusoidal signal after filter converges. The weights for the adaptive IIR bandpass filter is adjusted by

$$w(n) = w(n-1) + \mu e_r(n)\, \alpha(n)/\sigma(n)$$

where $\alpha(n)$ is the partial derivative of $e_r(n)$ with respect to filter weight and is written in a recursive way as

$$\alpha(n) = w(n)\alpha(n-1) - R_1\alpha(n-2) + R_3 x_r(n-1) + y_r(n-1)$$

and $\sigma(n)$ is the normalization factor. It can be calculated by

$$\sigma(n) = v\sigma(n) - (1 - v)\alpha^2(n).$$

with the forgetting number v at the range of $0 << v < 1$.

5. SIMULATION RESULTS

The computer simulations were done using a set of real speech signals corrupted by engine noise with different rpm. The schematic diagram of the experiment is given in figure 7.

All input signals were sampled at 4 kHz. Two omnidirectional microphone were used to pick up primary and reference signals, respectively. Anti-aliasing filters were used before the A/D convertors to prevent the aliasing signal caused by low sampling rate. A male speech corrupted by a four cylinder Quad 4 engine noise at different engine speed, from 1600 - 5000 rpm, was used for simulation. The cancellation result illus-

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

ADAPTIVE PERIODIC NOISE CANCELLATION FOR THE CONTROL OF ACOUSTIC HOWLING

J.B.Wright and J.B.Foley
Dept. of Microelectronics and Electrical Engineering,
Trinty College, Dublin 2, Ireland

Abstract

A new approach to dealing with acoustic echo is presented. The Adaptive Periodic Noise Canceller (APNC), based on the familiar adaptive line enhancer removes resonant peaks caused by acoustic feedback from a broadband speech signal. These resonances do not have to be harmonically related. The APNC has relatively few taps which while reducing the computational load also restricts residual noise. Simulation results which clearly illustrate the APNC operation are included.

1. Introduction

A long standing goal in telephony has been the replacement of the traditional handset with a free standing microphone and loudspeaker [1]. The hands free telephony situation has long suffered from the problem of acoustic echo which due to acoustical coupling between microphone and loudspeaker, (Fig. 1), may cause self-sustained oscillations which are heard as "howling". Solutions involving adaptive filtering techniques are almost invariably based on the standard echo canceller intended to eliminate echoes generated by impedance mismatches in the transmission path. The drawback of this method is that straight forward FIR implementation requires a filter with a large number of taps, up to 4000 being suggested by some [1]. Clearly a large number of taps increases the computational load of the filter as well as producing excessive residual noise. This contribution presents a new approach to the problem of acoustic echo cancellation.

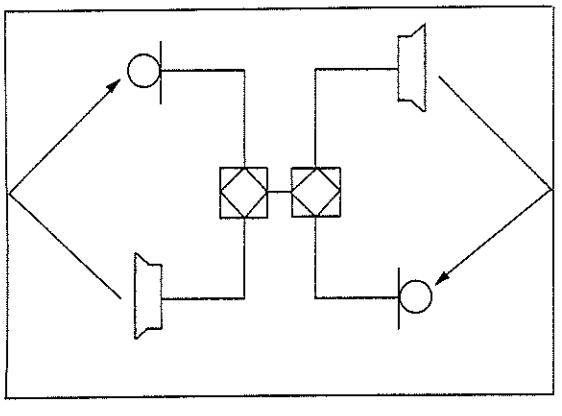


FIG.1 Generating acoustic echo

2. The adaptive periodic noise canceller (APNC).

Acoustical feedback is brought about by a signal repeatedly passing through the same loop (Fig. 1). With increasing system gain one particular frequency component (and possibly more) will start "growing" more rapidly than others and when a critical value of gain is reached the system resonates at this frequency. An acoustic echo can therefore be represented by the presence of an unwanted resonance in the system. If a signal processing system is able to identify this resonance it can then be subtractively removed from the signal path. The APNC, shown in block form in Fig. 2, is a two channel processor driven by the Widrow-Hoff LMS algorithm [2] which iteratively adjusts the filter weights towards their optimal values. During experimental work it was found

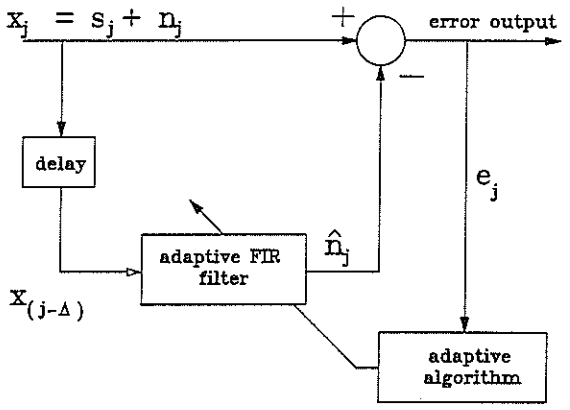


FIG.2 The Adaptive Periodic Noise Canceller (APNC)

necessary to use a modified version of the algorithm [3]. The "leaky" LMS algorithm is written as

$$W_{j+1} = \alpha W_j + 2\mu e_j X_j \qquad (1)$$

where α is a constant very close to 1, W_j is the vector of L filter coefficients, X_j is the contents of the tapped delay line of the FIR filter, μ is the user selected adaptation constant and e_j is the error signal. The notation associated with the APNC shown is defined as follows

$$x_j = s_j + n_j \qquad (2)$$

is the current input signal comprising of a speech component, s_j, and a resonant acoustic echo, n_j, while $x_{(j-\Delta)}$ is the input signal after a delay of Δ sample periods. The adaptive filter output, $\hat{n}_j$, will be shown to be an estimate of the resonance n_j contained in the APNC input. The error output is then

$$e_j = x_j - \hat{n}_j \qquad (3)$$

$$\cdot \cong s_j \qquad (3a)$$

i.e. the error output is a close approximation to the speech.

3. Filter time constant

The time constant associated with the adaptive process is inversely proportional to μ and hence a decrease in μ results in an increase in convergence time of the adaptive filter. If the value of μ is made small enough then the variation of W_j will be rendered far slower than the statistical variations of the incoming speech signal, i.e. the voiced components are of relatively short duration when compared to the tracking time constants and therefore from the point of view of the adaptive filter the speech signal may be considered weakly correlated or broadband.

Fortunately, a small value of μ also helps to restrict residual noise, due to coefficient variance, to a relatively low level.

4. Filter decorrelation delay.

Noting that correlation time is inversely proportional to bandwidth and if $R_{xx}(\Delta)$ is defined as

$$R_{xx}(\Delta) = E\left[x_j x_{j-\Delta}\right] \qquad (4)$$

then

$$R_{xx}(\Delta) = R_{ss}(\Delta) + R_{nn}(\Delta) \qquad (5)$$

since s_j and n_j are not correlated. As $R_{ss}(\Delta)$ is essentially zero, due to the broadband nature of the speech,

$$R_{xx}(\Delta) \cong R_{nn}(\Delta) \qquad (6)$$

The adaptive filter operates in such a manner as to form an optimal estimate of the primary correlated component, n_j in this case, therefore the filter weights will form a narrow bandpass transfer function about the centre frequency of n_j, the resonant acoustic echo.

5. Real time implementation of the APNC

The real time APNC presented here was developed using a Loughbourgh Sound Images Ltd. TMS320C25 PC Board. A 32 tap adaptive filter with an associated decorrelation delay of 8 sample periods proved sucessful. With f_s= 10kHz then 1000 instruction cycles are available and the APNC uses only approximately one third of this processing time.

An experiment simulating the "howling" condition was prepared with a broadband noise

signal replacing the speech while a periodic tone simulated the resonant acoustic echo. The experimental curves were obtained directly from an x-y plotter driven by a Marconi spectrum analyser. Fig. 3 shows an input consisting of a broadband signal combined with the unwanted resonant acoustic echo at $0.1f_s$. The output of the APNC with the periodic component suppressed is shown in Fig. 4. A notch is detectable in the broadband output at $0.1f_s$ and the width of this notch is inversely proportional to the number of filter taps used. A compromise must be reached between some slight degradation of the speech signal and the extra residual noise added when more taps are used. The input is also high pass filtered to remove any residual dc which may exist on a telephone line.

A plot of the magnitude of the filter gain frequency response, $H(\omega)$, was obtained by "freezing" the weights after adaptation and applying white noise to the stationary filter. Fig. 5 clearly displays the passband around the frequency of the periodic input and indicates good agreement with theory.

An input consisting of two sinusoids, $0.1f_s$ and $0.16f_s$, combined with the wideband signal as shown in Fig. 6 was also applied to the APNC input and the resulting output is displayed in Fig. 7. Finally the frequency response, $H(\omega)$, for a two tone input is shown in Fig. 8 with peaks corresponding to the locations of the periodic components.

6. Conclusion

A simple but effective solution to acoustic echo is proposed which is capable of being implemented in real time on a modern DSP chip. The APNC is computationally efficient and shows itself capable of supressing multiple acoustic resonances which are not harmonically related.

7. References

1. K. Murano, S. Unagami, F. Amano. "Echo cancellation and applications ", IEEE Communications Magazine, Vol.28, No.1, pp 49-55, January 1990.

2. B. Widrow and S.D. Stearns, Adaptive Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1985

3. J. Treichler, C. Johnson Jr., M. Larimore, Theory and Design of Adaptive Filters, Wiley, New York, 1987

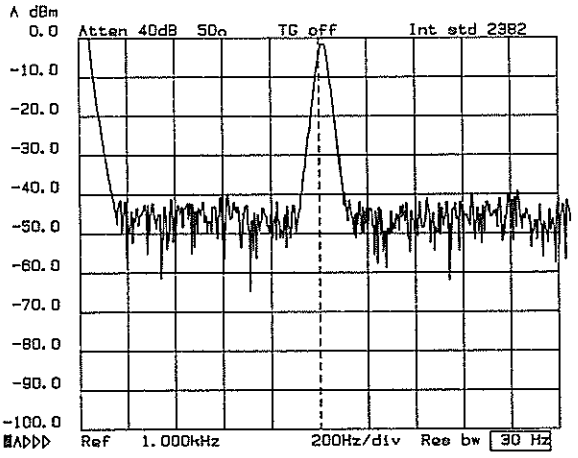


FIG.3 APNC input

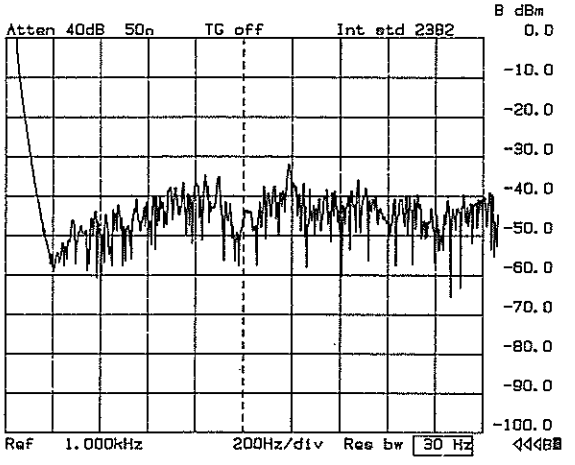


FIG.4 APNC output

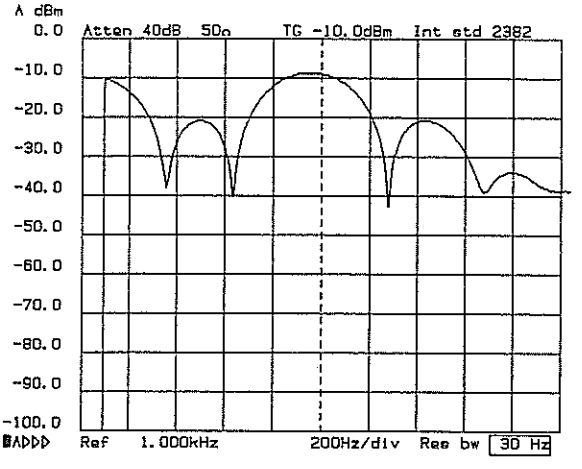


FIG.5 Filter frequency response, $H(\omega)$, of the adaptive filter after adaptation

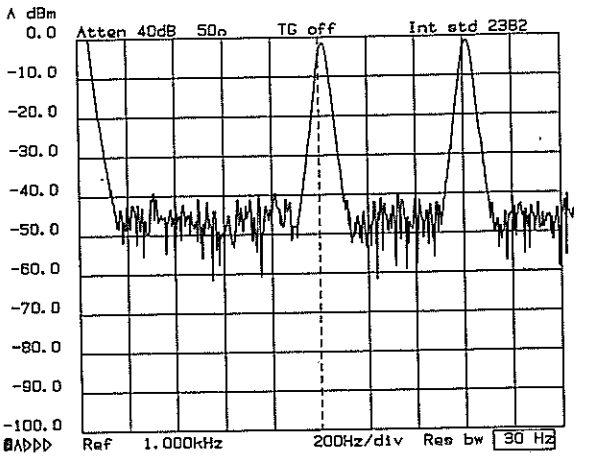


FIG.6 APNC input

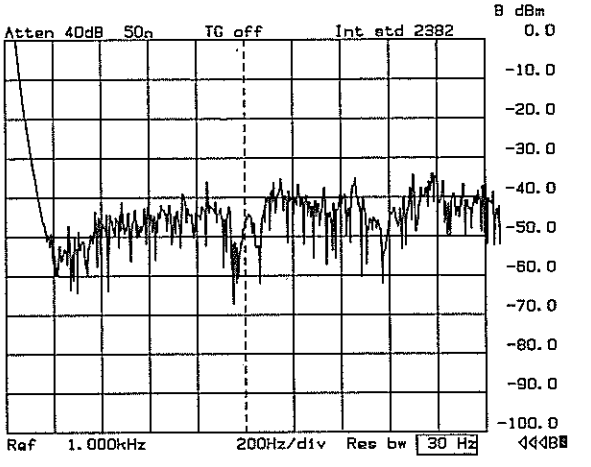


FIG.7 APNC output

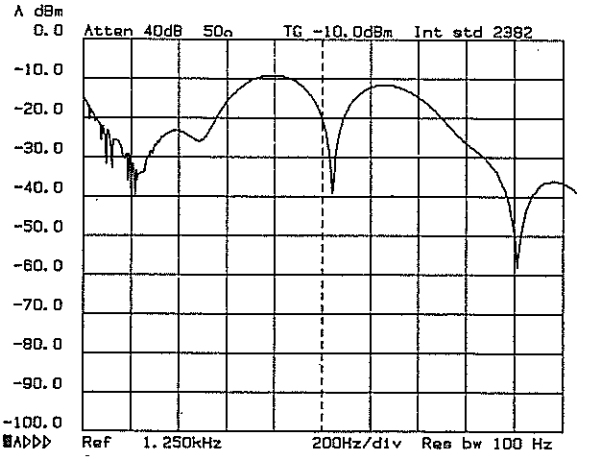


FIG.8 Filter frequency response, $H(\omega)$, of the adaptive filter after adaptation

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

ACOUSTIC ECHO CONTROLLER FOR WIDE-BAND HANDS-FREE TELEPHONY

Jean-Pascal JULLIEN, Grégoire LE TOURNEUR, André GILLOIRE
Centre National d'Etudes des Télécommunications
CNET LAA-TSS CMC BP 40 22301 LANNION CEDEX FRANCE

This paper describes an acoustic echo controller designed for a wide-band hands-free telephone prototype and the dedicated hardware implementation. This prototype complies with the CCITT recommendation P30 for the operation on wide-band speech (150 Hz to 7 kHz); it includes a speech codec (CCITT G722 standard) for digital transmission over the ISDN. The hardware uses commercial digital signal processors. The residual acoustic echo is inaudible and the conversational interactivity is good.

1. INTRODUCTION

The acoustic echo problem arises in audio terminals for telecommunications like teleconference systems, hands-free telephones, etc... This problem is now well identified and many solutions have already been proposed and tested [1],[2]. The control of the acoustic echo is necessary to limit the communication quality impairment due to echo; this impairment may be very high for communications having long transmission delays. This problem stands as a challenge, since efficient control requires complex signal processing such as very long adaptive filters whereas the cost of audio terminals should be as low as possible. Therefore tradeoffs between complexity and performances must be found. The solution described in this paper is based on the efficient combination of variable losses with a classical echo canceller; a global mechanism using common parameters and variables controls both functions [3]. The complete system has been implemented on a hardware using DSP chips; it is a part of a wide-band hands-free telephone prototype designed and built by CNET.

2. DESIGN RULES FOR THE PROTOTYPE

The quality of communications involving hands-free terminals depends on uncontrollable factors like the acoustic environment (room, noise) of the terminals and the physical positions (distance, movements) of the users. Since those factors may be very different depending on the context of use, quality specifications have to be defined according to general criteria.

In commercially available hands-free telephone sets which operate with narrow-band speech (300 Hz - 3400 Hz), the acoustic echo is generally controlled by variable losses. Some characteristics of such devices are specified in the CCITT recommendation P34 [4]. Wide-band speech (150 Hz - 7 kHz) can be transmitted over the ISDN thanks to the CCITT speech coding standard G722 [5]. Future hands-free terminals which will operate with wide-band speech need echo control devices with improved performances. The CCITT recommendation P30 [6] specifies quantities like acoustic echo return loss (AERL) in the context of wide-band group audio terminals; other quantities like speech attenuation during double talk, etc... are being discussed within ETSI and CCITT working groups. The recommendation P30 and some figures proposed in those discussions have been used as references for the design of the prototype.

In this project the minimum AERL objective has been fixed at 35 dB; the maximum speech attenuation on receive and send sides during double talk periods has been limited to 6 dB after echo canceller convergence. Those values correspond to the requirements specified in [6]. Another design rule was to avoid degradation of the echo canceller's convergence by local speech while keeping ability to track echo path variations. A new method to control the echo cancellation algorithm has been carried out to deal with those aspects.

3. ACOUSTIC ECHO CONTROL MECHANISM

The block-diagram of the audio section of the prototype (excluding the codec G722) is shown on figure 1. This section includes all the signal processing functions for the control of the acoustic echo.

The required attenuation of the echo is provided by an acoustic echo canceller AEC and additional variable

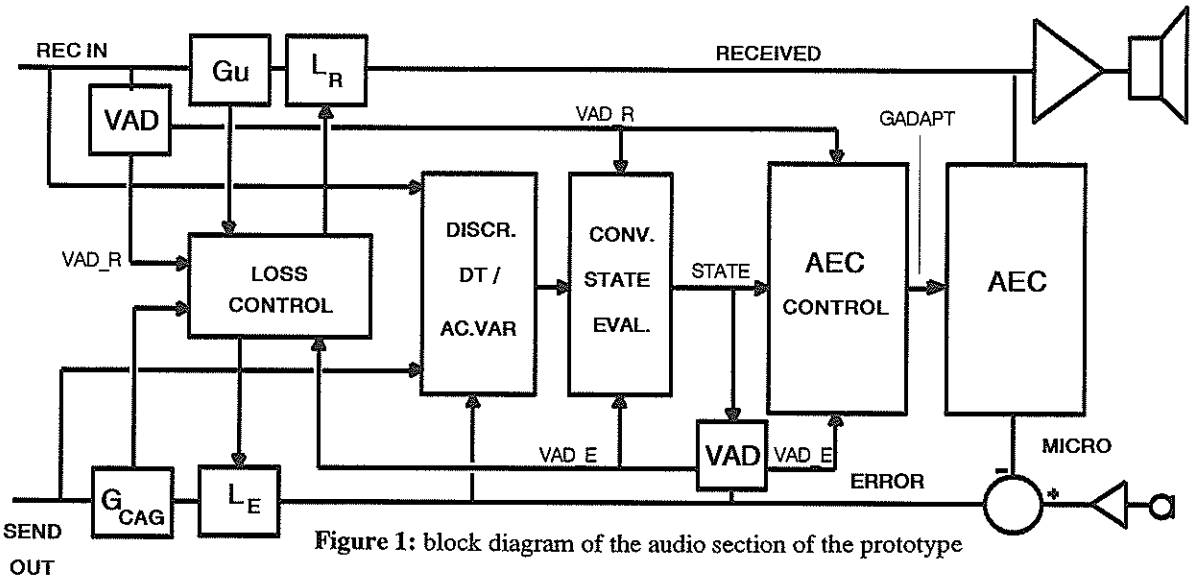


Figure 1: block diagram of the audio section of the prototype

losses L_R and L_E on the receive and send paths. The echo canceller operates on the full band of the speech signals (sampled at 16 kHz). It uses the classical normalized Least Mean Squares (NLMS) algorithm:

$$e(t) = y(t) - H^T(t-1)X(t)$$

$$P_x(t) = \alpha P_x(t-1) + (1-\alpha)x^2(t)$$

$$H(t) = H(t-1) + \frac{\mu}{NP_x(t)} e(t)X(t)$$

$x(t)$ is the received signal, $y(t)$ is the microphone signal and $X(t)$ is the vector of the N previously received samples. μ is the adaptation gain factor ($\mu < 2$); the adaptation gain is normalized by the short-term power of the input signal $P_x(t)$ ($\alpha \lesssim 1$).

The acoustic echo control mechanism is based on the "convergence state" concept. A variable STATE is elaborated in the module CONV. STATE EVAL. from inspection of the logical flags VAD_R, VAD_E and of the output of the Double Talk / Acoustic Variation discriminator DISCR DT /AC.VAR. This variable controls directly the adaptation gain of the echo canceller and the threshold of the voice activity detector module (VAD) on the send side, and it controls indirectly all the other modules.

When the terminal is switched on, the variable STATE has a low level since the convergence of the echo canceller has not yet started, and μ has a high value to help the adaptation. The level of STATE is increased as long as the convergence is improving and μ is reduced accordingly. The adaptation is frozen as soon as double talk or local speech are detected; it starts again only if a signal is received (VAD_R active) and no double talk is detected. The algorithm to compute the variable STATE is shown on fig.2.

The VAD on the send side, which detects local speech, should ideally be insensitive to the echo. Therefore its decision threshold is chosen large for low convergence states (large echo level) and low for high convergence states (low echo level). In addition the threshold depends

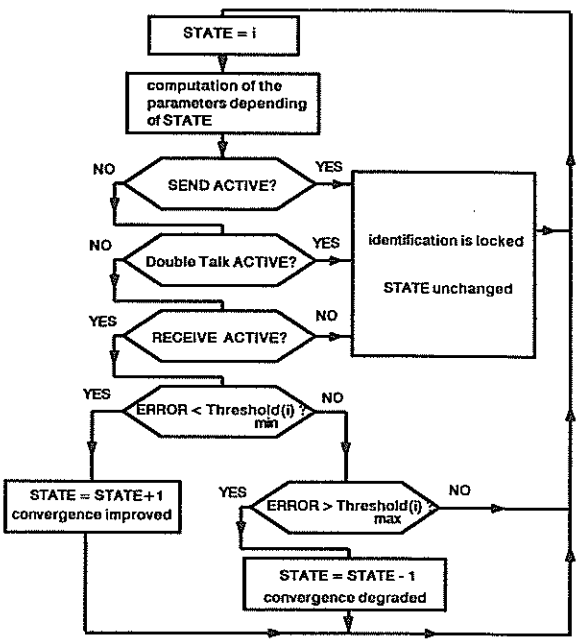


Figure 2: computation of the variable STATE

on the level of the room noise (in classical solutions, this threshold depends mainly of the noise level).

For proper operation of the mechanism, the echo which is high at the beginning of convergence and during echo paths variations should not be considered as local speech: the adaptation gain should be large for fast convergence whereas if local speech is present it should be zero. The decision is made by the DT / AC.VAR discriminator; it is obtained from the short-term correlation between the envelopes of the signals on the receive and on the send paths. This technique has been proven as efficient as more complicated ones which use the correlation between the received signal envelope and the adaptive filter coefficients increments.

The control of the additional losses L_R and L_E is achieved by a specific algorithm implemented in the LOSS CONTROL module. The sum $L = L_R + L_E$ is constant; the maximum loss is applied to the side detected as inactive by the VADs. When both sides are detected as active or inactive, both losses evolve towards $L_R = L_E = L/2$ with appropriate time constants. Thus, the loss control depends on the VADs decisions , and not on the relative levels of the signals on the receive and send sides as in classical solutions.

G_u is a user-adjustable gain on the receive path and G_{AG} is an automatic gain control on the send path. The values of both gains are taken into account by the algorithm in order to maintain the total loss $L = N - G_u - G_{CAG}$ constant (N is an attenuation dependent on the variable STATE; it guarantees the stability of the terminal).

4. IMPLEMENTATION

The hardware prototype of the acoustic echo controller is based on the general purpose digital signal processor Texas Instruments TMS 320C25 and the dedicated chip MOTOROLA DSP 56200. The main parts of the realization are indicated on Figure 3.

The DSP 56200 is an algorithm specific, digital signal processing peripheral designed to perform computationally intensive digital filtering. Two principal functions can be performed by this chip: fixed finite impulse response (FIR) filtering and adaptive FIR filtering using the LMS algorithm. The coefficients in the chip are coded on 24 bits, signal samples are coded on 16 bits and the accumulation is made on 40 bits. A fixed FIR with 256 taps can be implemented at the sampling rate of 37 kHz in one chip; the maximum sampling rate is 19 kHz for an adaptive FIR of the same length. A built-in interface allows cascading of several DSP's without external "glue" for higher sampling rates and larger numbers of taps (the cascaded chips are then seen as a

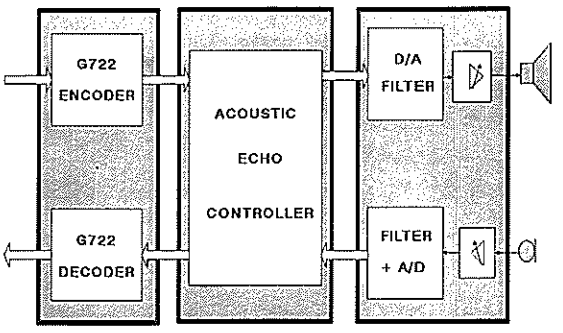


Figure 3: main audio parts of the realization

single powerful filtering peripheral). In the adaptive mode, the adaptation algorithm is exactly implemented without error or delay by the cascaded scheme; this is possible because each chip computes internally the error term *e(t)* and propagates it at the beginning of each cycle.

The hardware model uses from 2 to 4 DSP 56200 (depending on the performances required) for the echo canceller section. All the functions of the echo control mechanism: AEC control, VADs, CONV. STATE EVAL., DISCR. DT/ AC.VAR and LOSS CONTROL described in the previous section are implemented in one DSP 320C25. The computation of the adaptation gain, which is equal to μ times the inverse of the input signal power, involves careful scaling to take into account the scaling factor built in the DSP 56200. The figure 4 shows the links between the different chips.

The DSP 56200 are externally memory mapped in the TMS 320C25 with one wait state. Considering the time scale of speech events, the AEC control is performed every milliseconds, while the computation of the coefficient adaptation gain and the filterings in the VAD's are made at each sample period (62.5 μs). It takes about 500 cycles to perform these computations, and 40 cycles on the average for the control task; this fits in the 625 cycles available in one sampling period of 62.5 μs when using a TMS 320C25 running at 40 MHz.

The wide band speech codec implemented in the hardware is fully compatible with the G722 standard. It is a sub-band ADPCM algorithm with three modes of operation: 64 kbps for speech (mode 1), 56 kbps for speech and 8 kbps for data (mode 2), 48 kbps for speech and 16 kbps for data (mode 3). The full-duplex codec is implemented on a second TMS 320C25; the encoder requires about 590 cycles, the decoder 555 cycles and the I/O and interrupt routines 50 cycles (two successive signal samples are processed simultaneously in a time slot of 125 μs). The hardware implementation has been

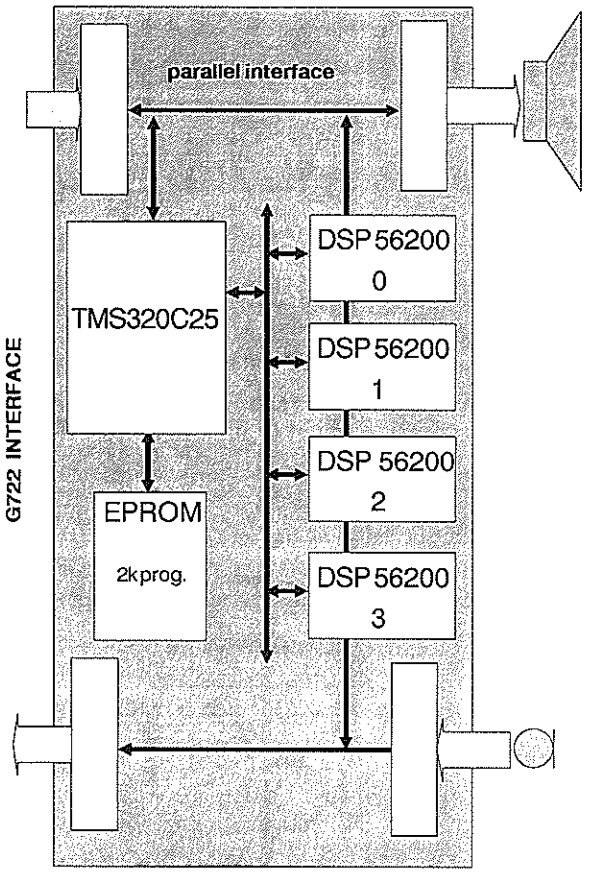


Figure 4: links between the chips

checked with the digital test sequences provided by the G722 standard. The audio parts have been carefully designed and implemented to comply with the G722 standard with regards to total harmonic distortion and noise specifications.

All those functions have been implemented on a PC-AT board, as shown on Figure 5. The PC supports the development software for the DSP's.

5. PERFORMANCES AND FURTHER IMPROVEMENTS

Typical ERLE's achieved by the acoustic echo canceller itself are typically in the order of 20 dB in a normal office; the canceller with 1024 taps (maximum available length) yields ERLE's of almost 30 dB in some situations. Since the attenuation of speech due to additional losses is no more than 6 dB during double talk periods, full duplex communication is possible when those periods occur, even with large transmission delays (500 ms). Speech is not noticeably disturbed; this shows that the mechanism correctly controls the adaptation of the canceller.

Further improvements can be expected from processing in frequency sub-bands for more efficient echo cancellation; the control mechanism could then use as well the signals in the sub-bands.

References

[1] A.Gilloire, J.F.Zurcher "Achieving the control of the acoustic echo in audio terminals" SIGNAL PROCESSING IV: Theories and Applications - Elsevier 1988

[2] Workshop on acoustic echo control, Berlin, 1989 (DBP, Dr. R.Wehrmann & Prof. E.Hänsler)

[3] French patent n° 89 11026 for "*Dispositif de traitement d'écho notamment acoustique dans une ligne téléphonique*" (Device for echo processing (especially acoustical echo) on telephone lines), inventors J.P.Jullien and G.Le Tourneur

[4] Recommendation P34 for Transmission Characteristics of Hands-free Telephones, CCITT Blue Book Vol.V, 1989

[5] Recommendation G722 for wide-band speech coding at 64 kbps, CCITT Blue Book, Vol.III, 1989

[6] Recommendation P30 for Group Audio Terminals, CCITT Blue Book Vol.V, 1989

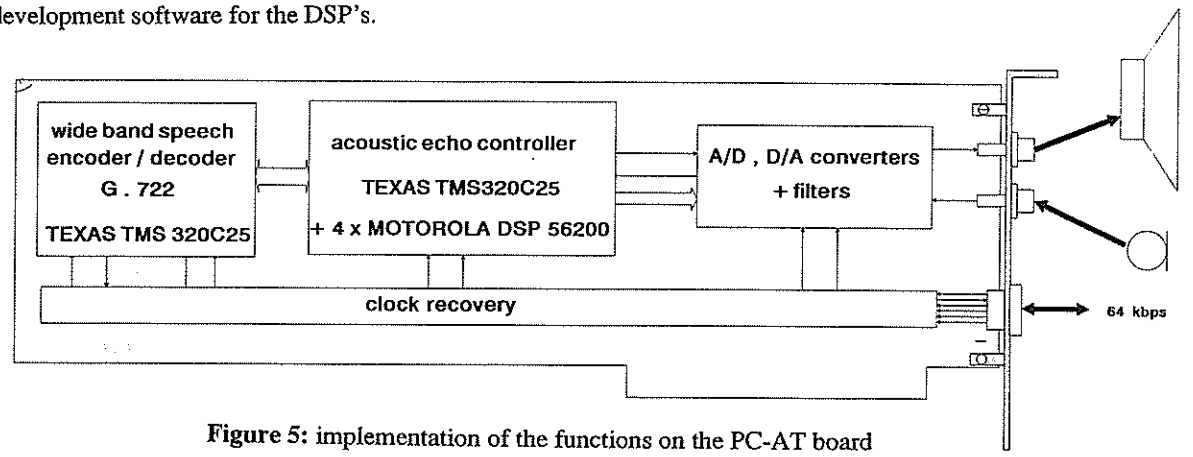


Figure 5: implementation of the functions on the PC-AT board

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

Considerations on Acoustic Echo Cancelling based on Realtime Experiments

Andreas von Zitzewitz, Siemens AG, Semiconductor Group, Munich, FRG

1 Introduction

Acoustic echo cancelling, or more precisely cancelling of acoustic echoes, is the accepted solution to realize the electrical decoupling of the loudspeaker and microphone signal of a hands-free telephone. According to Fig.1 the far end talker echo $y(t)$ is estimated by a so-called echo canceler (EC) and is substracted from the signal $y_n(t)$ which is the sum of $y(t)$ and the near end signal $n(t)$. It is well known

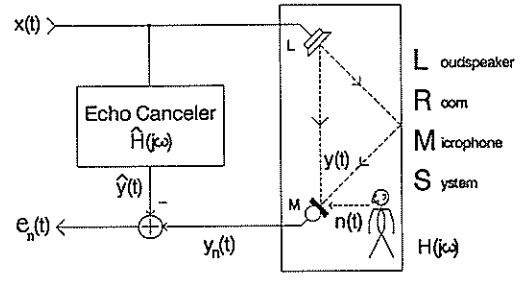


Figure 1: Cancelling of Acoustic Echoes

that the EC has to simulate the first 125 to 250 ms of the impulse response of the loudspeaker-room-microphone-system (LRMS) [1] in order to achieve 20 to 30 dB echo reduction.

Implying a sampling rate f_s of 8 kHz a process window of 125 to 250 ms is realizable by a transversal filter (TF) with N=1000 to 2000 taps. By using the normalized LMS algorithm [12] to adapt the filter coefficients, the computational power needed in terms of instructions per second to implement an adaptive TF (ATF) of such high order, e.g. in a TMS 320C25, is about 4N times the sampling rate and in this example 32 to 64 million instructions per second (MIPS).

Due to the availability of programmable digital signal processors that are able to calculate one tap in one cycle and with a cycle time of less then 100 ns, realtime experiments on EC of such high order becomes feasible by connecting only a few processors in parallel. Nevertheless, such multi-processor solutions, even implemented on a single chip by using advanced submicron CMOS technology are not acceptable in the price-sensitive telephone market. Furthermore, power consumption and convergence time have to be taken into consideration.

In Section 2 the question whether the high order TF can be replaced by a recursive structure of substantial lower order is discussed . In order to estimate the order of the recursive structure a heuristical method is used. In a second step a mixed structure is proposed and the results in comparison to the TF are given.

Another approach to reduce the computational effort and to achieve faster convergence is to divide the problem into subbands, as proposed by W.Kellermann, and this is presented in Section 3. A critical sub-sampled QMF-filter-bank, realized by bireciprocal lattice Wave Digital Filters (WDF) and the inherent aliasing effects are discussed and solutions are proposed.

Results based on realtime experiments are given in Section 4.

2 Structure of the Canceler

In the introduction it was mentioned that, due to the long impulse response of the LRMS, the filter order of the echo canceler (EC) must be chosen between 1000 and 2000 in order to achieve an echo reduction between 20 and 30 dB.

This fact leads to the obvious question, whether the long impulse response of the LRMS can be simulated by a recursive or infinite impulse response (IIR) filter of significant lower order.

Indeed, a direct and unbiased comparison between TF and recursive filters (RF) is not possible - it depends on the application you have in mind - but Rabiner and Gold give in [13] some numerical examples. They compare elliptic RF and minimax TF and from the given diagrams it can be seen that the order of the TF needs to be between a factor 3 and 30 higher than that of the RF to realize the specified low pass frequency responses. Taking the average we can probably use a gain factor of 8.

Furthermore, if the specification also includes a constant group delay the gain is reduced by a factor of 2. It is admitted that there is a great difference between the simulation of a LRMS and a low pass function but this is only a first attempt of estimation.

Another point of view is given by regarding the amplitude response of a typical LRMS over the full frequency range (0...4 kHz) which is shown in Fig.2. Two characteristics are remarkable: Firstly, besides some exceptions the amplitude fluctuations lie in a range between 10 and 20 dB. Secondly, the number of significant maxima is very high. Indeed, the number depends on the method of counting, but 80 maxima is found to be a reasonable lower bound.

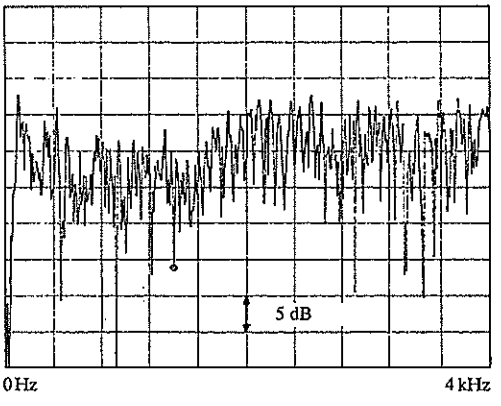


Figure 2: System Function of the LRMS

In order to estimate the order and the number of free parameters of an appropriate recursive filter we use the assumption, that a single maximum in the frequency domain can be simulated by a general second order recursive section. Such a section has 5 free parameters so that the over-all number of free parameters is about 400. If we cascade the second order section the 5th parameter (gain factor) can be summarized in a single parameter for the whole structure,

which results in a total number of $4\times80+1\approx320$ parameters.

Compared to the 1000 to 2000 free parameters of the TF the reduction factor in the number of parameters or number of coefficients that have to be adapted is 3 to 7 - the same order of magnitude that we have got by averaging the results of Rabiner and Gold. If we look at the bigger realization effort needed per parameter by using RF compared to TF nothing is gained. At this point it is emphasized that the foregoing discussion is quite speculative. If we assume, for example, a kind of "coupling" between the maxima and minima of the system function of the LRMS a further reduction might be achieved in the number of parameters needed. Based on such an assumption a mixed structure is proposed. To develop the structure we perform mental experiment by asking: How does the response of the LRMS arise from an impulse transmitted by the loudspeaker? Besides the direct sound, the microphone will receive echoes caused by reflections of the impulse at the ceiling, the walls and the furniture. An exponential decaying process can be simulated by

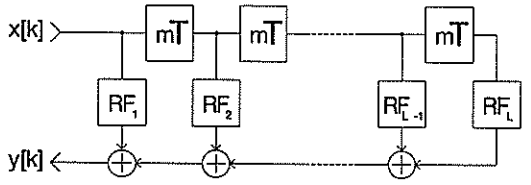


Figure 3: Transversal-Recursive-Filter

a simple first order recursive filter. To guarantee the stability of the recursive filter a passive and general first order Wave Digital Filter is used [15]. According to the above described model the TRF is built by simply taking the signals from a tap delay line, feeding the signals into the recursive sections RF_i, $i=1,2,\cdots L$, and adding the individual output signals. In other words, the TRF corresponds to the TF if the coefficients of the TF are replace by recursive sections. A further modification is given by replacing the single delays T (T is the sampling period) with m delays, $m=1,2,\cdots$, and by cascading an allpass section to the first order section. The intention of these modifications is a better adaptation to the group delay distortion of the individual echoes.

In realtime experiments several TRF configurations were tried out. In this paper the results of two versions are mentioned: Fistly, a structure with $L=30$ and first order allpass sections. Secondly, a structure with $L=64$ and no allpass section. In both cases $m=1,2$ and 3 were tried. The best result was achieved with the second structure and $m=2$. If we define the echo return loss enhancement as

$$ERLE = \frac{1}{N-N_0}\sum_{i=N_0}^{N-1} ERLE_{iL}, \qquad (1)$$

with

$$ERLE_{iL} = -10\log\frac{\sum_{n=0}^{M} e^2[iL-n]}{\sum_{n=0}^{M} y^2[iL-n]} \qquad (2)$$

($L\leq M$ for overlapping windows), then the achieved echo reduction is about 8 dB, comparable to a TF with $N=200$ taps. Considering the number of parameters or coefficients ($3L\approx N$) and the higher complexity in realizing the recursive section, compared to the TF, no reduction in terms of computational effort is achieved. Also notable is the observation, that by using the first structure (no cascaded allpass section) and $m=1$ the TRF always degenerates into a TF (the transfer function of the RFs is constant). This fact leeds us to the conclusion that the TF is the right structure to model the LRMS. In other words, the high order or the high number of coefficients of the TF is not only needed to model the long impulse response but also to correspond to the high number of free parameters of the LRMS which is manifested by the high sensitivity of the LRMS to changes in the room [1].

3 Echo Cancelling in Subbands

3.1 Principle

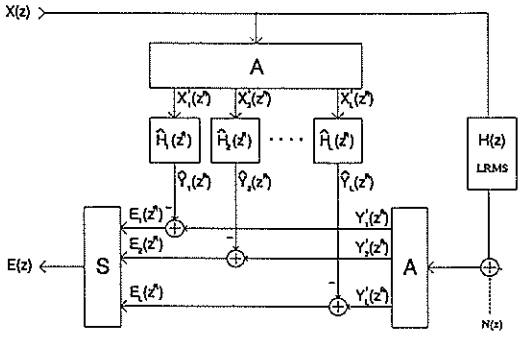


Figure 4: Echo Cancelling in Subbands

Another approach to reduce the computational effort and to achieve faster convergence is given by dividing the problem of cancelling of acoustic echoes into subbands, as proposed by Kellermann [10]. According to Fig.4 the main characteristics of this method are:

1. The input signal $x[k]\circ\!\!-\!\!\bullet X(z)$ and the echo signal $y[k]\circ\!\!-\!\!\bullet Y(z)$ are divided into L subbands with identical bandwidth by a so called analysis block A. (An unequal partitioning was not tested.)
2. In the subbands the sampling rate is reduced by a factor of $R=1(1)L$.
3. The echo cancelling is performed by L idependently working EC $\hat{H}_i(z^R)$, $i=1(1)L$.
4. The subband-error signals $e_i[m]\circ\!\!-\!\!\bullet E_i(z^R)$ are interpolated in the synthesis block S and combined to the residual error signal $e[k]\circ\!\!-\!\!\bullet E(z)$.

If the effort for realizing the filterbank is not taken into account and if the subband-EC is realized by an adaptive transversal filter (ATF) then the gain G in reducing the computational effort is [15]

$$G = R^2/L. \qquad (3)$$

The maximum $G=L$ is reached for a critical subsampled filterbank ($R=L$). In this paper a critical subsampled QMF-

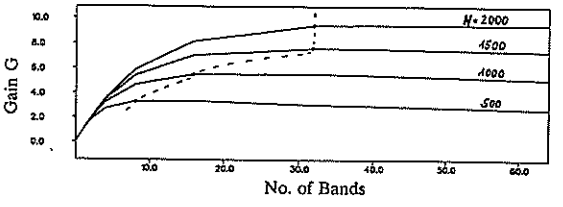


Figure 5: Gain Subband vs. Fullband

filterbank realized by bireciprocal Lattice-WDF is presented. By using this kind of filterbank the gain is approximately

$$G \approx \frac{4N}{k_{A,S}\cdot\log_2 L + 4N/L}, \qquad (4)$$

where $k_{A,S}$ is a parameter which depends on the dimensioning of the filterbank and on the number of subbands. For a constant $k_{A,S}$, (worst case value) the gain G is plotted

in Fig.5. The parameter N is the order of the (reference) TF in the fullband. It is emphazied that equation (4) and the diagram of Fig.5 are based on the instruction set of the TI-Processor TMS 320C25. If a more convenient processor architecture is used the WDF filterbank can be implemented more efficiently and the gain G will be higher.

3.2 Realization of the Filterbank with bireciprocal Lattice Wave Digital Filters

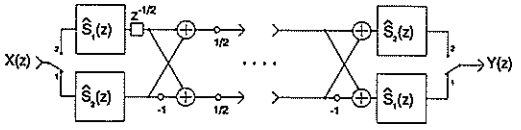


Figure 6: QMF-Section

There are a lot of possibilities to build a filterbank [2]. Very often a polyphase structure is used. Because of its modular structure and the inherent flexibility a QMF-filterbank is chosen. An efficient way to realize the QMF-sections is given by using bireciprocal Lattice WDFs [4]. The design of the WDF is supported by explicit formulas [7], a filter cataloque [9] and the powerful CAD environment FALCON [6]. Furthermore, if the recursive WDFs are properly designed stability will be guaranteed under all conditions [3]. In Fig.6 the signal flow graph of a QMF-section is shown. $\hat{S}_1$ and $\hat{S}_2$ are allpass functions. If D_1 and D_2 are the orders of $\hat{S}_1$ and $\hat{S}_1$, respectively, then the overall order of one section is

$$D = 2 \cdot (D_1 + D_2) + 1. \tag{5}$$

The factor 2 originates in the birecipricity of the system function. Besides the fact that the WDF runs at the lower rate, the implementation is very efficient. A further feature of the presented structure is that after linking the analysis and synthesis block the overall system function corresponds to an allpass function of the form

$$\frac{Y(z)}{X(z)} = \hat{S}_1(z^2)\hat{S}_2(z^2)z^{-1}. \tag{6}$$

This means that the aliasing terms are completely eliminated. According to Fig.4 this is only valid for the signal path of the near end signal $n[k]$ whereas in the path of the EC the elimination of the aliasing term requires further signal processing. This is descriped in the next section.

3.3 Elimination of Aliasing Errors

In order to study the above mentioned aliasing effects we calculate the output signal of the subband EC in the z-domain. By using the 2-band configuration of Fig.7 we get [15]

$$E(z) = H_\Delta(z) \cdot X(z) + H_A(z) \cdot X(-z), \tag{7}$$

with

$$\begin{aligned} H_\Delta(z) &= S_{11}^2(z)(H(z) - \hat{H}_o(z^2)) \\ &\quad -S_{11}^2(-z)(H(z) - \hat{H}_u(z^2)), \\ H_A(z) &= S_{11}(z)S_{11}(-z)(\hat{H}_u(z^2) - \hat{H}_o(z^2)), \\ S_{11}(z) &= \frac{1}{2}(\hat{S}_1(z^2)z^{-1} + \hat{S}_1(z^2)), \end{aligned} \tag{8}$$

where $H(z)$ is the transfer function of the LRMS, $\hat{H}_{o,u}(z)$ are the subband-EC and $S_{11}(z)$ and $S_{11}(-z)$ are the lowpass and highpass function of the QMF-block, respectively. Using this result it can be shown [15] that due to the aliasing term

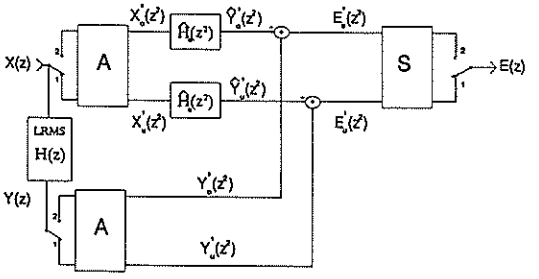


Figure 7: 2-Band Echo Canceller Configuration

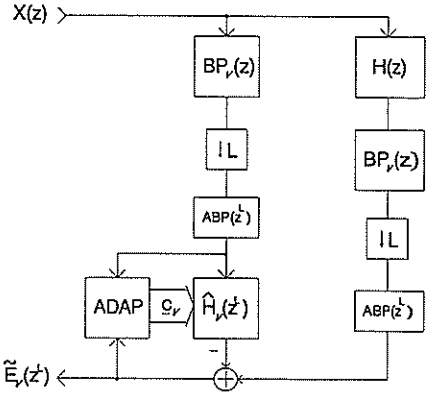


Figure 8: ν-th Subband with Anti-Alias-Bandpass

(8) there is no solution to the system idenfication demand $E(z)$=0. In reference [15] 5 methods to handle the aliasing problem are listed. 3 of them are mentioned here:

1. The aliasing area in the subbands is cut out by a so called Anti-Alias-Bandpass ABP(z^L). This is shown in Fig.8 where the ν-th subband of an L-channel filterbank is depicted. The effect of this operation is that the transmitted near end signal will show band gaps at the frequencies ω=$\nu\pi f_s/L$, ν=1(1)L-1.
2. A posteriori elimination (subtraction) of the aliasing term $H_A(z)X(-z)$ (aliasing signal $h_A[k]*x[k](-1)^k$). The bandpass function $S_{11}(z)S_{11}(-z)$ in $H_A(z)$ can be approximated by a simple equalizer [14]. It is emphasized that this kind of alias cancelling needs a lot of extra computational effort, especially because $H_A(z)$ has to be realized at the higher sampling rate.
3. Compensation of the aliasing term by using adaptive cross term filters as proposed by Gilloire and Vetterli [8]. But also here the additional effort is significant. Furthermore, problems in the convergence behaviour of the structure must be expected, although they are not discussed in [8].

This paper is restricted to the discussion of the method mentioned in item 1: The energy of the aliasing terms in the signal $E(z)$ (Fig.4 and 7) depends on the width of the transition band and on the stop band attenuation of the QMF-sections. The application of WDFs enables the design of extremly narrow transition bands and high stop band attenuations. The experimental results shown in the next section were obtained by using QMF-sections with 2.5 Hz transition bands and 60 dB stop band attenuation. The corresponding Anti-Alias-Bandpasses are symmetric and realize a band gap of 2×7 Hz (7 Hz is the width of the stop band). This band gap could not be detected in acoustic tests.

3.4 Additional Remarks

Besides considering the aliasing effects, the designer of a subband-EC has also to be aware that due to the downsampling the subband systems $H_\nu(z^L)$ have to simulate a noncausal impulse response [11],[15]. A fraction of the noncausal part can be realized by introducing a delay into the echo path. The length of this delay depends on the required accuracy and the allowed delay in telephone terminals.

Furthermore, the extremly narrow transition bands of the QMF-sections cause considerable group delay distortions around the subband boundaries which add a kind of reverberation to the near end signal n. If this (low) distortion is not acceptable a group delay equalization has to be performed in the QMF-sections and/or the Anti-Alias-Bandpasses.

4 Experimental Results

The tests were performed in a laboratory room of 50 m^2 size. The loudspeaker stood on the floor and the microphone on a desk at a distance of ca. 2 m. The used signal processor used had 16 bit accuracy for data (coefficient and state variables) and the A/D and D/A converters had a resolution of 12 bit.

Fig.9 shows the achievable ERLE for a fullband EC (top) and a subband EC with 8 bands (bottom). The graphs depend on the order of the TF where the parameter N_v stands for the number of delays inserted into the echopath to realize the noncausality of the subband EC.

The last two figures, Fig.10 and 11, show the convergence behaviour of a fulland EC with N=2000 (top) and an 8-band EC with N=250 in each band (bottom). In Fig.10 noise is used as input signal and in Fig.11 natural speech. The plots demonstrate that the subband EC converges significantly faster.

5 Conclusion

In the first part of the paper the question whether the high order (1000-2000) transversal filter (TF) can be replaced by a recursive structure of substantial lower order is discussed. In a first step the order of a general recursive filter (RF) is estimated by using the results from Rabiner and Gold and by analysing the characteristics of a recorded system response of a typical Loudspeaker-Room-Microphone-System. The result of this estimation is that the order of the RF must be higher than 300. Taking into account the additional effort for realizing a general RF, compared to a simple TF, nothing is gained. In a second step a mixed structure - a so called Transversal-Recursive-Filter (TRF) - is proposed. But also with this structure no reduction of the computational effort is achieved. Futhermore, it is notable is that in some cases the TRF degenerates into a TF. Hence a possible conclusion might be that the TF optimally describes the LRMS.

In the second part a critical downsampled Subband-EC is presented where the filterbank is efficiently realized with bireciprocal Lattice Wave Digital Filters. The inherent effects such as aliasing and group delay distortion are briefly discussed and simple solutions are proposed.

In the last part the results of realtime experiments are given.

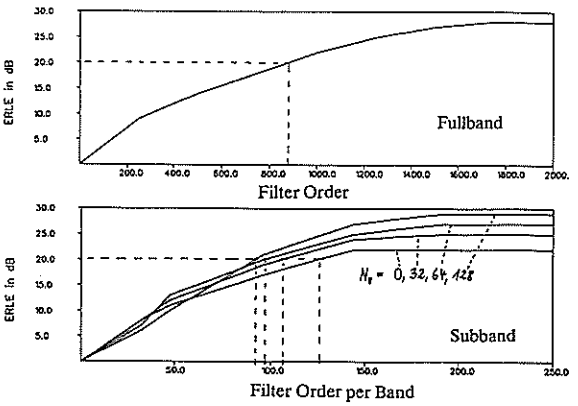


Figure 9: Achievable ERLE depending on the TF-Order

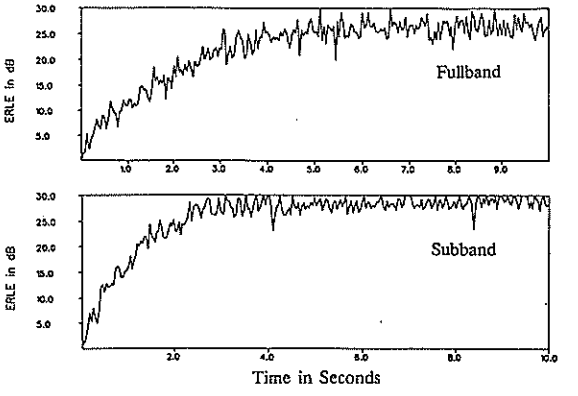


Figure 10: Achievable ERLE with Noise

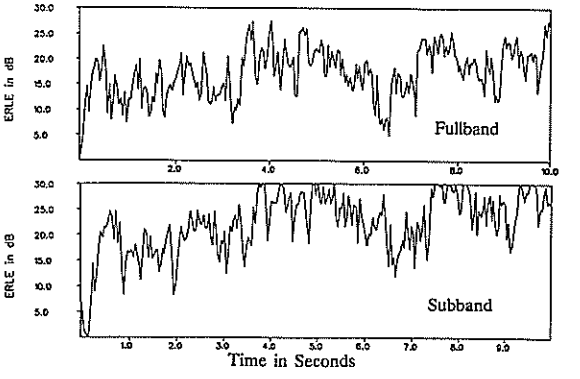


Figure 11: Achievable ERLE with natural Speech

References

[1] Becker, T.: "Untersuchungen zur Kompensation akustischer Echos", Dissertation, TH Darmstadt, 1986
[2] Crochiere, R.E. and L.R. Rabiner: "Multirate Digital Signal Processing", Prentic Hall, Inc., 1983
[3] Fettweis, A.: "Pseudopassivity, Sensitivity, and Stability of Wave Digital Filters", IEEE CT 19(1972)6
[4] Fettweis, A., et.al.: "Reconstruction of Signals after Filtering and Sampling Rate Reduction", IEEE ASSP 33(1984)4 , pp.893-902
[5] Fettweis, A.: "Wave Digital Filters: Theory and Practice", Proc. IEEE 74(1986)2, pp.270-327
[6] Gazsi, L.: "Reference Manual for FALCON Program", Ruhr-Univ. Bochum, July 1984
[7] Gazsi, L.: "Explicit Formulas for Lattice Wave Digital Filters", IEEE CAS 32(1985)1, pp.68-88
[8] Gilloire, A. and M. Vetterli: "Adaptive Filtering in Sub-Bands", ICASSP 1988, pp.1572-1575
[9] Güllüoglu, S.N.: "Diskrete Optimierung von Wellendigitalfiltern ...", Dissertation, Univ. Bochum, 1986
[10] Kellermann, W.: "Kompensation akustischer Echos in Frequenzteilbändern", Frequenz 39(1985)7/8
[11] Kellermann, W.: "Analysis and Design of Multirate Systems for Cancellation ...", ICASSP 1988
[12] Nagumo, J., A. Noda: "A learning method for system identification", IEEE AC 12(1967), pp.282-287
[13] Rabiner, L.R. and B. Gold: "Theory and Application ...", Prentice Hall, Inc., N.J., 1975
[14] Sauvagerd, U.: "A Ten-Channel Equalizer for Digital Audio Applications", IEEE CAS 36(1989)2
[15] Zitzewitz, A.v.: "Annäherung an das ideale Freisprechtelphon ....", Dissertation, Ruhr-Univ. Bochum, 1989

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A SYSTEM FOR ACOUSTIC ECHO CONTROL

José R. Casar Corredera and Gonzalo de Miguel Vela

Dep. Señales, Sistemas y Radiocomunicaciones
ETSI de Telecomunicación - UPM
28040 MADRID

A solution which combines the operation of a pair of adaptive FIR filters with a band-by-band center clipper is proposed to handle the problem of echo suppression in hands-free communication systems. Preliminary experiments show that the architecture is capable of effectively reducing to an acceptable level the acoustic and electrical echoes without significantly distorting the local speech signal.

1. INTRODUCTION

As in most voice communication systems, in hands-free / teleconference applications, the operational objective is to transmit and receive high quality speech in a simultaneous two-way conversation. However, the hands-free characteristic faces the system with some specific well known problems [1]. Some of these can be described in terms of the signals in Figure 1:

i) Room reverberation does not allow a perfect transmitted speech quality: the signal at the microphone is the result of convolving local speech with the near end speaker-to-microphone room impulse response.

ii) Acoustic echo (e_2) through far-end room is a factor which can severely degradate the conversation and result in feedback instability (acoustic coupling).

iii) Electrical echo (e_1) across the hybrid can also produce a feedback instability.

Both electrical and acoustic echoes could theoretically be compensated by using the well known concept of (adaptive) echo cancelling [2], [3], as it is practical in a variety of telecommunication applications. Being the concept directly applicable for the cancellation of the electrical echo, it appears to be not practical for acoustic echoes, at least in wide band applications: the room reverberation time (echo path duration) is typically very long, making the classical FIR-canceller inefficient, expensive and, above all, difficult to adapt: 2.800 taps would be necessary to duplicate a room with 200 ms. impulse response at a sampling rate of 14 Khz.

A number of alternative and complementary alternatives have been proposed which range from voice-switching based mechanisms and band-by-band cancellation to multimicrophone strategies (see [4] and [5]).

In this contribution we propose and preliminarly evaluate an echo control configuration which combines the action of two short FIR echo cancellers with a signal-driven center-clipping of the residual signal. To the authors knowledge,

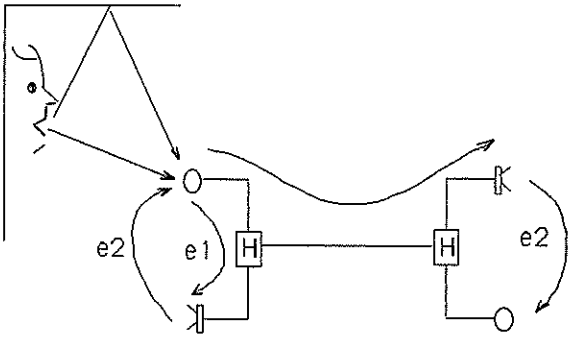


Figure 1: Acoustic and electrical paths in a Hands-Free System

Mitchell and Berkley were the first in proposing a center clipper to handle the problem of echo suppression in [6]. We now propose to combine that old idea with the even older idea of adaptively identifying an impulse response with an LMS adaptive FIR filter. When properly designed, the system is able to suppress the echoes without significantly distorting the local speech signal.

The rest of the paper is organized as follows:

Section 2 contains the qualitative description of the proposed architecture; the role of each block is there identified.

Section 3 reports on some parameter design issues and in particular describes the design and performance of a preliminary system.

Section 4 will contain the conclusions of the paper and recommendations for further work.

2. AN ECHO CONTROL SOLUTION

The solution to the echoes problem which we will discuss in this paper is based on the observation that cancelling the "maximum energy" section of the room impulse response alleviates the acoustic echo problem and, in particular, it avoids acoustic coupling. This means that a relatively short FIR canceller is able to reduce the acoustic problem to one of compensating the "reverberant tails" of the speech. That further compensation is performed by a center clipping processor in the proposed architecture, as it is shown in Figure 2.

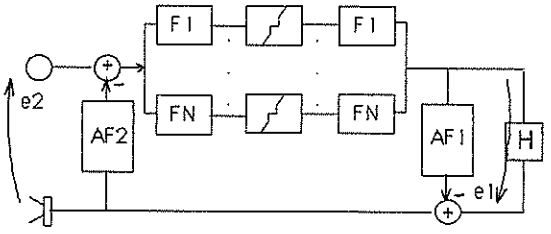


Figure 2: Basic Echo Control Architecture

i) The adaptive filter AF1 is a very short FIR filter intended to identify the electrical echo path across the hybrid transformer H and whose output therefore will ideally be a perfect reply of the echo e_1. These echo would be a major cause of ringing.

ii) The adaptive filter AF2 is a moderately short FIR filter, whose mission is to cancel the "most important" contribution of the acoustic echo e_2, which appears as the far-end speech signal at the loudspeaker propagates through the local room to the local microphone (it appears as a convolution with the, probably varying, room impulse response). As it is well known, when it is not compensated it can, on the one hand, produce a fatal acoustic coupling in the local end, and, on the other hand, as it propagates back to the far-end, manifests itself there as a disturbing medium-long delay echo.

Since a typical room impulse response is very long, it would be very expensive to exactly duplicate it with an adaptive filter. Moreover, the updating of the coefficients would not be possible in practical situations. However, a moderately short adaptive filter (AF2) is capable of preventing the feedback coupling and reducing the echo problem, by identifying and cancelling the most contributing section of the impulse response of the room. This varies from room to room, but it is basically contained in the first one third of the response.

iii) The residual echo (remaining after partial cancellation) is still to be further reduced by a band-by-band center clipper, which suppresses the low level reverberant tails. It is apparent that being the clipping level properly set, any residual echo would be ideally suppressed, whether the clipping operation is made either on the full signal or in a band-by-band scale. However this, the center clipping operation has to be made separately for adjacent frequency band signals (which are then postfiltered) to avoid the strong harmonic distortion which a direct clipping process would introduce to the local speech during double talking periods, when both the far-end echo and the local speech are picked up by the microphone.

When only the local speaker is speaking, the clipping level at each band should ideally be set to zero to avoid any residual distortion to be produced on his/her speech. Thus we are proposing to use a far-end signal-driven control of the clipping levels: the clipping level at each band should be set proportional to the level of the signal at the loudspeaker in that same band. In each band, the far-end residual echo signal is

clipped according to its expected level; and when no echo signal is present, the clipping threshold is automatically set to zero, thus allowing an undistorted local speech signal to be transmitted. It is evident that both the delays of the control loops and the proportionality gain constant have to be carefully designed.

iv) A conventional double-talking detector is still necessary to freeze the adaption of AF2 when the near-end speaker is active. Both AF1 and AF2 are LMS-based adaptive filters which have to be updated to produce respective minimum residual errors. AF1 should be updated only when there is no far-end signal and AF2, conversely, when there is no local speech. Therefore some kind of double talking decision should be devised.

3. A PRELIMINARY SYSTEM DESIGN AND SOME PERFORMANCE ISSUES

A number of different system parameter designs have been realized. By no means, the one described below is pretended to be the 'optimal'. It is known to be suboptimal both in performance and complexity. However it will allow us to illustrate some performance issues to seriously consider the design of a full, practical architecture.

The clipping system and its control is the most sensitive part of the whole suppressor. The results below are for a bank of 16 uniformly frequency spaced bandpass filters. The individual frequency responses together with the combined response are shown in Figure 3. The same bank of filters is used before and after the clippers. The sampling frequency is 10 KHz for a signal band between 200 Hz and 4800 Hz.

Figure 4 shows the percentage of distortion as a function of the clipping level for a sinusoidal signal. Distortion was measured as the ratio of the harmonic power over the fundamental power at the output. The clipping level is represented normalized with respect to the signal amplitude. The broken line curve represents the distortion for a sinusoid in the middle of one of the 16 bands and the continuous line curve represents the distortion for a sinusoid just in between two contiguous bands. It can be appreciated that a 10% clipping threshold produces a worst-case signal-to-distortion ratio of 43 dB.

A number of experiments were realized to measure the distortion for non harmonic signals, too. A measure of distortion (as a measure of relative distance between the input spectrum and the clipper output spectrum) was determined as a function of the normalized (with respect to the input signal power) clipping level, which was taken the same in each clipped band. The obtained profiles of distortion were similar to those computed for harmonic signals, but it is understood that at the end, only perceptual experiments, during double talking periods, would authorize the use of a given clipping threshold.

Finally, the signal-driven control of the clipping levels was simulated and evaluated to determine its actual operation, and gain a preliminary idea of which loop gain should be used which guaranteeing an acceptable suppression of the residual far-end echo still gives a basically undistorted local speaker' speech (during double talking periods).

Therefore, the full system was simulated (except the electrical echo path). The acoustics of a local room were simulated by using the image method described in [7] and the adaptive filter AF2 (which was taken of different lenghts in different experiments) led to converge before starting clipping control. An AR signal (with poles at 1 and 3 KHz) was taken to be the far-end signal and passed through the simulated room and the canceller. The residual echo thus formed was processed by the band-clipper. As was stated before, the clipping level at each band was controlled in real time by the local level of the far-end full signal at the corresponding band. The clipping level (CL) at time t, at band k (k=1,...16) was set at the value $CL(k,t)=\alpha\, E(k,t-T)$, where E(k,d) is the energy of the far-end signal in the band k, at time d. T is a conveniently selected delay parameter, which for the simulated stationary signal has not appeared to be critical, but which may be a determining factor when dealing with nonstationary signals as real speech.

Figure 5 shows the basic result of the experiments: it represents the ratio of clipper output power over clipper input power as a function of the normalized α^2 (The x-axis represents α^2 normalized by the power of the clipper input signal). Notice that $\alpha=0.93$ guarantees an additional suppression of 30 dB.

4. CONCLUSIONS AND FURTHER WORK

A revival of the idea of center clipping in combination with the classical LMS FIR canceller is suggested to probably be a reasonable solution to the difficult problem of acoustic echo compensation in hands-free teleconference applications. By no means the authors are claiming to have found the final solution. However, the results of a number of experiments (some of which have been presented here) seem to indicate that, when conveniently optimized, the architecture could be a serious candidate to consider, either in competition or cooperation with other solutions. A number of open issues are still to be considered. Not the least is its dynamic behaviour when faced with the nonstationarity of the speech and the variant characteristics of the room. Someone should realize some perceptual experiments, too.

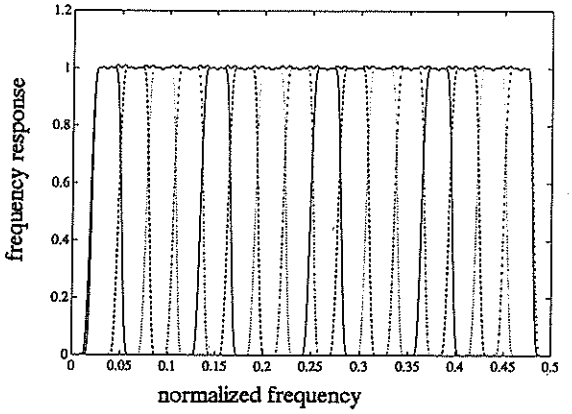


Figure 3: Filter bank frequency responses. Global response is also shown.

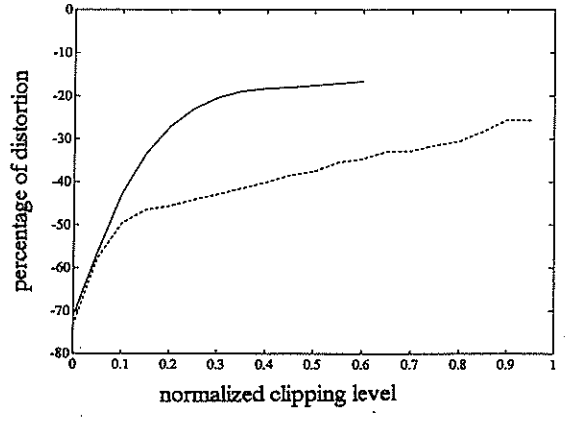


Figure 4: Band-by-band clipper distortion for sinusoidal inputs.Broken line: sinusoid in the middle of one band. Continuous line: sinusoid in between of two bands.

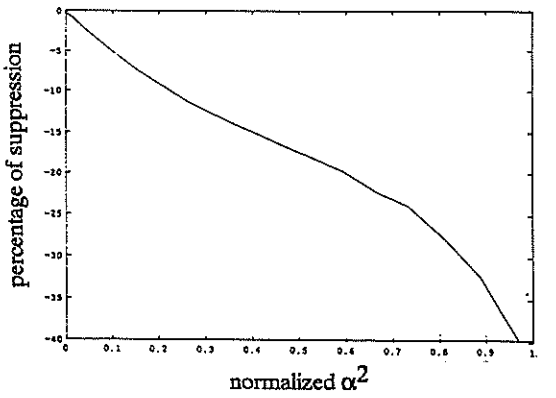


Figure 5: Additional residual echo suppression achieved by the clipper as a function of the squared control gain factor.

REFERENCES

[1] D.A. Berkley and O.M.M. Mitchell, Seeking the Ideal in Hands-Free Telephony, reprinted in [4], pp.215-222.

[2] M.M. Sondhi and D.A. Berkley, Silencing Echoes on the Telephone Network, Proc. IEEE, Vol 68, pp. 948-963, 1980.

[3] D.G. Messerschmitt, Echo Cancellation in Speech and Data Transmission, IEEE-SAC, Vol, SAC-2, pp. 283-297, 1984.

[4] J.S. Lim (Ed), Speech Enhancement, Prentice-Hall, pp. 211-281, 1983.

[5] Abstracts of Int. Work. Acoustic Echo Control, Berlin, RFA, 14-15 Sept 89, pp.18-52.

[6] O.M.M.Mitchell and D.A. Berkley, A Full Duplex Echo Suppressor Using Center Clipping, BSTJ, Vol.40, pp. 1619-1630, 1971.

[7] J.B. Allen and D.A. Berkley, Image Method for Efficiently Simulating Small-Room Acoustics, JASA, Vol. 65, pp. 943-950, 1979.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)
Elsevier Science Publishers B.V., 1990

An Iterative Algorithm for the Estimation of Echoes of a Loudspeaker-Room-Microphone System

Jürgen Cezanne

Institut für Netzwerk- und Signaltheorie, TH Darmstadt, Merckstr.25, D-6100 Darmstadt, FRG

This paper deals with the estimation of echoes of a multipath structure, e.g. the echo path of a hands-free telephone. The problem is formulated as a least squares approximation of a function by exponentials. Due to the complicate error surface there exists no exact method. Like Prony's algorithm the approach here substitutes the error surface by a quadratic form. Here, moreover its coefficients are adapted to the area where the minimum is currently searched. The quadratic form and its adaptation are constructed on the base of four requirements. Special modifications necessary due to the acoustic nature of the echo path will be described.

1 Introduction

In hands-free telephone systems the sound of the loudspeaker considerably influences the microphone signal via the so called echo path and thus leads to disturbing echoes or even howling. Signal processing methods that reduce these effects often require a model of the echo path. A simple model for its acoustic part is based on the assumption that the loudspeaker signal reaches the microphone via different paths through the room [2]. Thus its time continuous impulse response $\tilde{h}_R(t)$ is modeled by a train of weighted dirac impulses:

$$\tilde{h}_R(t) = \sum_{\mu=1}^{n} a_\mu \cdot \delta(t - t_\mu). \tag{1}$$

This paper adresses the estimation of the parameters a_μ and t_μ of this model on the base of measured data.

Let $h_{LRM}(k)$ be the sampled impulse response of the entire echo path (loudspeaker, room, microphone, antialiasing filter, etc.) and $h_{LM}(k)$ be the sampled impulse response of all electric components (loudspeaker, microphone, antialiasing filter, etc.). Both have been measured and are linearly related via the equation $h_{LRM}(k) = h_{LM}(k) * h_R(k)$. Thereby the sequence $h_R(k)$ contains the acoustic properties of the room and can be derived from $\tilde{h}_R(t)$ by the simulation theorem [1]:

$$h_R(k) = \int_{-\infty}^{\infty} \tilde{h}_R(\tau) \cdot \mathrm{sincp}\left(k - \frac{\tau}{T}\right) d\tau \tag{2}$$

$$= \sum_{\mu=1}^{n} a_\mu \cdot \mathrm{sincp}\,(k - d_\mu), \tag{3}$$

where T denotes the sampling time, $\mathrm{sincp}\,(x) := \sin(\pi x)/(\pi x)$, and $d_\mu := t_\mu/T$. Now the parameters a_μ and d_μ are to be estimated such that they fit the measured data $h_{LRM}(k)$ and $h_{LM}(k)$. Therefore the l_2-norm of the estimation error

$$\begin{aligned} e(k) &= h_{LRM}(k) - h_{LM}(k) * \sum_{\mu=1}^{n} a_\mu \cdot \mathrm{sincp}\,(k - d_\mu) \\ &= h_{LM}(k) * \left[f(k) - \sum_{\mu=1}^{n} a_\mu \cdot \mathrm{sincp}\,(k - d_\mu) \right] \end{aligned} \tag{4}$$

may be minimized, where $f(k)$ results from deconvolving $h_{LRM}(k)$ by $h_{LM}(k)$. Since $e(k)$ depends linearly on the a_μ, the minimization with respect to the a_μ is straightforward. Concerning the d_μ, however, it can be shown by a simple example that

$$Q(\underline{d}, f) := \min_{\underline{a} \in \Re^n} \sum_{k=-\infty}^{\infty} e^2(k) \tag{5}$$

with $\underline{a} := [a_1, a_2 \ldots a_n]$ and $\underline{d} := [d_1, d_2 \ldots d_n]$, may have infinitely many minima. Thus ordinary numerical algorithms searching minima iteratively cannot be used. Instead, an appropriate approximation of $Q(\underline{d}, f)$ has to be found.

2 Prony Method

Formulating (5) in the frequency domain yields

$$Q(\underline{d}, f) = \min_{\underline{a} \in \Re^n} \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| H_{LM}(e^{j\Omega}) \right|^2 \cdot \left| F(e^{j\Omega}) - \sum_{\nu=1}^{n} a_\nu \cdot e^{-j\Omega d_\nu} \right|^2 d\Omega \tag{6}$$

where $F(e^{j\Omega})$ is the Fourier transform of $f(k)$. This means that $F(e^{j\Omega})$ has to be approximated by a linear combination of exponentials. Such approximation problems are frequently solved by the Prony method. For example it has been applied to a very similar problem, namely for the estimation of echoes of a mobile radio channel [3]. The Prony method exploits the fact that a linear combination of exponentials can be seen as the homogeneous solution of a linear differential equation with constant coefficients:

$$\sum_{\mu=0}^{n} b_\mu \cdot j^\mu \cdot \frac{d^\mu}{d\Omega^\mu} \left(\sum_{\nu=1}^{n} a_\nu \cdot e^{-j\Omega d_\nu} \right) = 0 \qquad (7)$$

$$\text{with} \quad \sum_{\mu=0}^{n} b_\mu x^\mu := \prod_{\nu=1}^{n} (x - d_\nu) \text{ and } b_n = 1. \qquad (8)$$

This observation leads to Prony's algorithm. The coefficients b_μ of the differential operator in (7) are chosen such that

$$Q_P(\underline{b}, f) := \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| \sum_{\mu=0}^{n} b_\mu \left(j \cdot \frac{d}{d\Omega} \right)^\mu F(e^{j\Omega}) \right|^2 d\Omega \qquad (9)$$

becomes minimum, where $\underline{b} = [b_0, b_1 \ldots b_{n-1}]$ and $b_n = 1$. Subsequently the zeros of the characteristic polynomial (8) are evaluated and the amplitudes a_μ are determined by minimizing $\|e(k)\|_2^2$. The obvious advantage of the Prony method is that the objective function $Q_P(\underline{b})$ depends quadratically on the parameter $\underline{b}$. Hence, there exists a unique minimum that can be easily found. However, the difference between $F(e^{j\Omega})$ and the desired exponential sum is weighted by a data dependent differential operator in contrast to (6), where $|H_{LM}(e^{j\Omega})|^2$ is used as weighting function. This is important since $|H_{LM}(e^{j\Omega})|$ varies over a wide range of values due to the fact that it covers the transition regions of several filters.

3 Principle of a new algorithm

In this section an approximation $G(\cdot)$ of $Q(\underline{d}, f)$ is seeked that is as easy to minimize as the Prony objective function $Q_P(\underline{b}, f)$ and that provides for an appropriate weighting of the approximation error. In the following both intentions are formulated mathematically by four requirements (G1) - (G4) that will determine the function $G(\cdot)$ uniquely.

(G1) The simplicity of the Prony method relies on the fact that $Q_P(\underline{b}, f)$ depends quadratically on $\underline{b}$. To copy this idea equation (8) is interpreted as a general parameter transformation between $\underline{b}$ and $\underline{d}$. By that $G(\cdot)$ can be introduced as a function of $\underline{b}$ instead of $\underline{d}$. Now it is required that $G(\cdot)$ depends quadratically on $\underline{b}$, i.e., there exists a $(n+1) \times (n+1)$ matrix $\underline{G}(\cdot)$ such that

$$G(\underline{b}, f) = [\underline{b}^T, 1] \cdot \underline{G}(f) \cdot \begin{bmatrix} \underline{b} \\ 1 \end{bmatrix}. \qquad (10)$$

(G2) $Q(\underline{d}, f)$ and $Q_P(\underline{b}, f)$ are quadratic forms with respect to $f(k)$. Thus, it appears reasonable to prescribe that $G(\underline{b}, f)$ also equals a quadratic form of $f(k)$, i.e., there exist matrices $\underline{G}_{kl}$ such that

$$\underline{G}(f) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} \underline{G}_{kl} \cdot f(k) \cdot f(l). \qquad (11)$$

(G3) The functions $Q(\underline{d}, f)$ and $Q_P(\underline{b}, f)$ are nonnegative for every $\underline{d}$ and $f(k)$. They become zero, iff $f(k)$ is a linear combination of n echoes without noise and $\underline{d}$ and $\underline{b}$ correspond to the propagation times of the echoes. Obviously this property is essential for echo estimation by minimizing those objective functions. Therefore $G(\underline{b}, f)$ should also suffice these relations:

$$G(\underline{b}, f) \geq 0 \quad \forall f(k), \underline{b}, \qquad (12)$$

$$G(\underline{b}, f) = 0$$

$$\iff f(k) = \sum_{\mu=1}^{n} a_\mu \cdot \text{sincp}\,(k - d_\mu). \qquad (13)$$

(G4) The properties (G1) - (G3) do not determine the function $G(\underline{b}, f)$ uniquely. Thus, at least one more useful property, namely the appropriate error weighting, can be realized. Because of (G1), the difference between $Q(\underline{d}, f)$ and $G(\underline{b}, f)$ cannot be small for all $\underline{b}$. Therefore (G4) restricts to require that the difference is small in the vicinity of some point $\widehat{\underline{b}}$. This is achieved by demanding :

$$G(\underline{b}, f)|_{\underline{b}=\widehat{\underline{b}}} = Q(\underline{d}, f) \quad \forall f. \qquad (14)$$

As long as $\underline{b}$ operates in the vicinity of $\widehat{\underline{b}}$, error weighting will be appropriate. To keep $\underline{b}$ within this area, $\widehat{\underline{b}}$ is viewed as a parameter of $G(\underline{b}, f)$ that has to be kept tracking $\underline{b}$. To emphasize this fact $G(\cdot)$ is redefined as a function depending on $\underline{b}$,f and $\widehat{\underline{b}}$. Hence it will be denoted by $G(\underline{b}, f, \widehat{\underline{b}})$.

These considerations lead quite obviously to the following algorithm:

1. Calculate $f(k)$ by deconvolving $h_{LRM}(k)$ by $h_{LM}(k)$.

2. Choose a start vector $\widehat{\underline{b}}^{(0)}$ and assign $n := 0$.

3. Determine $\underline{b}^{(n)}$ such that $G(\underline{b}^{(n)}, f, \widehat{\underline{b}}^{(n)})$ becomes minimum.
4. Assign $\widehat{\underline{b}}^{(n+1)} := \underline{b}^{(n)}$
5. Is $\left|Q(\underline{d}^{(n)}, f) - Q(\underline{d}^{(n-1)}, f)\right|$ sufficiently small? If then determine the amplitudes a_μ by minimizing $\|e(k)\|_2^2$ and stop else increment n and continue with step 3.

It can be shown that there is only one function $G(\underline{b}, f, \widehat{\underline{b}})$ meeting the requirements (G1) - (G4). Defining $\widehat{d}_1, \widehat{d}_2 \ldots \widehat{d}_n$ as the roots of $x^n + \sum_{\mu=0}^{n-1} \widehat{b}_\mu x^\mu$ the function $G(\underline{b}, f, \widehat{\underline{b}})$ can be formulated by:

$$G(\underline{b}, f, \widehat{\underline{b}}) = Q\left(\widehat{\underline{d}}, f(k) \cdot \frac{\sum_{\mu=0}^{n} b_\mu \cdot k^\mu}{\sum_{\mu=0}^{n} \widehat{b}_\mu \cdot k^\mu}\right) \qquad (15)$$

4 Modifications

As will be pointed out in the next section the algorithm described above yields satisfying results as long as the frequency response $H_{LM}(e^{j\Omega})$ exhibits low-pass character. In practice, unfortunately, spectral components around $\Omega = 0$ are always cut off. So in this inner stopband $F(e^{j\Omega})$ does not contain any information about $h_R(k)$. This lack of information can be interpreted as a kind of additional noise. Recalling (15) this noise will be multiplied by $\sum_{\mu=0}^{n} b_\mu \cdot k^\mu / \sum_{\mu=0}^{n} \widehat{b}_\mu \cdot k^\mu$ and thus modulated parts of it fall into the passband of $H_{LM}(e^{j\Omega})$. These components considerably influence the value of $G(\underline{b}, f, \widehat{\underline{b}})$ and prevent the convergence of the algorithm. A careful analysis shows that an outer stopband around $\Omega = \pm\pi$ does not lead to such effects and that the difficulties arising from an inner stopband can be circumvented by allowing complex amplitudes for the echoes. That is, the desired objective function $Q(\underline{d}, f)$ has to be modified to

$$Q_{mod}(\underline{d}, f) = \min_{\underline{a} \in C^n} \frac{1}{\pi} \int_0^\pi \left|H_{LM}(e^{j\Omega})\right|^2 \cdot \cdot \left|F(e^{j\Omega}) - \sum_{\nu=1}^{n} a_\nu \cdot e^{-j\Omega d_\nu}\right|^2 d\Omega. \qquad (16)$$

Accordingly a modified approximation $G_{mod}(\underline{b}, f, \widehat{\underline{b}})$ may be formulated, which reads as

$$G_{mod}(\underline{b}, f, \widehat{\underline{b}}) = Q_{mod}\left(\widehat{\underline{d}}, f(k) \cdot \frac{\sum_{\mu=0}^{n} b_\mu \cdot k^\mu}{\sum_{\mu=0}^{n} \widehat{b}_\mu \cdot k^\mu}\right). \qquad (17)$$

This approximation has been applied to the iterative algorithm derived above and simulations show that it converges in the bandpass case. However, the amplitudes a_μ of the echoes and even some of the propagation times d_μ become complex. Moreover the tendency to summarize a nest of echoes by fewer echoes has increased, which might be due to the greater number of parameters. Primarily complex amplitudes and propagation times do not allow a physical interpretation. Rather they are seen as a parameter set that allows a simple description of the usual long LRM-impulse response $h_{LRM}(k)$.

5 Simulation Results

Firstly the algorithm described in section 3 has been tested with an artificially created echo sequence for $\widetilde{h}_R(t)$ (figure 1). Thereby a 21-tap low-pass filter with a cut-off frequency of $\Omega = 0.9\pi$ has been chosen for $h_{LM}(k)$. After convolving both a noise process has been superposed to simulate measurement noise with a signal to noise ratio of 20dB. The resulting sequence $h_{LRM}(k)$, the number n of echoes, and $h_{LM}(k)$ were given to the algorithm. Figure 2 shows the result of the deconvolution and figure 4 depicts the echoes found after 30 iterations. Finally the quality of the solution in every iteration has been measured by the so called system mismatch a_{SYS} defined by

$$a_{SYS}^{(n)} = 10\text{dB} \cdot \lg \frac{Q(\underline{d}^{(n)}, f)}{\|h_{LRM}(k)\|_2^2}.$$

Its behaviour is displayed in figure 3. For most of the steps the system mismatch decreases until the final value of about $-22dB$ is reached. Since $-a_{SYS}$ slightly exceeds the generated S/N of 20dB, the solution is a little bit closer to the data than the original echoes shown in figure 1. Roughly spoken, the algorithm has tried to detect some echo structure within the noise. Moreover, echoes having propagation times, which differ more than the sample time, have been resolved clearly, whereas the group of four echoes starting at about $t = 60T$ has been merged into three echoes. Therefore a small echo remains and appears at about $t = 72T$.

Secondly, the modified algorithm of the section 4 has been applied to measured data shown in figure 5 and 6. Thereby $h_{LRM}(k)$ is the impulse response of a simple acoustic environment that consisted of two panels, the main echoes of which can be discerned quite clearly. The echo estimate found after 30 iterations is depicted in figure 7. Thereby the absolute values of the complex amplitudes are displayed. One echo appeared at a complex arrival time. It has been inserted at the time given by the real part of the estimated propaga-

tion time. The system mismatch of this solution is $-23.5dB$.

6 Conclusions

The concept of substituting the complicate error surface by a quadratic nonnegative form has lead to encouraging results. Spectral weighting of the approximation error could be improved upon the Prony method. A drawback is that due to the bandpass character of the loudspeaker-microphone system the amplitudes of the echoes and some of their propagation times have to be allowed to become complex. Interesting questions for further research are whether the chosen parameter set for the quadratic form is optimum and whether it is possible to return to real echo parameters at the cost of a more complicate objective function.

References

[1] A. Papoulis, *Signal Analysis*. New York: McGraw-Hill, 1988.

[2] J.B. Allen and D.A. Berkley, "Image Method for Efficiently Simulating Small-Room Acoustics," *Journal of Acoustic Society of America*, Vol.65, No.4, April 1979, pp.943-950.

[3] U.Martin, R.Reng, H.W.Schüßler and K.Schwarz, "Determination of Wide Band Impulse Responses," *URSI: ISSSE 18.-20.September 1989*, Erlangen, pp.393-396.

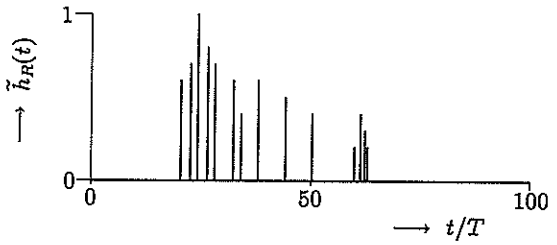


Figure 1: Artificially chosen echo sequence $\tilde{h}_R(t)$.

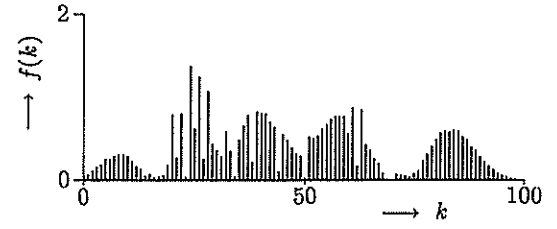


Figure 2: Result $f(k)$ of the deconvolution.

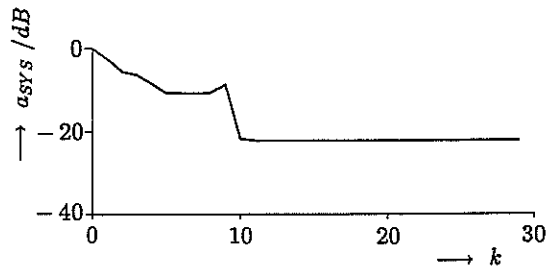


Figure 3: System mismatch a_{SYS} in dB.

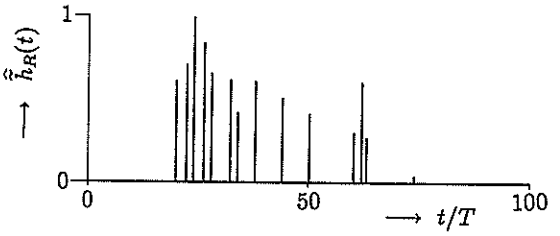


Figure 4: Estimated echoes after 30 iterations.

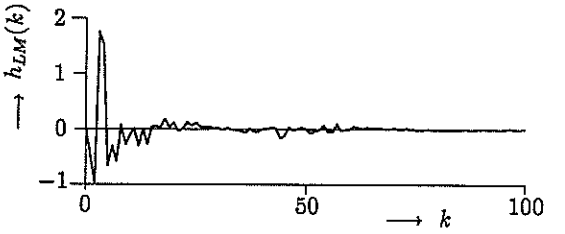


Figure 5: Measured LM-impulse response.

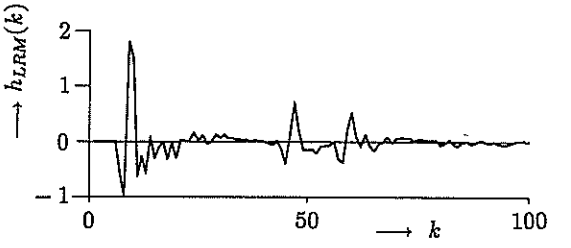


Figure 6: Measured LRM-impulse response.

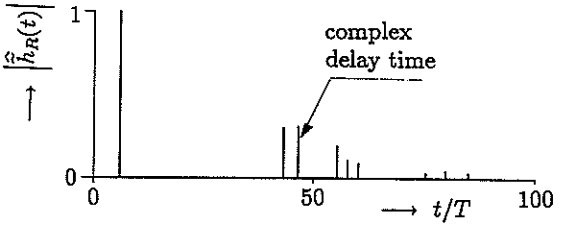


Figure 7: Absolute value $\left|\tilde{h}_R(t)\right|$ of the found echoes.

ACOUSTIC CANCELLATION OF ENGINE NOISE BY FAST ADAPTIVE IIR FILTERING

E. Masgrau, J.A. Rodríguez-Fonollosa

Dpto. Teoria de la Señal y Comunicaciones. U.P.C.
E.T.S.I. Telecomunicación. Apdo. 30.002
08080 Barcelona, Spain

ABSTRACT

Digital control of acoustic noise is an application area of digital signal processing with increasingly interest along the last years. This work tackles the reduction of the noise inside a motorcar cabin using a loudspeakers array drived by fast adaptive filtering of a reference noise source. Firstly, we propose an adaptive method for the deconvolution of the, in general, non-minimum phase radiation path by using an aditional filter. Secondly, we propose the use of some fast adaptive IIR lattice algorithms. They are very adequated for the very long impulse response of the acoustic system. Finally, a real scene is tested and the obtained results are analyzed.

INTRODUCTION

Passive industrial silencers work like acoustical low pass filters in noisy environments and, therefore, they work badly at low frequencies. Thus, digital acoustic noise cancellation has centred high interest and attention in the last years. Other applications related with acoustic cancellation, such as acoustic equalization of rooms, are object of broad analysis at present. The cabin of a motorcar represents a common scene for interesting and promising application of the refered acoustical processing.

In the figure 1 is shown a Widrow's cancellation scheme applied to an acoustic scene, i.e., a cabin car. The reference noise source is the motor of the car. It generates the primary noise inside the cabin across a complex acoustic and mechanic path. It can be modeled in a very simplified way with a pole-zero transference function G(z). The reference noise is used to feed one or several adaptive filters $W_m(z)$ in order to obtain an estimation of the primary noise. This estimation is then radiated in opposed-phase by the secondary sources or loudspeakers and the primary noise cancellation is obtained.

* This work has been supported by PRONTIC Grant nº 105/88

In the figure 2 a monochannel signal model of the acoustic system is shown. An additional

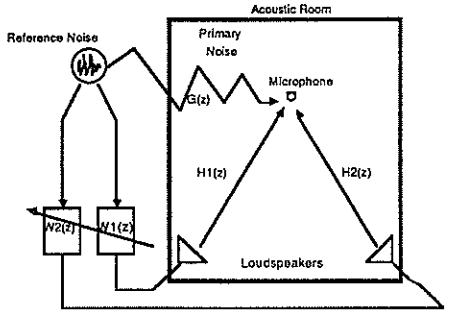


Fig. 1. Cancellation scheme

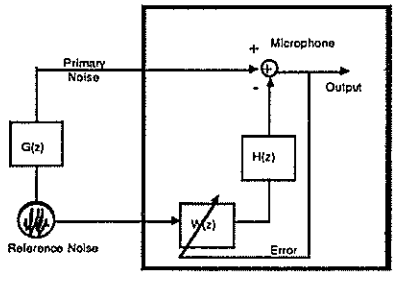


Fig. 2. Signal model.

microphone records the cancellation residual noise and it drives the adaptive filter adaptation. The theoretical optimum transfer function W(z) is G(z)/H(z), where G(z) is the above refered noise path model and H(z) is the radiation path model of the secondary source. This latter includes the loudspeaker response and the antialiasing and the interpolator filters response corresponding to A/D and D/A conversors in the secondary paths. Generally, a good approximation of W(z) can be obtained with a FIR filter, depending on the zeroes location of H(z).

Unfortunately, the acoustic radiation path H(z) usually has nonminimum phase character, and therefore, the optimum W(z) filter becomes unstable. A possible solution consists in the use of an aditional loudspeaker or secondary source [1], such as is shown in figure 1. In this case, the inverse filtering is carried out by the following equation:

$$H_1(z)\,W_1(z)+H_2(z)\,W_2(z) = G(z) \qquad (1)$$

where $H_1(z)$ and $H_2(z)$ must be relatively prime in the unit circle of z-plane. That is, they do not have any common zero outside or on the unit circle. On the contrary, the above problem is reproduced. Therefore, a good location selection for the secondary sources is a key factor in this context. The G(z) acoustic model presents, in general, a pole-zero structure and, in this case, a long impulse response for $W_m(z)$ is required; therefore, it is expected that the choice of IIR filters will be more adequated. Unfortunately, the typical IIR adaptive algorithms present a slow speed of convergence due to the multimodal error surface, and they can become unstable in the adaptation process.

In this paper, we use some fast IIR lattice algorithms developped by the authors [2,3]. These algorithms work very better than the classical IIR algorithms, specially in nearly unstability conditions, that is, when the magnitude of some lattice parcor coefficient is near to the unity.

In the steady-state, the converged $W_m(z)$ adaptive filters and its corresponding radiation paths $H_1(z)$ can be exchanged. Thus, the filter inputs are the filtered reference noises across the $H_1(z)$ paths, named $r_{1m}(n)$. The covariance matrix of this filtered signals defines the optimum Wiener solution of the $W_m(z)$ [4]. As it is well known, the eigenvalues spread of this matrix determines in part the convergence properties of the least squares adaptive algorithms. However, this statistic does not describe properly the convergence of the adaptive algorithms. It is only true if the convergence time-scale of these algorithms is very slower than the introduced delays of the radiation paths. This convergence dependence on the magnitude delay will be shown clearly later. In the adaptive algorithms, the correction terms of the coefficients update are determined by the filtered data $r_{1m}(n)$ instead of by its input signals [4].

MUTICHANNEL-MULTIERROR ADAPTIVE ALGORITHMS

The adaptive algorithms considered in this work are the multichannel-multierror extensions of the corresponding scalar algorithms, and they are described in detail in reference [5]. We consider a transversal FIR algorithms, the Normalized LMS, and several lattice IIR algorithms.

The multichannel-multierror LMS has been studied in reference [4]. The NLMS here used introduces the known normalization of the step-size by a variance estimation of the filtered data $r_{1m}(n)$.

The lattice IIR algorithms are the multichannel extensions of the scalar ones and they are described in [2,3,6]. They present a improved convergence with respect to the IIR direct form due to the known uncoupling properties of the lattice structure. Also, they make use of a corrected gradient term based in the inverse Hessian matrix. Many lattice IIR algorithms can be defined according to the accurate of the Hessian matrix calculation [6].

RESULTS

The empirical performance analysis of the proposed algorithms has been carried out in a simulated environment of a real scene. It consists in a cancellation system like that of the figure 1 working with real signals obtained in a motorcar

cabin operating in a anecoic room. Only the radiation paths $H_1(z)$ have been simulated. The primary and reference noises were recorded in an analogic tape, low pass filtered to 1,7 KHz and sampled to 4 KHz. Later, they were decimated to 500 Hz sampling frequency and the band pass signal was up to 230 Hz. It is the frequency range of the second order boom noise of the motor, the more harmful component of the cabin noise (typical frequencies range of 70-200 Hz corresponding to engine speeds between 2100 rpm and 6000 rpm).

In-phase motor and cabin signals were recorded to 2000, 3000, 4000, 5000 and 6000 rpm. After the digital sampling and processing above mentioned, they were used in the simulated cancellation scheme.

As a preliminar analysis, four cases have been considered. Three monochannel-monoerror cases with radiation paths $H(z)=1,z^{-5}$ and $(1+4z^{-5})$, and a bichannel-monoerror case with radiation paths $H_1(z)=z^{-5}$ and $H_2(z)=1+4z^{-5}$ have been analyzed. The first case is the ideal null delay. It determines the best cancellation performance. The rest of the cases present a nonminimum phase character.

For the second order cancellation analysis we have filtered selectively the non-cancelled (primary) noise and the cancelled cabin noise by means a running 32-points FFT. Then, the convergence time and the steady-state cancellation ratio were measured over the FFT output corresponding to the second order frequency. Also, the global cancellation in the full band (0-230 Hz) was calculated.

The convergence parameters of the algorithms were chosen to provide the same (roughly) convergence speed and a good cancellation ratios. As a preliminar performance analysis, the obtained results for 3000 rpm using the NLMS FIR and a lattice IIR algorithms are exposed in the Table I. The convergence time (T_C) in samples, the second order cancellation ratio (R_{SO}) in dB and the full band cancellation (R_B) in dB are shown. These cancellation ratios are calculated over 2500 samples beginning from the sample 500. The lattice IIR algorithm uses the LAR (Log Area Ratio) coefficients and a simplified gradient term where the Hessian matrix is diagonal (LAR-DH) [6]. The FIR order is 10 and the IIR order is 5 (5 zeroes and 5 poles).

In the Figure 3 is shown the second order energy evolution of the cabin (primary) noise and the cancelled cabin noise in the convergence period for the LAR-DH IIR lattice algorithm (case 4 of the Table I). In the Figure 4 is shown the steady-state spectra of the both primary and the cancelled noise for the same case. It can be observed the high second order cancellation ratio and the good tracking of another high noise component.

From the Table I the following facts can be remarked: the NLMS algorithm tracks very well the second order component of the noise, specially in the two first case; however, it provides a low cancellation ratio over the full band. The LAR-DH IIR lattice algorithm provides high second order cancellations and it improves the NLMS full band cancellation in 1,5-2 dB for the nonminimum phase H(z) cases. Also, the use of an additional adaptive filter in the nonminimum phase cases is shown very efficient in both NLMS and lattice-IIR algorithms. In both cases, the full band noise cancellation ratio is improved in 1,5-2 dB.

	T_C	R_{SO}	R_B	T_C	R_{SO}	R_B	T_C	R_{SO}	R_B	T_C	R_{SO}	R_B
FIR NLMS	50	20	6	100	14	3	50	16	3,5	50	16	4,5
IIR LATTICE	50	18	7	100	12	3	50	18	5	50	18	6,5
	$H(z)=1$			$H(z)=z^{-5}$			$H(z)=1+4z^{-5}$			$H_1(z)=z^{-5}, H_2(z)=1+4z^{-5}$		

Table I. Cancellation performance for NLMS FIR and LAR-DH IIR lattice algorithm.

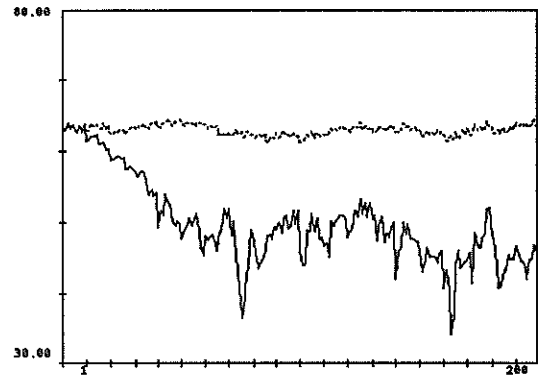


Figure 3.- Second order energy evolution of the primary noise (dashed line) and the cancelled primary noise (solid line) for the LAR-DH IIR lattice algorithm.

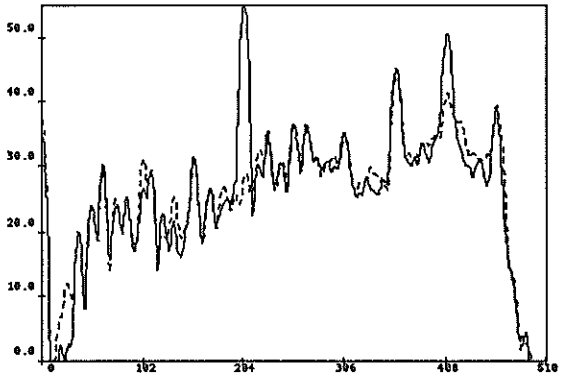


Figure 4.- Steady state spectra of both the primary noise (solid line) and the cancelled primary noise (dashed line) for the same case shown in figure 3.

CONCLUSIONS

In this paper, the acoustic noise reduction inside a motorcar cabin is tackled. We propose to use an additional adaptive filter to the nonminimum phase radiation path deconvolution. Also, we introduce some fast adaptive IIR lattice algorithms to improve the acoustic model tracking. Preliminary results have shown some large improvements in the noise cancellation using the proposed techniques. The additional filter provides an improvement of about 2 dB in the full band cancellation. The IIR lattice filter increases 1,5-2 dB the full band cancellation. By using both techniques we observe a improvement of about 3 dB. Also, the second order cancellation is improved in 2 dB in this case.

Acknowledgements

The authors acknowledge the assistant of the R. Leyva in obtaining the experimental results and the collaboration of the Laboratorio de Acústica of Seat-Volkswagen at Martorell (Spain) in getting the real signals.

REFERENCES

[1] M. Miyoshi, Y. Kaneda. "Inverse filtering of room acoustics". Trans. on ASSP, Vol. 36, pp. 145-152, February 1988.

[2] J.A. Rodríguez-Fonollosa, E. Masgrau. "A new algorithm for adaptive IIR filtering based on the log-area-ratio parameters". Proc. EUSIPCO-90, Barcelona (Spain), September 1990.

[3] J.A. Rodríguez-Fonollosa, E. Masgrau. "Spectral sensitivity an convergence rate in adaptive IIR filtering". IEEE Proc. ISCAS'90, pp. 1959-1962, New Orleans (USA), May 1990.

[4] S.J. Elliot et al. "A multiple error LMS algorithm and its application to the active control of sound and vibrarion". Trans. on ASSP, Vol. 35, No. 10, pp. 1423-1434, October 1987.

[5] E. Masgrau, J.A. Rodríguez-Fonollosa. "Multichannel-multierror adaptive algorithms for acoustic control noise". To be published.

[6] J.J. Shynk. "On Lattice-form algorithms for adaptive IIR filtering". IEEE Proc. ICASSP'88, pp. 1547-1557. New York (USA), 1988.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

PERFORMANCE COMPARISON OF ADAPTIVE ALGORITHMS FOR ACOUSTIC ECHO CANCELLATION

M. BERGER and F. GRENEZ

Telenorma S.A., 1120 brussels, Belgium
and
University of Brussels, 1050 Brussels, Belgium

1. INTRODUCTION

The acoustic coupling between the loudspeaker and the microphone is a severe problem in hands-free telephony. The resulting echo is send back to the source and degrades the quality of the communication. Moreover a loop is formed with the electric coupling across an imperfectly matched hybrid (4 wire/2 wire). This can cause the Larsen effect if this loop is becoming unstable.

Echo cancellation is based on the simulation of the acoustical path by an adaptive filter. The estimated echo signal is then substracted from the real echo.

The acoustic echo path is characterized by a very long impulse response and consequently very long adaptive filters are necessary.
Another difficulty comes from the properties of the speech signals : non-stationarity and spectral envelope.

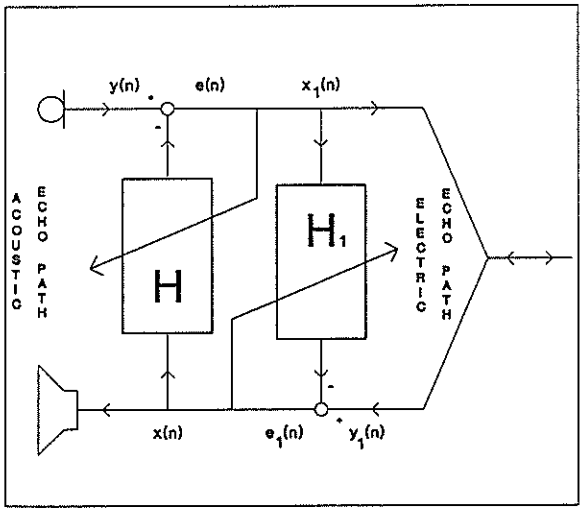


Fig. 1 Block diagram of acoustic and electric echo cancellation.

2. ALGORITHMS

A commonly used algorithm for the update of the adaptive taps is the normalized least mean square algorithm (NLMS) :

$$e(n+1) = y(n+1) - \underline{H}^t(n)\,\underline{X}(n+1)$$

$$\underline{H}(n+1) = \underline{H}(n) + \delta(n+1)\, e(n+1)\, \underline{X}(n+1)$$

$$\delta(n+1) = \delta' \,/\, N\, \sigma_x^2(n+1)$$

where

$\underline{H}(n+1)$ are the adaptive taps

$\underline{X}(n+1)$ is the input signal

$y(n+1)$ is the echo signal

$\delta(n+1)$ is the adaptation constant

N is the adaptive filter length

A variation of the NLMS algorithm can be achieved by reusing the data at each iteration. The quantity $e(n+1)$ is the a priori error. The a posteriori error can be computed by :

$$\varepsilon(n+1) = y(n+1) - \underline{H}^t(n+1)\,\underline{X}(n+1)$$

and can be used to update the coefficients a second time :

$$\underline{H}(n+1) = \underline{H}(n) + \delta(n+1)\, \varepsilon(n+1)\, \underline{X}(n+1)$$

This algorithm is refered as LMS2 in the figures.

Other types of algorithm are the fast version of the recursive least squares algorithms. One such FLS algorihm can be found in [4].

The LMS and related algorithms perform quite well with white noise, but degrade with colored signal like speech. Some solutions have been proposed for the whitening of the received signal [5].

We consider here an elementary technique of preemphasis. Preemphasis is used in LPC

analysis of speech to improve the quantization and stability properties.

It consists of a simple filter :

$$F_p(Z) = 1 - \mu\, z^{-1}$$

$$x'(n) = x(n) - \mu\, x(n-1)$$

which emphasizes the high frequencies before processing.

We use a value $\mu = 0.9$. At the output, the signal is deemphasized by $F_d(Z)$:

$$e(n) = e_p(n) + \mu\, e(n-1)$$

The structure is shown in fig. 2.

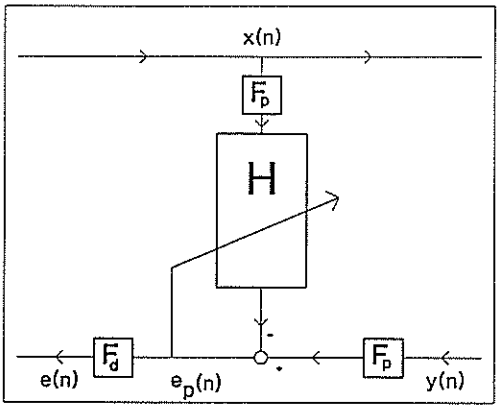


Fig. 2 Preemphasis.

3. RESULTS WITH ACOUSTIC ECHO

Results have been obtained for real speech signals and real echo in office room (fig. 3).

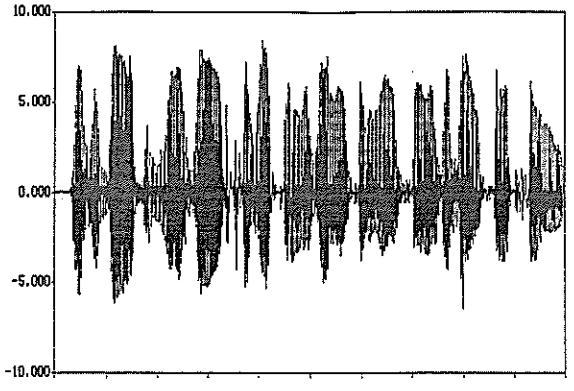
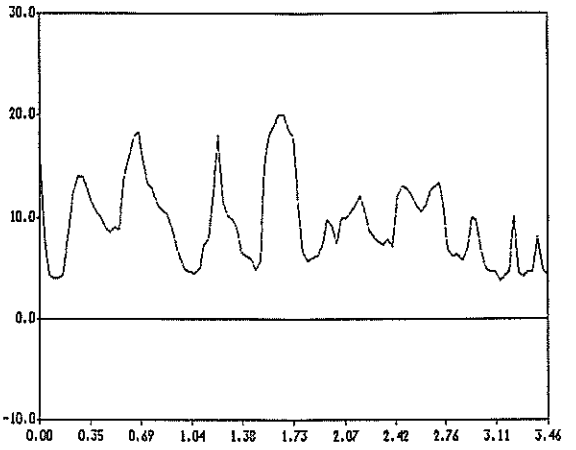


Fig. 3 Speech signal (3.6 Sec).



Signal Length (Sec)

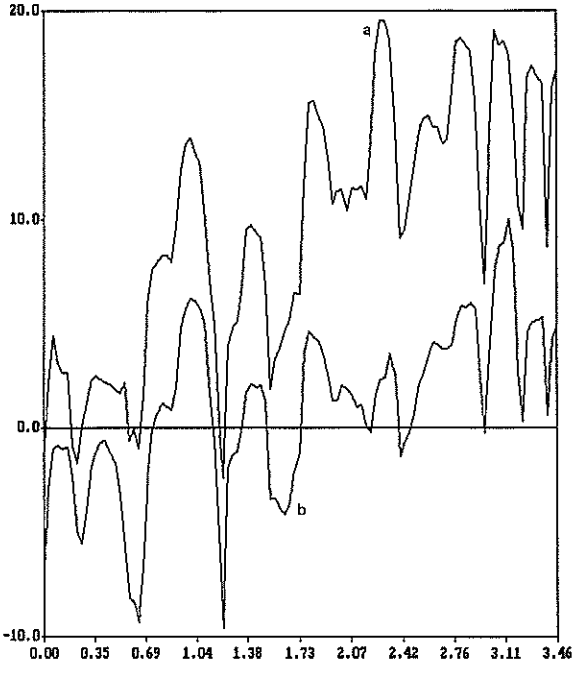
Fig. 4 ERL (dB) of the acoustic path.

The echo return loss (ERL) of the acoustic echo path is shown in fig. 4 :

$$ERL = 10\ Log(\ E[x^2]\ /\ E[y^2]\ )$$

The following figures give the adaptation gain or echo return loss enhancement (ERLE) :

$$ERLE = 10\ Log(\ E[y^2]\ /\ E[e^2]\ )$$



Signal Length (Sec)

Fig. 5 Effect of preemphasis on the NLMS algorithm. With preemphasis (a). Without preemphasis (b).

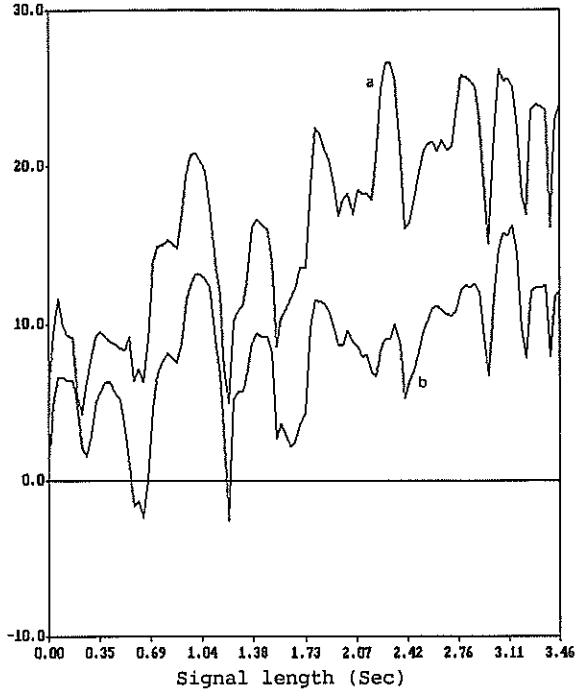


Fig. 6 Effect of preemphasis on the LMS2 algorithm. With preemphasis (a). Without preemphasis (b).

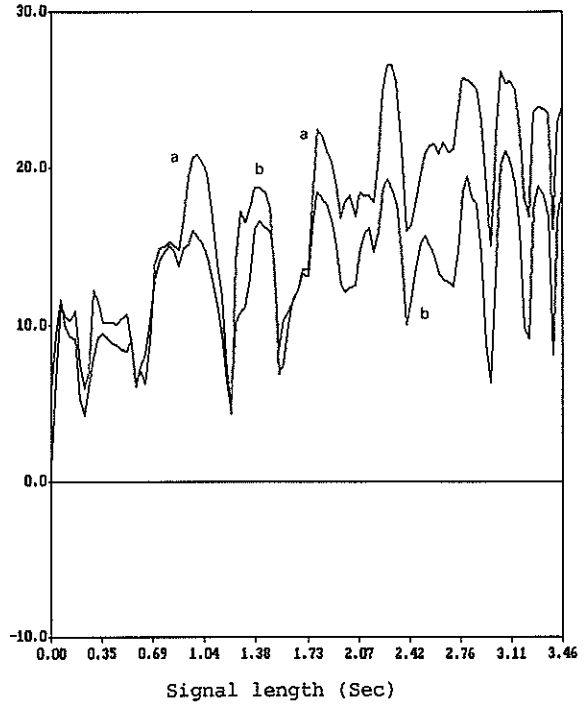


Fig. 7 Comparaison of LMS2 with preemphasis and FLS algoritms. LMS2 with preemphasis (a). FLS (b).

The filter length is N = 600 and the filter coefficients are zero at time t = 0.

From these results, it can be seen that the preemphasis improve the ERLE of about 10 dB, for both the NLMS and LMS2 algorithms.

On the other hand, the preemphasis has no effect on the FLS algorithm as could be expected. Further, there is no significant gain in using the FLS algorithm in place of the LMS2.

4. RESULTS FOR REAL TIME ELECTRICAL ECHO

A 4 wire/2 wire hybrid is characterized by an electrical echo. The impulse response of this echo is quite shorter. This echo can also be cancelled by an adaptive FIR filter (fig. 1). Such an echo cancellation has been implemented with a TMS 320C25 processor.

Real time measurements have been realized with a white noise [signal $x_1(n)$] of 19 dBm power in the telephonic band (fig. 8).

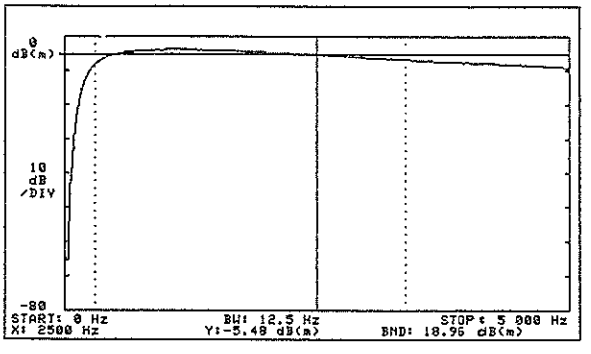


Fig. 8 White noise signal power.

An adaptative FIR filter with 100 taps gives the residual power of fig. 9, where the NLMS algorithm is used for adaptation. The steady state ERLE is about 47 dB.

For a 64 taps filter with the LMS2 algorithm (see fig. 10), the determined ERLE is about 50 dB.

But the most important advantage of the LMS2 algorithm is its convergence speed.

In fig. 10 and 11, we have the residual signal as a function of time, starting from all coefficients equal to zero. The convergence speed is about 6.8 dB/100 ms for the NLMS (100 taps). The speed has been multiplied by a factor of two for the LMS2 (64 taps) algorithm : 15.2 dB/100 ms.

This clearly shows the advantage of the LMS2 algorithm with double adaptation, even if the computational load is increased by a factor of two for the same number of taps, as better performances can be achieved with a shorter filter.

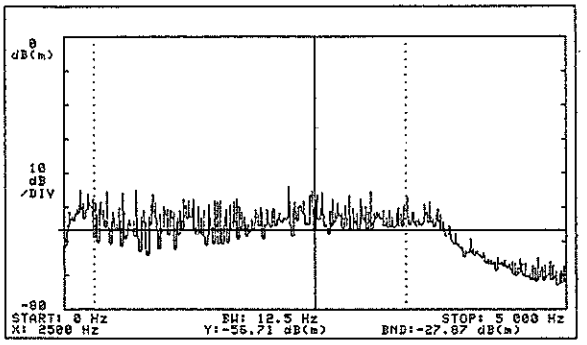


Fig. 9 Residual power with 100 taps and NLMS algorithm.

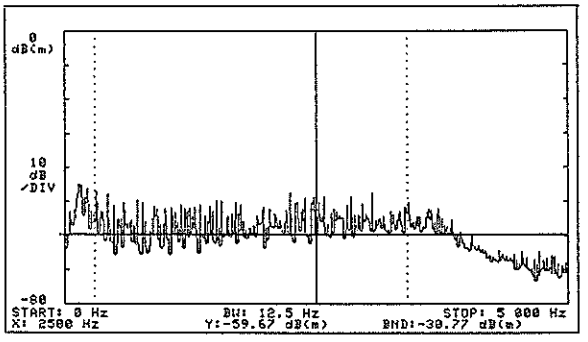


Fig. 10 Residual power with 64 taps and LMS2 algorithm.

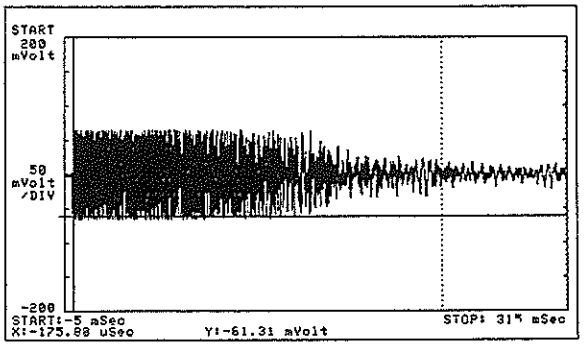


Fig. 11 Convergence speed of the NLMS filter (100 taps).

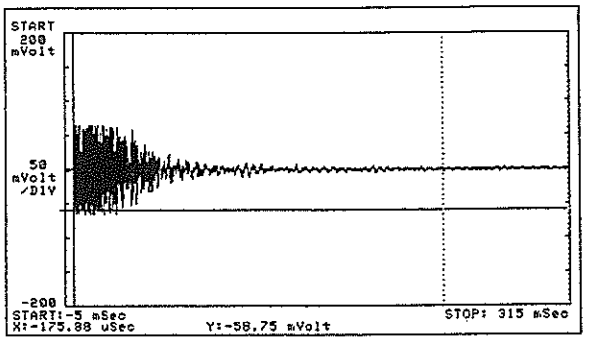


Fig. 12 Convergence speed of the LMS2 filter (64 taps).

REFERENCES

[1] M. BELLANGER, "Adaptive Digital Filters and Signal Analysis", Marcel Dekker, 1987.

[2] A. GILLOIRE and J.F. ZURCHER, "achieving the control of the acoustic echo in audio terminals", Proc. Eusipco 1988, pp. 491-494.

[3] W. ARMBRUSTER, "High quality hands-free telephony using voice switching optimised with echo cancellation", Proc. Eusipco 1988, pp. 495-498.

[4] M. BELLANGER, "Engineering aspects of fast least squares algorithms in transversal adaptive filters", Proc. ICASSP, 1987, pp. 2149-2152.

[5] H. YASUKAWA, S. SHIMADA, I. FUSUKAWA, "Acoustic echo canceller with high speed quality", Proc. ICASSP, 1987, pp. 2125-2128.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A NOVEL CFAR DETECTOR FOR MULTIPLE TARGET SITUATIONS IN SPATIALLY CORRELATED CLUTTER

Stelios D. Himonas

Department of Electrical Engineering
New York Institute of Technology
1855 Broadway, New York 10023

Mourad Barkat

Department of Electrical Engineering
SUNY at Stony Brook
Stony Brook, New York 11794-2350

In this paper, we consider the problem of constant false alarm rate, CFAR, detection for multiple target situations and spatially correlated clutter. We propose a novel detection scheme in which the interfering targets are censored by performing successive consistency tests on the degree of correlation between all the adjacent pairs of range cells in the reference window of the cell under test. As a result, two preliminary decisions about the contents of each range are made. These decisions are then combined according to the "AND" or the "OR" fusion rules in order to determine whether a particular cell contains interference or not. The cells that are decided to contain only clutter are combined to set the detection threshold against which the output of the test cell is compared in order to make a decision about the presence or the absence of a target in the cell under test.

1. INTRODUCTION

In radar automatic detection, the received signal is sampled in range by the range resolution cells. The clutter background in the cell under test is estimated by combining the outputs of the nearby resolution cells. The detection threshold is obtained by scaling the noise level estimate with a constant, T, to achieve the design probability of false alarm, α. One such detector is the Cell-Averaging CFAR, CA-CFAR, detector in which the reference samples are summed together [1].

In the presence of interfering targets, the probability of detection of the target in the test cell is seriously degraded. A number of signal processing algorithms in which the interfering targets are censored have been proposed in the literature [2-7]. In many practical situations, as in the case of whether clutter and chaff, the clutter samples may not be statistically independent but they may be partially correlated [3]. In [8], Himonas and Barkat considered the problem of partially correlated and identically distributed samples.

In this paper, we consider the problem of constant false alarm rate, CFAR, detection for multiple target situations and spatially correlated clutter. We propose a novel detection scheme in which the interfering targets are censored by performing successive consistency tests on the degree of correlation between all the adjacent pairs of range cells in the reference window of the cell under test. As a result, two preliminary decisions about the contents of each range cell are made. These decisions are then combined according to the "AND" or the "OR" fusion rules in order to determine whether a particular cell contains interference or not.

* This work was supported in part by the National Science Foundation under Research Award No. ECS - 8907176.

2. CFAR DETECTOR WITH CONSISTENCY TESTS

Consider a radar operating in a multiple target environment, and assume that L pulses hit the targets. The received signal, $r(t)$, is processed by the L channels, as shown in Figure 1. In the ith channel, the signal is processed by an in-phase and a quadrature channel. $r(t)$ is sampled every τ seconds to yield $2M+1$, $M = N/2$, in-phase samples and $2M+1$ quadrature samples. τ denotes the transmitted pulse width, and the $2M+1$ samples represent the returns from the $2M+1$ range cells. The outputs of any two different channels are assumed to be statistically independent. The Lx$(2M+1)$ in-phase samples and the Lx$(2M+1)$ quadrature samples are then processed by the censoring algorithm (which is described later) that censors the samples which may correspond to interfering targets. The censoring procedure yields the samples that have not been censored, as shown in Figure 2. Note that, in Figure 2, the censoring procedure is shown to output $(2M+1)$ samples. The samples that have not been censored are set equal to zero. The outputs of the cells surrounding the test cell are then summed together to yield an estimate, q, of the noise level in the test cell. This estimate is scaled by a threshold multiplier, T, in order to achieve the desired probability of false alarm, P_F. As in the case of the GCA-CFAR detector [8], in order to maintain CFAR T is calculated based on an estimate of the correlation coefficient between the clutter returns of two adjacent range cells. The output of the test cell, q_{M+1}, is compared to the adaptive threshold Tq in order to make a decision about the presence or the absence of a target in the test cell.

The input to the censoring device are observations from the L channels, as shown in Figure 1. The sequences of observations are

$$\mathbf{x}_i = [x_{i1}\ x_{i2}\ \cdots\ x_{i,2M+1}]^T\ ,\ i = 1, 2, ..., L \qquad (1a)$$

and

$$\mathbf{y}_i = [y_{i1}\ y_{i2}\ \cdots\ y_{i,2M+1}]^T\ ,\ i = 1, 2, ..., L\ . \tag{1b}$$

Assuming Swerling II targets embedded in zero mean Gaussian noise, the in-phase and the quadrature samples from the jth range cell, $j = 1,2,...,2M+1$, of the ith channel, $i = 1,2,...,L$, are observations from zero mean Gaussian random variables with variance μ_j. Under hypothesis H_1, that is, in the presence of a target in the jth range cell

$$x_{ij} = n_{ij} + u_{ij} + f_{ij} \tag{2a}$$

while under hypothesis H_0, that is, in the absence of a target in the jth range cell

$$x_{ij} = n_{ij} + u_{ij}\ . \tag{2b}$$

n denotes the thermal noise process ($\mu = 1$), u denotes the clutter process ($\mu = C$), and f denotes the target return ($\mu_j = I_j$). The clutter process is assumed to be a first order Markov process with correlation matrix A, that is,

$$[\mathrm{A}]_{kl} = \rho^{|k-l|}\ ,\ k,l = 1,2,...,2M+1\ . \tag{3}$$

The actual value of the correlation coefficient, ρ_a, between the outputs of any two adjacent range cells, x_{ij}, $x_{i,j+1}$, depends on the contents of the jth and the $(j+1)$th range cells. That is, four possible cases arise, H_{00}, H_{01}, H_{10}, and H_{11}. H_{00} means that both cells contain only noise, while H_{11} means that both cells contain a target return. H_{01} means that the jth cell contains only noise while the $(j+1)$th cell comprises a target. H_{10} means that the jth cell contains a target while the $(j+1)$th cell contains only noise. It can be shown in a straightforward manner that

$$\rho_a \approx \begin{cases} \rho\ , & H_{00} \\ \rho(1+\bar{I}_j)^{-1/2}\ , & H_{10} \\ \rho(1+\bar{I}_{j+1})^{-1/2}\ , & H_{01} \\ \rho[(1+\bar{I}_j)(1+\bar{I}_{j+1})]^{-1/2}\ , & H_{11} \end{cases} \tag{4}$$

where $\bar{I}_j$ denotes the signal to total noise (thermal noise plus clutter) ratio. In order to obtain equation (4), we have assumed that the clutter power is much greater than the thermal noise power, that is, $C >> 1$. We observe from equation (4) that in the absence of interfering targets from two adjacent range cells the correlation coefficient is high. In the presence of targets, however, the actual correlation coefficient is reduced significantly since it is inversely proportional to the target(s) SNR(s). Hence, using this fact, we propose the following algorithm to censor the interfering targets.

The censoring procedure starts by computing the maximum likelihood estimates, MLEs, of the variances in all the range cells, and the MLEs of the correlation coefficients between all adjacent pairs of range cells. The MLE of the variance of the output of the jth cell, q_j, is obtained to be [8]

$$q_j = \frac{1}{2L}\sum_{i=1}^{L}(x_{ij}^2 + y_{ij}^2)\ ,\ j = 1, 2, ..., 2M+1\ . \tag{5}$$

Note that the ith estimate of the variance is determined by averaging the $2L$ observations (L in-phase channels and L quadrature channels) obtained from the jth cell. The MLE of the correlation coefficient between the jth and the $(j+1)$th cell, $\hat{\rho}_j$, is obtained to be [8]

$$\hat{\rho}_j = \frac{1}{2L}\sum_{i=1}^{L}\frac{(X_{ij}X_{i,j+1} + Y_{ij}Y_{i,j+1})}{Q_j^{1/2}Q_{j+1}^{1/2}} \tag{6}$$

Observe that the estimates $\hat{\rho}_M$ and $\hat{\rho}_{M+1}$ involve the cell under test. Therefore, an overall estimate of the correlation coefficient between any two adjacent reference range cells, $\hat{\rho}_0$, is obtained by averaging the $2M-2$ estimates of ρ without including the two estimates between the test cell and its adjacent cells. That is, $(2M-2)$ estimates are used to obtain

$$\hat{\rho}_0 = \frac{1}{2M-2}\left(\sum_{j=1}^{M-1}\rho_j + \sum_{j=M+2}^{2M}\rho_j\right)\ . \tag{7}$$

In order to determine whether the jth cell may contain an interfering target return, the censoring procedure performs the following tests on the two pairs of cells that contain the jth cell, that is, the $(j-1,j)$th pair and the $(j,j+1)$th pair of range cells. First, consider the $(j-1,j)$th pair. The estimated value of the correlation coefficient between the $(j-1)$th and the jth cells, $\hat{\rho}_{j-1}$, is compared to a correlation coefficient threshold, $T_\rho\hat{\rho}_0$. If $\hat{\rho}_{j-1} > T_\rho\hat{\rho}_0$, we decide that both cells contain noise (clutter plus thermal noise) and the test ends. That is, we decide that $D_1(j-1) = D_2(j) = 0$, where $D_1(j-1)$ and $D_2(j)$ denote the two preliminary decisions about the $(j-1)$th and the jth cells, respectively. If $\hat{\rho}_{j-1} < T_\rho\hat{\rho}_0$, three possible cases arise, H_{10},H_{01},H_{11}. That is, we perform the test

$$\hat{\rho}_{j-1} \underset{H_{10},H_{01},H_{11}}{\overset{H_{00}}{\gtrless}} T_\rho\hat{\rho}_0 \tag{8}$$

where T_ρ is chosen so that the design false alarm probability at the preliminary decisions is maintained at the desired value α. Next, we need to determine which of the three cases H_{01},H_{11},H_{10} is true. In this case, we perform the classical CFAR test in which we compare the variance estimate (average power) of the $(j-1)$th cell to a scaled variance estimate of the jth cell. That is, we perform the test

$$q_{j-1} \underset{H_{01},H_{11}}{\overset{H_{10}}{\gtrless}} T_v q_j \tag{9}$$

where T_v is selected so that the probability of falsely declaring only one interfering target, under hypothesis H_{11}, does not exceed a desired value α. If $q_{j-1} > T_v q_j$ then we decide H_{10} and the algorithm stops. The decisions are $D_1(j-1) = 1$ and $D_2(j) = 0$. Otherwise, the two other possible cases (H_{01}, or H_{11}) will arise. We then perform the test

$$q_j \underset{H_{11}}{\overset{H_{01}}{\gtrless}} T_v q_{j-1} \tag{10}$$

If $q_j > T_v q_{j-1}$ then we decide H_{01} and the consequent decisions are $D_1(j-1) = 0$ and $D_2(j) = 1$. Otherwise, H_{11} is true and $D_1(j-1) = D_2(j) = 1$.

After the preliminary decisions $D_1(j-1)$ and $D_2(j)$ are made, we consider the next pair of adjacent range cells, the jth and the $(j+1)$th range cells, and carry the test in a similar manner. This test will then yield the two preliminary decisions $D_1(j)$ and $D_2(j+1)$. Thus, by considering the $(j-1,j)$th and the $(j,j+1)$th pairs of adjacent range cells, we obtain two preliminary decisions about the contents of the jth range cell, $D_2(j)$ and $D_1(j)$, respectively. These two preliminary decisions are then combined according to the "AND" or the "OR" fusion rules in order to decide whether the jth cell contains an interfering target or not. If the jth cell contains an interfering target, it is then censored.

The remaining (non censored) reference samples are then added together to yield an estimate, q, of the noise level in the cell under test. The output of the test cell, q_{M+1} is compared to a scaled version of q, Tq, according to

$$q_{M+1} \underset{H_0}{\overset{H_1}{\gtrless}} Tq \tag{11}$$

in order to make a decision about the presence or the absence of a target in the test cell.

The parameters of the proposed detector are determined as follows. The scaling constant T_v is selected so that the probability of falsely declaring only one interfering target under hypothesis H_{11} does not exceed a desired value, α. That is,

$$Pr(Q_j > T_v Q_{j-1} \,|\, H_{11}) = \sum_{k=0}^{L-1} \binom{L+k-1}{k} \frac{T_v^k}{(1+T_v)^{L+k}} \leq \alpha. \tag{12}$$

where we have assumed that $I_j = I_{j-1}$. The correlation threshold multiplier, T_ρ, is chosen so that, in the test of expression (8), the probability that $\hat{\rho}_{j-1}$ is less than $T_\rho \hat{\rho}_0$ under hypothesis H_{00} does not exceed a desired value α. That is,

$$Pr(\hat{\rho}_{j-1} < T_\rho \hat{\rho}_0 \,|\, H_{00}) \leq \alpha \ . \tag{13}$$

A mathematical analysis to obtain an expression for $Pr(\hat{\rho}_{j-1} < T_\rho \hat{\rho}_0 \,|\, H_{00})$ is extremely cumbersome. Consequently, in order to calculate the value of T_ρ computer simulations were conducted. Finally, the threshold multiplier T which is chosen to achieve the design probability of false alarm, α, in the test cell is calculated based on the estimate of ρ, $\hat{\rho}_0$, by using the procedure suggested in [8].

3. RESULTS AND CONCLUSIONS

The detection performance of the proposed detector has been evaluated by means of computer simulation, as shown in Figures 3 and 4. The detection performance when no censoring scheme is employed (without consistency tests) is also shown for comparison. The parameter ρ, shown on the figures denotes the value of the correlation coefficient used to generate the data for conducting the simulations. The processor assumes no knowledge of the value of ρ, but it estimates ρ and then computes the threshold multiplier, T, as described earlier. We observe that by processing the preliminary decisions according to the "OR" fusion rule, we obtain a robust performance. However, with the "AND" fusion rule while we still have a good performance by using the consistency tests, a slight capture effect is observed. For heavily correlated clutter returns, the "AND" fusion rule yields robust performance as well.

REFERENCES

[1] H.M. Finn and R.S. Johnson, "Adaptive Detection Mode with Threshold Control as a Function of Spatially Sampled Clutter Estimates", *RCA Review*, Vol. 29, No. 3, pp. 414-464, 1968.

[2] M. Barkat, S.D. Himonas and P.K. Varshney, "CFAR Signal Detection for Multiple Target Situations", *IEE Proceedings*, Part F, No.5, pp. 193-209, 1989.

[3] H. Rohling, "Radar CFAR Thresholding in Clutter and Multiple Target Situations", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-19, No. 4, pp. 608-621, 1983.

[4] J.T. Rickard and G.M. Dillard, "Adaptive Detection Algorithms for Multiple Target Situations", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-13, No. 4, pp. 338-343, 1977.

[5] J.A. Ritcey, "Performance Analysis of the Censored Mean Level Detector", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-22, No. 4, pp. 443-454, 1986.

[6] S.D. Himonas and M. Barkat, "A robust radar CFAR detector for multiple target situations", *Proceedings of the 1989 IEEE/AESS National Radar Conference*, pp. 85-90, 1989.

[7] B. Barboy, A. Lomes and E. Perkalski, "Cell-Averaging CFAR for Multiple Target Situations", *IEE Proceedings*, Vol. 133, Pt. F, No. 2, pp. 176-186, 1986.

[8] S.D. Himonas and M. Barkat, "On Adaptive CFAR Detection in Spatially Correlated Clutter", *IEE Proceedings*, Part F, to appear.

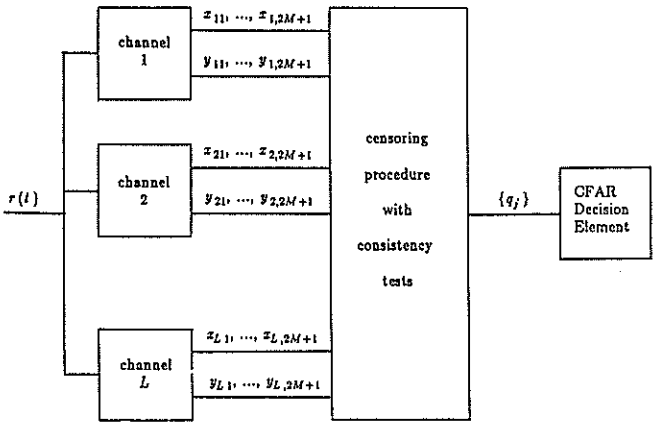


Fig. 1. Processing of the received observations

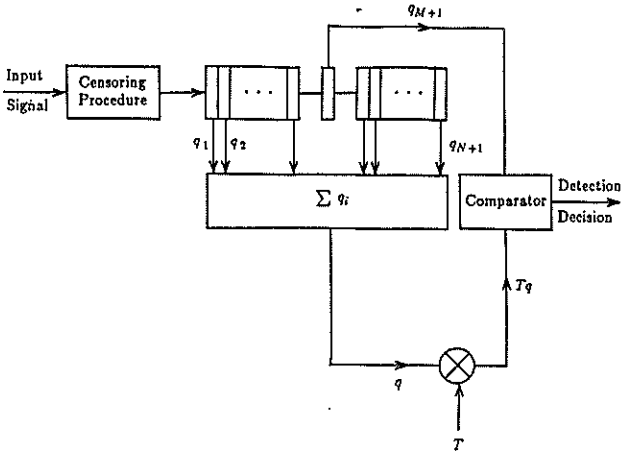


Fig. 2. CFAR Decision Element

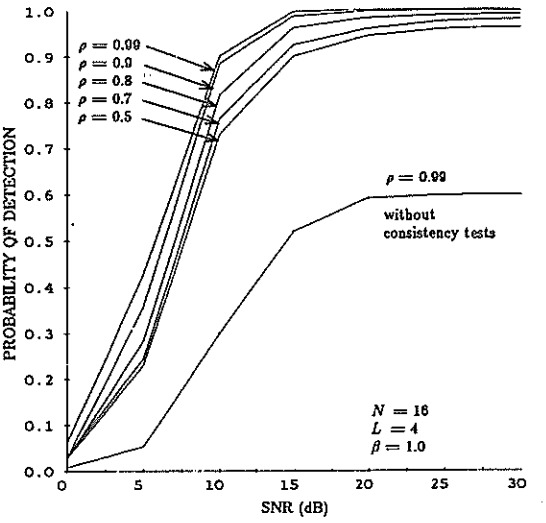


Fig. 3. Simulated probability of detection versus SNR(dB) when one interfering target is present; censoring by "AND" fusion rule.

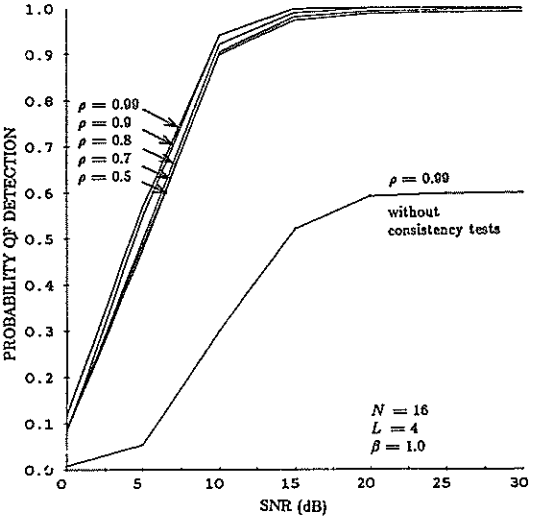


Fig. 4. Simulated probability of detection versus SNR(dB) when one interfering target is present; censoring by "OR" fusion rule.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

TIME–FREQUENCY PROPERTIES OF SIX CLASSES OF CONGRUENTIAL FREQUENCY HOP SIGNALS

JEROME R. BELLEGARDA

IBM RESEARCH
T. J. WATSON RESEARCH CENTER
YORKTOWN HEIGHTS, NEW YORK 10598

Frequency hop pulse train signals used in applications such as coherent multi-user or multi-beam echolocation must be selected on the basis of good time-frequency characteristics in *both* auto- and cross-ambiguity domains. We present a comparative analysis of five classes of such signals, all algebraically constructed via three types of congruences on a finite field. A sixth class, of slightly different type, is also developed to underscore the generality of the congruential frequency hop framework. For each class, an uniform upper bound is placed on the entire cross-ambiguity function surface, and bounds are placed on the amplitude of spurious peaks in the auto-ambiguity functions. These bounds depend on time-bandwidth product and code length exclusively, which allows for a meaningful comparison across classes. Typical aspects of time-frequency behavior are discussed, and examples of representative auto- and cross-ambiguity functions are shown to illustrate the time-frequency properties of the signals considered.

I. INTRODUCTION

To perform reliable target and/or channel scattering function measurements, frequency hop pulse train signals used in high resolution radar or sonar systems must be chosen such that their auto-ambiguity functions exhibit a narrow "thumb tack" and adequately low spurious peaks [1]. In contrast, in applications like multiple-access communication, the emphasis is on designing a sequence of frequency-hopped coded waveforms with mutually small cross-correlation functions for any time-frequency shift, as intercode rejection requires the entire cross-ambiguity function surface to be uniformly bounded across the signal set considered [2]. Since a basic trade-off is involved between these two objectives, difficulties arise in situations where both are desirable, such as multi-user radar or multi-beam sonar imaging [3], [4].

To exemplify one particular design compromise, we recently proposed a class of frequency hop signals constructed upon an extension of the theory of quadratic congruences [5]. This class was specifically defined for easy comparison with so-called "*full*" frequency hop pulse trains (such as those based upon, e.g., linear congruences (LC) [6], or Welch–Costas arrays (WC) [8]), meaning that all the available frequency channels are occupied. Subsequently [9], we extended the analysis to "*non-full*" frequency hopping patterns as well, thus enabling comparisons with, e.g., ordinary quadratic congruential (QC) signals [7], for which the elements of each signal set do not span all available frequencies.

The purpose of this paper is to further consolidate the above approach into a unified congruential frequency hop analysis framework, and compare a few representative classes in terms of their time-frequency properties. The next section specifies all signal sets considered and the associated correlation properties. They are used in Section III to place bounds on the amplitude of spurious peaks in each auto-ambiguity function, as well as a uniform upper bound on the cross-ambiguity function between any two elements of each signal set. This leads to a discussion of the design trade-offs involved and an analysis summary for six different classes. Finally, Section IV considers the issue of class selection and illustrates the time-frequency behavior of selected frequency hop signals.

II. SIGNAL SETS

We consider a rectangular pulse of length T seconds divided into N equal segments. For the purpose of analysis, we choose $N = P - 1$, where P is an odd prime, which in turn implies that N is even. (From a practical point of view, this restriction entails little loss of generality.) Let B be the (radian) bandwidth available, so that each signal considered occupies a time-bandwidth product of approximately $2BT$. In each segment of the pulse (time slot) we place one, and only one, subpulse written as:

$$s_k(t) = p\Big(t - \frac{T}{N}k\Big)\, e^{\,j(\omega_k t + \theta_k)}, \qquad \text{for } 1 \le k \le N, \quad (1)$$

where $p(t) = 1/\sqrt{T}$ if $-T/N \le t \le 0$ and 0 otherwise, and:

$$\omega_k = \omega_0 + y_k \frac{B}{N}, \qquad \text{for } 1 \le k \le N, \quad (2)$$

with some suitable ω_0. The (ordered) set of integers $\{y_k\}_{k=1}^{N}$ is obtained through a *placement operator*, which can be of exponential, polynomial, or rational type, as detailed below.

For WC signals, the placement is of *exponential* type:

$$y_k = \big[aR^k\big]_P \qquad \text{for } 1 \le k \le N, \quad (3)$$

where R is any primitive root of P [11], and a belongs to the set $J_P = \{1, 2, \ldots, P-1\}$. (Throughout this paper, expressions of the form $y = [x]_P$ should be read, "y is congruent to x, modulo P.") Since P is an odd prime, J_P forms an abelian group under multiplication modulo P; hence, $\bar{J}_P = J_P + \{0\}$ is a Galois field of order P [11]. An example of WC code array for $P = 11$, $R = 2$ and $a = 1$ is shown in Fig. 1.a.

A placement of *polynomial* type can be written as:

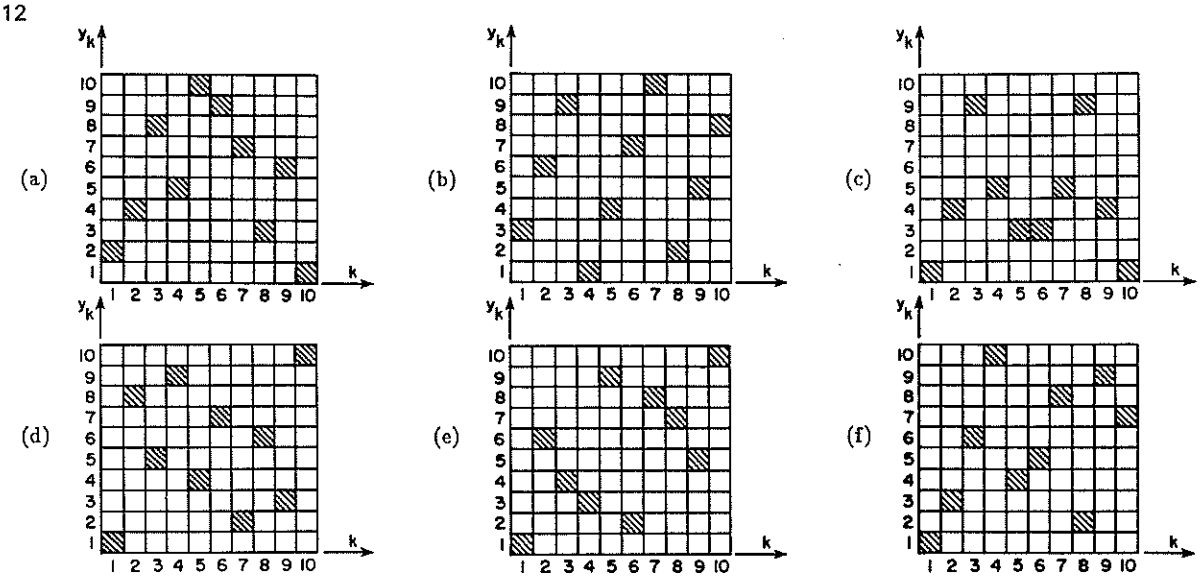


Fig. 1. $N \times N$ Grid Representation of Placements $\{(k, y_k) : 1 \le k \le N\}$, for $P = N + 1 = 11$.
(a) Welch–Costas, $R = 2$, $a = 1$; (b) Linear Congruential, $c_1 = 3$; (c) Quadratic Congruential, $c_2 = 1$;
(d) Cubic Congruential, $c_3 = 1$; (e) Hyperbolic Congruential, $c_1 = 1$; (f) Extended Quadratic Congruential, $a = 1$, $b = 2$.

$$y_k = \left[\sum_{i=0}^{Q} c_i k^i\right]_P \qquad \text{for } 1 \le k \le N, \tag{4}$$

for some integers c_i members of the set $\bar{J}_P = \{0, 1, \ldots, P-1\}$. The non-negative integer Q defines the order of the placement: the values $Q = 1$ and $Q = 2$ obviously correspond to LC and QC signals, respectively; $Q = 3$ yields the class of cubic congruential (CC) signals [10], etc. Fig. 1.b-d depicts a typical pattern for each of the LC, QC, and CC classes.

Finally, a placement of *rational* type is given by:

$$y_k = \left[\sum_{i=0}^{Q} \frac{c_i}{k^i}\right]_P \qquad \text{for } 1 \le k \le N, \tag{5}$$

where the division by k^i is to be interpreted on the finite field, i.e., $x = [1/k^i]_P \iff [xk^i]_P = 1$. For $Q = 1$, the placement (5) corresponds to the class of hyperbolic congruential (HC) signals, a member of which is represented in Fig. 1.e [10].

Variations on the above three types can also be included in the unified framework. For example, the EQC placement operator:

$$y_k = \begin{cases} \left[a\,\dfrac{k(k+1)}{2}\right]_P & \text{if } 1 \le k \le \dfrac{N}{2}; \\ \left[b\,\dfrac{k(k+1)}{2} + (a-b)\,\dfrac{P^2-1}{8}\right]_P & \text{if } \dfrac{N}{2} \le k \le N; \end{cases} \tag{6}$$

where a and b are in J_P, is a piecewise version of (4). Note that this definition slightly modifies the original development of [12], albeit without affecting the signal properties. Fig. 1.f depicts a typical EQC code array.

The placement operators defined in (3)–(6) do not necessarily induce a permutation over $\bar{J}_P$. To wit, notice in Fig. 1.c the symmetry with respect to the center of the array, which demonstrates a many-to-one correspondence over J_P. Physically, this means that we may send two pulses through certain frequency channels (1, 3, 4, 5, and 9 in this example) while other channels never get utilized (hence the name non-full).

As for the above classes, QC signals are always non-full, CC signals are full only if $P = 3L + 2$ [10], while WC, LC, EQC, and HC signals are always full.

While (3)–(6) determine the signals themselves, what determines the size of the corresponding auto- (respectively cross-) ambiguity function(s) is the set of *placement differences* for any time-frequency shift of one signal with respect to itself (respectively another signal). If y and $\tilde{y}$ denote two placements belonging to the same class, the *placement difference function* for $1 \le k \le N$ can be expressed as:

$$(\tilde{y}\Delta y)_{k,i,m} = \left[\tilde{y}_{k+i} + m - y_k\right]_P \qquad \text{for } 0 \le i, m \le N-1, \tag{7}$$

where i and m correspond to any horizontal and vertical shift. Through (2), this can be translated into any time delay and Doppler shift, respectively. A coincidence, or "hit," is a time-frequency shift for which one element of each of the two sequences $\{y_k\}$ and $\{\tilde{y}_k\}$ occupy the same time-frequency position. In general, the set of placement differences is a subset of the finite field (or complete residue system) $\bar{J}_P$; a hit is simply the observation that the number 0 is included within the subset for a particular time-frequency shift.

The maximum number of hits that can occur simultaneously for a given (cyclic) shift is obtained by solving $(\tilde{y}\Delta y)_{k,i,m} = 0$ in the cross-ambiguity case or simply $(y\Delta y)_{k,i,m} = 0$ in the auto-ambiguity case, for $i \neq 0$. Let H_C and H_A, respectively, be the resulting values. Since the maximum cross-correlation between two subpulses is known to be, for $0 \le |\tau| \le T/N$ [12]:

$$|C_{k\ell}(\tau)| = \begin{cases} \dfrac{1}{N} - \dfrac{|\tau|}{T} & \text{if } k = \ell; \\ \dfrac{2N}{BT|y_k - y_\ell|} & \text{if } k \neq \ell; \end{cases} \tag{8}$$

the total hit contribution to the entire maximum cross-correlation between two code words is bounded by H_j/N, $j = A, C$.

For the non hit contribution (corresponding to $y_k \neq y_L$), let us set $k = \ell + i$. Then the issue is to determine all the (non-zero) values taken on by $(\tilde{y}\Delta y)_{\ell,i,m}$ (in the cross-ambiguity case) or simply $(y\Delta y)_{\ell,i,m}$ (in the auto-ambiguity case) as $1 \leq \ell \leq N$. Using the general methodology developed in [12], it is fairly straightforward to realize that this quantity always assume *at least* $\lceil N/H_j \rceil$ $(j = A, C)$ distinct values over J_P, where $\lceil \cdot \rceil$ denotes the next largest or equal integer. Clearly, as far as cross-correlation is concerned, the worst possible case is when these $\lceil N/H_j \rceil$ distinct values are the only ones taken, in which situation the placement difference has to go H_j *times* through a residue system congruent only to $J_{\lceil P/H_j \rceil}$ modulo $\lceil P/H_j \rceil$. The upper bound is attained for a minimal residue system, i.e., $\{-\lceil N/(2H_j) \rceil, \ldots, -1, 1, \ldots, \lceil N/(2H_j) \rceil\}$. Defining:

$$M_j = \left\lceil \frac{N}{2H_j} \right\rceil , \qquad \text{for } j = A, C , \tag{9}$$

the total non-hit contribution to the entire maximum cross-correlation between two code words is seen to be bounded by $2H_j \sum_{k=-M_j}^{M_j} 2N/BT|k|$.

III. AMBIGUITY BOUNDS

Because all of the above code words have virtually identical time delay and frequency shift characteristics, the correlation properties described in Section II directly extend to the narrowband auto-ambiguity function for any class considered. Furthermore, a uniform upper bound on the entire cross-ambiguity function surface can be determined on a class-by-class basis by computing the cross-correlation function between pairs of signals, for every time-frequency shift, and taking the largest value that these functions can achieve. Hence, taking into account both hit and non-hit contributions in the manner of [12], we obtain the upper bounds:

$$A_j(N) = \frac{H_j}{N} + \frac{8H_jN}{BT}\left(1 + \ln M_j\right) \qquad \text{for } j = A, C , \tag{10}$$

where we have slightly overbounded the non-hit contribution when N/H_j is odd. For each prime $P = N + 1 \geq 3$ (and constant BT), the expression $A_A(N)$ represents a uniform cyclic upper bound on the spurious peaks away from the mainlobe of any auto-ambiguity function in a given class (with the obvious exception of the Doppler axis where all placement differences are identical). Similarly, the expression $A_C(N)$ represents a uniform cyclic upper bound on the entire cross-ambiguity function surface for a given class. Although (10) does not take into consideration the fact that, if we actually delay the signal instead of just shifting cyclically, there will be fewer overlaps, this can easily be incorporated into the bound computation by simply assuming that the smallest terms in the sum are dropped [12].

Ignoring the placement-specific hit factors H_j and M_j, the upper bounds obtained in (10) depend only on the code length, N, and the time-bandwidth product, BT. It is of interest to investigate the relative influence of these two parameters, and in particular to calculate the specific value of BT which minimizes the bound $A_A(N)$. Differentiating (10) with respect to N and solving for BT results in the optimal value:

$$(BT)_{\text{opt}} = 8N^2\left(2 + \ln M_A\right) . \tag{11}$$

This relationship characterizes the trade-off between time-bandwidth product and number of elements in the code set. Observe that this optimal BT is asymptotically independent of H_A, and therefore, for sufficiently large N, is the same *for all the classes* considered above. The corresponding minimum value of $A_A(N)$ is class-dependent, however, since:

$$A_A(N)_{\min} = \frac{H_A}{N}\left(1 + \frac{1 + \ln M_A}{2 + \ln M_A}\right) . \tag{12}$$

In turn, the bound on the cross-ambiguity surface becomes:

$$A_C(N)_{\min} = \frac{H_C}{N}\left(1 + \frac{1 + \ln M_C}{2 + \ln M_A}\right) . \tag{13}$$

The bracketed terms in (12) and (13) admit the N-asymptotic expressions $2 - 1/\ln N$ and $2 - (1 + \ln H_C/H_A)/\ln N$, respectively. Consequently, if BT grows as specified by (11), the upper bound on the spurious peaks of the auto-ambiguity function goes to zero as $2H_A/N$, and the uniform upper bound on the cross-ambiguity function surface goes to zero as $2H_C/N$. Note that these expressions are valid across *the whole class* of signals considered. The resulting asymptotic bound values for some of the classes of frequency hop congruential signals defined in Section II are summarized in Table I.

IV. CLASS SELECTION

The selection of a particular class of congruential signals for use in a specific application obviously depends on many factors. For instance, non-full signals such as QC (and some CC) may not be desirable in some applications because they violate the one-pulse-per-time-slot-and-one-pulse-per-frequency-channel rule required to optimize performance under clutter-limited conditions [1]. Yet, the framework just developed is useful to aid in the selection process: Table I shows LC signals, for example, to be well suited for inter-code rejection but not recommended for accurate range and Doppler measurements. In a situation such as multi-user echolocation, any one of the last four classes in Table I would seem indicated, with QC signals apparently the best but EQC signals still reasonably attractive. To illustrate the point, let us compare the typical behavior of, say, WC, QC, and EQC signals in time-frequency space.

For the values $P = 31$ and $BT = 512$, Fig. 2.a-c and 2.d-f show typical auto- and cross-ambiguity functions, respectively, for the three classes of signals. Somewhat surprisingly,

Class of Signals	Auto-Ambiguity Bound	Cross-Ambiguity Bound
Welch-Costas	$2/N$	N/N
Linear Congr.	N/N	$2/N$
Quadratic Congr.	$2/N$	$4/N$
Cubic Congr.	$4/N$	$6/N$
Hyperbolic Congr.	$4/N$	$4/N$
Extended Quadratic Congr.	$8/N$	$12/N$

Table I. N-Asymptotic Auto- and Cross-Ambiguity Bounds for 6 Classes of Frequency Hop Congruential Signals.

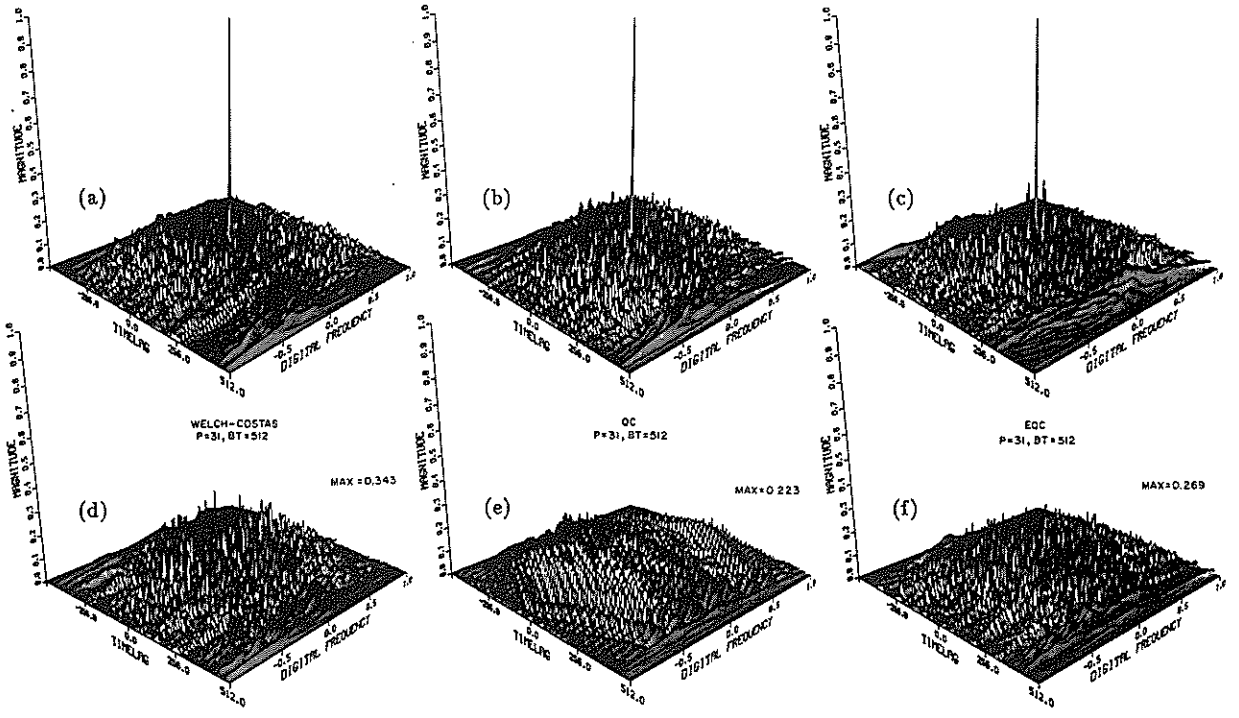


Fig. 2. Typical Time–Frequency Behavior for Three Classes of Frequency Hop Signals, for $P = N + 1 = 31$. Auto–Ambiguity Functions: (a) Welch–Costas, (b) Quadratic Congruential, (c) Extended Quadratic Congruential. Cross–Ambiguity Functions: (d) Welch–Costas, (e) Quadratic Congruential, (f) Extended Quadratic Congruential.

there is very little difference between WC/QC and EQC auto-ambiguity functions, except for (marginally) higher sidelobes in the latter; in this case the EQC departure from the near ideal characteristics of WC and QC signals seems to be fairly limited. This tends to suggest that all signals with bounded H_A behave quite similarly in the auto-ambiguity domain. In the cross-ambiguity domain, the maximum peak present in the respective ambiguity surface reaches 0.223, 0.269, and 0.343 in the QC, EQC, and WC cases, respectively. This ranking closely reflects the cross-ambiguity bound values obtained in Table I, and further emphasizes that signals with "sufficiently small" H_C should be selected for best performance in a multi-user environment.

V. CONCLUSION

We have presented a comparative analysis of several classes of frequency hop signals, based upon exponential, polynomial, and rational congruences on a finite field. For each class, the unified framework led to uniform upper bounds on (i) the amplitude of spurious auto-ambiguity peaks, and (ii) the level of the cross-ambiguity surface. These bounds allow for a meaningful comparison between whole classes of frequency hopping patterns, especially of interest for applications where both auto- and cross-ambiguity properties are of importance, such as active multi-user radar or multi-beam sonar imaging.

REFERENCES

[1] J.P. Costas, "A Study of a Class of Detection Waveforms Having Nearly Ideal Range–Doppler Ambiguity Properties," *Proc. IEEE*, Vol. 72, pp. 996–1009, August 1984.

[2] R.M. Mersereau and T.S. Seay, "Multiple Access Frequency Hopping Patterns with Low Ambiguity," *IEEE Trans. AES*, Vol. AES–17, pp. 571–578, July 1981.

[3] P.M. Cassereau and J.S. Jaffe, "Frequency Hopping Patterns for Simultaneous Multiple–Beam Sonar Imaging," *Proc.* 1987 *ICASSP*, pp. 1704–1707, April 1987.

[4] J.S. Jaffe *et al.*, "Incoherent Coding Techniques and Performance Characterization for MultiBeam Sonar Systems," *Proc.* 1988 *ICASSP*, pp. 2709–2712, April 1988.

[5] J.R. Bellegarda, "Time–Frequency Properties of Extended Quadratic Congruential Frequency Hop Signals," in *Proc.* 1989 *ICASSP*, pp. 2669–2672, May 1989.

[6] E.L. Titlebaum, "Time–Frequency Hop Signals Part I: Coding Based Upon the Theory of Linear Congruences," *IEEE Trans. AES*, Vol. AES–17, pp. 490–493, July 1981.

[7] E.L. Titlebaum and L.H. Sibul, "Time–Frequency Hop Signals Part II: Coding Based Upon Quadratic Congruences," *IEEE Trans. AES*, Vol. 17, pp. 494–499, July 1981.

[8] S.W. Golomb and H. Taylor, "Constructions and Properties of Costas Arrays," *Proc. IEEE*, Vol. 72, pp. 1143–1163, September 1984.

[9] J.R. Bellegarda, "Congruential Frequency Hop Signals in Multi–User Environments: A Comparative Analysis," in *Proc.* 1990 *ICASSP*, April 1990.

[10] S.V. Marić, private communication, August 1989.

[11] M.R. Schroeder, *Number Theory in Science and Communication*, Berlin: Springer, 1986.

[12] J.R. Bellegarda and E.L. Titlebaum, "Time–Frequency Hop Codes Based Upon Extended Quadratic Congruences," *IEEE Trans. AES*, Vol. 24, pp. 726–742, Nov. 1988.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

NON-PARAMETRIC SERIAL DECISION FUSION

A. Elías Fusté, A. Broquetas Ibars, R. Castro Fouz

E.T.S.I.Telecomunicación de Barcelona, P.O.Box 30002 08080 Barcelona, SPAIN

The study of a distributed serial data fusion system for a network of several CFAR receivers is presented. Rank fusion rules and independent control of each receiver detection threshold have been used in order to obtain a CFAR operation of the network. A recursive algorithm has been formulated allowing to choose the optimum rank rule for a given network with the objective of maximize the global probability of detection. The results are compared to those obtained for a parallel configuration with a unique concentrated fusion center which is used as a reference.

1. Introduction

In radar surveillance often several spacially distributed sensors can detect a target with different probabilities of detection and false alarm rates, range, interference rejection, etc. By combining the detected signals in a fusion center better detection characteristics and interference rejection can be obtained. For example, a parallel concentrated fusion center can process the data from all the sensors of a network, as shown in fig.1.

An alternative is to use a distributed serial fusion configuration in which each sensor performs the fusion of its signals with the information of the preceding sensors. This architecture has several advantages:

- The amount of information that must be transmitted along the network is reduced, requiring narrower channel bandwiths

- The processing is simpler and fastest because is performed in multiple parallel processors

- Because the fusion is distributed, the network is more immune to malfunction or destruction of its parts.

However even a serial network must have a central control center performing the following tasks:

- Modify the detection thresholds of the the sensors

- Choose the best fusion rule according to the network status

- Reconfigure the network in case of malfunction or partial destruction

The present study assumes a network of Cell Averaging Constant False Alarm Rate (CA-CFAR) sensors with different characteristics of detection. A parallel concentrated fusion center has also been studied to be used as a reference. The global probability of detection has been evaluated for both serial and parallel configurations. From these results the detection thresholds of the different receivers have been optimized. Finally some relevant results of this optimization are presented outlining the main conclusions of this work.

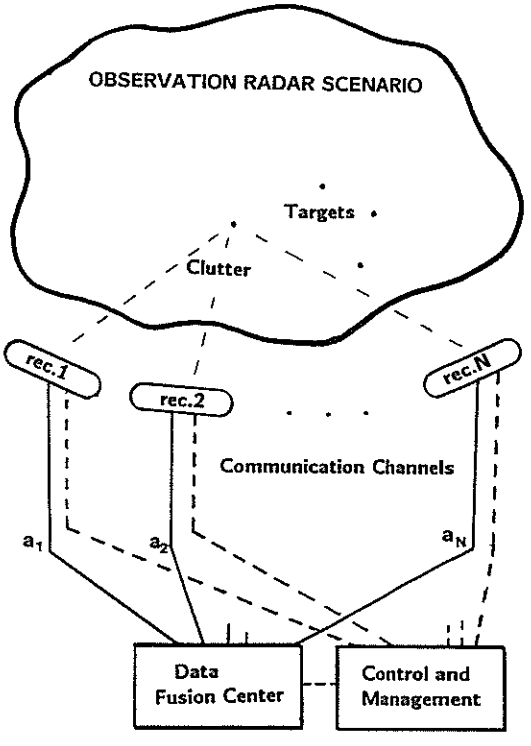


Fig.1 Fusion of a distributed sensor network data

2. Parallel and Serial Configurations

The objective of a given network will be to maximize the probability of detection while keeping the false alarm ratio constant. The targets are assumed to be Swerling-1 type, with noise or interference due to clutter having similar Rayleigh statistics. In this case, the probabilities of detection and false alarm can be obtained with the following expressions [1]:

$$P_D = \left[\frac{1 + SNR}{1 + SNR + T} \right]^M ; \quad P_{FA} = \left[\frac{1}{1 + T} \right]^M$$

where:
SNR is the signal to noise or interference ratio
M is the number of cells of the CA-CFAR receiver
T is the detection threshold

The serial network with N receivers consists of N-1 binary fusion centers, this configuration is depicted in fig.2 a). A parallel structure which has been evaluated as a reference is shown in fig.2 b).

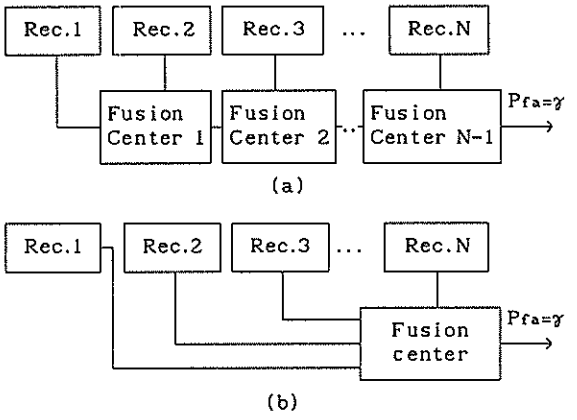


Fig. 2
a) Distributed Fusion in a serial configuration
b) Fusion in a parallel configuration

In both configurations a non-parametric 'k of N' decision rule is used, in this way the fusion center decides the presence of a target if at least k of the N receivers have detected the target. For the serial structure the fusion is binary and the possible rules are the 'AND' (2 of 2) or the 'OR' (1 of 2). In a parallel structure it is possible to weigh the individual decisions with a coefficient a_i with values between 0 and 1. The unweighted case using equal coefficients has been studied in [2].

3. Evaluation of the Global Probability of Detection

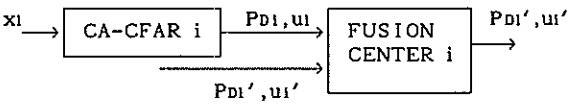
The expressions obtained in [3] for a system consisting of two sensors can be generalized to the case of N sensors and 'k of N' rules, obtaining the probability of detection and false alarm of the parallel configuration. In this case the expressions of the total probabilities of detection or false alarm are the following:

$$\begin{aligned} P_T = {} & R(0,\ldots,0)\cdot(1-P_0)\cdot(1-P_1)\cdots(1-P_{N-1}) + \\ & R(0,\ldots,0,1)\cdot(1-P_0)\cdots(1-P_{N-2})\cdot P_{N-1} + \\ & R(0,..,0,1,0)\cdot(1-P_0)\cdots P_{N-2}\cdot(1-P_{N-1}) + \\ & \ldots\ldots \\ & R(1,\ldots,1)\cdot P_0\cdot P_1\cdots P_{N-1} \end{aligned}$$

R() gives the fusion decision according to the decision rule, the individual decision vector $\vec{u}$ and the weighting vector $\vec{a}$ as follows:

$$R(\vec{u}) = \begin{cases} 1 \text{ (target detected)} & \text{if } \vec{a}\cdot\vec{u} = \sum a_i\cdot u_i \geq k \\ 0 \text{ (not detected)} & \text{if } \vec{a}\cdot\vec{u} < k \end{cases}$$

The serial structure suggest a recursive evaluation based on the basic fusion cell as shown bellow



From the basic cell j the following recursive expression can be deduced

$$P'_{DJ+1} = (-1)^{R_{J+1}} \cdot P_{DJ}\cdot P'_{DJ} + \left[P_{DJ} + P'_{DJ} \right] \delta(R_J)$$

where:

P'_{DJ+1} is the probability of detection obtained in the decision center j
P_{DJ} is the probability of detection in the CA-CFAR j

$R_J = \left\{ \begin{matrix} 1 \text{ for the AND rule} \\ 0 \text{ for the OR rule} \end{matrix} \right\}$ is the decision rule in the cell j

$$\delta(R_J) = \begin{cases} 1 \text{ if } R_J = 0 \\ 0 \text{ if } R_J = 1 \end{cases}$$

The probability of false alarm can be obtained in the same way substituting P_D by P_{FA}.

4. Network Optimization Method

The network optimization consists on finding the detection thresholds of the sensors that maximize the total probability of detection while keeping the total probability of false alarm P_{FAT} constant. The optimization of a function with the restriction of keeping a parameter (P_{FAT}) constant is solved by the Lagrange multipliers method. In this way we obtain a system of N+1 non-linear equations and N+1 unknowns which has been solved with the Newton-Raphson second order fixed-point method. An initial solution is calculated from the restriction imposed to P_{FAT} and the bisection method [4].

5. Results

Several probabilities of detection have been evaluated for different network configurations and SNR values, after threshold optimization. Figure 3 shows the probability of detection against the SNR in three receivers for a fixed total false alarm probability of 10^{-5} and for different decision rules O: OR, A: AND. The receivers use a similar Mi=8 cell CFAR. The results obtained with a unweighted parallel decision center using a 2 of 3 rule are also shown as a reference, which in this case gives the optimum probability of detection. For high SNR the OR-OR rule give similar results to the optimum, for intermediate SNR the best serial rule is the AND-OR and finally for low SNR the AND-AND rule gives better probability of detection. Figs. 4 and 5 show the dependence of the probability of detection with the number of sensors when a 12 cell CFAR is used in the receivers. Fig.4 corresponds to the case of an OR decission rule and fig.5 shows the results of using AND rules in the network, it can be seen that the probabilities of detection are generally higher when the OR rule is used in the fusion centers. Fig.6 show the results obtained for an inhomogeneous four sensor network with different CFAR lengths and SNR ratios, the results corresponding to a unweighted parallel architecture using 2 of 4 and 3 of 4 rules are also shown. The optimum configuration in this case is a serial network using OR rules in the fusion centers.

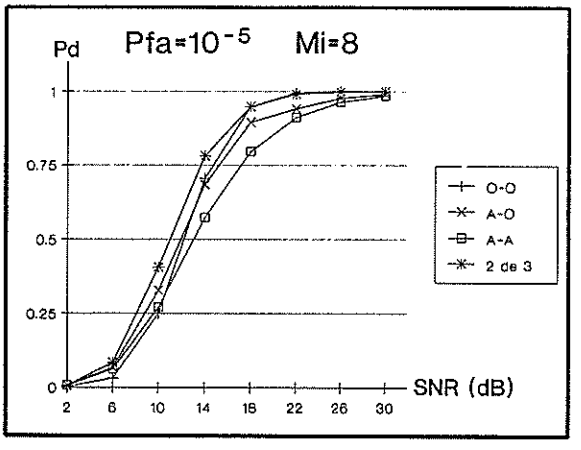


Fig.3 Probability of detection vs. signal to noise ratio for a serial configuration using diferent decision AND (A) or OR (O) rules, and for a parallel configuration using a 2 of 3 decision rule. The receivers use a 8 cell CFAR.

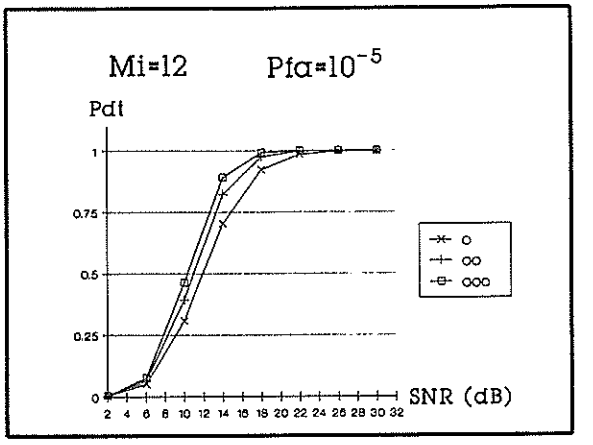


Fig.4 Probability of detection vs. signal to noise ratio for a serial configuration using 2, 3 and 4 sensors for an OR decision rule. The receivers use a 12 cell CFAR.

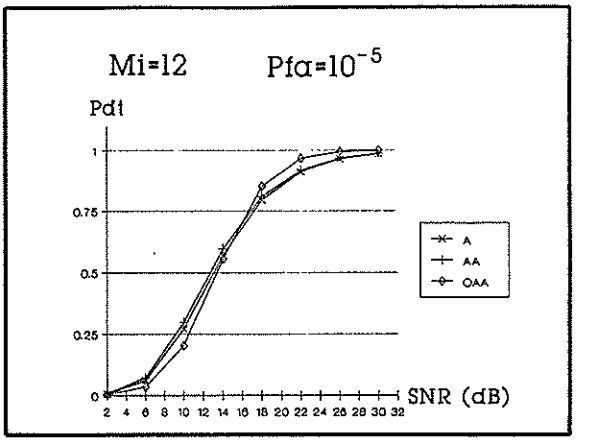


Fig.5 Same as fig.3 but using AND, AND-AND, OR-AND-AND rules

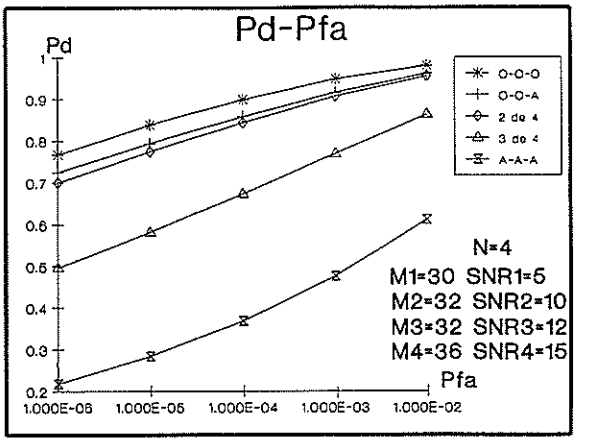


Fig.6 Probability of detection vs. probability of false alarm for an inhomogeneous network of 4 receivers with different CFAR lengths and SNR. Several serial and parallel fusion modes have been evaluated.

6. Conclusions

From the results obtained the following conclusions can be outlined:

A serial structure can always be simulated by a conveniently weighted parallel configuration, therefore this configuration is always the optimum although similar results can be obtained in some cases with an optimized serial structure.

In general for a serial network better results are obtained using OR rules for high SNR whereas for low SNR the probability of detection is higher for AND rules.

In a serial network the last receivers have more influence in the total detection characteristics, for this reason it is convenient to place the best sensors at the end of the network. In this case the serial configuration can have comparable or even better performance than a unweighted parallel network.

References

[1] N. Levanon, "Radar principles", Chap. 12, John Wiley & Sons, New York 1988.

[2] A. Elías, J.Puga, "Estudio de un procesador CFAR con receptores distribuidos", IV Simposium Nacional del Comité Español de la URSI, pp.875-879, Santander 25-27 Sept. 89.

[3] M. Barkat, P.K. Varshney "CA-CFAR Detection with distributed radars and data fusion". Proceedings of the Radar 87 Conference, pp. 165-169, London 19-21 October.

[4] R.J. Castro Fouz, "Estudio de un centro de fusión de datos con estructura serie". Proyecto Final de Carrera, E.T.S.I. Telecomunicación Barcelona, 1990.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A CFAR AR-BASED METHOD FOR RADAR DETECTION IN CLUTTER

José R. Casar Corredera and Gonzalo de Miguel Vela

Dep. Señales, Sistemas y Radiocomunicaciones
ETSI de Telecomunicación - UPM
28040 Madrid

A CFAR detector for radar signals in clutter, which operates by successively thresholding the complex partial correlation coefficients of an AR model of the signal, is proposed. The method capitalizes on the spectral differences between the only-noise and the signal+noise cases, as they manifest themselves in distances between the corresponding correlation coefficients. As in other CFAR schemes, the noise parameters reference for the cell under test is obtained from the spatially neighbouring radar resolution cells. We present some preliminary performance curves, for a number of illustrative cases.

1. INTRODUCTION

Detection of radar signals in a background noise by frequency methods has traditionally relayed upon using thresholds on the computed (estimated) spectrum of the data. This is true both for DFT-based detectors and for parametric (such as AR-based) analyzers.

Detectors based on the DFT are being widespreadly used in radar applications, since their operation is well known and predictable, and combine a maximum coherent gain for narrow-band targets in white noise with an extremely high computational convenience.

Detectors based on an AR model for the signal under analysis have received recent attention (see for example [1]) due, among other reasons, to their potential capability of providing higher spectral resolution and therefore increased doppler discrimination, particularly with short lenght data records.

A series of reasons (including stability and complexity) has motivated the study of alternative spectrum-based processing schemes which do not use thresholds on the spectrum but on the corresponding reflection coefficients (parameters of the lattice implementation of the prediction error filter or partial correlation coefficients) (see for example [2]). When the clutter spectral shape is known or can be estimated, the on-the-reflection coefficients thresholding methodology can be extended to deal also with the coloured noise (clutter) case [3].

The point which first motivated this contribution is the fact that both in the white noise case and in the coloured noise case, noise parameters have to be known or otherwise previously estimated to set the thresholds which give a specified false alarm probability: those parameters are the noise power in the white noise case and the spectrum parameters (reflection coefficients, for example) in the coloured case (clutter). However they are rarely known in practice; moreover, they are time and space varying, thus preventing the use of fixed detection thresholds

In a practical radar detector (which operates either in time or frequency), the problem is overcome by resorting to the Constant-False-Alarm-Rate (CFAR) concept, where the unknown local noise characteristics (whichever they be) are derived from a number of adjacent space cells in an appropriate way, so that the detection thresholds are locally and automatically computed, thus making the relevant detection characteristics (i.e. probability of false alarm) independent to a degree of the noise statistics.

This paper extends that CFAR concept to detectors which operate by thresholding on the complex reflection coefficients of the AR model of the signals, thus providing a robust model-oriented signal detector for radar signals in coloured noise (clutter), which has the sensitivity of the AR spectral modelling and the advantages of structural detection.

The rest of the paper is organized as follows:

Section 2 will introduce the basics of the in-clutter detection procedure as a thresholding operation in the parameters of the lattice-prediction-error-filter of the data (reflection coefficients or partial correlation coefficients).

Section 3 contains the description of the whole operating detection algorithm.

Section 4 will elaborate on decision region design and will give some performance measures: a number of curves of Probability of False Alarm versus 'size' of the decision region and a number of Receiver-Operating-Characteristics will be provided.

Section 5 concludes the paper.

A final remark: we will avoid any discussion on the well known derivation and general properties of the lattice as a prediction error filter, which can be found elsewhere.

2. BASIC DETECTION STRATEGY [1,2,3]

Let us suppose a radar environment where we are concerned with the detection of a complex signal from a set of N complex data x(n) corrupted by a background noise and clutter. We want to distinguish between the hypothesis H0 (signal absent) and H1 (signal present).

The lattice detection scheme is based upon computing the lattice parameters k(m), m=1,...,M of the M-th order Prediction Error Filter (PEF) of the data x(n) (for example using the Levinson-Durbin recursion) and comparing them with their expected values in the noise-alone case (we will include both background and clutter in the term 'noise').

Let us temporarily assume that we know the noise spectral density and therefore that we can have precomputed the true values of the lattice PEF parameters for the noise-alone data: $r^0(m)$, m=1,...,M. Clearly, under the hypothesis H0 (signal absent), the mean value of k(m) will be $r^0(m)$, for, under H0, x(n) is an only noise record. However, under the hypothesis H1 (signal present), the mean value of k(m) will no longer be $r^0(m)$ but a different value (say $r^1(m)$), since the presence of the signal in the data gives rise to a bias in its spectrum and, in turn, a bias in the reflection coefficients. Therefore, it is possible to decide presence / absence of the radar signal by comparing the lattice-parameters of the measured data with the only-noise lattice parameters.

Let us point out that such an scheme uses essentially spectral information, since designing an M-th order PEF is equivalent to computing an AR(M) estimate of the signal spectrum.

3. THE OPERATING CFAR DETECTOR.

As it was stated in the Introduction, in practical radar operation, noise statistics are very rarely known. Using the CFAR concept, estimates of the relevant noise characteristics are derived from the cells in the neighborhood of the one under test (cells in the reference space window), by making the assumption of spatial homogeneity: i.e., by assuming that the statistics of the noise are basically unchanged from cell to cell in a reasonably wide area, which is taken as the reference window. Thus the statistics of the noise (in our detector, the noise reflection coefficients) in the cell under test are computed by averaging the reflection coefficients computed in the cells in the reference: this is known as the Cell-Averaging CFAR (CA-CFAR). Another possibility is to take the sample median of the reflection coefficients, instead of the sample mean: this is known as the Median OS-CFAR, which is a particular case of the family of OS-CFAR's (Ordered-Statistics CFAR), in which a given percentil of the noise distribution is used to set the detection threshold).

Having established this inference procedure to derive an estimate of the (coloured) noise statistics, the operation of the proposed detection mechanism can be summarized as follows:

a) the lattice complex prediction error filter

(reflection coefficients) is computed for each received complex data set: i.e., an AR model of the radar complex signal (real part, the in-phase signal; imaginary part, the in-quadrature signal) is computed.

b) a detection test is performed successively on the computed reflection coefficients, k(m), m=1,...,M. (which can therefore also be computed successively), by checking whether or not the complex coefficient is within or outside a 'decision region' in the complex plane. Whether any of them is outside the region, a target is declared, and eventually its doppler spectrum computed, if necessary.

c) for a given geometrical shape of the 'decision region' in the complex plane (see below), its 'central' point is obtained:

-either averaging the corresponding reflection coefficients in the adjacent cells (CA-CFAR),

-or taking the sample median of the coefficients in the adjacent cells (OS-CFAR).

It is then understood that a data record is declared as an only-noise record if every partial correlation coefficient is close (within a decision region) to the corresponding estimated only-noise reflection coefficient. In other words, a target is declared whether *any* of the reflection coefficients is outside its decision region.

In both CFARs the 'size' of the decision region is designed to give a specified false alarm rate (but it also depends on where in the plane the region is centered).

Under the hypothesis of spatial homogeneity, both schemes above can be proven to provide CFAR characteristics to a certain extent (although they present a different behaviour under non-homogeneity conditions).

4. DECISION REGION SHAPE AND SOME EXPERIMENTS

For a given estimated only-noise partial correlation coefficient, in a given lattice stage, r, the decision test is realized according to the following rule:

$$\frac{(1-\mathrm{Re}(r^{*}k))^{2}}{(1-|r|^{2})(1-|k|^{2})} \underset{H0}{\overset{H1}{\gtrless}} 1+T$$

where k is the corresponding partial correlation coefficient in the cell under test. T is the decision threshold which should be designed to give a prespecified probability of false alarm.

It can be easily verified that for a given r the 'decision region' has an ellipsoidal shape. This form has been obtained in [4] by computing the approximate probability density function of the complex random variable k and taking equiprobability contours.

A number of experiments have been realized to preliminarly establish some performance measures. We have realized Monte Carlo simulations with records of 64 points (it should be noticed that the size of the decision region to achieve a given probability of false alarm depends on the available number of points to estimate the partial correlation coefficients). In all the cases, the first order prediction error filter was computed (just one correlation coefficient used in the detection process). Thus, the Probability of False Alarm(P_{FA}) and Probability of Detection (P_D) curves below should be interpreted as per-stage probabilities. In all the cases, an 8 cell reference window was taken. The coloured noise was always simulated as a Gaussian AR(1) process.

The profiles of threshold T versus P_{FA} for two nominal values of the true noise partial correlation coefficient, $r^{o}=0.3$ and $r^{o}=0.7$, are represented in Figure 1 for the CA-CFAR.

To evaluate the detection power of the schemes, some Receiver-Operating-Characteristics (ROC) curves were plotted for some illustrative cases. Signals were generated as AR(1) processes with $|r^{1}|=0.8$ and varying phases $\mathrm{Arg}(r^{1})$. These signals are by no means taken as good models of typical narrowband target radar returns in noise and/or clutter. The objective at this point was to gain an idea of how spectral differences (and therefore differences in the values of the correlation coefficients) can be capitalized by the proposed detection mechanism. The signals at the cells in the space reference window, were still generated as only-noise signals with nominal partial correlation coefficient $r^{o}=0.3$ and $r^{o}=0.7$.

Figures 2 and 3 show, for the CA-CFAR, the P_D

versus P_{FA} profiles for several values of $Arg(r^1)$, and r^o=0.3 and r^o=0.7, respectively.

No significant differences were found when using the OS-CFAR instead of the CA-CFAR, in an homogeneous environment.

Notice that, as it could be expected, the detection capabilities improve as the spectral distance between the only-noise case (H0) and the with-signal record case (H1) increases (as it is measured by increasing differences between r^o and r^1).

5. CONCLUSIONS AND FURTHER WORK

A new CFAR scheme which provides a model-oriented signal detector for radar signals in coloured noise (clutter) has been proposed. By its nature, the detector should have the sensitivity of the AR spectral modelling, the robustness of CFAR schemes and the advantages of structural detection (thresholding is performed on a reduced set of bounded parameters).

Further work on their performance in realistic radar scenarios should be realized. Also a comparison with DFT-based CFAR detectors has to be carried out for both narrow and wide (doppler) band targets.

ACKNOWLEDGMENTS

Credit has to be given to Mr. Mariano García to have worked out a usable closed approximate expression for the probability density function of the correlation coefficients for Gaussian processes (see [4]).

REFERENCES

[1] S. Haykin, Radar Signal Processing, IEEE ASSP Mag., Vol 2, pp.1-18, 1985.

[2] A.S. Arslanian and T.S. Durrani, Target-Clutter Identification by Lattice Processors, Proc. ICASSP-84, pp.47.4.1-47.4.4.

[3] J.R.Casar, F.J. Casajús and T.S. Durrani, A Complete Lattice-Ladder Scheme for Detecting and Estimating Radar Signals in Clutter, Proc. MELECON-85, pp.335-338.

[4] M. García Otero, Ph.D. Thesis, UPM, 1990.

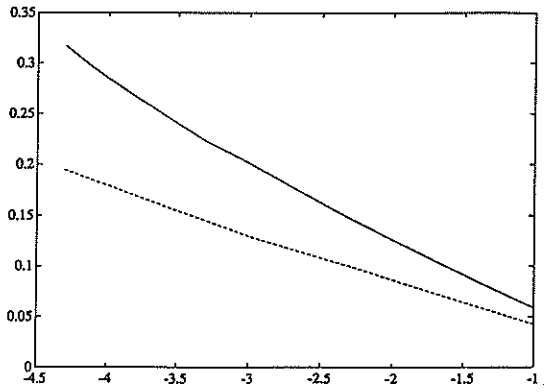


Figure 1: CA-CFAR threshold T versus $\log(P_{FA})$ for r^0=0.3 (continuous line) and r^0=0.7 (broken line).

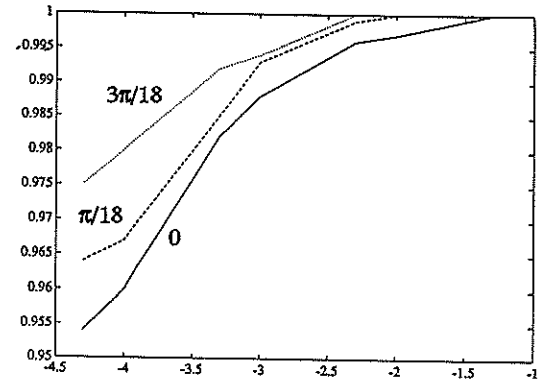


Figure 2: CA-CFAR P_D versus $\log(P_{FA})$ for r^0=0.3, $|r^1|$=0.8 and $Arg(r^1)$=0, π/18, 3π/18.

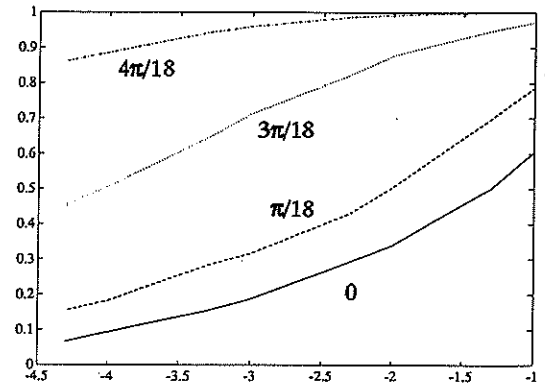


Figure 3: CA-CFAR P_D versus $\log(P_{FA})$ for r^0=0.7, $|r^1|$=0.8 and $Arg(r^1)$=0, π/18, 3π/18, 4π/18.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

TWO-DIMENSIONAL FILTERS FOR RADAR AND SONAR APPLICATIONS

R. Klemm, J. Ender

FGAN-FFM, Neuenahrer Str. 20, D 5307 Wachtberg 7, F.R. Germany

The problem of suppression of clutter (or reverbs) received by *moving* radar or sonar is addressed. The sensor velocity causes clutter returns to exhibit an azimuth dependant Doppler shift. The so obtained clutter Doppler bandwidth prevents detection of slow targets. The use of two-dimensional filters for detection of slow targets via time-space clutter rejection is addressed. Potential applications are real and synthetic aperture radar, and sonar.

GLOSSARY

CNR	clutter/noise ratio
φ	azimuth
FIR	finite impulse response
MTI	moving target indicator
N_t	temporal filter dimension
N_s	number of sensors
SAR	synthetic aperture radar
SCNR	signal/(clutter+noise) ratio
v	velocity

1. INTRODUCTION

1.1 Problem Statement

The problem of discriminating moving targets before a stationary background by a *moving* sensor array is addressed. In stationary systems such as ground based radar Doppler discrimination is a well-known tool to separate moving targets from the stationary background. The background energy concentrates at zero Doppler frequency and can be removed efficiently by well-known MTI techniques, such as pre-filters and/or Doppler analysis (FFT). If the sensor is moving echoes from the stationary background are Doppler shifted. This Doppler shift is proportional to the sine of the angle of arrival. For an omnidirectional background the reflected energy received by a moving sensor is spread over a Doppler band whose width is determined by the sensor beamwidth and the platform velocity.

If the target is fast compared with the sensor platform the problem is easily solved since the target Doppler frequency falls outside the interfering Doppler band. If, however, the radial target velocity component is smaller than the platform velocity, then the target echo falls into the Doppler band of the background and is difficult to detect. What is the difference between a moving target and the background, which target feature can be exploited for moving target detection inside the Doppler band of the background? For the background there is a one-to-one correspondence between Doppler and azimuth. For a moving target there is no such correspondence since its Doppler frequency depends also on the unknown target velocity and the direction of target motion.

Two dimensional filters applied to the data of a linear sensor array in conjunction with coherent Doppler filtering proves to be appropriate for discrimination between the different azimuth-doppler dependencies. The principle is illustrated in Fig. 1. The clutter power spectrum is plotted versus azimuth φ and velocity v. It appears to be a knife edge along the diagonal which reflects the one-to-one correspondence between azimuth and Doppler frequency of stationary targets. The thickness of the knife edge is determined by the array pattern and the spectral width of the Doppler filter (array aperture, length of the data sequence), respectively. The shape of the spectrum is determined by the sensor directivity pattern and the spatial clutter distribution.

On the left and right side of Fig. 1 the projections of the clutter spectrum on the azimuth and velocity axes are shown. One-dimensional (*temporal* or *spatial* only) filters would have to detect Doppler targets in the projected spectrum. A *time-space filter*, however, forms a narrow ditch along the diagonal of the φ-v-plane as indicated by the dashed line, thus leaving sufficient range for detection of moving targets.

The time-space filter techniques are particularly suited for detection of *slow* targets by a *moving* sensor array. Such techniques are referred to as TASTE (techniques for airborne slow target extraction), s. Ref. 8.

1.2 Survey of literature

In radar the DPCA (displaced phase centre antenna) techniques have been used to compensate for the platform motion such as to reduce the clutter bandwidth. The principle is that the phase centre of the receive antenna is moved opposite to the platform motion direction so that the motion effect is eliminated. This can be done either by use of a monopulse antenna [1] or by switching between two subapertures of an array antenna [2]. In [4] the likelihood-ratio-test is applied to two-dimensional (temporal-spatial) data fields to suppress, i.e., nulls in the antenna beam pattern are created for the associated clutter Doppler frequencies only. A refined technique for real-time operation based on this principle has been proposed in [5]. For special radar configura-

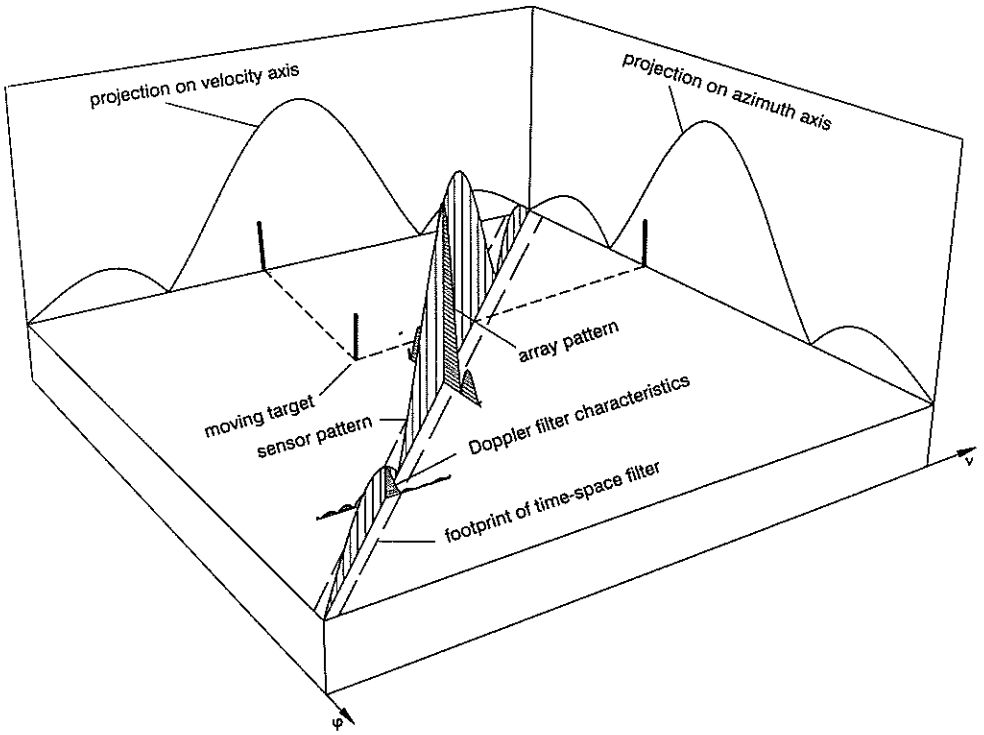


Fig. 1 Time-space spectrum of moving clutter

tions an even simpler technique based on 2D-FIR filters is given in [6].

To our knowledge such techniques have not yet been applied to sonar systems. The problem of motion induced reverberation bandwidth is well recognized ([3], pp. 470) and is countered to some extent by transmitting appropriate waveforms.

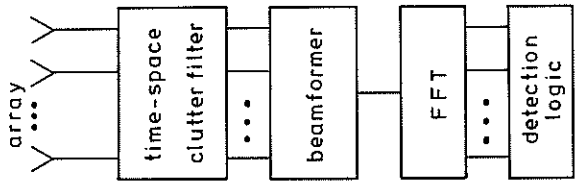


Fig. 2 Time-space radar receiver

2. VECTOR VALUED LEAST SQUARES FILTERS

The general scheme of a time-space radar receiver is depicted in Fig. 2. The data received by a sensor array are pre-whitened by a time-space clutter filter. Subsequent beamforming and an FFT provide signal match in space and time. A detection logic decides on "target" or "no target". For short data sequences (airborne surveillance radar) the clutter filter is a matrix, operating on finite data sets (Ref. 5). For long (in particular infinite) data sequences the time-space clutter filter may operate as a vector valued FIR-filter.

The filters addressed in this paper are vector valued generalisations of the prediction error filter treated by Burg [7] and subsequent publications on maximum entropy spectral analysis. Consider a stationary data sequence x_t. Then any data sample x_i can be predicted by a linear filter $\mathbf{g}$ applied to a data segment vector $\mathbf{x}(t)$ so that the resulting average prediction error $E\{|x_i-\mathbf{g}^*\mathbf{x}(t)|^2\}$ is minimum. The result is the well-known discrete Wiener-Hopf equation $\mathbf{R}_{(N)}\mathbf{g}=\mathbf{r}$, where $\mathbf{R}_{(N)}$ is the NxN correlation matrix of the interference and $\mathbf{r}$ the vector of cross-correlation values between $x_i(t)$ and $\mathbf{x}(t)$. The prediction error filter is defined by $\mathbf{h}^*\equiv(1,-\mathbf{g}^*)$ and can be obtained from the equation $\mathbf{h}=\mathbf{R}_{(N+1)}^{-1}\mathbf{e}_i$, where $\mathbf{e}_i$ is the i-th column of the unity matrix. For an input process specified by $\mathbf{R}_{(N+1)}$ the output sequence of the filter $\mathbf{h}$ has minimum power.

Consider a linear equispaced sensor array with its axis in motion direction. The m-th complex echo sample received by the n-th sensor due to a point source is (after demodulation and A/D conversion)

$$s(m,n) = \int a(mT,\varphi)\exp[j\beta(2v_r mT+n\, d\, \sin\varphi)]d\varphi$$

where a is amplitude, v_r radial velocity between target and sensor, d sensor spacing, T sampling interval, $\beta=2\pi/\lambda$ and φ is azimuth. For stationary scatterers v_r

means the radial component of the platform velocity: $v_r = v\sin\varphi$.

In case of clutter or reverbs the total of received signals is an integral of the above expression over φ. The two-dimensional (time-space) cross covariance coefficient is $r_{mn} = E\{s(l,q)s^*(i,k)\}$ with m=l-i and n=q-k. The total of coefficients can be written as a time-space covariance matrix R. In analogy to the one-dimensional prediction error filter we now use the first N_s rows of R^{-1} as filter matrix H. This approach is justified since all of these rows have the minimum power property addressed above. Multiplying the filter matrix by a beamformer vector b gives again a filter vector h=Hb which now operates in both the spatial and temporal dimension. The filter operation is illustrated in Fig. 3.

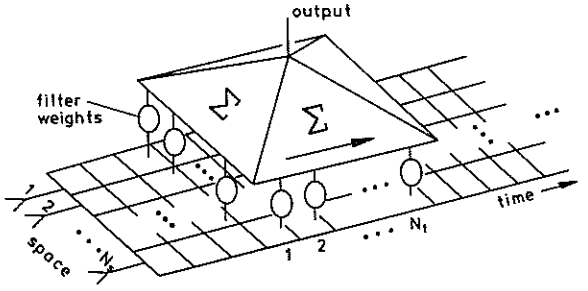


Fig. 3 Two-dimensional filter scheme

3. SIMULATION RESULTS

The following numerical examples are to illustrate the operation of *space-time* FIR *filters* compared with one-dimensional *temporal* or *spatial* filters. The effectiveness is judged by calculating the gain in SCNR. The look direction was assumed to be φ=0 (sideways). The length of the Doppler filter is 33. In the following figures the gain is plotted versus the target velocity normalised to the Nyquist velocity. The target is assumed to be in the look direction. The CNR is 20 dB.

In Fig. 4 we assumed a single stationary point reflector in the look direction φ=0. It would appear in Fig. 1 as a peak in the centre of the plot. If *no clutter filter* is applied (N_s,N_t=1) the gain is determined by the beamformer and the Dopplerfilter bank. The curve shows the inverse transmission characteristics of the Doppler filter bank which is a sinx/x function. The gain minimum indicates the radial velocity component of the clutter point (v=0).

The *spatial* filter (N_s=5, N_t=1) causes pure spatial suppression of the clutter signal. Since we are dealing with least squares filters the clutter portion is suppressed only down to the noise level (no exact null is formed). Since the gain of a Doppler filter is independent of the CNR we obtain a similar gain curve as if no clutter filter were used. It is obvious that the spatial filter cannot operate if the interference is located in the look direction.

The *temporal* filter (N_s=1,N_t=5) causes a notch at the clutter velocity and a smooth gain curve elsewhere. The

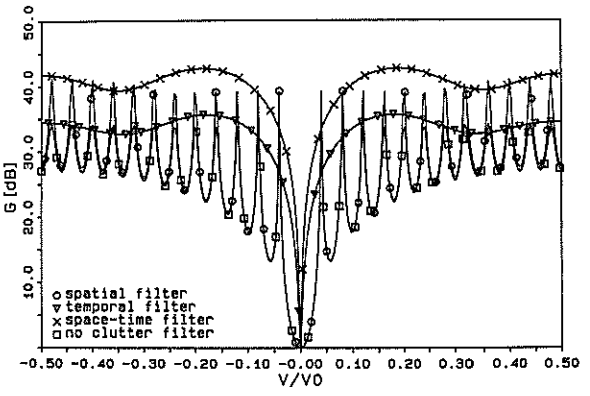


Fig. 4 Single point reflector in look direction

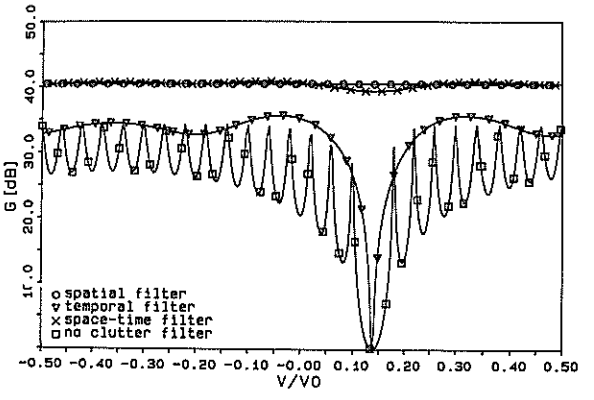


Fig. 5 Single point reflector outside look direction

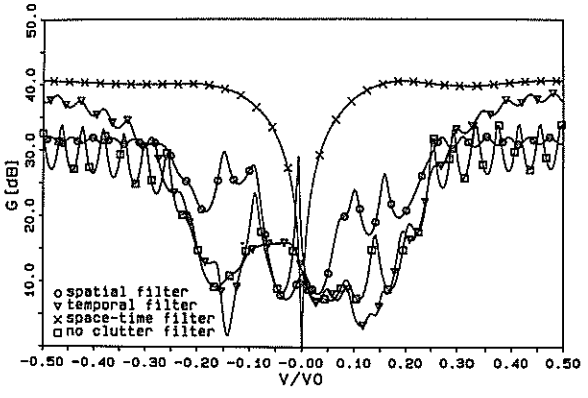


Fig. 6 Omnidirectional clutter

time-space filter yields more or less the same curve as the temporal filter except for the additional array gain. Obviously the temporal and time-space filters lead to comparable results if the interference is a single line in frequency and azimuth.

In Fig. 5 we assumed again a single point shaped reflector, however being located outside the look direction, i.e., in the sidelobe area of the beamformer. This causes a shift of the clutter velocity relative to the radar which can be recognised as a shift of the minima of the gain curves.

If *no clutter filter* is applied the gain curve shows again the sidelobe structure of the inverse Doppler filter. The gain achieved by the *temporal* filter is similar to the one shown in Fig. 4. The *spatial* filter now operates as sidelobe canceller. As this filter has no temporal dimension the gain is independent of the velocity (or Doppler frequency). Perfect clutter cancellation is achieved. The *time-space* filter shows a slight response at the clutter velocity.

In Fig. 6 a realistic clutter scenario is shown. For $N_s,N_t=1$ (*no clutter filter*) we obtain the inverse clutter spectrum. As can be seen the clutter energy is distributed over a band v/v_0=-0.25...0.25. This corresponds to the projection of the clutter spectrum onto the v-axis in Fig. 1.

The transfer function of a *spatial* filter (N_s=5, N_t=1) is the inverse spectrum projected on the φ-axis (s. Fig. 1). The associated clutter velocity spectrum is shown in Fig. 6. As the spatial filter causes attenuation in the look direction no significant gain is obtained even outside the clutter band.

The *temporal* filter suppresses the clutter components over the entire clutter band but approaches high gain outside the clutter bandwidth. Such filter might be useful to detect targets faster than the sensor platform. The *time-space* filter shows an ideal smooth gain curve with a notch at v=0 which is the clutter velocity in the look direction. Here the target coincides with the clutter with respect to both azimuth and velocity so that no distinction is possible.

4. APPLICATIONS

4.1 Airborne search radar

In a coherent pulse Doppler radar subsequent echoes form the temporal dimension of the two-dimensional data field. This means that T in equ. 1 denotes the interval between successive pulses. For clutter suppression with a 2-D least squares FIR filter an array antenna is required whose sensors are equispaced in the horizontal.

4.2 Synthetic aperture radar (SAR)

The synthetic aperture principle is based on long coherent pulse trains which are received while the platform is moving at constant speed. Again T in equ. 1 denotes the pulse interval. Imaging radars such as SAR need high bandwidth for high range resolution. High bandwidth however may degrade the performance of spatial filters (e.g., sidelobe cancellers for anti-jamming). It was shown in Ref. 9 that the clutter suppression performance of least-squares time-space FIR filters is basically not affected by the system bandwidth.

4.3 Active Sonar

Due to irregularities of the medium and long travel time of an acoustic pulse subsequent reverberant echoes of a pulse train are usually uncorrelated. In this case T in equ. 1 may denote the interval between samples taken from the single echo. This is also justified because the Doppler bandwidth and the system bandwidth can be made the same order of magnitude.

5. CONCLUSIONS

Detection of slow targets by a moving sensor array before a clutter (or reverbs) background was addressed. Time-space (or velocity-azimuth) filters are capable of detecting slow targets buried in the clutter (or reverbs) band. The improvement in SCNR compared with one-dimensional (spatial or temporal) filters is dramatic. Real-time adaptive filtering is possible. Adaptive implementations can compensate for instantaneous platform motion errors.

REFERENCES

[1] Skolnik, M., Radar Handbook, McGraw-Hill (1970), pp. 18-7

[2] Tsandoulas, G.N., Unidimensionally scanned phased arrays. IEEE Trans. AP-28, No 1 (1980), pp. 86-99

[3] Van Trees, H., Detection, estimation and modulation theory. John Wiley (1971), Pt. 3.

[4] Brennan, L.E., Mallett, J.D., Reed, I.S., Adaptive arrays in airborne MTI. IEEE Trans. AP-24, (1976), pp. 607-615

[5] Klemm, R., Adaptive airborne MTI: an auxiliary channel approach. IEE Proc., Vol 134, Pt. F, No 3,June 1987, pp. 269-276.

[6] Ender, J., Klemm, R., Airborne MTI via digital filtering. IEE Proc., Vol 136, Pt. F,N0 1, Febr. 1989, pp. 22-28

[7] Burg, J.P., A new analysis technique for time series data, in: Proc. of the NATO Advanced Study Institute, Enschede, Netherlands, 1968

[8] Ender, J., Klemm, R., TASTE - techniques for airborne slow target extraction, Pt. I + II, to be published.

[9] Klemm, R., Ender, J., New aspects of airborne radar, in: Proc. of IEEE RADAR 90, 7-10 May (1990), Arlington, VA, USA.

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

A UNIFIED APPROACH TO NON-LINEAR PROCESSING OF MULTIPLICATIVE NOISE WITH APPLICATIONS TO RADAR IMAGES

A. HILLION, J.M BOUCHER

Groupe Traitement d'Images
Département Mathématiques et Systèmes de Communication
ENST BR - B.P. 832 - 29285 BREST CEDEX - FRANCE

Abstract : This paper is devoted to the use of statistical non linear filters for speckle noise reduction on radar images. It assumes a multiplicative noise model for the speckle and gives a general solution on the basis of a pixel by pixel analysis. It shows how known classical filters are related to such a general estimator and computes the accurate solution for various types of backscattering coefficient distributions. An adaptive procedure for the processing of radar images is then described.

1 - Introduction

The radiometric resolution of radar images is disturbed by a particular granular noise, called speckle, due to the interferences of many individual scatterers in a resolution cell. A statistical study of this noise has shown that it could be seen as a multiplicative noise, whose distribution is a Gamma law [4], the law parameter N being the number of looks used in a first average process for reducing the speckle. Many kinds of filter have been created in the past [1,2,6] to solve this problem.
They process the image adaptively by weighting a given pixel according to the local image statistic with a mean square error criterion and they assume a priori knowledge of the noise moments. In this paper, it is shown that these filters belong to a more general class of non linear estimators and that the best filter can be found in this set depending on the backscattering coefficient distribution.

2 - The mathematics of generalized linear estimation

If the classical speckle model [4] is written as

$$(1)\ I = S.Z$$

the filtering problem is the estimation of an unobserved random variables (The reflectivity of the earth) by a suitable function S of the observed random variable I (the intensity of the image). Z is the speckle noise, which is assumed to be a random variable independent of reflectivity.

Various filterings are defined by looking for the best estimator, according to the **mean-square error criterion**
$\varepsilon^2(\hat{S}) = E[(\hat{S} - S)^2]$ within certain sets of estimators.

When confining to the set of linear combinations of 1 and I^t, $\hat{S} = a\, I^t + b$, we get the **best t-linear estimator,** $\hat{S}(t)$ [5]

(2) $\hat{S}(t) = a(t)\, I^t + b(t)$ is an unbiased estimator of S ($E(\hat{S}(t)) = E(S)$)

(3) The error estimation $\hat{S}(t) - S$ and I^t are uncorrelated ($\mathrm{cov}(\hat{S}(t) - S, I^t) = 0$)

The mean square error is found to be

$$(4)\ \Delta(t) = \varepsilon^2(\hat{S}(t)) = \mathrm{var}(\hat{S}(t) - S(t))$$
$$= \mathrm{var}\, S - \{\mathrm{var}\, I^t\}^{-1} \cdot \mathrm{cov}^2(S, I^t)$$

A special case occurs when t vanishes, leading to the definition of the **log-linear estimator** $\hat{S}(0^+)$
$\hat{S}(0^+) = a(0^+)\, L_I + b^+(0)$ is the best estimator within the set of linear combinations of 1 and $L_I = \mathrm{Log}\, I$ and verifies

$$(4\text{ bis})\ \Delta(0^+) = \varepsilon^2(\hat{S}(0^+)) = \lim_{t \to 0^+} \Delta(t)$$
$$= \mathrm{var}\, S - \{\mathrm{var}\, L_I\}^{-1}\, \mathrm{cov}^2(S, L_I)$$

Finally, we define the **best generalized-linear estimator** $\hat{S}_G$ as any estimator achieving

$$(5)\ \varepsilon^2 = \varepsilon^2(\hat{S}_G) = \inf_{0 < t \leq 1} \Delta(t)$$

Deriving (Eq 4) leads to

$$(6)\ \Delta'(t) = 2\, \mathrm{cov}(\hat{S}'(t), \hat{S}(t) - S)$$

As $\hat{S}'(t) = a'(t)\, I^t + a(t)\, L_I \cdot I^t + b'(t)$ holds and since $\hat{S}(t) - S$ is uncorrelated with 1 and I^t (cf Eq 2 and Eq 3), one gets

(6 bis) $\Delta'(t) = 2\, a(t)\, \mathrm{cov}(\hat{S}(t) - S, L_I . I^t)$, which entails the following result :

(7) If, for some t_o belonging to]0,1[, $\hat{S}(t_o)$ - S and $L_I.I^{to}$ are uncorrelated, then, as a rule, $\hat{S}(t_o)$ is the best generalized linear estimator
($\hat{S}(t_o) = \hat{S}_G$ is uncorrelated with 1, I^{to}, $L_I.I^{to}$).

3 -Classical filters

Various classical estimators are particular instances of the t-linear procedure.

. $\hat{S}$ (1) is known as the LEE-algorithm [6] : it is the best estimator which is a linear function of the intensity

. The homomorphic filtering [6], $\hat{S}_H$, the logarithm of which is the best linear function of L_S = Log S,may be written as $\hat{S}_H = a\, I^{t1}$, where $t_1 = (\text{var } L_I)^{-1}$. var (L_S).

.C.R. Moloney and M.E. Jernigan [7] have recently introduced another filter, the multiplicative estimator $\hat{S}_M$. $\hat{S}_M$ minimizes the mean-square error within the set of estimators of the form I^t
$\hat{S}_M = I^{t2}$, where t_2 verifies $\varepsilon^2(I^{t2}) = \inf_{0<t\le 1} \varepsilon^2(I^t)$
The very definition of the best generalized linear estimator entails that $\hat{S}_G$ is better than all these previous filters.

4 -Actual derivation of the generalized-linear estimator

. Putting s (t) = E (S^t), z (t) = E (Z^t), i (t) = E (I^t), one immediately gets from (Eq 1)

(8) i (t) = s (t) z (t), which enables us to write (Eq 4) as

$$(9)\quad \text{var } S - \Delta(t) = \frac{z^2(t)\{s(t+1) - s(1)\,s(t)\}^2}{s(2t)\,z(2t) - s^2(t)\,z^2(t)}$$

or

9(bis)

$$\text{var } S - \Delta(t) = \{\text{var } I^t\}^{-1} . z^2(t)\left\{\frac{E(I^{t+1})}{z(t+1)} - \frac{E(I).E(I^t)}{z(1)\,z(t)}\right\}^2$$

(Eq 9 involves the moments of noise and reflectivity whereas Eq 9 bis involves moments of noise and image intensity).

As a rule, some statistics of the noise are known. One generally makes the assumption that the mean of Z is equal to one (z (1) = 1) and that the variance, var Z = v, is closely related to the number of looks used to generate the images (v = z (2) - z^2 (1)). Moreover, the probability distribution of Z is often considered as belonging to a classical family, gamma distributions or lognormal distributions. In the case of log-normal distribution [2], for instance, the evaluation of the moments of Z is easy, leading to

$$(10)\quad z(t) = \exp\left\{-\frac{w}{2}\,t\,(t-1)\right\}$$

where w = Log (1 + v)

As for reflectivity, the situation is more intricate. If one knows that the distribution of S belongs to a classical family of distributions (uniform, triangular, gaussian, log normal...), the moments of S can be evaluated, leading to an analytic form for Δ (t) in (Eq 9). Then standard numerical methods of minimization produce the optimal value of t achieving the lower bound in (Eq 5).

That method has been used in [3] and [5]. In [5], reflectivity is assumed to obey a uniform distribution ranging from 0 to 2 μ. In [3], the hypothesis that S is a two-valued random variable allows applications to edge detection.
If Z being log-normal as in (10) reflectivity obeys a log-normal distribution (with mean value μ and variance σ^2 = var (S)), one finds, with standard calculus using properties of gaussian distributions, that the conditional mean of S given I is written as

$$(11)\quad E^I(S) = I^{to} . \mu^{1-to} e^{\frac{to\,w}{2}}$$

where, putting $v = \text{Log}(1 + \sigma^2.\mu^{-2})$, one has

$$t_o = \frac{v}{v+w}$$

Since the conditional mean of S given I is the best estimator among all the possible estimators, one has necessarily
$(12)\ \hat{S}_G = E^I(S) = \hat{S}(t_o)$
(which can be verified by studying the variations of

$$(13)\quad \text{var } S - \Delta(t) = \mu^2 . \frac{(e^{vt} - 1)^2}{\left(e^{t^2(v+w)} - 1\right)}$$

(By a suitable choice of v and w, one can exhibit a best generalized estimator $\hat{S}_G = \hat{S}(t_o)$ for any t_o belonging to]0,1[)

5 -Practical implementation

For real images, two approaches are possible for the derivation of the generalized estimator.

In the first approach, one has at hand a prior knowledge about reflectivity. The mathematical form of the distribution of S is known, except for some parameters such as mean, variance, skewness which can be estimated from the data by any efficient parametric statistical procedure.
Then, one gets closed formulas for the moments s (t) involved in (Eq 9), which enables a numerical minimization of Δ (t).

In the second approach, when one has no prior knowledge about the distribution of reflectivity, one is compelled to use (Eq 9 bis) which involves the moments of the intensity image. These moments can be estimated from the data for a

significant set of values of t and a sub-optimal procedure of minimization applied to the interpolated function Δ (t) [5]

The moments of I such as E [$I^{3/2}$], E [I], which are unknown, must be estimated from the data, by evaluating estimators such as $M^{-2}\sum_{i=1}^{M^2} I_i^{3/2}$ over a M x M window centred at the considered pixel. The window length must be large enough to make the estimation accurate, but cannot be extended too much because of the image's non stationarities. A length of 5 or 7 must be chosen. In this case, only an approximate value of t_0 giving the best filter can be obtained. As t varies between 0 and 1, it will not be far from 0, 1/2 or 1 : the improvement in mean square error will be poor between t_0 and the nearest value of t. It will be sufficient to compare the mean square errors in these three cases and to retain the best filter.

Practically, it is more realistic to reduce the set of competing optimal values for t_0 to a given number of significant values, say $\left(t = 0, t = \frac{1}{2}, t = 1\right)$
The quality of the three estimators is measured by the values of f (t) = var S - Δ (t), which is written for t = 0, $\frac{1}{2}$,1 as
(14)

$$f(0) = \{\text{var } L_1\}^{-1} . \{z(1)\}^{-2} \left\{\text{cov}(I, L_1) - \frac{E(I)}{z(1)} \text{cov}(Z, L_Z)\right\}^2$$

where cov (Z, LZ) = E (Z LZ) - E (Z) E (LZ)
= z' (1) - z (1) z' (0)

(14 bis)

$$f\left(\tfrac{1}{2}\right) = \left\{\text{var } I^{\frac{1}{2}}\right\}^{-1} . z^2\left(\tfrac{1}{2}\right) . \left\{\frac{E(I^{3/2})}{z(3/2)} - \frac{E(I).E(I^{1/2})}{z(1).z\left(\frac{1}{2}\right)}\right\}^2$$

(14ter)

$$f(1) = \{\text{var } I\}^{-1} . z^2(1) \left\{\frac{E(I^2)}{z(2)} - \frac{\{E(I)\}^2}{z^2(1)}\right\}^2$$

In Eq 14, 14 bis, 14 ter, quantities such as z (1), $z\left(\frac{1}{2}\right)$... which only depend on the distribution of speckle, are known.
One replaces, in Eq 14, 14 bis, 14 ter, the theoretical moments of I by the empirical moments and get estimates

$\hat{f}$ (0), $\hat{f}\left(\frac{1}{2}\right)$, $\hat{f}$ (1)

Finally, one selects the t-estimator S (t) $\left(t = 0, 1, \frac{1}{2}\right)$ corresponding to the greatest of the three values $\hat{f}$ (0), $\hat{f}\left(\frac{1}{2}\right)$, $\hat{f}$ (1)

This algorithm has been tested, in a simpler version [5] on real SEASAT images of Brittany (France).
As no prior knowledge of the reflectivity distribution is assumed, the algorithm may be applied to unhomogeneous images and may be a first rough step to segmentation by putting together in the same class the pixels corresponding to the same values of t.

Conclusion

An exact solution for the non linear filtering of a multiplicative noise with a mean square error criterion has been given. Various cases of noise and backscattering coefficient distributions have been considered leading to different optimal filters. The problems involved in a practical use of such filters have been outlined and a simplified strategy established.

REFERENCES

[1] H. ARSENAULT and M. DENIS - "Image processing in signal-dependent noise" Com. J. Phys. - vol 61 - 1983 - p. 309-317.
[2] J.M BOUCHER, A. HILLION - "α-linear processing of a multiplicative noise with application to radar images" - IAESTED Symp., Genève, April 1987, pp 42-4
[3] J.M BOUCHER, A. HILLION - "Non-linear filtering and edge detection in speckled radar images" IGARSS'88, Edinburgh, pp 1267-1268.
[4] J.W GOODMAN - "Some fundamental properties of speckle" J. Opt. Soc. Amer. vol 66 - Nov. 76 - pp. 1145-1149.
[5] A. HILLION, J.M BOUCHER - "A new non-linear filtering algorithm with application to radar images". Radar 88, Ann Arbor, MI - April 1988, pp 177-181
[6] P.T KUAN and al. - "Adaptive restoration of images with speckle" IEEE trans. on ASSP. vol 35 - n° 3 - mars 87 - pp 373-383
[7] C.R MOLONEY and M.E JERNIGAN - "Nonlinear adaptive restoration of images with multiplicative noise" - ICASSP 89 - Vol 3 - Glasgow - mai 89-pp. 1433-1436

SIGNAL PROCESSING V: Theories and Applications
L. Torres, E. Masgrau, and M.A. Lagunas (eds.)

MULTIVARIATE SIGNAL PROCESSING IN POLARIMETRIC RADARS

Dr. G. Wanielik

Daimler-Benz AG, Research Center, Wilhelm-Runge-Str.11, D-7900 Ulm, West Germany

Polarimetric radars offer the possibility to work on multivariate measurement sequences instead of scalar ones, which conventional radars use. The multivariate target information given in the form of the scattering matrix (SM) can be used to enhance the performance of a detector and to solve the target classification problem.

1. INTRODUCTION

A conventional pulse radar transmits, for example, a horizontally polarized wave and receives only the horizontally polarized part of the reflected signal. A polarimetric radar however transmits additionally a second, vertically polarized signal and receives both, the horizontally and vertically polarized parts of the two reflected signals. These four reflected signals at the time index i build the scattering matrix (SM) $\underline{S}(i)$ with the elements $S_{i,j}(i)$,i,j=1,2. An alternative description of the SM $\underline{S}(i)$ is its vectorial form $\underline{s}(i) = (S_{11}(i),S_{21}(i),S_{12}(i),S_{22}(i))^T$, which I call the multivariate signal vector $\underline{s}(i)$. For some applications it is more suitable.

2. MEASURED SM-DATA

When dealing with polarimetric signal processing it is of essential interest to determine the differences that occur in the SM-data when looking at different reflecting objects. I present these differences using measured SM-data of rain clutter and of a small jet observed at a specific aspect angle. The data were measured by the operational DLR-polarimetric weather radar, [1].

To visualize the information contained in one complex SM $\underline{S}(i)$ we use a feature vector constructed from the SM, [2]. We choose the horizontal, vertical and left-circular polarizations for the illumination of the reflecting objects and take the three reflected polarizations as well as the intensity and the phase of the first reflected wave to get the feature vector.

As a first measurement result we see in figure 1 typical SM-data of rain clutter. Notice that we get a whole sequence of SM which results in the three displayed point clouds a,b,c of the reflected polarization vectors on separate Mollweide-projections of the Poincarè-sphere. The intensity and phase as a function of the time of the first reflected wave are also shown in part d,e of figure 1.
Due to the limitation of the phase interval to $[-\pi,\pi]$ in figure 1e we get the 'sweep-like' wave

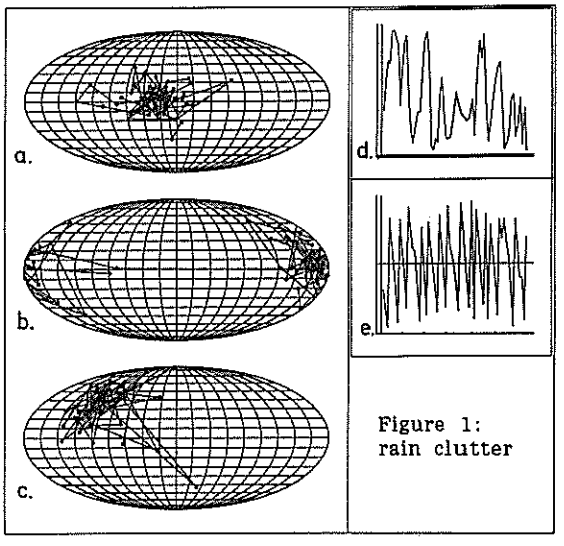


Figure 1: rain clutter

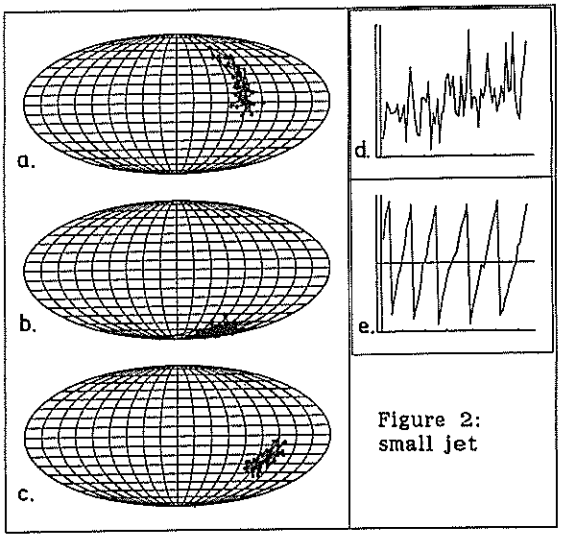


Figure 2: small jet

form which contains as information the radial velocity of the reflecting object.

The next target is an aircraft, a small jet, for which I present measurement data at a specific aspect angle. It is shown in figure 2 in the same way as before. Comparing both figures we can see the different places and spread of the point clouds on the Poincarè-sphere as well as differences in the wave intensities and wave phases.

This target dependent differences in the SM-data motivate the following study of polarimetric signal processing, the target detection and the target classification.

3. THE POLARIMETRIC CFAR (P-CFAR)

Radar detectors use small parts of a limited number of Doppler-filtered radar raster pictures to decide whether a cell under test is an outlier compared to the local surrounding data or not. Special types of detectors are those that guarantee a given constant probability of false alarms, the so-called CFAR, which we are interested in. In the following I discuss a polarimetric CFAR, which is able to detect in situations where a non-polarimetric CFAR fails.

In contrast to the non-polarimetric CFAR-detector, which uses a sequence of scalar amplitudes for a decision, the recently proposed polarimetric CFAR-detector [3] renders a decision based on a sequence of multivariate vectors $[\underline{s}(i)]$, i=1,...,2N+M as shown in figure 3. We distinguish here between the M cells under test $\underline{s}(N+1)$ to $\underline{s}(N+M)$ and the surrounding vectors. The P-CFAR has to answer the question whether the M cells under test $\underline{s}(N+1)$ to $\underline{s}(N+M)$ contain similar vectorial information to that of the surroundings or whether they are outliers.

In the following I extend this P-CFAR detector [3] to the case that the target information is concentrated not only in one Doppler-filtered radar raster picture as shown in figure 3 but in several pictures.

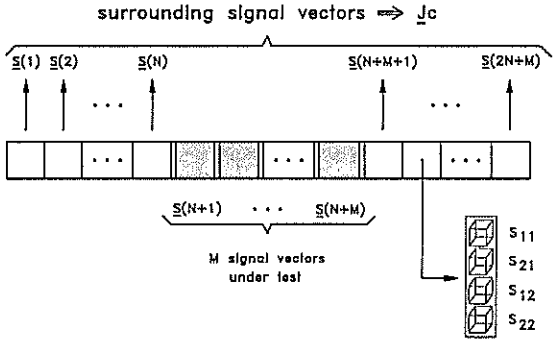


Figure 3: P-CFAR environment for one Doppler-channel

This implies that my recently proposed results are special cases of the more general formalism discussed now. The detection decision is based on the quadratic forms:

$$\begin{aligned} Q_{1\nu} &= \underline{s}_\nu(N+1)\cdot\underline{J}_{c\nu}^{-1}\cdot\underline{s}_\nu^{*T}(N+1) \\ &\vdots \\ Q_{M\nu} &= \underline{s}_\nu(N+M)\cdot\underline{J}_{c\nu}^{-1}\cdot\underline{s}_\nu^{*T}(N+M) \end{aligned} \qquad (1)$$

The matrices $J_{c\nu}$ in equation (1) are the covariance matrices (COV) of the surrounding stochastic vector processes in the ν-th Doppler-filtered channel. To build the test variable $Q(\mu,\Delta,M)$ in equation (3) some 'single-channel' test variables Q_ν are used:

$$Q_\nu = \sum_{i=1}^{M} Q_{i\nu} \qquad (2)$$

$$Q(\mu,\Delta,M) = \sum_{\nu=\mu-\Delta}^{\mu+\Delta} w_\nu\cdot Q_\nu \qquad (3)$$

$Q(\mu,\Delta,M)$ is the weighted sum of $2\Delta+1$ single-channel test variables Q_ν which lie around the central Doppler-filtered channel μ. Target detection occurs if

$$Q(\mu,\Delta,M) \;>\; T_{M\Delta} \qquad (4)$$

where $T_{M\Delta}$ is a threshold which guarantees a given probability of false alarms. The weights w_ν in equation (3) must be selected in a proper way. One possibility is to use $w_\nu=1$, another is to spend higher weights to those channels that contain a greater part of the spectral density of the target process.
The recently discussed detectors [3] are special cases of equation (4):

If the bandwidth of the target process is 'small enough' compared to that of the Doppler filters, then Δ can be chosen to be zero ($\Delta=0$). Then a detection occurs in the Doppler channel μ if:

$$Q(\mu,0,M) = \sum_{i=1}^{M} Q_{i\mu} \;>\; T_{M0} \qquad (5)$$

If additionally the target information occurs only in one cell (M=1) then a detection occurs in channel μ if:

$$Q(\mu,0,1) = Q_{1\mu} \;>\; T_{10} \qquad (6)$$

A more complete and quantitative description of the P-CFAR, which includes some performance evaluation, can be found in [3],[4], the robust P-CFAR is discussed in [5].

4. POLARIMETRIC TARGET CLASSIFICATION

4.1. Principles of Target Classification

The task of classification algorithms is to decide to which of the K classes the measurement data belong. Typically classification algorithms need a priori knowledge of the K processes. Such a priori knowledge may be described by statistical

parameters of the processes, which are estimated in a learning session using labelled representative samples of the K processes, [2].

The typical signal processing structure of a classifier uses two cascades. Cascade 1 evaluates a feature vector $\underline{m}$ from the measurement vectors $\underline{s}(i)$, $i=1,\ldots,N_m$, which then is used as input to cascade 2 which first evaluates an estimation vector $\underline{d}=(d_1,\ldots,d_j,\ldots,d_K)^T$ using properly selected estimation functions $f_j(\underline{m})=d_j$. Then, based on this estimation vector $\underline{d}$, the decision function generates a decision vector $\underline{e}^T=(0,\ldots,1,\ldots,0)$ which contains a value 1 only in that vector component j to which class the feature vector $\underline{m}$ belongs to and zeros elsewhere.

4.2. Singlelook Target Classification

A singlelook classifier works on one feature vector $\underline{m}$ generated from a measurement sequence of SM-data $\underline{s}(i)$, $i=1,\ldots,N_m$.

In the following I want to discuss a special classifier which is based on the multivariate complex normal distributions of the K class dependent signal vectors $\underline{s}(i)|_k$. This means that I directly use one measured SM-vector for the classification without any feature generation ($\underline{m}=\underline{s}$). The density functions of the zero-mean vectors $\underline{s}(i)|_k$ of class k is then given by

$$d|_k = \frac{1}{\pi^L \cdot \det(\underline{J}_k)} \cdot e^{-\underline{s}^{*T} \cdot \underline{J}_k^{-1} \cdot \underline{s}}, \quad k=1,\ldots,K \qquad (7)$$

where $L=\dim(\underline{s})$. I take into account the prior probabilities of class membership P_k which is the probability of drawing at random a member of class k from a mixed population of all K classes. As otherwise stated, we assume that the relative frequencies of all groups are different. Using the rule of Bayes the a posteriori probability that a given vector $\underline{s}$ belongs to class k can be used to construct an estimation function:

$$d_k = f_k(\underline{s}) = P(k|\underline{s}) = \frac{P_k \cdot d|k}{\sum_{j=1}^{K} P_j \cdot d|j} \qquad (8)$$

This estimation function has the special property that

$$\sum_k d_k = 1 \qquad (9)$$

which is sometimes useful in applications. For many purposes however there is no need to have such a normalization property so that we get a simpler classifier than in equation (8).

$$\hat{d}_k = \underline{s}^{*T} \cdot \underline{J}_k^{-1} \cdot \underline{s} + \ln(\det \underline{J}_k) - \ln P_k \qquad (10)$$
$$k=1,\ldots,K$$

The value $\hat{d}_k$ is the minimum value of all $\hat{d}_j$ if the value d_k of equation (8) is the maximum value of all d_j. We use the following estimation function $\hat{e}_k(\underline{d})$:

$$\hat{e}_k(\underline{d}) = \begin{cases} 1 & \text{if } \hat{d}_k = \min_j(\hat{d}_j) \\ 0 & \text{elsewhere} \end{cases} \qquad (11)$$

Equation (10) uses the quadratic forms

$$Q_k = \underline{s}^{*T} \cdot \underline{J}_k^{-1} \cdot \underline{s} \qquad (12)$$

which can be reformulated to get a result which is easier to interpret. As shown in [2] we can use Cloude's matrix decomposition vectors $\underline{A}_K$ and $\underline{B}$ for this, so we get:

$$\hat{d}_k = \underline{A}_k^T \cdot \underline{B} + \ln(\det \underline{J}_k) - \ln P_k \qquad (13)$$

by using the following relationship:

$$Q_k = \underline{A}_k^T \cdot \underline{B}$$

$$\underline{J}_k^{-1} = \sum_{i=0}^{15} A_{ki} \cdot \underline{\eta}_i \quad ; \quad \underline{J} = \underline{s} \cdot \underline{s}^{*T} = \sum_{i=0}^{15} B_i \cdot \underline{\eta}_i \qquad (14)$$

$$A_{ki} = \frac{1}{2} \cdot \mathrm{tr}(\underline{J}_k^{-1} \cdot \underline{\eta}_i); \quad B_i = \frac{1}{2} \cdot \mathrm{tr}(\underline{J} \cdot \underline{\eta}_i)$$

Equation (13) evaluates a scalar product of the vector $\underline{A}_k$ and the vector $\underline{B}$, which is actually evaluated from the vector $\underline{s}$, which we have to classify, and adds to it a constant $\ln(\det \underline{J}_k)-\ln P_k$. Because of the well-known properties of scalar products as distance measures between vectors, equation (13) gives more insight into the working mechanism of a classifier than the normally used result of equation (8).
Another polarimetric singlelook classifier, which is not based on the assumption of multivariate complex normal distributed observation data, can be found in [2].

4.3. Multilook Target Classification

In the previous chapter I have shown a singlelook classifier which makes a decision based on one feature vector $\underline{m}(i)$. Based on such 'singlelook' classifier results I now want to discuss a recently proposed new multilook classifier, [2]. This new multilook classifier evaluates a number N of singlelook decision and estimation vectors $\underline{d}(i)$, $\underline{e}(i)$ along a track of the target as shown in figure 4.

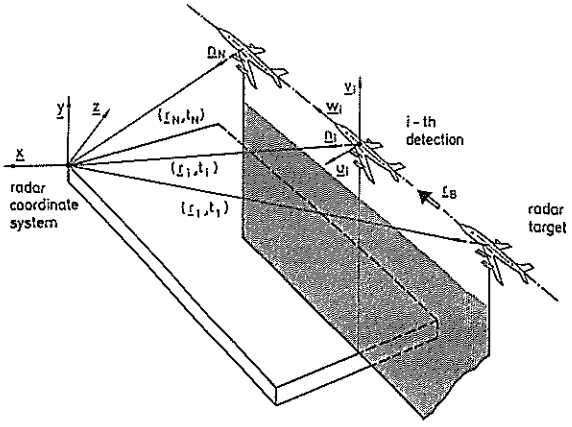


Figure 4: scenario of a multilook target classifier

An advantage of this multilook classifier is that it can compensate errors which occur in singlelook classifiers and that it therefore produces better classification results. I assume that the radar has evaluated a stable track and that it has detected the target at N time instances t_i, furthermore that all SM $\underline{S}(t_i,\underline{n}_i)$, i=1,...,N are stored. Based on the knowledge of the flight direction and on the radar resolution cell we can estimate the direction vector $\underline{n}_i$ under which the monostatic radar observes the SM-data of the target. This direction vectors $\underline{n}_i$ and the related direction areas of uncertainty are described in the target coordinate system. The areas are described by an index l_i so that each direction vector $\underline{n}_i$ is functionally related to a direction index l_i.

$$\underline{n}_i \longrightarrow l_i \in (1,\ldots,\Lambda) \qquad (15)$$

Such a direction knowledge l_i is necessary because the SM depend on the direction. This implies that we have to use direction dependent estimation functions for the classification. One set for each estimation is described by the direction index l_i.

$$f_{l_i k}(\underline{m}(i)), \quad k=1,\ldots,K; \; l_i \in (1,\ldots,\Lambda) \qquad (16)$$

The working mechanism of the multilook classifier can be described as follows. For each detection i along the target track a direction estimation gives us the direction index l_i which is used to choose the direction dependent estimation functions $f_{l_i k}(\underline{m}(i))$ which evaluate an estimation vector and a decision vector.

$$(\underline{d}(i)),(\underline{e}(i)); \quad i=1,\ldots,N \qquad (17)$$

These vector sequences are used to evaluate an overall estimation vector $\underline{D}$ and an overall decision vector $\underline{E}$ using properly selected combining rules. The first combining possibility evaluates the estimation vector components D_{1j} based on a weighted sum of the direction dependent singlelook decision vector components:

$$D_{1j} = \sum_{i=1}^{N} w_\nu(i) \cdot e_j(i) \qquad (18)$$

A second combination rule can be evaluated in the same manner but using the singlelook estimation vector components instead of the singlelook decision vector components:

$$D_{2j} = \sum_{i=1}^{N} w_\nu(i) \cdot d_j(i) \qquad (19)$$

The weights $w_\nu(i)$ can be chosen in several ways [2] such as:

$$w_1(i) = w_1 = \frac{1}{N} \qquad (20)$$

It should be mentioned that for multilook classifier purposes the used singlelook classifier results should be normalized in the same way as, for example, in equation (8).

This new multilook classifier is much more powerful than a singlelook classifier because errors in the estimation of the aspect angle can confuse a singlelook classifier much more than a multilook classifier. A further argument to use a multilook classifier is that there may be aspect angles where the class dependent feature vectors $\underline{m}(i)|_k$ do not differ very much. In such situations it happens that the performance of a singlelook classifier is quite poor. Because the multilook classifier uses a great number N of singlelook results he has a good chance that only a few bad singlelook results are used.

5. CONCLUSIONS

Because different targets have different polarimetric signatures this target information can be used to enhance the performance of the signal processing. As an example the P-CFAR is able to detect in situations where a non-polarimetric CFAR fails. Furthermore the polarimetric signatures can be used as input to the target classification which even works if only point targets are present.

ACKNOWLEDGEMENTS

The measured SM-data were taken with the DLR-polarimetric weather radar in Oberpfaffenhofen. I wish to thank this group headed by Dr. Schroth as well as the group at TST headed by Prof. D.J.R. Stock for their good cooperation.

REFRENCES

[1]A.Schroth, M.Chandra, P.Meischner: "Kohärente polarimetrische Radartechniken zur Forschung in der Mikrowellenausbreitung und Wolkenphysik", DFVLR-Forschungsbericht 88-47, 1988, Oberpfaffenhofen

[2]G. Wanielik, "Measured Scattering Matrix Data and their Use in Polarimetric Classifiation and Clustering Algorithms", Intern. Conference on Radar Polarimetry, IRESTE Nantes, March 1990

[3]G. Wanielik, "Single- and Multilook Polarimetric CFAR-Detectors", Intern. Conference on Radar Polarimetry, IRESTE Nantes,March 1990

[4]G. Wanielik, D.J.R.Stock: "Measured Scattering Matrix Data and a Polarimetric CFAR Detector, which works on this Data", IEEE Intern. Radar Conference, Washington, May 1990

[5]G. Wanielik, D.J.R. Stock, "A New Clutter Rejection Method using a Robust Polarimetric CFAR-Detector", Intern. Symposium on Noise and Clutter Rejection in Radars and Imaging Sensors, ISNCR-89, Kyoto, Japan, Nov. 1989