Vandewalle, J.;
Boite, R.; Moonen, M.;
Oosterlinck, A.;
Vol. 2.
FOREWORD

EUSIPCO-92, the European Signal Processing Conference, is the sixth in the sequence of the International Conferences promoted and organized by EURASIP, The European Association for Signal Processing. The Conference was held from 24 to 27 August 1992, in Brussels, the capital of Belgium, and the administrative heart of the European Community. These three volumes present the Proceedings of the Conference.

EUSIPCO-92 consisted of 48 separate sessions, organized in 6 parallel programs. The Program Committee reviewed over 640 submissions to select the 410 papers that were presented at the Conference. Each submission was assessed by at least two independent reviewers. In addition, 13 tutorials were given by well known experts in the field, and 7 special sessions were held, with 44 contributions on relevant actual topics.

The technical sessions were organized in 7 themes, containing the following topics:

1. Theory of Signals and Systems:
   a) Detection, b) Estimation, c) Filtering, d) Spectral Estimation, e) Adaptive Systems,
   f) Modeling, g) Digital Transforms, h) Digital Filtering.

2. Image Processing and Multidimensional Signal Processing:
   a) Coding, b) Enhancement, c) Restoration, d) Medical Image Processing.

3. Speech Processing:
   a) Coding, b) Synthesis, c) Recognition and Understanding, d) Enhancement.

4. Implementations:
   a) Hardware, b) Software, c) VLSI, d) Novel Architectures, e) Array Processing.

5. Knowledge Engineering and Signal Processing:
   a) Expert Systems, b) Pattern Recognition, c) Signal Interpretation,
   d) Image Understanding.

6. Neural Networks for Signal Processing:
   a) Theory, b) Speech, c) Vision, d) Implementations.

7. Applications:
   a) Radar, b) Sonar, c) Communications, d) Geophysics, e) Digital Audio, f) Biomedics,
   g) Sensing, h) Robotics, i) Astrophysics, j) Mechanics, k) other.

The volumes are organized as follows:

Volume I:
Tutorials, Special Sessions, Speech Processing, Knowledge Engineering and Signal Processing

Volume II:
Theory of Signals and Systems, Neural Networks for Signal Processing

Volume III:
Image Processing and Multidimensional Signal Processing, Implementations, Applications
We would like to thank all the EUSIPCO-92 participants and in particular those whose contributions made these proceedings so attractive. Our thanks are also due to all the co-sponsoring institutions for their financial support, and to Elsevier Science Publishers B.V. - North Holland for the realization of these Proceedings. We would be remiss not to thank the Conference Committee, the Scientific Committee, the Program Committee, the Session Chairmen, and the Tutorial Speakers. Last, but by no means least, we are most grateful to Ann Deforce, Rita De Wolf and Ingrid Tokka, for their outstanding clerical support.

Heverlee, May 1992

J. Vandewalle
R. Bolle
M. Moonen
A. Oosterlinck
CONFERENCE COMMITTEE

General Chairman
Oosterlinck A.
Katholieke Universiteit Leuven

Vice-Chairman
Van Eck J.L.
Université Libre de Bruxelles

Program Chairmen
Boite R.
Faculté Polytechnique de Mons

Vandewalle J.
Katholieke Universiteit Leuven

Finance
Govaerts R.
Katholieke Universiteit Leuven

Local Arrangements
Grenez F.
Université Libre de Bruxelles

Exhibition
Van Compernolle D.
Katholieke Universiteit Leuven

Publications
Moonen M.
Katholieke Universiteit Leuven
PROGRAM COMMITTEE

Acheroy M.  Hasler M.  Trullemans Ch.
Barbé A.  Hoffman G.  Vanbiessen L.
Boite R.  Kunt M.  Van Compernolle D.
Bruylant I.  Laloux A.  Vanderschoot J.
Catthoor F.  Leich H.  Vandewalle J.
David J.  Martens J.P.  Van Eycken L.
Delogne P.  Moonen M.  Van Huffel S.
Deprettere E.  Sas P.  Vanwormhoudt M.
Destiné J.  Schoentgen J.  Vanpoucke F.
Fawé A.  Schoukens J.  Vloeberghs C.
Fettweis A.  Slock D.  Wambacq P.
Gileen G.  Steyaert M.  Wellekens C.
Grenez F.  Suetens P.

SCIENTIFIC COMMITTEE

Almeida L.  Fliege N.  Mecklenbräuker W.
Atal B.S.  Galand C.  Mersereau R.M.
Bellanger M.  Gersho A.  Mitra S.K.
Benveniste A.  Granlund G.H.  Moschytz G.
Biemond J.  Gueguen C.  Neuvo Y.
Butterweck H.J.  Hasler M.  Niemann H.
Capellini V.  Haton J.P.  Peek J.B.H.
Carayannis G.  Herault J.  Pelliandini F.
Castanié F.  Hernandez-Gomez L.  Picinbono B.
Claassen T.A.C.M.  Herrmann O.  Sankur B.
Constantinides A.  Hess W.  Schüssler H.W.
Dellyannis Th.  Heute U.  Scharf L.
Deprettere E.  Kalouptsidis N.  Sicuranza G.L.
Devijver P.  Kunt M.  Vaidyanathan P.P.
Dewilde P.  Lacoste J.L.  Venetsanopoulos A.
Durrani T.S.  Lacroix A.  Vetterli M.
Escudié B.  Lagunas M.A.  Vich R.
Fallsida F.  Macchi O.  Weinrichter H.
Fettweis A.  McWhirter J.G.  Wolf D.
Figueiras-Vidal A.R.
ORGANIZATION

ORGANIZER:
European Association for Signal Processing
EURASIP
in co-operation with
IEEE Benelux Section

SPONSORS:
Belgian National Fund for Scientific Research (N.F.W.O.)
Egmontstraat 5
1050 Brussel
Belgium

CERA
Brusselsesteenweg 100
3000 Leuven
Belgium

LMS International
Interleuvenlaan 68
3001 Leuven
Belgium
EXHIBITORS:

Comdisco Systems, Inc.
First Floor, Aztec Centre
Aztec West, Almondsbury
Bristol BS12 4TD
England

Digital Equipment NV
Luchtschipstraat 1
1140 Evere
Belgium

DVC Digitalvideo Computing GmbH
Seestrasse 7
D-8036 Herrsching
Deutschland

ELSEVIER Science Publishers B.V.
Sara Burgerhartstraat 25
1055 KV Amsterdam
The Netherlands

European Development Center NV
Abdijstraat 34
3001 Heverlee
Belgium

John Wiley & Sons Ltd
Baffins Lane
Chichester, Sussex PO19 1UD
United Kingdom

KLUWER Academic Publishers Group
P.O. Box 989
3300 AZ Dordrecht
The Netherlands
OPENING SESSION

New challenges in signal processing for consumer electronics products
Dr. T.A.C.M. Claassen
- Philips Research Laboratories, Eindhoven, The Netherlands

TUTORIALS

Algorithmic engineering - A worked example
- Prof. J.G. McWhirter, I.K. Proudler
  - Defence Research Agency - RSRE, Malvern, UK

Wavelets in signal analysis
- Prof. I. Daubechies
  - Rutgers University, New Brunswick, USA

Design methodologies for application-specific signal processing architectures
- Prof. F. Catthoor
  - IMEC vzw., Heverlee, and Katholieke Universiteit Leuven, Belgium

Wavelet analysis in image processing
- Prof. J.-P. Antoine
  - Université Catholique de Louvain, Louvain-la-Neuve, Belgium

Recent trends in speech coding
- Prof. D. Wolf, U. Kipper, H. Reiningr
  - J.W. Goethe-Universität, Frankfurt am Main, Germany

Speech synthesis - A solved problem?
- Prof. W. Hess
  - Rheinische Friedrich-Wilhelms-Universität Bonn, Germany

Trends in adaptive filtering
- Prof. M. Bellanger
  - Conservatoire National des Arts et Métiers, Paris, France

Discrete passive modeling of physical systems described by PDEs
- Prof. A. Fettweis
  - Ruhr-Universität Bochum, Germany

Continuous speech recognition: from hidden Markov models to neural network
- Mr. H. Bourlard
  - Lernout & Hauspie Speech Products, Wemmel, Belgium

Reduction and approximation of linear computational networks
- Prof. P. Dewilde
  - Delft University of Technology, The Netherlands
Efficient signal modeling tools
Prof. G. Carayannis
National Technical University of Athens, Greece

Recent HDTV Systems
Prof. M. Kunt
Ecole Polytechnique Fédérale de Lausanne, Switzerland

From order 2 to HOS: new tools and applications
Prof. J.L. Lacoume, Gastel M., Amblaard PO.
Institut National Polytechnique de Grenoble, France

SPECIAL SESSIONS

Special Session 1
Imaging, Algorithms and Architectures

SS1-1 In vivo magnetic resonance imaging of metabolites
de Beer R., Mehikopf A.F., van Ormondt D., Pijnappel W.W.F., Schramp M.

SS1-2 A matrix description of synthetic aperture radar data
Bierens L.H.J., Deprettere E.F.

SS1-3 High resolution underwater acoustic imaging in time-domain
Bu J., Deprettere E.F., Dewilde P.

Special Session 2
Adaptive Filtering in Telecommunications

SS2-1 On the adequateness of IIR adaptive filtering for acoustic echo cancellation
Mboup M., Bonnet M.

SS2-2 Structure and convergence analysis of the generalized multi-delay adaptive filter
Alti Amrane O., Moulines E., Grenier Y.

SS2-3 A comparative study of efficient transversal algorithms for acoustic echo cancellation
Pétilion T., Gilloire A., Theodoridis S.

SS2-4 On the integration of subband echo cancellation into subband coding schemes
Kellermann W.

SS2-5 Adaptive infinite impulse response hybrids for primary rate ISDN
Dąbrowski A., Braszak A.

SS2-6 Symbol by symbol MAP detection and the Baum-Welch identification algorithm in digital communications
Vallet R.
SS2-7 A convergence study for the constant modulus algorithm leading to a normalized-CMA and a block-normalized-CMA
Hilal K., Duhamel P.
135
SS2-8 Equalization techniques in indoor radio environments
Casadevall F.J., Valdovinos A.
139
SS2-9 The piecewise linear adaptive filter for channel equalisation in digital communications
Callender C.P., Cowan C.F.N., Theodoridis S.
143
SS2-10 The RLS decision feedback equalizer with a finite duration exponential window
Liu H., He Z.
147

Special Session 3

Acoustic Articulatory Inversion
SS3-1 Some aspects of the physical and functional relation between articulation and acoustics
ten Bosch L.F.M.
151
SS3-2 The acoustic-area function inversion problem and the distinctive region model
Mrayati M., Carré R.
155
SS3-3 Inversion and speech recognition
Bailly G., Abry C., Boë J., Léboulterière P., Perrier P., Schwartz J.L.
159
SS3-4 Speech recognition by means of inverse problem solving
Sorokin V.N.
165
SS3-5 Estimation and generation of articulatory motion using neural networks
Shiral K., Kobayashi T.
167
SS3-6 Acoustical sensitivity functions and the control of a vocal tract model
Jospa P., Soquet A., Saerens M.
171

Special Session 4

Chaos in Signal Processing
SS4-1 Nonlinear dynamics, chaos and signal
Mira C.
175
SS4-2 Chaotic signal analysis and higher order statistics
Flandrin P., Michel O.
179
SS4-3 Effects of roundoff errors on chaotic dynamics
Beck C.
183
SS4-4 From order to disorder, an approach of chaos in musical signal
Gibiat W.
187
SS4-5 Chaos in digital recursive filters
Ogorzalek M., Hasler M., Baite R.

SS4-6 Chaos in a bed working under rapid fluidization regime
Borys A.

SS4-7 Subharmonic generation and chaos in adaptive filters
Darlington P.

Special Session 5

Multirate Digital Filter Banks - Theory and Applications

SS5-1 Orthonormality and paraunitarity in filter banks
Valayanathan P.P.

SS5-2 Multi-complementary filter bank: a new concept with aliasing-free subband
signal processing and perfect reconstruction
Fiege N.J., Zößer U.

SS5-3 The use of multirate filter banks for coding of high quality digital audio
Sporer T., Brandenburg K., Edler B.

SS5-4 Analysis/synthesis filter banks based on generalized sinusoidal transforms with
an application to speech coding
Gluth R., Heute U.

SS5-5 Multicarrier concept for digital mobile radio transmission
Kammeyer K.D., Tutsel U.

SS5-6 Perfect transmultiplexers using IIR filter banks
Mitra S.K., Creusere C.D., Babic H.

SS5-8 A hierarchical scheme for coding video at 4-10 Mb/s
Ansari R., Guillermot Ch.

SS5-9 Image subband coding using short-kernel cosine modulated filter banks
Aase S.O., Tanem J., Ramstad T.A.,

SS5-10 DFT polyphase transmultiplexer filter banks with effective reconstruction
Fiege N.J.

SS5-11 Applications of a resonator-in-a-loop filter-bank
Martin K.W., Padmanabhan M.

Special Session 6

Floating-Point Quantization Effects in Signal Processing

SS6-1 Floating-point sigma-delta D/A-converter
Kalliojärvi K., Kontro J., Neuvo Y.

SS6-2 Analysis of floating-point quantization errors using stochastic models
Hartwig F., Lacroix A.
SS6-3 How to determine the internal wordlength in floating-point implementations of recursive filters
Laakso T.I., Hartimo I.  

SS6-4 Correlation effects in digital filters operating with floating-point arithmetic
Boite R.  

SS6-5 Design of cascade- or parallel-form digital filters with floating point arithmetic and maximum SNR
Zurawka T., Lüder E.  

SS6-6 Implementation of modified wave digital filters by floating-point digital signal processors
Leickel T., Fettweis A.  

SS6-7 Floating-point realizations for minimum roundoff noise digital filters
Sanz-González J., López-Ferreras F., Andina D.  

Special Session 7
Transmission and Coding of HDTV

SS7-1 Coding of HDTV for multi-Gigabit networks
Johann J.  

SS7-2 Comparison of progressive, interlaced and quincunx-subsampled scanning for reversible predictive coding
Bruyland I., De Lamellileure J., De Vriendt J., D’Haeyer J., Knoll A.  

SS7-3 Subband Coding at 140 Mbit/s of interlaced HDTV signals

SS7-4 Investigations in COST 206 on motion adaptive filtering applied to HDTV
Knoll A.  

SS7-5 Motion compensated sub-nyquist sampling
Belfor R.A.F., Lagendijk R.L., Blemond J.  

SPEECH PROCESSING

Session S1.L

Speech Enhancement

S1.L-1 Speech retrieval in a multimedia system
Glavitsch U., Schäuble P.  

S1.L-2 Lateral inhibition in vowel processing
Grochulewski S., Krenz R.
S1.L-3 Speech enhancement: analysis and comparison of methods on various real situations
*Nguyen Thi H.L., Jutten C., Caelen J.*
303

S1.L-4 Enhancement of speech by using higher-order spectral modeling
*Masgrau E., Fonollasa J.A.R., Ardanuy A.*
307

S1.L-5 A wideband audio codec with a bitrate of 32 kbit/s for realtime implementation on a single DSP
*Fessler P., Thierer G.*
311

Session S2.L

Speech Analysis & Synthesis

S2.L-1 Full-band harmonic-sieve pitch extractor for a harmonic coder
*Carl H.*
315

S2.L-2 An efficient algorithm for pitch prediction using fractional delays
*O’Neill C., Murray B., Fagan A.D.*
319

S2.L-3 Signal processing using a model of the peripheral auditory system
*Fjällbrant T., Mekuria F.*
323

S2.L-4 On robust estimation of LP coefficients in high pitch speech analysis
*Crestel J., Guitton M.*
327

S2.L-5 Minimal rules for articulatory speech synthesis
*Kröger B.*
331

S2.L-6 Nonlinear synthesis of the glottal waveform
*Schoentgen J.*
335

S2.L-7 A simple mixed-excitation synthesizer for manipulating voice quality in mono-pulse excited vocoder applications
*Verhelst W.*
339

S2.L-8 Improving the TD-P SOLA text-to-speech synthesizer with a specially designed MBE re-synthesis of the segments database
*Dutoit T., Leich H.*
343

S2.L-9 Unvoiced speech analysis and synthesis using Poissonian random formant-wave-functions
*Richard G., d’Alessandro C., Grau S.*
347

S2.L-10 Signal synthesis in the time-frequency domain
*Asl M.K., Saleh B.E.A.*
351

Session S3.P

Speech Recognition

S3.P-1 A neural network approach to robust phoneme recognition
*Ming J., Hulhuang C.*
355
S3.P-2 An efficient procedure for robust speaker adaptation using a piecewise linear acoustic mapping  
S3.P-3 Chinese continuous speech recognition using neural network  
Yongsheng C., Biao L., Baozong Y.  
S3.P-4 ECGI stochastic learning and some results for difficult dictionary speech recognition  
Castaño M.A., Vidal E.  
S3.P-5 Reduction techniques for frames and frame dimensions in automatic speech recognition  
Le Cerf P., Van Compernolle D., Van Dijst M.  
S3.P-6 Speech recognizer training behaviour using distance and probabilistic frame scoring  
Van Dijst M., Van Compernolle D.  
S3.P-7 N-best hypotheses: an approach taking into account coarticulation  
Bonafonte A., Martíno J.B., Pardás M.  
S3.P-8 Post-recognizer language processing: applications for automatic speech and handwriting recognition  
Bellegarda J., Braden-Harder L., Jensen K., Kanevsky D., Zadrozny W.  
S3.P-9 Characteristic analysis of phonemes  
Takizawa Y., Oda K., Fukasawa A.  
S3.P-10 Environment adaptation for speech recognition in noise  
Blanchet M., Boudy J., Lockwood Ph.  
S3.P-11 SVD based parameter estimation as noise robust preprocessing for speech recognition  
Flk F.K., Madsen J.P., Micheelsen H., Pharao K.  
S3.P-12 Speaker verification over telephone lines based on digit strings  
Fakotakis N., Tsapanoglou A., Kokkinakis G.  
S3.P-13 Improved automatic recognition of speech in noise and Lombard effect  
Hansen J.H.L., Bria O.N.  
S3.P-14 A medium size vocabulary speech recognizer for Spanish language  
Vellgo M., García Gómez R., Gómez Mena J., Ruiz Mezou B., Gerbolés E.  
S3.P-15 Simple speech classifiers for improved speaker recognition  
Mason J.S., Eatock J.  

Session S4.L  

Speech Recognition  
S4.L-1 Training of hidden Markov models with a genetic algorithm  
Monte E., Ardevol H., Martíno J.B.
S4.1.2 Data reduction of speaker-dependent word recognition system for multi-speaker using CombNET-II
Kitamura T., Hui W., Iwata A., Suzumura N.

S4.1.3 A discriminative training algorithm for speech recognizer based on the neural prediction model
Jun H., Le loh H.

S4.1.4 Phoneme based isolated word recognition using neural networks for prediction and classification
Kasper K., Rehlinger H., Wolf D.

S4.1.5 Building blocks for a neural noise reduction network for robust speech recognition
Trompf M.

S4.1.6 Hidden Markov Models with combined state transition and duration probabilities
Vaseghi S.V., Conner P.

S4.1.7 Context-dependent modeling in a two-stage HMM word recognizer for continuous speech
Kuhn T., Niemann H., Schukat-Talamazzini E.G., Eckert W., Rieck S.

S4.1.8 Fast calculation of statistical properties for languages based on the new recursive Markov model
Nijhuis J.J.

S4.1.9 Phonetic bootstrapping of word models for speaker independent isolated word recognition
Smolders J., Van Campenhout D.

S4.1.10 Robust speech recognition for word spotting
Ortega J., Alvarez J., Lopez E., Hernandez L.

Session S5.1

Speech Coding

S5.1.1 Split-dimension piece-wise vector quantization of PARCOR coefficients
Chan C.F., Law K.W.

S5.1.2 A low data rate LPC vocoder using contour quantization
Chang S., Chen S.-H., Chung C.-J., Hong V.

S5.1.3 Upper bound on redundancy reduction for real speech D.P.C.M. systems
Bojković Z.S.

S5.1.4 On the use of a constant autocorrelation codebook for CELP coding
Steger J.
S5.L-5 Analysis, synthesis and quantization procedures for a 2.5 Kbps voice coder obtained by combining LP and harmonic coding
García C., Banga E.R., Alba J.L., Hernández L.
471

S5.L-6 A new quantization technique for lsp parameters and its application to low bit rate multi-band excited vocoders
Deketelaere S., Lech H., Wery B., Deman I., Dothe M.
475

S5.L-7 A long history quantization approach to low bit rate speech coding
Xydeas C.S., So K.K.M.
479

S5.L-9 A low delay wideband speech codec at 16 kbit/s
Drogo De Iacovo R., Montagna R., Sereno D.
483

S5.L-10 Hybrid harmonic coding of speech
Abrantes A.J., Marques J.S.
487

Session S6.L

Speech Coding

S6.L-1 A low delay adaptive pulse excitation coder
Liu F.-W., Ni W.-Z., Wang D.
491

S6.L-2 Low-delay algebraic CELP coding of wideband speech
Sanchez V., Laflamme C., Salimi R., Adoul J.-P.
495

S6.L-3 Pitch prediction in a wideband CELP coder
Fulseth A., Harborg E., Johansen F.T., Knudsen J.E.
499

S6.L-4 2.4 Kbps speech coding: a new strategy for coding voiced residuals
Haagen J., Nielsen H., Hansen S.D.
503

S6.L-5 Generating the excitation in multipulse coders: a deconvolution based approach
Docampo D., González F., Abreu V., Pérez F.
507

S6.L-6 A fast CELP vocoder with efficient computation of pitch
Yang G., Zaneilato G., Lech H.
511

S6.L-7 Speech quality assessment of a real-time RPE-LTP vocoder
Hansen J.H.L., Nandkumar S.
515

S6.L-8 Subjective evaluations of error masking techniques for bursty channels
Perkis A.
519

S6.L-9 Reduced complexity CELP coders with multistage search
Mauc M., Baudoin G., Jelinek M., Jardin P.
523

S6.L-10 Embedded algebraic CELP coders for wideband speech coding
le Guyader A., Lozach B., Moreau N.
527
Session K1.P

Knowledge Based Engineering

K1.P-1 Secant distance analysis for corner point detection
Möschken J., Hosticka B.J.  

K1.P-2 Low-level segmentation of complex color images
Schettini R.  

K1.P-3 A new fast algorithm to identify the nearest neighbour
Bakamidis S.G., Boutalis Y.S.  

K1.P-4 Solving the ambiguity reject in pattern recognition by multi-step prediction of the membership function
Fredico C., Dubuisson B.  

K1.P-5 Automatic extraction of language models from a linguistic knowledge base
Fink G.A., Kummert F., Sagerer G.  

K1.P-6 On the design of test images for segmentation evaluation
Zhang Y.J., Gerbrants J.J.  

K1.P-7 Speech understanding by alternating control: a semantic network based approach
Seestadt B., Kummert F., Sagerer G.  

K1.P-8 A multi-agent model for signal interpretation
Charpillet F., Fehr D., Gong Y., Haton J.P.  

K1.P-9 Optimal clustering for vector quantizer design
Cocurullo F., Lavagetto F., Moresco M.  

K1.P-10 Classification of remote sensing images by an improved CF approach
Dambra C., Magnanelli L., Pellegretti P., Serpico S.B.  

K1.P-11 Invariant image recognition using triple-correlations
Delopoulos A., Tirakis A., Kollas S.  

K1.P-12 Discontinuity preserving optical flow estimation and segmentation
Geurtz A.M.  

K1.P-13 Integration of a voting scheme and a stochastic relaxation algorithm for edge labelling
Fosetti G.L., Murino V., Regazzoni C.S., Vernazza G.  

K1.P-14 Texture segmentation using 2-D Wigner analysis: an evaluation and comparison
Zhu Y.M., Gautre R., Gao Y., Amiel M.  

K1.P-15 Texture classification and image segmentation using bicumulants and fractals
Nandl A.K., Moe J.P.
Session K2.1

Knowledge Based Engineering

K2.1-1 A comparative study of grey level corner detectors
    Eryurtlu F., Kittler J. 591

K2.1-2 Distributed artificial intelligence techniques in image interpretation
    Roberto V. 595

K2.1-3 Recognition and location of two-dimensional partially occluded objects
    Schettini R. 599

K2.1-4 Estimating ramp-edge attributes from scale-space
    Neumann H., Ottenberg K. 603

K2.1-5 Syntactic parser for pattern recognition of sampled data signals
    Grant P.M., Hamilton R.J., Pringle R.D. 607

K2.1-6 Image identification using two descriptions alternately
    Lin W., Fraser D.A. 611

K2.1-7 Rule-based system for Chinese newspaper segmentation
    Chang W-T., Chang C-W. 615

K2.1-8 Extraction of road networks from digitized maps
    Ogier J.M., Olivier C., Lecourtier Y. 619

K2.1-9 Pitch analysis of noisy speech
    Moreno A., Fonollosa J.A.R. 623

K2.1-10 Two-dimension template-based encoding for quadtree representation
    Chang K-C., Tso C-K. 627
THEORY OF SIGNALS AND SYSTEMS

Session T1.L

Array Processing: Detection

T1.L-1 Improved performance of root-music with sensor perturbations  
Mathur S.  

T1.L-2 Performance characteristics of a generalized Neyman-Pearson detector with fuzzy  
information  
Son J.C., Song I., Kim S., Ill Park S., Yun J.S., Jhee Y.K.  

T1.L-3 Tests for optimizing sensor positions using linear regression  
Zoubir A.M., Grewing T., Böhme J.F.  

T1.L-4 Pattern reconstruction using parametric time-frequency method  
Mars J.  

T1.L-6 Maximum likelihood source location estimation via EM algorithm  
Kraus D., Matwali D., Böhme J.F.  

T1.L-7 An improved algorithm for the spatial analysis of correlated sources  
Grenier D., Dellsle G.Y.  

T1.L-8 A complex adaptive eigensubspace algorithm for DOE or frequency estimation  
and tracking  
Delmas J.P.  

T1.L-9 Exploiting multivariate autoregressive model in the separation of sources  
Rantala S., Suoranta R.  

T1.L-10 A data-based enumeration technique of coherent signals  
Krim H., Proakis J.G.  

Session T2.L

Estimation

T2.L-1 On the equivalence of SVD and TLS based Yule-Walker approaches to frequency  
estimation  
Stolca P., Söderström T., Van Huffel S.  

T2.L-2 Computationally efficient multiple parameter estimation of composite signals in  
noise with periodic components  
Mertins A., Fliege N.J.  

T2.L-3 Threshold-free Bayesian estimation using censored marginal inference  
Quinn A.P.
T2.L-4 Efficient recursive "ML" bearing estimation in non stationary context
Larzabal P., Clérgeot H.

T2.L-5 Efficient implementation of nonlinear Volterra systems via Laguerre orthogonal functions
Aishebelli S.A., Mertzios B.G., Venetsanopoulos A.N.

T2.L-6 The "Z-slice": an identification method for noisy MA systems using higher order statistics
Constantinides A.G., Fleuret E.

T2.L-7 The discrete-time running cross-correlation function estimated from the Laguerre spectra
den Brinker A.C.

T2.L-8 Some robust stability results concerning sparse predictors
Docampo D., Pérez F., Fernández J.R., Abdallah C.

T2.L-9 The scale-angle representation in image analysis with 2-D wavelet transform
Antoine J.-P., Carrette P., Murenzi R.

T2.L-10 Lowering the threshold SNR of Fourier signal subspace
Joutsensalo J., Karhunen J.

Session T3.L

Estimation

T3.L-1 Estimation of the fundamental frequency of a noisy sum of cisoids with harmonic related frequencies - Application to sonar multipaths delay estimation
Ferrari A., Alergrin G., Theys C.

T3.L-2 What is a good estimate?
Combettes P.L., Edmonson W.W.

T3.L-3 Optimal reconstruction of transient signals using deconvolution algorithms
Dabóczi T., Kollár I.

T3.L-4 Unification of several approaches to mean frequency estimation for Doppler determination
Cah G.D., Yardim A.

T3.L-6 Optimal SVD-based determination of number of sinusoids
Tsujii H., Sano A.

T3.L-7 Application of the bootstrapped total least squares (BTLS) estimator in linear system identification
Van Hamme H., Pintelon R.

T3.L-8 Gaussian noise insensitive FIR system identification
Scarano S., Cortina F., Neri A., Jacovitti G.

T3.L-9 Iterative techniques for blind source separation using only fourth-order cumulants
Cardoso J.F.

T3.L-10 Phase reconstruction of autoregressive processes using third order cumulants
Nandi A.K., Mehlam R.
Session T4.L

Spectral Estimation

T4.L-1 Digital calculation of the fundamental sinewave of signals using methods
constrained in frequency domain
Zobos T., Ruczewski P., Szymańska J. 747

T4.L-2 Generalized Wigner-Ville distributions : examples and properties
Boashash B., Ristici B. 751

T4.L-3 Linear prediction, maximum flatness, maximum higher-order entropy on AR
polyspectral estimation
Chi C.Y. 755

T4.L-5 An adaptive ARMA lattice filter for spectral estimation with frequency weighting
Haseyama M., Nagai N., Miki N. 759

T4.L-7 Application of SVD to AR spectral estimation
Rands Jensen P.M., Hermansen K. 763

T4.L-8 A new approach to estimation of the frequency of a noisy cisoid
Ferrari A., Alengrin G., Menez J. 767

T4.L-9 Separation of a mixture of independent sources through a maximum likelihood
approach
Pham D.T., Garat Ph., Jutten C. 771

T4.L-10 A framework for autoregressive theory
Broersen P.M.T., Wensink H.E. 775

Session T5.P

Estimation

T5.P-1 A new identification algorithm for allpass systems by higher-order statistics
Chi C.Y., Kung J.Y. 779

T5.P-2 Optimal Bayesian detection of spectral changes in random signals
Le Carpentier E., Doncarli C., Vozel B. 783

T5.P-3 Lowering the SNR threshold in SVD-based frequency estimation algorithms
Portillo-Garcia J.I., Casar-Corredera J.R. 787

T5.P-4 Degeneracy in non-linear predictors
Mulgrew B., Nisbet K., McLaughlin S. 791

T5.P-5 Polynomial phase signal estimation
Faraj A., Castanie F. 795

T5.P-6 The MV spectral convergence applied to random fields
Lyon D.E., Sherman P.J., Frazho A.E. 799

T5.P-8 A recursive least square approach with exponential
weighting factor to short time Fourier analysis
Gruber P. 803
T5.P-9 Doubly normalized Schur RLS lattice filter and CORDIC array architecture  
   Strobach P.  
   807

T5.P-10 A RLS with an adaptive forgetting factor  
   Carter A., Ruiz P.  
   811

T5.P-11 A fourth order cumulant based adaptive algorithm for the identification of  
   non-linear Hammerstein models  
   Deerga Rao K., Reddy D.C.  
   815

T5.P-12 Bootstrapped spatial separation of wideband superimposed signals  
   Messer H., Bar-Ness Y.  
   819

T5.P-13 Hybrid mean/median Wiener filters  
   Lansford J.  
   823

T5.P-15 Study of MA model identification using order statistics in the presence of noise  
   Tourneret J.-Y., Lacaze B.  
   827

Session T6.L

Detection and Filtering

T6.L-1 A neural network for online estimation of parameters of  
   noisy sinusoidal signals  
   Chaohaki A., Tobos T.  
   831

T6.L-2 Blind Identification in presence of noise  
   Comon P.  
   835

T6.L-3 The $L^\infty$ Wiener problem: an application of the $L^\infty$ extension problem  
   Fajre C.J., Mullis C.T.  
   839

T6.L-4 A non-parametric approach for detecting changes in the autocorrelation structure  
   Tsatsanis M., Giannakis G.B.  
   843

T6.L-5 Signal representation using operators  
   Nordberg K., Knutsson H., Granlund G.  
   847

T6.L-6 Instantaneous frequency and its measurement using digital filtering  
   Čižek V.V.  
   851

T6.L-7 Multichannel multirate sampling of low-pass stochastic signals  
   Kida T., So-Ngunkotchakorn S.  
   855

T6.L-8 Optimization of multiresolution analysis for finite sequences  
   Pesquet J.C.  
   859

T6.L-9 Noise cancellation with uncorrelated reference - a non linear filter approach  
   Cal M., Ball J.  
   863

T6.L-10 Detection of deterministic transient signals  
   Maguer A.  
   867
Session T7.L

Modeling and Interpolation

T7.L-1 A class of recursive Newton-type polynomial extrapolation filters
Ovaska S.J., Vainio O.

T7.L-2 Gibbs random field texture model
Czarnecki W.

T7.L-3 Novel Schur type algorithm for near to Toeplitz linear systems with multichannel entries
Ilavas A., Theodoridis S.

T7.L-4 Using hidden variable fractal interpolation to model one-dimensional signals
Vines G., Hayes Ill M.H.

T7.L-6 All-poles modelling using a linear combination of cumulants
Vidal J., Fonollosa J.A.R.

T7.L-7 Transfer function of linear continuous time-varying systems via non-commutative algebra
Guglielmi M.

T7.L-8 A time-varying Prony model
Molinaro F., Castanie F.

T7.L-9 AAK model reduction for time-varying systems
van der Veen A.-J., Dewilde P.

T7.L-10 Optimum interpolation for time-limited and band-limited signals
Neagoe V.

Session T8.L

Digital Transforms

T8.L-1 Nonuniform discrete Fourier transform and its applications in signal processing
Mitra S.K., Chakrabarti S., Abreu E.

T8.L-2 Uniqueness and stability of discrete zero-crossings and maxima wavelet representations
Berman Z., Baras J.S.

T8.L-3 A simple and reliable phase unwrapping algorithm
Krajnik E.

T8.L-4 2-D FFT algorithms with parallel bit reversion on hypercube computers
Angelopoulos G., Pitas I.

T8.L-5 Time-warped polynomials for signal representation and coding
Phillips W.

T8.L-7 Design of signal-matched wavelets - theory and results
Greiner T., Eberle E., Pandit M.
T8.L-8  A new fast algebraic convolution algorithm  
      Minkwitz T., Creutzburg R.  
      933

T8.L-9  Simple, optimal regularity estimates for wavelets  
      Rioul O.  
      937

T8.L-10 Certain aspects of fixed point FFT and WFTA rounding errors comparison  
      Łukasik E., Stasiński R.  
      941

Session T9.L

Digital Filtering

T9.L-1  A novel fast FIR filtering algorithm and its implementation  
      Chen S-G., Tsai R.J.  
      945

T9.L-2  A study of the filter roundoff noise in subband coding systems  
      Renard J.P., Leich H.  
      949

T9.L-3  Design of finite wordlength 2-D IIR filters using simulated annealing  
      Radecki J., Konrad J., Dubois E.  
      953

T9.L-4  A simple design method for FIR filters with integer coefficients using a personal computer  
      Nakamura S., Morishita Y., Hario Y., Kawahashi A., Kadomaki Y.  
      957

T9.L-5  Synthesis methods for wave digital filters exhibiting arbitrary amplitude characteristics  
      Yaseen M., Henk T.  
      961

T9.L-6  Efficient temporal band-limitation based on motion adaptive spatial filtering  
      Kim J.H., Kim S.  
      965

T9.L-7  Generalized theory of multidimensional M-band filter bank design  
      Shah I.A., Kaliker A.A.C.  
      969

T9.L-8  A new WLS Chebyshev approximation method for the design of FIR digital filters with arbitrary complex frequency response  
      Chi C.Y., Chiou S.L.  
      973

T9.L-10  Modulated filter banks in multidimensions with arbitrary sampling lattices  
      Tay D.B.H., Kingsbury N.G.  
      977

Session T10.L

Digital Filtering

T10.L-2  Oversampled digital leapfrog filters  
      Corthay F., Pellandini F.  
      981

T10.L-3  Using the root datasieve to recognise and locate patterns in a data sequence  
      Bangham J.A.  
      985
XXX

T10.L-5 A Bezout resultant based stability test for 2-D digital recursive filters
Benidir M., Barret M. 989

T10.L-6 FIR digital filters using a least-squares derived recursive section
Horrocks D.H., Bull D.R. 993

T10.L-7 Explicit description of lattice state-space wave digital filters
Ansorge M., Pellandini F. 997

T10.L-8 Design of two-dimensional video filters with spatial constraints
Ouvrard V., Slohan P. 1001

T10.L-9 Improved frequency-sampling two-dimensional FIR filter design
Lightstone M., Mitra S.K. 1005

T10.L-10 Optimal implementation of multi-channel filter banks
Vertest E.L., Li G. 1009

Session T11.P

Filtering and Transforms

T11.P-1 Computing the cardinality of root signal sets of morphological filters
Wang G., Gabbouj M., Neuvo Y. 1013

T11.P-2 A new moment-based method for low noise ordering of cascade form FIR
digital filter
Khare A., Yoshikawa T. 1017

T11.P-3 Multirate discrete Fourier transform
Tadokoro Y., Yamanaka S., Ishihara A. 1021

T11.P-4 Analysis of quantization noise in frequency sampling FIR filters and its reduction
via error feedback
Gold D., Ur H. 1025

T11.P-5 Sensitivity and roundoff noise in wave digital filters versus optimal state-space
realizations
Sanz-González J., Rodrigo-Aparicio E., Lema F. 1029

T11.P-6 Lumped, sampled-data and digital filters with maximum number of constraints
on amplitude and phase characteristics
Henk T., Abo-Zahhad M. 1033

T11.P-7 Digital filters based on complex transmission line networks
Takahashi A., Nagal N., Miki N. 1037

T11.P-9 Hints on the relationships between subband coding,
wavelet and vaguelette decompositions and cosine sine
altemned transformations
Maes S. 1041

T11.P-11 Prime factor real-valued Fourier, Cosine and Hartley
transforms
Chan S.C., Ho K.L. 1045
Session T12.L

Adaptive Systems

T12.L-1 Active noise cancellation using a modified form of the filtered-X LMS algorithm
       Bjarmanson E.

T12.L-2 A pipelined adaptive lattice filter architecture: theory and applications
       Shanbhag N.R., Parhi K.K.

T12.L-3 Blind deconvolution of multivariate signals using adaptive FIR lossless filters
       Loubaton P., Regalia Ph.

T12.L-4 Relation between reduced dimension time and frequency domain adaptive algorithm
       Egelmeers G.P.M., Sommen P.C.W.

T12.L-5 An alternative scheme for adaptive nonlinear prediction using radial basis functions
       Vesin J.M.

T12.L-6 Convergence analysis of stochastic gradient adaptive algorithms for arbitrary error surfaces
       Páez Borralló J.M., Casajús Quirós F.J.

T12.L-7 Nonlinear effects in adaptive filters of the LMS type
       Butterweck H.J.

T12.L-8 Feedforward and feedback in a symmetric adaptive noise canceler: "stability analysis in a simplified case"
       Van Gerven S., Van Compernolle D.

T12.L-9 High order learning in temporal reference array beamforming
       Lagunas M.A., Pagés-Zamora A., Pérez-Neira A.

T12.L-10 An adaptive morphological filtering method
       Kuosmanen P., Koskinen L., Astola J.

Session T13.L

Adaptive Systems

T13.L-1 A realization method of 2-D separable-denominator digital filter based on reduced-dimensional decomposition
       Kawasaki S.-J., Hamada N.

T13.L-2 Adaptive prediction and interpolation filters for radar data processing
       Tuszyński M., Wojtklewicz A.
XXXII

T13.L-3 Efficient algorithms for second order polynomial system identification
Glentis G.O., Kalouptsidis N. 1101

T13.L-4 Fast linear phase adaptive filtering using constrained least squares algorithm
Resende L.S., Romano J.M.T., Bellanger M. 1105

T13.L-5 A blind adaptive equalizer based on a lattice/all-pass configuration
Jelonnek B., Kammeyer K.D. 1109

T13.L-6 On the reduction of nesting algorithms rounding errors
Stański R. 1113

T13.L-7 Wavelet transform based adaptive filtering
Erdöl N., Basbuğ F. 1117

T13.L-8 An adjustable constraint approach for robust adaptive beamforming
Castedo L., Tseng C-Y., Griffiths L.J. 1121

T13.L-9 Transient and steady-state behavior of the 2D fast RLS lattice algorithm
Liu X., Mandridake L., Najim M. 1125

T13.L-10 A comparative fixed-point error analysis of the Schur and split-Schur algorithms
Glaros N., Carayannis G. 1129

Session T14.L

Filtering

T14.L-1 Multidimensional wave digital filters by use of transformation principle
Wenzhe L. 1133

T14.L-2 Adaptation of grey level structuring elements for morphological filters with
application to shape detection
Salember P. 1137

T14.L-3 Use of bitonic sorters for rank order filtering
Akaran L., Haddad R.A., Ersoy C. 1141

T14.L-4 Noise reduction with correlated signals using higher order statistics
Serviere Ch., Baudois D. 1145

T14.L-5 On the use of instantaneous phase variance for clutter suppression
Albtol A., Vergara L. 1149

T14.L-6 Study of generalized linear interpolation approximation for multi-dimensional
waveforms
Kida T., Mochizuki H. 1153

T14.L-7 Detail-preserving filters with improved lowpass characteristics
Defée I., Soininen R., Neuvo Y. 1157

T14.L-8 Synthesis of inherently stable and low-sensitive 2-D IIR filters using the $l_1$-passivity
concept
Domański M. 1161
NEURAL NETWORKS FOR SIGNAL PROCESSING

Session N1.L

Neural Networks

N1.L-1 Classification of texture by an association between a perceptron and a self-organizing feature map
Maillard E., Zerr B., Merkle J.

N1.L-2 Two-layer learning vector quantizer for color image quantization
Kotropoulos C., Augé E., Pitas I.

N1.L-3 Selection of hidden layer nodes in neural networks by statistical tests
Ciftaloglu O., Türkcan E.

N1.L-4 Convergence analysis of the hamming network
Kotropoulos C., Kalouptsidis N.

N1.L-5 A novel algorithm for training a bidirectional associative memory
Melia-Predoitiu M., Stoica P.

Session N2.P

Neural Networks

N2.P-1 A bank of multilayer perceptrons for the on-line recognition of handwritten characters
Bellegarda E., Kim J., Bellegarda J.

N2.P-2 Left ventricular boundary detection in echocardiographical images using artificial neural networks
Xin Y., Boekaerts P., Bister M., Taeymans Y., Cornelis J.

N2.P-3 A specific processor for the computation of TDNN algorithms with applications in phonetic coding
Pérez-Castellanos M., Rodellar V., Pelnado V., Díaz A., Gómez-Vilda P.

N2.P-4 A sample protocol for the Hopfield neural network
Klijn Hesselink H.W., de Bruin F.F.G., Veelenturf L.P.J.
N2.P-5  A comparison of error diffusion and the Hopfield neural network as methods to quantize hologram distributions
van den Bulcke P.  1211

N2.P-6  Neural prediction in image vector quantization
De Natale F.G.B., Desoli G.S., Floravanti S., Giusto D.D.  1215

N2.P-7  New approach for multi-perceptron architecture design applied to the edge detection problem
Molina Ch., Bastard D., Baylou P., Najim M.  1219
VOLUME III

IMAGE PROCESSING AND MULTIDIMENSIONAL SIGNAL PROCESSING

Session II.1

Image Coding

II.1.1 Hierarchical finite state vector quantization for image coding
   Yu P., Venetsanopoulos A.N.

II.1.2 A fast algorithm for vector quantization: application to codebook generation in image subband coding
   Akrout N., Allart C., Diab C., Prost R., Goutte R.

II.1.3 An efficient encoding algorithm for vector quantization of images
   Lo K., Cham W.

II.1.4 A generalized product-code model for structured vector quantization of audio and visual signals
   Chan W-Y., Gersho A.

II.1.5 Adaptive multi-stage vector quantization of images
   Salillas J., Farvardin N.

II.1.6 Variable block-size vector quantization in the transform domain
   Plansky H.

II.1.7 On the Hadamard-Haar transforms and their multiresolution properties
   Drygajlo A.

II.1.8 Adaptive transform coding of images based on variable-shape-blocks
   Itoh S., Matsuda I., Utsunomiya T.

II.1.9 An adaptive orthogonal transform for image data compression
   Phillips W.

II.1.10 Comparative analysis and simulation of image signal decomposition techniques
       Fazekas K., Csillag P.

Session II.2

Image Coding

II.2.1 A 3-dimensional subband coding scheme with motion-adaptive subband selection
       Queluz M.P.

II.2.2 Context algorithm applied to sub-band coding of images
       Furlan G., Galiani C., Lanoon E., Menez J.

II.2.3 Signal-adapted subband coding for video distribution
       Cornes S., Macq B., Van Cailie B.
II. L-4  CQF filter banks optimized for subband zonal sampling and distortion minimization  
Vandendorpe L.  
1275

II. L-6  Frame rate conversion of motion picture films for progressive -scan HDTV  
Sezan M.I., Lagendijk R.L.  
1279

II. L-7  An algorithm for line registration of TV images based on a 2-D AR model  
Kokaram A., Rayner P.  
1283

II. L-8  Intraframe processing of interfaced video: analysis and applications  
Shah I.A., Beuker R.  
1287

II. L-9  Constrained motion estimators for 3DTV sequence coding  
Tarifaoui A., Labit C.  
1291

II. L-10  Motion and disparity analysis of a stereoscopic sequence application to 3DTV coding  
Dugelay J.-L., Pelé D.  
1295

Session II. L

Image Coding

II. L-1  Perceptually-based multi-channel representation for image sequence coding  
Wegmann B., Zetzsche C.  
1299

II. L-2  A 3-dimensional morphological image sequence coding  
Chen R.J., Chieu B.C.  
1303

II. L-3  Image sequence coding using adaptive weighted median prediction  
Song X., Yin L., Neuvo Y.  
1307

II. L-4  A modification of the conjugate direction method for motion estimation  
Paúndez Zanuy M., Tarrés Ruiz F.  
1311

II. L-5  Motion compensated recursive filtering of noisy displacement fields  
Estratialis S.N., Katsaggelos A.K.  
1315

II. L-6  Displacement field estimation in noisy image sequences  
Brailean J.C., Katsaggelos A.K.  
1319

II. L-7  Multigrid based motion estimation for interframe image sequence coding  
Dufaux F., Kunt M.  
1323

II. L-8  Adaptive regularization for motion field estimation  
Zheng H., Blostein S.D.  
1327

II. L-9  An image sequence coder based on vector quantization and progressive transmission  
de Geus P.L.  
1331

II. L-10  Wavelet transform and lattice V.Q. for image sequence coding  
Galdon T., Barlaud M., Mathieu P.  
1337

II. L-11  A new pel-recursive Kalman-based motion estimation method  
Tziritas G.  
1341
Session 14.1

Image Coding

14.1-1 Design of perfect reconstruction, finite word length, nonuniform, and parallel-structured filter banks
   Li W., Basso A., Nicolau A., Kunt M. 1345

14.1-2 Adaptive DCT color image sequence coding
   Chen R.J., Chieu B.C. 1349

14.1-3 Reducing the complexity of a fractal-based image coder
   Ølen G.E., Lepsøy S., Ramstad T.A. 1353

14.1-5 Motion adaptive vector quantization for video coding
   Moresco M., Lavagetto F., Cucurullo F. 1357

14.1-6 A video codec based on perceptually derived and localized wavelet transform for mobile applications
   Ebrahimi T., Kunt M. 1361

14.1-7 A pyramidal vector quantization approach to transform domain
   Mocagatta L., Kunt M. 1365

14.1-8 Predictive control of variable bit rate coders
   Ronda J.I., Ortega A., Fernández A., Garola N. 1369

14.1-9 An efficient image compression algorithm based on filter bank analysis and fractal theory
   Tenehnao M., Kozarev A., Trpovski Z., Simslo B. 1373

14.1-10 Adaptive image coding with harmonic angular filters
    Cusani R., Jacovitti G. 1377

Session 15.1

Image Enhancement & Restoration

15.1-1 2-D Adaptive nonlinear equalizers
   Lin J., Unbehauen R. 1381

15.1-2 Motion compensated noise filtering of image sequences
   Kleihorst R.P., de Haan G., Lagendijk R.L., Biemond J. 1385

15.1-3 Generalized iterative algorithm for spectral signal deconvolution
   Van Droogenbroeck M. 1389

15.1-4 Adaptive nonlinear filter for nonstationary noise
   Abdurahim A., Dobrowiecki T. 1393

15.1-5 Infrared image enhancement with nonlinear spatio-temporal filtering
   Meier W., vom Stein H.D. 1397

15.1-6 Multiform enhancement by means of locally adaptive rank-order filters for the preprocessor of natural images
   Amiri H., Zamperoni P. 1401
XXXVIII

15.L-7 A simple synthesis method for weighted median filters
Sun T., Neuvo Y. 1405

15.L-8 An approach to the automatic filtering of images
Chehdi K. 1409

15.L-9 Working on image data using fuzzy rules
Russu F., Ramponi G. 1413

15.L-10 Achieving super-resolution with band-pass images
Salerno E.A. 1417

Session 16.L

Medical Image Processing

16.L-1 A unified approach to tomographic image reconstruction from incomplete
projections
Saito T., Kudo H. 1421

16.L-2 Quantitative description and reconstruction of intensity functions using
scale-space and multiresolution processing
Ottenberg K., Neumann H., Stiehl H.S. 1425

16.L-3 A blackboard system for analysing brain CT-scans
LI H., De Cuyper B., Bister M., Nyssen E., Cornells J. 1429

16.L-4 Multiresolution texture analysis-tissue characterization based on ultrasonic images
Greiner T., Mettach H., Pandit M., Kumaresan R. 1433

16.L-5 Three dimensional image registration based on parameter accumulation
Chen G-S., Defrise M., Deconinck F. 1437

16.L-6 Image registration by a region-based approach and by correction of elastic
deformations
Semin A., Trucco P., Venturi G., Dellepiane S. 1441

16.L-7 Edge preserving tomographic reconstruction from few projections
Bedini L., Benvenuti L., Salerno E.A., Tonazzini A. 1445

16.L-8 2-D maximum entropy spectrum estimation in computerized tomography - an
approach of closed form solution
Yuan Y., Durrani T.S. 1449

16.L-9 Fuzzy voxel labeling of magnetic resonance images by global optimization
Verbeeck R., Vandermeulen D., Delaere D., Suetens P., Marchal G. 1453

Session 17.P

Image Processing

17.P-1 On computing multifractality for texture discrimination
Arduini F., Floravanti S., Glusto D.D. 1457
17.P.2 A 2-D joint process lattice filter for adaptive change detection in image sequence
Youali H., Janati M., Najim M.

1461

17.P.3 Local and isotropic error diffusion: why and how ?
Bernard T.

1465

17.P.4 Multichannel histogram equalization with applications in the HSI color space
Pitas I., Kiriklis P.

1469

17.P.5 Non-linear techniques for image interpolation
Gasull A., Marqués F., Fernández A., Torres L.

1473

17.P.6 A new construction of Trellis waveform coders
van der Vleuten R. J., Weber J. H.

1477

17.P.7 Multivariate ordering and filtering with application to color image processing
Tang K., Astola J., Neuvo Y.

1481

17.P.8 The use of different metrics in vector median filtering: application to fine arts and
paintings
Barl M., Cappellini V., Mecocci A.

1485

17.P.9 The recursive implementation of the optimal filter for the detection of roof edges
and thin lines
Petrou M., Kolomvats A.

1489

17.P.10 Optimisation of distortion-rate in image coding: an application of wavelet
packets
Haddad P., Barlaud M., Mathieu P.

1493

17.P.11 Control of the image quality for digital TV and HDTV codecs
Leduc J.-P.

1497

17.P.12 Image data compression using nonlinear pyramid vector quantization
Song X., Neuvo Y.

1501

17.P.14 Delayed-decision pattern-matching lossy image coding via copying
Salto T., Moteki M., Takemasa E., Komatsu T.

1505

IMPLEMENTATIONS

Session V1.P

Implementations

V1.P.1 Alternative DPCM architecture for high sampling rates
Sundsbø I., Timmermann D.

1509

V1.P.2 A new systolic array for 2-D FIR digital filtering and its applications
Wang C.L., Peng H.L.

1513

V1.P.3 VLSI implementations for a new Video Codec
Kowalczuk J., Hervigo R., Ebrahimi T., Dufaux F., Maccagatta L., Miynek D., Kunt M.

1517
V1.P-4 VLSI implementation of a modified-VQ encoder suitable for image/video applications
Lee C.Y., Juan S.C. 1521

V1.P-5 Software implementation of 2D FFT on DSP 96002 digital signal processor
Chan S.C., Wong Y.K. Ho K.L. 1525

V1.P-6 DPCM video signal/image cosine transform
Aggoun A., Ibrahim M.K. 1529

V1.P-7 Design and realization of high sampling rate quadrature demodulators
Eiterich A., Kleinheppel W., Stammler W. 1533

V1.P-8 Interference cancellation using a switched-capacitor adaptive filter (LMS algorithm)
Campos P.F. 1537

V1.P-9 Real-time implementation of a CELP codec with unequal error protection
Clüver K., Gries T., L H., Noll P. 1541

V1.P-10 ASIC solution for real-time inverse kinematic and Jacobian computations
Koosts F., Ort K., Böhme J.F. 1545

Session V2.1

Novel Architectures for Signal Processing

V2.1-1 Digital filter bank multiplier block implementation
Wells I. 1549

V2.1-2 An RNS implementation of a length-8 DCT based on a cyclic convolution algorithm
Wolter S., Laur R. 1553

V2.1-3 A codec chip architecture for (24,12,8) Golay code with the permutation decoding
Guo N., Cao W., Spinnler W., Eberlein E., Seitzer D. 1557

V2.1-4 Very high speed CORDIC implementation : algorithm transformation and novel carry-save architecture
David H., Meyr H. 1561

V2.1-5 A very fast algorithm for image display with a limited number of colors
Lancini R., Migliorati P., Pecora D. 1565

V2.1-6 An application specific multi-precision multiplier architecture for a dynamic range compressor for audio
Pauwels M., Thoen F., Beeckmans E., Goossens G., Catthoor F., De Man H. 1569

V2.1-7 Randomizing of array element spacing and processing array signals
Bilinskis I., Lagunas M.A. 1573

V2.1-8 Analog implementation of fixed-coefficient and adaptive FIR filters based on Σ – Δ oversampling
Qinuting H. 1577
APPLICATIONS

Session A1.1

Spread Spectrum

A1.1-1 The effect of interpolation on the BER performance of bandlimited direct-sequence spread-spectrum systems

Moeneclaey M., Bucket K. 1587

A1.1-2 Adaptive CDMA co-channel interference cancellation

Mowbray S., Pringle R.D., Grant P.M. 1591

A1.1-3 Spread spectrum receiver design for a multipath environment

Povey G.J.R., Grant P.M., Pringle R.D. 1595

A1.1-4 Communication using spread spectrum modulation on electric power distribution circuits

Vuori J., Skyttä J. 1599

Session A2.1

Echo Cancellation and Equalization

A2.1-1 An investigation of near-optimal receiver structures using the M-algorithm for equalization of dispersive fading channels

Fechtel S.A., Meyr H. 1603

A2.1-2 Pre-echo reduction by optimum filtering

Mahieux Y., Petit J.P., Le Guyader A. 1607

A2.1-3 Wideband acoustic echo canceller with two filter structure

Armbruster M. 1611

A2.1-4 A fast RLS look-up-table plus FIR adaptive echo canceller

Weruaga-Prieto L., Cid-Suelto J., Figueiras-Vidal A.R. 1615

A2.1-5 On efficient IIR equalization of subscriber loops exploiting a priori knowledge of channel characteristics

Ebringer L., Young G. 1619

A2.1-6 Blind equalization using a fast self-orthogonalizing algorithm

Wu W.R., Hsieh M.J. 1625

A2.1-7 Acoustic echo canceller for hands-free mobile radiotelephony

Jensen S.H. 1629

A2.1-8 Study of a noise cancellation system based on the coherence function

Le Bouquin R., Faucon G. 1633
A2.L-9 Echo cancellation in the multicarrier modulation data transmission systems
   Bogucka H., Wesołowski K.

A2.L-10 Combined implementation of predictive speech coding and acoustic echo cancellation
   Acker C., Vary P.

Session A3.L

Data Communications

A3.L-1 Data-pump Implementation for automatic interworking between automode modems and other CCITT & Bell modems
   Alvarez A.M., S.I.S.M.

A3.L-2 Real time simulation for the ISDN U-transceiver adaptive filters
   Begusíc D., Grenèz F.

A3.L-3 A new technique of data over voice based on time-warping
   Wullisch D., Bukris M.

A3.L-4 A novel frequency domain method for symbol timing recovery and its application in multi-carrier demodulators
   Al-Jailli Y.O., Barton S.K.

A3.L-5 On the classification of the multilevel phase shift keying signal
   Jovanović S., Djurić P.

A3.L-6 Simultaneous frequency and timing estimation from signal subspace in digital modulations
   Cabrera M., Lagunas M.A.

A3.L-7 Equivalent channel for audio-hidden digital transmission
   Roviras D., Castañé F.

A3.L-8 Adaptive predictors for carrier jitter suppression in highly efficient digital modems
   Diez del Rio L., García Gómez R.

A3.L-9 DSP based carrier recovery technique for OQPSK mobile satellite communications
   Ahmad J., Jeans T.G., Evans B.G.

A3.L-10 Improvement of non-data aided phase estimation by thresholding
   Gnythe K.

Session A4.L

Data Communications

A4.L-1 On the performance of vector-quantizer under channel errors
   Ben-David G., Malah D.

A4.L-2 Adaptive algorithms for MCD applications over Rician fading channels
   Coloma J., Carrasco R.A.
A4.L-3 Digital modulation trade-offs for modern systems
   Acha V., Carrasco R.A.

A4.L-4 An improved adaptive MLSE receiver for fading communications channels
   Del Re E., Castellini G., Pierucci L., Conti F.

Session A5.L

Digital Audio

A5.L-1 Analysis of drum sounds using higher order spectra
   Hoveil S.A., Mulgrew B.

A5.L-2 A digital signal processing algorithm for the linearisation of PWM based DACs for
digital audio applications
   Goldberg J.M., Sandler M.B.

A5.L-3 A novel filtered-x LMS algorithm and its application to active noise control
   Bao C., Sas P., Van Brussel H.

A5.L-4 Simulation of room acoustics using a digital signal processor
   Berkhoff A.P., Slump C.H., Herrmann O.E., Hoenders B.J.

A5.L-5 Implementation of real-time synthesis of string and wind instruments on a
   floating-point signal processor
   Karjalainen M., Välimäki V.

Session A6.L

Biomedical Applications

A6.L-1 Improved quantitative time-domain analysis of NMR
data by total least squares
   Van Huffel S., Aerts L., Bervoets J., Vandewalle J., Decanniere C., Van Hoeke P.

A6.L-2 3D Modeling of left ventricle tomograms using Kohonen feature maps
   Manhaeve C., Lemahieu I., Vogelaers D.

A6.L-3 Detection, extraction and spectral analysis of sleep spindles in human EEG
   Dorr C., Cecchin T., Sauter D. Di Renzo N., Mouze-Amady M.

A6.L-4 Spectral maps by AR models as a support to conventional ultrasonic imaging
   Fort A., Manfredi C., Masotti L., Rocchi S.

A6.L-5 Nonlinear interference cancelling in biomedical systems
   Karrakchou M., Loup F., Vesin J.M., Feith F., Perret C., Kunt M.

A6.L-6 A new class of filters to remove artifacts from physiological signals
   Väätä A., Neeljärvi J., Neuvo Y.

A6.L-7 Comparison of AR-based classifiers for respiratory sounds
   Engin T., Güler C., Sankur B., Kahya Y.
A6.L-8 An intelligent spin imaging signal processor (ISISP) for the in vivo MR domain
van Tongeren B.P.O., de Beer R., Mehlkopf A.F., van Ormondt D.

A6.L-9 Comparison of three cluster distances in a supervised recognition system for
campimetric diagnosis
Captantlo L., Fillipoj G.C.

Session A7.P

Miscellaneous Applications

A7.P-1 Digital signal processing for an open-loop fibre-optic gyroscope
Göckler H.G., Gebauer T.

A7.P-2 Some robustness aspects of local validation algorithms. Application to the
monitoring of an eddy-currents sensor
Brüe D., Cecchini T., Sauter D.

A7.P-3 Time-frequency analysis of knock in spark ignition engines
Bossmeyer T.M., Hansknecht D.J., König D.

A7.P-4 OQAM systems: digital techniques and synchronization
Mendia J., Carrasco R.A.

A7.P-5 Controlling three-dimensional beam-pattern by “spatial frequency modulation”:
theoretical and simulation results
Roeckerath-Rles M.-Th., Cibris J.

A7.P-6 Signal-processing based acquisition of very high resolution images using multiple
cameras
Saito T., Igarashi T., Komatsu T. Alzawa K.

A7.P-7 3D Image analysis for agricultural robotics. The potato operation
Gil S., Lefebvre M., Glassey M.-A., Baur C., Pun T.

A7.P-9 A pragmatic approach to resolution threshold of beamspace MUSIC
Ming Lù

A7.P-10 Wide band antenna processing in the case of industrial noise source
Schmitt F., Gonneau E., Guelhote J.-P.

A7.P-11 An FFT-based speed detector for inverter fed induction motors
Ferrah A., Bradley K.J., Asher G.M.

A7.P-12 Ground resistivity measurement with transputers
Lamarque G., Gasnier S.

Session A8.L

Radar, Sonar and Seismic Applications

A8.L-1 Signal parameter estimation for seismic data
Maiwald D., Kraus D., Böhme J.F.
A8.I.2 A cyclic beamforming method for DOA estimation
Gelli G., Izzo L. 1805

A8.I.3 Target tracking from distributed sensors and simultaneous sensors field calibration
Ywanne F., Martinet F., Rabadar C. 1809

A8.I.4 Reflectivity study of time-varying radar targets
Bertrand J., Bertrand P. 1813

A8.I.5 Detection of hydroacoustic transients by means of higher-order statistics
Persson L., Sangfelt E. 1817

A8.I.6 Data association and tracking from distributed sensors using hidden Markov models
and dynamic programming: the diamant algorithm
Martinet F., Forster P. 1821

A8.I.7 Estimation of the autocorrelation coefficients of band limited signals: Applications
to the sources localization problem
Chonavel T., Loubaton P. 1825

A8.I.8 Radar signal processing for vehicle speed measurements
Kleinheppel W., Stammmer W., Bergmann D. 1829

A8.I.9 Cepstrum analysis of the radar return signal from rotating aircraft blades
Martin J., Mulgrew B. 1833

A8.I.10 Direction of arrival estimation under perturbed conditions
Tseng C-Y., Feldman D.D., Griffiths L.J. 1837

A8.I.11 A multichannel adaptive filter for the suppression of surface related multiple
reflections in marine seismograms
Rosenberger A. 1841
Index

Aase S.O., 231
Abdallah C., 697
Abdulrahim A., 1393
Abo-Zahhad M., 1033
Abrantes A.J., 487
Abreu E., 909
Abreu V., 507
Abry C., 159
Acha V., 1693
Acker C., 1641
Adoul J.-P., 495
Aerts L., 1721
Aggoun A., 1529
Ahmad J., 1677
Ait Amrane O., 115
Aizawa K., 1777
Akarun L., 1141
Akrout N., 1227
Alba J.L., 471
Albiol A., 1149
Alegrín G., 709
Alegrín G., 767
Al-Jalili Y.O., 1657
Allard C., 1227
Alshebeli S.A., 685
Alvarez A.M., 5645
Alvarez J., 451
Ambard P.O., 91
Amiel M., 583
Amiri H., 1401
Andina D., 269
Angelopoulos G., 921
Ansari R., 227
Ansorge M., 997
Antoine J.-P., 23
Antoine J.-P., 701
Ardanuy A., 307
Ardevol H., 415
Arduini F., 1457
Armbruster M., 1611
Asher G.M., 1793
Ask M.K., 361
Astola J., 1089
Astola J., 1481
Augé E., 1177
Babic H., 223
Bahl L., 369
Bailly G., 159
Bakamidis S.G., 539
Ball J., 863
Bang A.R., 471
Bangham R.A., 985
Bao C., 1709
Baozong Y., 363
Baras J.S., 913
Barlaud M., 1337
Barlaud M., 1493
Bar-Ness Y., 819
Barni M., 1485
Barret M., 989
Barton S.K., 1667
Bassu E., 1117
Basso A., 1345
Bastard D., 1219
Baudoin G., 523
Baudoin D., 1145
Baur C., 1781
Baylou P., 1219
Beck C., 183
Bedini L., 1445
Beeckmans E., 1569
Begus D., 1649
Belfor R.A.F., 291
Bellanger M., 47
Bellanger M., 1105
Bellegarda E., 1195
Bellegarda J., 359
Bellegarda J., 383
Bellegarda J., 1195
Ben-David G., 1685
Benidir M., 989
Benvenuti L., 1445
Bergmann D., 1829
Berkhoff A., 1713
Berman Z., 913
Bernard T., 1465
Bertrand J., 1813
Bertrand P., 1813
Bevoets J., 1721
Beuker R., 1287
Biemond J., 1385
Biemond J., 291
Bierens L.H.J., 103
Bilinski L., 1573
Biqing L., 363
Bister M., 1199
Bister M., 1429
Bjarnason E., 1053
Blanchet M., 391
Blöstein S.D., 1327
Boashash B., 751
Boë J., 159
Boekaerts P., 1199
Bogucka H., 1637
Böhme J.F., 1545
Böhme J.F., 1801
Böhme J.F., 641
Böhme J.F., 649
Boîte R., 191
Boîte R., 265
Bojković Z.S., 463
Bonafonte A., 379
Bonnet M., 111
Borys A., 195
Bossmeyer T.M., 1765
Boudy J., 391
Bourlard H., 63
Boutalis Y.S., 539
Braden-Harder L., 383
Bradley K.J., 1793
Brallean J.C., 1319
Brandenburg K., 211
Braszak A., 127
Bria O.N., 403
Brie D., 1761
Broersen P.M.T., 775
Bruijland I., 277
Bruyland I., 281
Bu J., 107
Bucket K., 1587
Bukris M., 1653
Bull D.R., 993
Butterweck H.J., 1077
Cabrera M., 1665
Caeelen J., 303
Cai M., 863
Cain G.D., 723
Callender C.P., 143
Campos P.F., 1537
Cao W., 1557
Capitanio L., 1753
Capodiferro L., 281
Cappellini V., 1485
Carayannis G., 1129
Carayannis G., 81
Cardoso J.F., 739
Carl H., 315
Carrasco R.A., 1689
Carrasco R.A., 1693
Carrasco R.A., 1769
Carre R., 165
Carrette P., 701
Carter A., 811
Casadevall F.J., 139
Casajús Quiró F.J., 1073
Casar-Corredor J.R., 787
Castanie F., 1669
Castanie F., 795
Castanie F., 897
Castaño M.A., 367
Castedo L., 1121
Castellini G., 1697
Catthoor F., 15
Catthoor F., 1569
Cecchin T., 1729
Cecchin T., 1761
Chakrabarti S., 909
Cham W., 1231
Chan C.-F., 455
Chan S.C., 1045
Chan S.C., 1525
Chan W-Y., 1235
Chang C.-W., 615
Chang K-C., 627
Chang S., 459
Chang W-T., 615
Charpillet F., 559
Chehdi K., 1409
Chen G-S., 1437
Chen R.J., 1303
Chen R.J., 1349
Chen S-G., 945
Chen S.-H., 459
Chi C.Y., 755
Chi C.Y., 779
Chi C.Y., 973
Chieu B.C., 1303
Chieu B.C., 1349
Chieu S.L., 973
Chift N.N., 1049
Chonavel T., 1825
Chung C.-J., 459
Cibils J., 1773
Cichocki A., 831
Cid-Sueiro J., 1615
Ciftcioglu O., 1181
<table>
<thead>
<tr>
<th>Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Čžek V.V.</td>
<td>851</td>
</tr>
<tr>
<td>Claassen T.A.C.M.</td>
<td>1</td>
</tr>
<tr>
<td>Clergeot H.</td>
<td>681</td>
</tr>
<tr>
<td>Cluver K.</td>
<td>1541</td>
</tr>
<tr>
<td>Cocurullo F.</td>
<td>1357</td>
</tr>
<tr>
<td>Cocurullo F.</td>
<td>563</td>
</tr>
<tr>
<td>Collette T.</td>
<td>1581</td>
</tr>
<tr>
<td>Coloma J.</td>
<td>1689</td>
</tr>
<tr>
<td>Combettes P.L.</td>
<td>713</td>
</tr>
<tr>
<td>Comes S.</td>
<td>1271</td>
</tr>
<tr>
<td>Comon P.</td>
<td>835</td>
</tr>
<tr>
<td>Conner P.</td>
<td>435</td>
</tr>
<tr>
<td>Constantinides A.G.</td>
<td>689</td>
</tr>
<tr>
<td>Conti F.</td>
<td>1697</td>
</tr>
<tr>
<td>Corina F.</td>
<td>735</td>
</tr>
<tr>
<td>Cornelis J.</td>
<td>1199</td>
</tr>
<tr>
<td>Cornelis J.</td>
<td>1429</td>
</tr>
<tr>
<td>Corthay F.</td>
<td>981</td>
</tr>
<tr>
<td>Cowan C.F.N.</td>
<td>143</td>
</tr>
<tr>
<td>Cretei J.</td>
<td>327</td>
</tr>
<tr>
<td>Creusere C.D.</td>
<td>223</td>
</tr>
<tr>
<td>Creutzburg R.</td>
<td>933</td>
</tr>
<tr>
<td>Csillag P.</td>
<td>1259</td>
</tr>
<tr>
<td>Cusani R.</td>
<td>1377</td>
</tr>
<tr>
<td>Czarnecki W.</td>
<td>875</td>
</tr>
<tr>
<td>Dabóczy T.</td>
<td>717</td>
</tr>
<tr>
<td>Dąbrowski A.</td>
<td>127</td>
</tr>
<tr>
<td>d’Alessandro C.</td>
<td>347</td>
</tr>
<tr>
<td>Dambra C.</td>
<td>567</td>
</tr>
<tr>
<td>Darlington P.</td>
<td>199</td>
</tr>
<tr>
<td>Daubechies I.</td>
<td>13</td>
</tr>
<tr>
<td>David H.</td>
<td>1561</td>
</tr>
<tr>
<td>de Beer R.</td>
<td>1749</td>
</tr>
<tr>
<td>de Beer R.</td>
<td>99</td>
</tr>
<tr>
<td>de Bruin F.F.G.</td>
<td>1207</td>
</tr>
<tr>
<td>De Cuyper B.</td>
<td>1429</td>
</tr>
<tr>
<td>De Geus P.L.</td>
<td>1331</td>
</tr>
<tr>
<td>de Haan G.</td>
<td>1385</td>
</tr>
<tr>
<td>De Larriolilleure J.</td>
<td>277</td>
</tr>
<tr>
<td>De Larriolilleure J.</td>
<td>281</td>
</tr>
<tr>
<td>De Man H.</td>
<td>1569</td>
</tr>
<tr>
<td>De Natale F.G.B.</td>
<td>1215</td>
</tr>
<tr>
<td>De Souza P.V.</td>
<td>359</td>
</tr>
<tr>
<td>De Vriendt J.</td>
<td>277</td>
</tr>
<tr>
<td>Decaniere C.</td>
<td>1721</td>
</tr>
<tr>
<td>Deconinck F.</td>
<td>1437</td>
</tr>
<tr>
<td>Deergha Rao K.</td>
<td>815</td>
</tr>
<tr>
<td>Defée L.</td>
<td>1157</td>
</tr>
<tr>
<td>Defrise M.</td>
<td>1437</td>
</tr>
<tr>
<td>Deketelaere S.</td>
<td>475</td>
</tr>
<tr>
<td>Del Re E.</td>
<td>1697</td>
</tr>
<tr>
<td>Delaere D.</td>
<td>1453</td>
</tr>
<tr>
<td>Delisle G.Y.</td>
<td>653</td>
</tr>
<tr>
<td>Dellepiane S.</td>
<td>1441</td>
</tr>
<tr>
<td>Delmas J.P.</td>
<td>657</td>
</tr>
<tr>
<td>Delopoulos A.</td>
<td>571</td>
</tr>
<tr>
<td>Deman L.</td>
<td>475</td>
</tr>
<tr>
<td>den Brinker A.C.</td>
<td>693</td>
</tr>
<tr>
<td>Deprettere E.F.</td>
<td>103</td>
</tr>
<tr>
<td>Deprettere E.F.</td>
<td>107</td>
</tr>
<tr>
<td>Desoli G.S.</td>
<td>1215</td>
</tr>
<tr>
<td>Dewilde P.</td>
<td>107</td>
</tr>
<tr>
<td>Dewilde P.</td>
<td>71</td>
</tr>
<tr>
<td>Dewilde P.</td>
<td>901</td>
</tr>
<tr>
<td>D’Haeyer J.</td>
<td>277</td>
</tr>
<tr>
<td>Di Renzo N.</td>
<td>1729</td>
</tr>
<tr>
<td>Diab C.</td>
<td>1227</td>
</tr>
<tr>
<td>Diaz A.</td>
<td>1203</td>
</tr>
<tr>
<td>Diez del Rio L.</td>
<td>1673</td>
</tr>
<tr>
<td>Djurić P.</td>
<td>1661</td>
</tr>
<tr>
<td>Dobrowiecki T.</td>
<td>1393</td>
</tr>
<tr>
<td>Docampo D.</td>
<td>507</td>
</tr>
<tr>
<td>Docampo D.</td>
<td>697</td>
</tr>
<tr>
<td>Donaar M.</td>
<td>1161</td>
</tr>
<tr>
<td>Doncarli C.</td>
<td>783</td>
</tr>
<tr>
<td>Dorr C.</td>
<td>1729</td>
</tr>
<tr>
<td>Dothev M.</td>
<td>475</td>
</tr>
<tr>
<td>Drogo De Jacovo R.</td>
<td>483</td>
</tr>
<tr>
<td>Drygajlo A.</td>
<td>1247</td>
</tr>
<tr>
<td>Dubois E.</td>
<td>953</td>
</tr>
<tr>
<td>Dubuisson B.</td>
<td>543</td>
</tr>
<tr>
<td>Dufaux F.</td>
<td>1323</td>
</tr>
<tr>
<td>Dufaux F.</td>
<td>1517</td>
</tr>
<tr>
<td>Dugelay J.-L.</td>
<td>1295</td>
</tr>
<tr>
<td>Duhamel P.</td>
<td>135</td>
</tr>
<tr>
<td>Durrani T.S.</td>
<td>1449</td>
</tr>
<tr>
<td>Dutoit T.</td>
<td>343</td>
</tr>
<tr>
<td>Eatock J.</td>
<td>411</td>
</tr>
<tr>
<td>Eberle E.</td>
<td>929</td>
</tr>
<tr>
<td>Eberlein E.</td>
<td>1557</td>
</tr>
<tr>
<td>Ebrahimi T.</td>
<td>1361</td>
</tr>
<tr>
<td>Ebrahimi T.</td>
<td>1517</td>
</tr>
<tr>
<td>Ebringer L.</td>
<td>1619</td>
</tr>
<tr>
<td>Eckert W.</td>
<td>439</td>
</tr>
<tr>
<td>Edler B.</td>
<td>211</td>
</tr>
<tr>
<td>Edmonson W.W.</td>
<td>713</td>
</tr>
<tr>
<td>Efstratiadis S.N.</td>
<td>1315</td>
</tr>
<tr>
<td>Egelmeers G.P.M.</td>
<td>1065</td>
</tr>
<tr>
<td>Elterich A.</td>
<td>1533</td>
</tr>
<tr>
<td>Engin T.</td>
<td>1745</td>
</tr>
<tr>
<td>Erdöl N.</td>
<td>1117</td>
</tr>
<tr>
<td>Ersoy C.</td>
<td>1141</td>
</tr>
</tbody>
</table>
Kim S., 637
Kim S., 965
Kingsbury N.G., 97
Kinkis P., 1469
Kipper U., 31
Kitamura T., 419
Kittler J., 591
Kleihorst R.P., 1385
Kleinhenkel W., 1533
Kleinhenkel W., 1829
Knoll A., 277
Knoll A., 281
Knoll A., 285
Knudsen J.E., 499
Knutsson H., 847
Kobayashi T., 167
Kocsis F., 1545
Kokaram A., 1283
Kokkinakis G., 399
Kollár I., 717
Kollas S., 571
Kolumvass A., 1489
Komatsu T., 1505
Komatsu T., 1777
Komatsu T, Saito T., 1505
König D., 1765
Korad J., 953
Kontro J., 243
Koskinen L., 1089
Kotropoulos C., 1177
Koutroumbas K., 1185
Kowalczyk J., 1517
Kozarev A., 1373
Krajnik E., 917
Kraus D., 1801
Kraus D., 649
Krenz R., 299
Krim H., 665
Kröger B., 331
Kudo H., 1421
Kuhn T., 439
Kumaresan R., 1433
Kummert F., 547
Kummert F., 555
Kung J.Y., 779
Kunt M., 1323
Kunt M., 1345
Kunt M., 1361
Kunt M., 1365
Kunt M., 1517
Kunt M., 1737
Kunt M., 83
Kuosmanen P., 1089
Laakso T.I., 251
Labit C., 1291
Laboissière P., 159
Lacaze B., 827
Lacourne J.L., 91
Lacroix A., 247
Laflamme C., 495
Lagendijk R.L., 1279
Lagendijk R.L., 1385
Lagendijk R.L., 291
Lagunas M.A., 1085
Lagunas M.A., 1573
Lagunas M.A., 1665
Larmerque G., 1797
Lancini R., 1565
Lancon E., 1267
Lansford J., 823
Larabial P., 681
Laur R., 1553
Lavagetto F., 1357
Lavagetto F., 563
Law K.W., 455
Le Bouquin R., 1633
Le Carpentier E., 783
Le Cerf P., 371
Le Guyader A., 1607
Le Guyader A., 527
Lecourtier Y., 619
Leduc J.-P., 1497
Lee C.Y., 1521
Lefebvre M., 1781
Leich H., 343
Leich H., 423
Leich H., 475
Leich H., 511
Leich H., 949
Leickel T., 265
Lema F., 1029
Lemahieu I., 1725
Lepšy S., 1353
Li G., 1009
Li Hui, 1541
Li Wei, 1345
Li Wenzhe, 879
Llaves A., 879
Lightstone M., 1005
Lin J., 1381
Lin W., 611
Liu F.-W., 491
Liu H., 147
Liu X., 1125
Lo K., 1231
Táboas T., 747
Táboas T., 831
Lockwood Ph., 391
López E., 451
Lopez-Ferreras F., 269
Loubaton P., 1825
Loubaton P., 1061
Loup F., 1737
Lozac B., 527
Lüder E., 259
Lukasik E., 941
Lyon D.E., 799
Macq B., 1271
Madsen J.P., 395
Maes S., 1041
Magnanelli L., 567
Maguer A., 867
Mahieux Y., 1607
Mailard E., 1173
Maiswald D., 1801
Maiswald D., 649
Malah D., 1685
Mandlakale L., 1125
Manfredi C., 1783
Manhaege C., 1725
Marchal G., 1453
Mariño J.B., 379
Mariño J.B., 415
Marques F., 1473
Marques J.S., 487
Mars J., 645
Martin J., 1833
Martin K.W., 239
Martinerie F., 1809
Martinerie F., 1821
Masgrau E., 307
Mason J.S., 1049
Mason J.S., 411
Masotti L., 1733
Mathieu P., 1337
Mathieu P., 1493
Mathur S., 633
Matsuda I., 1261
Mauc M., 523
Mboup M., 111
McLaughlin S., 791
McWhirter J.G., 5
Mecocci A., 1485
Mehlan R., 743
Mehlkopf A.F., 1749
Mehlkopf A.F., 99
Meier W., 1397
Mella-Predoviciu M., 1189
Mekuria F., 323
Mendia J., 1769
Menez J., 1267
Menez J., 767
Merkle J., 1173
Mertins A., 673
Mertzios B.G., 685
Messer H., 819
Metzlich H., 1433
Meyr H., 1561
Meyr H., 1603
Michteisen H., 395
Michel O., 179
Migliorati P., 1566
Miki N., 1037
Miki N., 759
Ming J., 355
Ming Lu, 1785
Minkwitz T., 933
Mitra C., 175
Mitra S.K., 1005
Mitra S.K., 223
Mitra S.K., 909
Mlynek D., 1517
Mocagagatta I., 1365
Mocagagatta I., 1517
Moehlzu H., 1153
Moe J.P., 587
Moeneclaey M., 1587
Molina Ch., 1219
Molinaro F., 897
Montagna R., 483
Monte E., 415
Moreau N., 527
Moreno A., 623
Moresco M., 1357
Moresco M., 563
Morishita Y., 957
Möschmen J., 531
Moteiki M., 1505
Moteiki M., 1505
Moulines E., 115
Mouze-Amady M., 1729
Mowbray S., 1591
Mrayati M., 155
Mulgrew B., 1701
<table>
<thead>
<tr>
<th>Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mulgrew B., 1833</td>
<td></td>
</tr>
<tr>
<td>Mulgrew B., 791</td>
<td></td>
</tr>
<tr>
<td>Mullis C.T., 839</td>
<td></td>
</tr>
<tr>
<td>Murenzi R., 701</td>
<td></td>
</tr>
<tr>
<td>Mutino V., 579</td>
<td></td>
</tr>
<tr>
<td>Murray B., 319</td>
<td></td>
</tr>
<tr>
<td>Nádas A., 359</td>
<td></td>
</tr>
<tr>
<td>Nagai N., 1037</td>
<td></td>
</tr>
<tr>
<td>Nagai N., 759</td>
<td></td>
</tr>
<tr>
<td>Nahamoo D., 359</td>
<td></td>
</tr>
<tr>
<td>Najm M., 1125</td>
<td></td>
</tr>
<tr>
<td>Najm M., 1219</td>
<td></td>
</tr>
<tr>
<td>Najm M., 1461</td>
<td></td>
</tr>
<tr>
<td>Nakamura S., 957</td>
<td></td>
</tr>
<tr>
<td>Nandi A.K., 587</td>
<td></td>
</tr>
<tr>
<td>Nandi A.K., 743</td>
<td></td>
</tr>
<tr>
<td>Nandkumar S., 515</td>
<td></td>
</tr>
<tr>
<td>Neagoe V., 905</td>
<td></td>
</tr>
<tr>
<td>Neelajrvi J., 1741</td>
<td></td>
</tr>
<tr>
<td>Neri A., 735</td>
<td></td>
</tr>
<tr>
<td>Neumann H., 1426</td>
<td></td>
</tr>
<tr>
<td>Neumann H., 603</td>
<td></td>
</tr>
<tr>
<td>Neuvo Y., 1013</td>
<td></td>
</tr>
<tr>
<td>Neuvo Y., 1157</td>
<td></td>
</tr>
<tr>
<td>Neuvo Y., 1307</td>
<td></td>
</tr>
<tr>
<td>Neuvo Y., 1405</td>
<td></td>
</tr>
<tr>
<td>Neuvo Y., 1481</td>
<td></td>
</tr>
<tr>
<td>Neuvo Y., 1501</td>
<td></td>
</tr>
<tr>
<td>Neuvo Y., 1741</td>
<td></td>
</tr>
<tr>
<td>Neuvo Y., 243</td>
<td></td>
</tr>
<tr>
<td>Nguyen Thi H.L., 303</td>
<td></td>
</tr>
<tr>
<td>Ni W.-Z., 491</td>
<td></td>
</tr>
<tr>
<td>Nicoulin A., 1345</td>
<td></td>
</tr>
<tr>
<td>Nielsen H., 503</td>
<td></td>
</tr>
<tr>
<td>Niemann H., 439</td>
<td></td>
</tr>
<tr>
<td>Nijmans J.J., 443</td>
<td></td>
</tr>
<tr>
<td>Nisbet K., 791</td>
<td></td>
</tr>
<tr>
<td>Noll R., 1541</td>
<td></td>
</tr>
<tr>
<td>Nordberg K., 847</td>
<td></td>
</tr>
<tr>
<td>Nyssen E., 1429</td>
<td></td>
</tr>
<tr>
<td>Oda K., 387</td>
<td></td>
</tr>
<tr>
<td>Ogier J.M., 619</td>
<td></td>
</tr>
<tr>
<td>Ogorkalez M., 191</td>
<td></td>
</tr>
<tr>
<td>Øljen G.E., 1353</td>
<td></td>
</tr>
<tr>
<td>Olivier C., 619</td>
<td></td>
</tr>
<tr>
<td>O'Neill C., 319</td>
<td></td>
</tr>
<tr>
<td>Ort K., 1545</td>
<td></td>
</tr>
<tr>
<td>Ortega A., 1369</td>
<td></td>
</tr>
<tr>
<td>Ortega J., 451</td>
<td></td>
</tr>
<tr>
<td>Ottenberg K., 1425</td>
<td></td>
</tr>
<tr>
<td>Ottenberg K., 603</td>
<td></td>
</tr>
<tr>
<td>Ouvrard V., 1001</td>
<td></td>
</tr>
<tr>
<td>Ovaska S.J., 871</td>
<td></td>
</tr>
<tr>
<td>Padmanabhan M., 239</td>
<td></td>
</tr>
<tr>
<td>Páez Borrallo J.M., 1073</td>
<td></td>
</tr>
<tr>
<td>Pagès-Zamora A., 1085</td>
<td></td>
</tr>
<tr>
<td>Pandit M., 1433</td>
<td></td>
</tr>
<tr>
<td>Pandit M., 929</td>
<td></td>
</tr>
<tr>
<td>Pardès M., 379</td>
<td></td>
</tr>
<tr>
<td>Parhi K.K., 1057</td>
<td></td>
</tr>
<tr>
<td>Pauwels M., 1569</td>
<td></td>
</tr>
<tr>
<td>Peinado V., 1203</td>
<td></td>
</tr>
<tr>
<td>Pélè D., 1295</td>
<td></td>
</tr>
<tr>
<td>Pellandini F., 981</td>
<td></td>
</tr>
<tr>
<td>Pellandini F., 997</td>
<td></td>
</tr>
<tr>
<td>Pellegretti P., 567</td>
<td></td>
</tr>
<tr>
<td>Peng H.L., 1513</td>
<td></td>
</tr>
<tr>
<td>Pérez F., 507</td>
<td></td>
</tr>
<tr>
<td>Pérez F., 697</td>
<td></td>
</tr>
<tr>
<td>Pérez-Castellanos M., 1203</td>
<td></td>
</tr>
<tr>
<td>Pérez-Neira A., 1085</td>
<td></td>
</tr>
<tr>
<td>Perkins A., 619</td>
<td></td>
</tr>
<tr>
<td>Perret C., 1737</td>
<td></td>
</tr>
<tr>
<td>Perrier P., 159</td>
<td></td>
</tr>
<tr>
<td>Persson L., 1817</td>
<td></td>
</tr>
<tr>
<td>Pesquet J.C., 859</td>
<td></td>
</tr>
<tr>
<td>Peterec D., 1565</td>
<td></td>
</tr>
<tr>
<td>Pétillon T., 119</td>
<td></td>
</tr>
<tr>
<td>Petit J.P., 1607</td>
<td></td>
</tr>
<tr>
<td>Petrov M., 1489</td>
<td></td>
</tr>
<tr>
<td>Pham D.T., 771</td>
<td></td>
</tr>
<tr>
<td>Pharao K., 395</td>
<td></td>
</tr>
<tr>
<td>Phillips W., 1255</td>
<td></td>
</tr>
<tr>
<td>Phillips W., 925</td>
<td></td>
</tr>
<tr>
<td>Pichony M.A., 359</td>
<td></td>
</tr>
<tr>
<td>Pierucci , 1697</td>
<td></td>
</tr>
<tr>
<td>Pijnappel W.W.F., 99</td>
<td></td>
</tr>
<tr>
<td>Pinteon R., 731</td>
<td></td>
</tr>
<tr>
<td>Pitas I., 1177</td>
<td></td>
</tr>
<tr>
<td>Pitas I., 1469</td>
<td></td>
</tr>
<tr>
<td>Pitas I., 921</td>
<td></td>
</tr>
<tr>
<td>Plansky H., 1243</td>
<td></td>
</tr>
<tr>
<td>Portillo-García J.J., 787</td>
<td></td>
</tr>
<tr>
<td>Povey G.J.R., 1595</td>
<td></td>
</tr>
<tr>
<td>Pringle R.D., 1591</td>
<td></td>
</tr>
<tr>
<td>Pringle R.D., 1595</td>
<td></td>
</tr>
<tr>
<td>Pringle R.D., 607</td>
<td></td>
</tr>
<tr>
<td>Proakis J.G., 665</td>
<td></td>
</tr>
<tr>
<td>Prost R., 1227</td>
<td></td>
</tr>
<tr>
<td>Proudler I.K., 5</td>
<td></td>
</tr>
<tr>
<td>Pun T., 1781</td>
<td></td>
</tr>
<tr>
<td>Queluz M.P., 1263</td>
<td></td>
</tr>
</tbody>
</table>
Quinn A.P., 677
Rabadan C., 1809
Radecki J., 953
Ramponi G., 1413
Ramstad T.A., 1353
Ramstad T.A., 231
Rands Jensen P.M., 763
Rantala S., 661
Rayner P., 1283
Reddy D.C., 815
Regalia Ph., 1061
Regazzoni C.S., 579
Reininger H., 31
Reininger H., 427
Renard J.P., 949
Resende L.S., 1105
Richard G., 347
Rieck S., 439
Rioul O., 937
Ristic B., 751
Roberto V., 595
Rocchi S., 1733
Rodellar V., 1203
Rodrigo-Aparicio E., 1029
Roeckerath-Ries M.-Th., 1773
Romano J.M.T., 1105
Ronda J.I., 1369
Rosenberger A., 1841
Rovias D., 1669
Ruczewski P., 747
Ruiz Mezcua B., 407
Ruiz P., 811
Russo F., 1413
Saerens M., 171
Sagerer G., 547
Sagerer G., 555
Saito T., 1421
Saito T., 1505
Saito T., 1777
Salami R., 495
Saleh B.E.A., 351
Salembier P., 1137
Salerno E.A., 1445
Salerno E.A., 1417
Sailias J., 1239
Sanchez V., 495
Sandler M.B., 1706
Sangfelt E., 1817
Sa-Ngaukotchakorn S., 855
Sankur B., 1745
Sano A., 727
Sanz-González J., 1029
Sanz-González J., 269
Sas P., 1709
Sauter D., 1729
Sauter D., 1761
Scaramo G., 735
Schamel G., 281
Schäuble P., 295
Schettini R., 535
Schettini R., 599
Schmitt F., 1789
Schoentgen J., 336
Schamp M., 99
Schukat-Talamazzini E.G., 439
Schwartz J.L., 1025
Seestaedt B., 555
Seltzer D., 1557
Semino A., 1441
Sereno D., 483
Serpico S.B., 567
Serviere Ch., 1145
Sezan M.I., 1279
Shah A.I., 1287
Shah A.I., 969
Shanbhag N.R., 1057
Sherman P.J., 799
Shirai K., 167
Si S.M., 1645
Simon S., 281
Sirsic B., 1373
Slohan P., 1001
Skyttä J., 1699
Slock D., 1169
Slump C.H., 1713
Smolders J., 447
So K.K.M., 479
Söderström T., 669
Soininen R., 1157
Sormmen P.C.W., 1065
Somogyi Th., 281
Son J.C., 637
Song I., 637
Song X., 1307
Song X., 1501
Soquet A., 171
Sorokin V.N., 165
Spinnler W., 1557
Sporer T., 211
Stammler W., 1533
Stammler W., 1829
Stamnitz P., 281
LVI

Stasiński R., 1113
Stasiński R., 941
Steger J., 467
Stenger L., 281
Stiehl H.S., 1425
Stroica P., 1189
Stroica P., 669
Strobach P., 807
Suetens P., 1453
Sun T., 1405
Sundsbø I., 1509
Suoranta R., 661
Suzumura N., 419
Szynarńda J., 747
Tadokoro Y., 1021
Taeymans Y., 1199
Takahashi A., 1037
Takemasa E., 1505
Takemasa E., 1505
Takizawa Y., 387
Tampaouti A., 1291
Tanem J., 231
Tang K., 1481
Tarrés Ruiz F., 1311
Tay D.B.H., 977
Temerinac M., 1373
ten Bosch L.F.M., 151
Terci A., 1049
Theodoridis S., 119
Theodoridis S., 143
Theodoridis S., 879
Theys C., 709
Thierer G., 311
Thoen F., 1569
Timmermann D., 1509
Tirakis A., 571
Tonazzini A., 1445
Torres L., 1473
Tourneret J.-Y., 827
Trompf M., 431
Tripovski Z., 1373
Trucco P., 1441
Tsai R.J., 945
Tsatscanis M., 843
Tseng C.-Y., 1121
Tseng C.-Y., 1837
Tsio C.-K., 627[B
Tsopanoglou A., 399
Tsuji H., 727
Tulsei U., 219
Türkcan E., 1181

Tuszyński M., 1097
Tziotas G., 1341
Unbehauen R., 1381
Ur H., 1025
Utsunomiya T., 1251
Vaidyanathan P.P.[B., 203
Valinio O., 871
Valdovinos A., 139
Vällimäki V., 1717
Vallet R., 131
Van Brussel H., 1709
Van Callie B., 1271
Van Compernolle D., 1081
Van Compernolle D., 371
Van Compernolle D., 375
Van Compernolle D., 447
van den Bulck P., 1211
van der Veen A.-J., 901
van der Vleuten R. J., 1477
Van Diest M., 371
Van Diest M., 375
Van Droogenbroeck M., 1389
Van Eycken L., 281
Van Gerven S., 1081
Van Hamme H., 731
Van Hecke P., 1721
Van Huffels S., 1721
Van Huffels S., 669
Van Ormondt D., 1749
Van Ormondt D., 99
Van Tongeren B.P.O., 1749
Vandendorpe L., 1275
Vandermeylen D., 1453
Vandewalle J., 1721
Värri A., 1741
Vary P., 1641
Vaseghi S.V., 435
Veeleutert L.P.J., 1207
Veiga M., 407
Venetsanopoulos A.N., 1223
Venetsanopoulos A.N., 685
Venturi G., 1441
Verbeeck R., 1453
Vergara L., 1149
Verhelst W., 339
Vernazza G., 579
Vertiest E.I., 1009
Vesin J.M., 1069
Vesin J.M., 1737
Vidal E., 367
Vidal J., 887
Vines G., 883
Vogelaers D., 1725
vom Stein H.D., 1397
Vozel B., 783
Vuori J., 1599
Wang C.L., 1513
Wang D., 491
Wang Q., 1013
Weber J. H., 1477
Wegmann B., 1299
Wells L., 1549
Wensink H.E., 775
Wenzhe L., 1133
Weruaga-Prieto L., 1615
Wery B., 475
Wesołowski K., 1637
Wojtkiewicz A., 1097
Wolf D., 31
Wolf D., 427
Wolter S., 1553
Wong Y.K., 1525
Wu W.R., 1625
Wulich D., 1653
Xin Y., 1199
Xydeas C.S., 479
Yamanaka S., 1021
Yang G., 511
Yardim A., 723
Yaseen M., 961
Yin L., 1307
Yongsheng C., 363
Yoshikawa T., 1017
Youlal H., 1461
Young G., 1619
Yu P., 1223
Yuan Y., 1449
Yun J.S., 637
Ywanne F., 1809
Zadrozny W., 383
Zamperoni P., 1401
Zanelato G., 511
Zerr B., 1173
Zetzsche C., 1299
Zhang Y.J., 551
Zheng H., 1327
Zhu Y.M., 583
Zöller U., 207
Zoubir A.M., 641
Zurawka T., 259
IMPROVED PERFORMANCE OF ROOT-MUSIC WITH SENSOR PERTURBATIONS

SHANLA MATHUR

Department of Electronics and Communications Engineering, Osmania University, Hyderabad - 500 007, India.

The performance of Root-Music, for estimating the direction of arrival (DOA) of plane waves in the presence of sensor gain and phase perturbations, in the case of a linear equispaced sensor array was studied. It is found that the performance of Root-Music with sensor perturbations can be improved by simple teoplitzization of the array covariance matrix (R) for DOA estimation. Also the teoplitzed R matrix tends to the true array covariance matrix with increase in number of sensors. Computer simulations are provided to substantiate the analysis.

1 INTRODUCTION

Most of the eigen decomposition based methods decompose the observed covariance matrix into two orthogonal spaces, commonly referred to as signal and noise subspaces, and estimate the DOA's from one of these spaces [1] - [4]. In Root-Music [5], the DOA's are determined form the roots of a polynomial formed from the noise subspace.

2. PROBLEM FORMULATION

Assume that D narrowband sources impinge on a linear array consisting of M equispaced omnidirectional sensors, and that the directions for arrival (DOAs) of these sources are \( \{ \Theta_1, \Theta_2, \ldots, \Theta_M \} \). The output of the m-th sensor can be expressed as:

\[
r_m(t) = \sum_{d=1}^{D} a_d(t) e^{-j\omega_0(m-1)\tau_d} + n_m(t)
\]

where \( a_d(.) \) denotes the signal emitted by the d-th source as observed at sensor one, \( \omega_0 \) is the centre frequency of the sources and \( n_m(.) \) is the additive noise, assumed to be a zero-mean stationary random process that is independent from sensor to sensor. The interelement path delay is given by \( \tau_d = \Delta \sin \Theta_d / c \), where \( \Delta \) denotes spacing of the sensor, \( \Theta_d \) is the direction of arrival of the d-th source and c is the velocity of the plane waves. Let

\[
r(t) = [r_1(t), r_2(t), \ldots, r_M(t)] \tag{2}
\]

be the simultaneously sampled vector (snapshot) of array signals, \( s(.) \) and \( n(.) \) are similarly defined. The array signal vector can then be expressed as:

\[
r(t) = As(t) + n(t) \tag{3}
\]

where \( A \) is a MxD direction matrix with the i-th column

\[
a_i = [1 e^{-j\omega_0 \tau_1} \ldots e^{-j\omega_0 (M-1)\tau_i}] \tag{4}
\]

The array covariance matrix can then be expressed as:

\[
R = E[r(t)r^+(t)] = ASA^+ + \sigma^2 I \tag{5}
\]

where \( S \) denotes the source covariance matrix, \( \sigma^2 \) is the variance of the additive noise.

In Root-Music, the DOAs can be estimated accurately only when the gain and phase characteristics of the elements in the array are identical, which may be difficult to ensure in practice. Any deviation in gain and phase characteristics can cause errors in the estimation of the DOAs.

3 EFFECT OF SENSOR GAIN AND PHASE PERTURBATIONS

Let the deviation of gain and phase of the m-th sensor be \( \delta_m \) and \( \phi_m \) respectively and assume that the array covariance matrix in the presence of gain and phase errors can be written as
\[ \hat{K} = G A^* G + \sigma^2 I \]  
(6)

where \( G \) is a diagonal matrix of size \( M \times M \) with elements

\[ g_m = (1 + \delta_{am}) e^{j \phi_m} \]  
(7)

Note that in equation (7), a nominal gain of unity and a nominal phase of zero for the sensors is assumed for the unperturbed array. We also assume that both \( \delta_{am} \) and \( \delta \) are small and zero - mean random variables with the assumption (7) can be simplified as:

\[ g = (1 + \delta_{am} + j \delta_{pm}) \]  
(8)

In the presence of sensor gain and phase errors, the noise subspace \( \mathbb{N}_n \) is modified to \( \tilde{\mathbb{N}}_n \) where \( \tilde{\mathbb{N}}_n \) is the noise subspace obtained from \( \tilde{R} \). Consequently, the polynomial obtained using \( \tilde{\mathbb{N}}_n \) will be perturbed and hence the DOA estimates. The degradation in the performance of the algorithm with sensor perturbations has been studied extensively in terms of bias and variance through computer simulations and is explained in the next section.

4 IMPROVED PERFORMANCE OF ROOT-MUSIC ALGORITHM IN THE PRESENCE OF SENSOR PERTURBATIONS

With sensor perturbations, the elements of the perturbed array covariance matrix \( \tilde{R} \) can be written as:

\[ \tilde{R}_{ij} = (g_i g_j^*)^* \tilde{Y}_{y_i y_j} \]  
(9a)

and \( \tilde{R}_{ii} = |g_i|^2 \tilde{Y}_{y_i y_i} + \sigma^2 \)  
(9b)

where \( \tilde{Y}_{y_i y_j} = \sum_{d=1}^{D} y_d(t)^* e^{-j \omega (m-1) T_d} \)  
(10)

The teoplitzization of \( \tilde{R} \) is done by obtaining the average estimate of each of the 1st diagonal of \( \tilde{R} \) and replacing that diagonal with this computed estimate. Let us denote the teoplitzized matrix by \( \tilde{R} \). To see how teoplitzization helps in improving the DOA estimates, for gain errors only consider the following.

The averaged estimate of the main diagonal of \( \tilde{R} \) is

\[ \tau(0) = \frac{1}{N} \sum_{m=1}^{N} (g_m^2 Y_{y_m}^* y_m) + \sigma^2 \]  
(11)

where \( g_m^2 = (1 + \delta_{am})^2 \)  
(12)

\[ \tau(0) = \frac{1}{N} \sum_{m=1}^{N} (1 + 2 \delta_{am} \delta_{am} Y_{y_m}^* y_m) + \sigma^2 \]  
(13)

Assuming a zero - mean, uniform random distribution, \( \chi(0) \) can be approximated as

\[ \chi^2 \]  
(14)

where \( \chi \) is the upper limit of the uniform distribution

\[ \chi(0) = \frac{1}{3M} \sum_{m=1}^{N} Y_{y_m}^* y_m \]  
(15)

As \( M \) increases, the contribution of the second term will decrease and hence \( \chi \rightarrow \infty \), \( \tau(0) \rightarrow \chi(0) \). Thereby reducing significantly the sensor gain perturbation effects.

Consider next the averaged estimate of the first sub - diagonal of \( \tilde{R} \), which is given by,

\[ \tau(1) = \frac{1}{M} \sum_{m=1}^{M-1} (g_m \delta_{m+1} Y_{y_m}^* y_{m+1}) \]  
(16)

where \( g_m \delta_{m+1} = (1 + \delta_{am})(1 + \delta_{am+1}) \)  
(17)

\[ \tau(1) = \frac{1}{M} \sum_{m=1}^{M-1} [1 + (\delta_{am} + \delta_{am+1}) + \delta_{am} \delta_{am+1}] \]  
(18)

\[ \frac{1}{M-1} \left( \sum_{m=1}^{M-1} (\delta_{a1} + \delta_{a2} + \ldots + \delta_{am+d}) \right) + (\delta_{a1} \delta_{a2} + \delta_{a3} \delta_{a4} + \ldots + \delta_{am+d}) \]  
(19)
where \( y(1) = \frac{1}{M-1} \sum_{m=1}^{M-1} y_m x_{m+1} \) \( (20) \)

with increase in \( M \), \( r(1) \rightarrow y(1) \), thereby reducing the effects of sensor gain perturbations.

Similarly, it can be shown that the averaged estimate of each of the 1th sub-diagonal of \( \hat{\mathbf{R}} \) can be reduced to \( y(1) \). Similar mathematical analysis can be carried out in the case of gain and phase errors.

5 SIMULATION STUDIES

The following scenario is assumed for the purpose of simulation. Two sources from 20° and 30° with respect to broadside direction of displacement vector impinge on a linear array whose interelement spacing is \( \lambda/2 \). The source and sensor noise powers were assumed to be zero db each. The nominal gain and phase values for the sensors are unity and zero respectively. Since the analysis of the previous section was based on the asymptotic covariance matrices, simulations were carried out for the same.

The perturbed covariance matrix was formed using \( \hat{\mathbf{R}} \). The DOAs were estimated using the teoplitized matrix, \( \hat{\mathbf{R}} \). The bias and variance in the angle estimates thus obtained were computed.

The above simulations were carried out for various values of \( a_{\text{max}} \) and \( p_{\text{max}} \). Fig. 1a and Fig. 1b show the estimates and variance of the DOAs as a function of \( a_{\text{max}} \), for \( M = 6 \) are plotted, while Fig. 2a and Fig. 2b show the plots for \( M = 8 \). The estimates and variance in the DOAs were also computed without teoplitization i.e., using \( \hat{\mathbf{R}} \) and these plots are shown Figs. 1 & 2 by dashed lines. From these plots, the significant improvement in the performance of Root-Music by teoplitization is apparent.

The combined effect of gain and phase perturbations, some sample results are presented in Table 1. With teoplitization of the array covariance matrix, the results are presented in Table 2. Comparing Tables 1 and 2, we can see the improved performance of Root-Music by teoplitization.

6 CONCLUSIONS

In this paper we have proposed a simple method to improve the performance of Root-Music algorithm in the presence of sensor gain and phase perturbations. It is found that, with teoplitization the bias and variance in the angle estimates reduces significantly with increase in the number of sensors.

References:


PERFORMANCE CHARACTERISTICS OF A GENERALIZED NEYMAN-PEARSON DETECTOR WITH FUZZY INFORMATION

Jae Cheol Son, Ickho Song, Sangyoub Kim, Seong Il Park
Jin Seon Yun, and Yoon Kyoo Jhee

Department of Electrical Engineering
Korea Advanced Institute of Science and Technology (KAIST)
373-1 Guseong Dong, Yuseong Gu, Daejeon 505-701, Korea
Tel: +82-42-869-3445, 5445
Fax: +82-42-869-3410
E-mail: isong@Sejong.kaiast.ac.kr

In this paper statistical characteristics of the locally optimum fuzzy detector test statistic for known signals are discussed. Specifically, the probability distribution of the locally optimum fuzzy detector test statistic is obtained from the exact and approximate approaches. Performance characteristics of the locally optimum fuzzy detector are also considered and compared to those of the locally optimum detector.

1. Introduction

In several studies it has been suggested that under certain circumstances the fuzzy set theoretic techniques can be applied to signal detection problems [e.g., 3, 7]. For example, in [7, 8] it was pointed out that techniques of statistical hypotheses testing with fuzzy information can be used in detecting signals when the detection processor produces a self-noise.

As an application of the fuzzy set theory in signal detection, the quantizer-detector scheme was considered in [7, 8] assuming that the observations can be interpreted as fuzzy informations since it is customary to implement the nonlinearity of the locally optimum (LO) detector by a quantizer [e.g., 5]. We will denote this concatenation of a quantizer and the LO detector by the LOQ detector. The rationale to consider the fuzzy information in the quantizer-detector structure can be explained as follows. In digital signal processing [e.g., 6] it is usually assumed that the probability distribution of the quantization error process is uniform over the range of the quantization error, and this assumption is known to yield reasonable results in some situations. In other cases, however, this assumption does not give an accurate description of the behavior of the quantizer [e.g., 9]. In addition, the assumption is not appropriate either when the signals corrupted by a noise (e.g., self-noise) are quantized. Based on this observation there have been some studies on statistical analysis of the quantized noisy signals [e.g., 1]. However, it seems that to employ fuzzy testing of statistical hypotheses is more convenient and practical for signal detection since exact analysis of the quantization error is cumbersome in practice.

In this paper, we will investigate the statistical characteristics and performance of the locally optimum fuzzy (LOF) detector test statistic as a natural extension of our previous study [7]. Two approaches will be used to obtain the distribution of the LOF detector test statistic in Section 3. In addition, performance characteristics of the LOF detector will be investigated and an important observation will be obtained in Section 4.

2. Basics

Let us consider the frequently encountered signal-detection problem which can be expressed by the following two hypotheses \(H_0\) and \(H_1\):

\[
H_0: \quad Y_i = W_i, \tag{1}
\]

versus

\[
H_1: \quad Y_i = \theta e_i + W_i, \tag{2}
\]

for \(i = 1, 2, \ldots, n\). In (1) and (2), \(Y_i\) is the observation, \(\theta\) is the known signal component, and \(W_i\) is the purely-additive noise (PAN) component at the \(i\)-th sampling instant. \(\theta\) is the amplitude parameter which controls the signal strength. The PAN components \(W_i\), \(i = 1, 2, \ldots, n\), are assumed to be independent and identically distributed (i.i.d.) with the common continuous probability density function (pdf) \(f\). It is also assumed that the pdf \(f\) is zero-mean and even.

In order to handle the observation \(Y_i\) as fuzzy information based on the descriptions in Section 1, let us introduce some definitions. Let \((X^*, B_{X^*}, F)\) be a probability space where \(X^*\) is a Euclidean \(n\)-dimensional space, \(B_{X^*}\) is the Borel \(\sigma\)-field, and \(F\) is a probability measure over \(X^*\). A fuzzy information system \(\tau\) is a fuzzy partition of the real line \(X\) by means of fuzzy events [2]. An \(n\)-tuple of elements in \(\tau\), \(\tau = (\kappa_1, \kappa_2, \ldots, \kappa_n)\), representing the algebraic product of \(\kappa_1, \kappa_2, \ldots, \kappa_n\), is called the sample fuzzy information of size \(n\). The set consisting of all algebraic products of \(\kappa_1, \kappa_2, \ldots, \kappa_n\) is called the fuzzy random sample of size \(n\) and is denoted by \(\xi^n\). The probability distribution of \(\xi^n\) is given by

\[
P(\xi^n) = \int_{X^n} \lambda_{\kappa_1}(\gamma_1) \cdots \lambda_{\kappa_n}(\gamma_n) \, dF(\gamma_1, \ldots, \gamma_n), \tag{3}
\]

where the integral is the Lebesgue-Stieltjes integral and \(\lambda_{\kappa}(\gamma)\) is called the membership function of \(\kappa\). For notational convenience, we will denote by \(P(\mathcal{R}) = P(\mathcal{R} | H_0)\) and \(P_1(\mathcal{R}) = P(\mathcal{R} | H_1)\) the probability distributions of \(\xi^n\) under \(H_0\) and

* This research was supported by Korea Advanced Institute of Science and Technology (KAIST) and by Korea Science and Engineering Foundation (KOSEF), for which the authors would like to express their thanks.
for a given false-alarm probability based on the classical definition of probability. Similarly, we will denote the subspace of the ordered fuzzy information space in which a fuzzy detector accepts $H_1$ for a given false-alarm probability based on the Zadeh's definition of probability (3) by $\Psi_f$.

3. Distributions of the Test Statistic

It was already shown in [7] that the LOF detector test statistic for known signals can be expressed as

$$T_{LOF}(\mathbf{Q}) = \sum_{i=1}^{n} \theta_i \gamma_i \phi_i(\xi_i),$$

where

$$\gamma_i(\xi_i) = \frac{\mathbb{E}(\lambda_i | \xi_i)}{\mathbb{E}(\lambda_i)}$$

with $\mathbb{E}(\cdot)$ denoting the statistical expectation with respect to the pdf $f$.

The practical methods in most radar (or sonar) signal detection problems relies upon the Neyman-Pearson procedure. When we use the Neyman-Pearson procedure in detection systems, it is necessary to obtain the probability distribution of the test statistic under the null hypothesis to find the exact detection threshold. In this section, we will consider two approaches to obtain the distribution function of the LOF detector test statistic for known signals.

3.1. Exact Distribution

Let us consider here the 2m-level MRQ only, since a similar approach can be applied to the MTQ. To obtain the distribution of the test statistic (4), let us first define $\{d_1, d_2, \ldots, d_{2m}\}$ as a set of nonnegative integers such that $\sum_{i=1}^{2m} d_i = n$. Then the probability of the set $\{A_1 \cap A_2 \cap \ldots \cap A_{2m}\}$, where $A_i$, $i = 1, \ldots, 2m$, are the events defined as $A_1 \Delta \{\text{the fuzzy information } \tau_i \text{ appears exactly } d_1 \text{ times in } n \text{ observations}\}$, $A_2 \Delta \{\text{the fuzzy information } \tau_i \text{ appears exactly } d_2 \text{ times in } n \text{ observations}\}$, $\ldots$, and $A_{2m} \Delta \{\text{the fuzzy information } \tau_i \text{ appears exactly } d_{2m} \text{ times in } n \text{ observations}\}$, is easily obtained to be

$$\frac{n!}{d_1! \cdot d_2! \cdot \ldots \cdot d_{2m}!} \cdot \frac{d_1}{\tau_1} \cdot \frac{d_2}{\tau_2} \cdot \ldots \cdot \frac{d_{2m}}{\tau_{2m}},$$

where $0 \leq d_i \leq n$ and $p(\mathbf{C})$ is the probability that the fuzzy information $\tau_i$ is observed when a sample fuzzy information of size 1 is taken from the experiment.

The distribution of the test statistic can now be easily obtained since the probability that the test statistic takes a specific value is just one value or the sum of two or more values evaluated by (6). It should be emphasized, however, that the above procedure becomes unmanageable as the sample size grows. To obtain the probability distribution of the LOF detector test statistic, we should calculate the joint probability for all elements in $\tau_i$ (6). If we assume that quantization level is 256 and the sample size is 20, then the joint probability for 2560 ($= 1.46 \times 10^{50}$) cases should be obtained which seems to be too large to handle.

3.2. Approximate Approach

If we would like to find the exact threshold to satisfy a given false-alarm restriction, the procedure in the previous subsection is unavoidable. If the threshold is not required to be set exactly, however, the calculation burden due to the
large sample size can be alleviated by exploiting the well-known central limit theorem. For brevity, let us assume that \( \alpha_i, i = 1, \ldots, n \), are all equal to 1, which is a common and reasonable assumption.

We first see that the mean and variance of the LOF detector test statistic under the null hypothesis are obtained to be

\[
\mu_0 = \sum_{\mathbf{x}} P(\mathbf{x}) D(\mathbf{x}),
\]

and

\[
\sigma_0^2 = \sum_{\mathbf{x}} P(\mathbf{x}) \{ D(\mathbf{x}) - \mu_0 \}^2,
\]

respectively. The following theorem describes how we can find the approximate value of the detection threshold for large sample size.

**Theorem 1.** Assume that the mean and variance of \( D(\mathbf{x}) \) under \( H_0 \) are finite. Then the fuzzy test accepting \( H_1 \) when \( \mathbf{x} \in \Xi \) and rejecting it when \( \mathbf{x} \in \Xi^c \) is asymptotically locally optimum for testing \( H_1 \) against \( H_0 \) at level \( \alpha \), where

\[
\Xi = \{ \mathbf{x} \mid \mathbf{x} \in \mathbb{R}^n \} \text{ with } T^*_\alpha = \frac{T_{lof}(\mathbf{x}) - \mu_0}{\sigma_0} > r \}.
\]

In (9) \( r \) is the \( 100(1-\alpha) \)th percentile of the standard normal distribution.

**Proof:** Denoting \( D(\mathbf{x}) \) by \( G_1 \), we see that \( G_1, G_2, \ldots \) are clearly i.i.d. nondegenerate random variables. In addition, the mean and the variance of \( G_2 \) under \( H_0 \) are \( \mu_0 \) and \( \sigma_0^2 \) respectively. Since both \( \mu_0 \) and \( \sigma_0^2 \) are assumed to be finite, we see from the central limit theorem that \( T^*_\alpha \) converges in law to \( T \) as \( n \to \infty \), where \( T \) is a standard normal random variable. Then the statement of the theorem is immediately obtained.

Strictly speaking, Theorem 1 can be applied when the sample size \( n \) approaches to infinity. If we consider a sufficiently large value of \( n \), however, we see that Theorem 1 can be used to find the approximate value of the threshold. Since it is easy to see that \( \mu_0 = 0 \) because of the odd-symmetry property of \( D(\mathbf{x}) \), the approximate distribution of the LOF detector test statistic is normal with mean zero and variance \( n \sum P(\mathbf{x}) D(\mathbf{x}) \). Thus the approximate value of the threshold for level \( \alpha \) is the \( 100(1-\alpha) \)th percentile of this normal distribution.

Now we concisely discuss the usefulness of the approximate approach. As we have seen from the discussion in Subsection 3.1, the probability distribution of the test statistic is available only after the joint probabilities for all \( (2m)^n \) cases are calculated. In the approximate approach, however, what is required to be known is the probabilities of \( 2m \) fuzzy informations, not the joint probabilities of \( (2m)^n \) cases.

### 4. Performance Characteristics

Since intensive calculations are required when the quantization-level is large, we will investigate the performance characteristics of the LOF detector when the quantization level is 2 and 3.

#### 4.1. 2-level MrQ Detector

We first consider the 2-level MRQ detector. Since it was found [7] that we can not detect weak known signals when \( \Delta = \infty \), it is assumed here that \( \Delta < \infty \). Note that the 2-level MRQ detector based on crisp information is the sign detector in the classical detection theory. Let us now consider the following theorem.

**Theorem 2.** If the pdf \( f \) is zero-mean and even, then the 2-level MRQ detector is a nonparametric detector.

**Proof:** Since \( f \) is zero-mean and even-symmetric, we have \( P(\mathbf{x}) = P(-\mathbf{x}) \) for any value of \( \Delta \). Hence we see that the probability for any element in \( W^c \) under \( H_0 \) is always \( 2^m \). Therefore, \( P_\alpha \) does not depend on the noise pdf \( f \) and the theorem is proved.

It is well-known that the classical sign detector is a nonparametric detector. We can therefore expect that the two detection schemes the 2-level LOF and sign detectors would have quite similar characteristics. This anticipation becomes more clear if we consider the following theorem.

**Theorem 3.** The 2-level LOF detector based on crisp information and the 2-level LOF detector based on fuzzy information make the same decision for the same observation when the sizes of the two detectors are equal.

**Proof:** Note that a test problem on the fuzzy information space can be converted into a test problem on the crisp information space. Converting \( \Psi_\alpha \) with a given value of \( \alpha \) into the corresponding subspace of the ordered crisp information space, we see that it is equal to \( \Psi_\alpha \) with the same value of \( \alpha \), since the probability for any element of the crisp and information spaces under \( H_0 \) is \( 2^m \) and the order of the ordered fuzzy information space is preserved after the conversion. Hence the theorem is proved.

Theorem 3 implies that it makes no difference between employing fuzzy information and crisp information in making a decision when the level is two.

#### 4.2. 3-level MTQ Detector

Let us now assume that the input signals are quantized by a 3-level MTQ which provides output values -1, 0, and 1. Suppose that the experimentation provides us with the following fuzzy information: \( \lambda_{-1} = \text{"approximately less than -1"} \), \( \lambda_0 = \text{"approximately 0"} \), and \( \lambda_1 = \text{"approximately greater than 1"} \), where the membership functions associated with the fuzzy informations \( \lambda_{-1}, \lambda_0, \) and \( \lambda_1 \) are

\[
\lambda_{-1}(y) = \begin{cases} 1 & y < -1 \\ -y & -1 \leq y < 0 \\ 0 & \text{otherwise}, \end{cases}
\]

\[
\lambda_0(y) = \begin{cases} 1 + y & -1 \leq y < 0 \\ 1 - y & 0 \leq y < 1 \\ 0 & \text{otherwise}, \end{cases}
\]

and

\[
\lambda_{-1}(y) = \begin{cases} 1 & y > 1 \\ y & 0 \leq y \leq 1 \\ 0 & \text{otherwise}, \end{cases}
\]

respectively. In (10), it is easy to see that \( \Delta = 2 \).

We assume that a decision is made for every three observations, and that \( \alpha = 0.03155 \), \( W^c = (0, 1) \), that is, each of the noise components is normally distributed with zero mean and unit variance. The false-alarm probabilities \( P_{fa} \) and detection probabilities \( P_d \) of the LOF and LOQ detectors in various cases are obtained and shown in Tables 1-4.

We now consider two situations. Let us first consider the ideal case where there is no self-noise. From Table 1, we see that the characteristic of the LOQ detector is almost the same as that predicted by theoretical values which were obtained by an exact calculation under the assumption that we have all the information about the whole "noise" environment. On the contrary we see that the characteristic of the LOF detector
deviates from that predicted by the theoretical values since there is no fuzziness in the observation.

In Tables 2-4, however, the errors associated with the LOF detector are reduced whereas those associated with the LOQ detector are increased, when the distributions of the self-noise are $N(0,0.5), N(0,1)$, and $N(0,3)$. We see that for a preassigned size $\alpha$ the LOF detector has a similar characteristic to that of the LOQ detector obtained with perfect information about the noise environment, while the LOQ detector without information about the self-noise sacrifices as the self-noise becomes important to the overall noise characteristics. When the variance of the self-noise is 0.5, it is hard to say whether the LOF detector is better than the LOQ detector. This is due to the fact that the credibility is chosen to be so large relative to the fuzziness resulting from the self-noise. When the variance of the self-noise is 3, however, we may conclude that the credibility is chosen appropriately. This shows that the choice of the credibility and the shape of the membership function is an important factor in the design and performance analysis of fuzzy detectors.

Now note that the size of the LOQ detector in Tables 2-4 is greater than 0.05155 although the threshold was set to satisfy the constraint of size $\alpha$, since the threshold was calculated without considering the self-noise. The reason for the low $P_d$ of the LOF detector is, hence, due to this low $P_{fa}$ compared to that of the LOQ detector.

As a final observation, let us consider the following. As it is known, when we compare the performance of two tests, it is required to set the false-alarm probabilities of the two tests equal in the Neyman-Pearson procedure. Let us now consider the generalization of Theorem 3.

**Theorem 4.** If we make the LOQ and LOF detectors have the same size $\alpha$ based on the classical probabilistic definition, the two detectors will make the same decision for the same observation provided that the order of the ordered fuzzy information space is preserved after the fuzzy test problem is converted into a test problem on the crisp information space.

**Proof:** If $W_f$ is converted into the corresponding subspace of the ordered crisp information space under the assumption that the order is preserved, then the only way to achieve the same size for the two detectors based on the classical probabilistic definition is to make the converted $W_f$ equal to $W_c$. Then the two detectors make the same decision for the same observation.

It is immediate from Theorem 4 that the two detectors have the same probability of detection when the order of the ordered fuzzy information space is preserved after the fuzzy test problem is converted into a test problem on the crisp information space. It should be remarked that preservation of the order is not a very strong condition and we can find many examples which satisfy the assumption.

**5. Conclusion**

In this paper, statistical characteristics of the locally optimum fuzzy detector for known signals were investigated, and the performance of the detector was analyzed. We showed that the locally optimum fuzzy detector provides performance similar to that of the locally optimum detector which uses crisp information for a preassigned false-alarm probability with perfect information about the whole noise environment: the crisp locally optimum detector without complete information about the self-noise, however, sacrifices as the self-noise becomes more important to the overall noise characteristics. In other words, the locally optimum fuzzy detector has a robustness property.

**References**


Tests for Optimizing Sensor Positions Using Linear Regression

A.M. Zoubir, T. Grewing and J.F. Böhme

Department of Electrical Engineering, Ruhr University Bochum, D-4630 Bochum, FRG

In this study, we propose a method for finding optimum sensor positions in a group of vibration sensors for knock detection in spark ignition engines. It differs from other techniques in that only signal processing and statistical tests are used. Our method is based on linearly predicting a reference signal from the output of an arbitrary sensor group. We derive a linear regression model in the frequency domain and discuss rank tests for various hypotheses that are tested with respect to the model parameters. This leads us to a technique for testing sensor irrelevancy. Simulation and experimental results emphasize the usefulness of the method.

1. Introduction

The efficiency of a spark ignition engine can be improved by an increase of its compression ratio. However, for high compression ratio engines the angle of ignition for minimum fuel consumption lies in a region where knock occurs. Knock is known as an abnormal combustion of the gas mixture causing rapid rise of temperature and pressure. This may lead to engine damage, particularly when it occurs at high speed. Knock detection systems with structural vibration sensors are used to control high compression engines for better fuel economy while avoiding engine damage as well as noise annoyance to the vehicle occupants.

The use of vibration sensors on the surface of the engine is easy and economical but makes the detection of knock difficult. This is due to the poor signal to noise power ratio (SNR) of the vibration signals. It is obvious that knock detectability can be improved if the vibration sensors are placed optimally on the engine surface. Holographic techniques for imaging knock centres have been proposed, but their application is complicated and does not necessarily lead to a practical solution [9]. Up to now, the sensor positions have been chosen heuristically.

In this contribution, we present a method for finding optimum positions of sensors within a sensor group being positioned for example heuristically on the engine. The method is less complex than holographic techniques because only signal processing and statistical tests are used. Alternatively to [10], we propose nonparametric tests.

In section 2., we introduce the data model. In section 3., we propose rank based tests for identifying optimum sensor positions in a group of sensors. Section 4. concerns test results with simulated data. We discuss experimental results in section 5. before concluding.

2. Data Model

Let \( S(t) \) be a reference signal and \( Z_i(t), i = 1, \ldots, q/2 \) sensor signals, all zero-mean and stationary, \( t = 0, \pm 1, \pm 2, \ldots \). The series \( S(t) \) and \( Z_i(t) \) model the cylinder pressure signal recorded via a pressure transducer during a combustion cycle for reference purposes and a vibration sensor signal, respectively. Consider the linear prediction model

\[
S(t) = \sum_{i=1}^{q/2} \sum_{u=0}^{\infty} h_i(u) Z_i(t-u) + \mathcal{E}(t),
\]

where \( h_i(\cdot) \) is the \( i \)th impulse response of the prediction filter and \( \mathcal{E}(t) \) is the prediction error series. Suppose we are given \( N/2 \) data blocks of length \( T \) each, \( S(t) = (S^1(t), \ldots, S^{N/2}(t))^T \), \( Z_i(t) = (Z_i^1(t), \ldots, Z_i^{N/2}(t))^T \), \( i = 1, \ldots, q/2 \), \( t = 0, \ldots, 2T-1 \). We compute the finite Fourier transforms of the time series data, and (1) becomes

\[
d_\mathcal{E}(\omega) \approx \sum_{i=1}^{q/2} H_i(\omega) \cdot d_{Z_i}(\omega) + d_\mathcal{E}(\omega),
\]

approximately a complex regression at each frequency \( \omega \). For the sake of simplicity, we drop the argument \( \omega \). Assuming independence and the same distribution function for the real and imaginary part of \( d_\mathcal{E} \), we obtain the real valued linear regression

\[
Y = V \cdot B + E.
\]

Herein, \( Y = (\text{Re} \{d_{Z_1}\}, \text{Im} \{d_{Z_1}\})^T \) and \( B, V \) contain the suitable arranged real and imaginary parts of the prediction filter frequency responses and the Fourier transformed sensor signals, respectively.

The aim of this work is to present an approach to reduce the dimension of \( Z(t) \), leading to outputs of sensors that are optimally positioned for knock detection. For this purpose, we use a nonparametric test to first decide whether there is no sensor giving relevant information on the reference signal by testing the corresponding coefficient \( B \) to be zero. In the case of rejection, we then apply a test to decide the relevancy of additional sensors.
3. Quadratic Rank Tests

Parametric methods to test the hypothesis $H: B = 0$ against the alternative $K: B \neq 0$ assume the vector $E$ to be normally distributed. Though, this assumption holds asymptotically for large $T$ [3], it is not necessarily guaranteed in our case. Therefore, we use the quadratic rank test proposed by Adchiche [1]. Following [1], we calculate

$$M_N = \frac{(V'\psi_N(\mathcal{R}_Y(Y)))(V'V)^{-1}(V'\psi_N(\mathcal{R}_Y(Y)))}{A^2(\psi)},$$

where $\psi_N(\mathcal{R}_Y(Y))$ is a function of the ranks $\mathcal{R}_Y(Y)$ of $Y$ with $\psi_N(i) = \psi(i/(N + 1))$ and $A^2(\psi) = \int \psi(u)^2 \, du - \bar{\psi}^2$, $\bar{\psi} = \int \psi(u) \, du$. For the Wilcoxon scores generated by $\psi(u) = u$ that are used in this contribution, we have $A^2(\psi) = 1/12$ [4].

The exact distribution of $M_N$ under the hypothesis $H$ can be calculated by permutation technique for any given matrix $V$. However, this requires $N!$ computations of the statistic. This can be inpracticable in some cases where $N$ is large. Thus, we use the asymptotic distribution of $M_N$ that has been proposed in [1]. Under $H$ and some regularity conditions, the statistic in (4) has asymptotically for large $N$ a chi-square distribution with $q$ degrees of freedom [1].

For the decision, whether additional sensors improve detection, it is necessary to test only some components of $B$. For this, it is more convenient to rewrite (3) in the form

$$Y = V_1B_1 + V_2B_2 + E,$$

where $B = (B_1', B_2')'$ and $V = (V_1, V_2)$. The vectors $Y$ and $E$ are as before $N$ vector-valued. $B_1$ is $q_1$ vector-valued and the matrix $V_1$ has dimension $N \times q_1$ for $s = 1, 2$. It is of interest to test $H_2$: $B_2 = 0$, $B_1$ unspecified, against the alternative $K_2: B_2 \neq 0$. Herein, $B_1$ corresponds to already chosen sensors and $B_2$ to further ones. A test similar to the one given above has been proposed by Adchiche [1] and is summarized below.

To eliminate the influence of the unknown vector $B_1$, an estimate $\hat{B}_1$ for $B_1$ is calculated. In this contribution, we use the least squares estimate. Though, any other estimate that is consistent and translation invariant might be used. An estimate $\hat{B}_1$ for $B_1$ is translation invariant if for all $B_1$, $\hat{B}_1(Y - V_1B_1) = \hat{B}_1(Y) - B_1$, where $\hat{B}_1(Y)$ denotes estimates computed from $Y$ [1]. With this estimate, the cleared data

$$\hat{Y} = Y - V_1\hat{B}_1$$

are computed. Let us now define the matrices

$$C_1 = V_1(V_1'V_1)^{-1}V_1',$$

$$\hat{T}_2 = V_2(I_N - C_1)\psi_N(\mathcal{R}_\hat{Y}(\hat{Y})), \quad C_2 = V_2(I_N - C_1)V_2,'$$

where $I_N$ is the identity matrix of dimension $N \times N$. For testing the hypothesis $H_2$ against $K_2$, Adchiche proposes the statistic

$$\hat{M}_N = \frac{\hat{T}_2C_2^{-1}\hat{T}_2}{A^2(\hat{\psi})}$$

Under the hypothesis $H_2$, $\hat{M}_N$ has asymptotically for large $N$ a chi-square distribution with $q_2$ degrees of freedom. This holds under some regularity conditions stated in [1,8].

To compare the suggested tests with classical likelihood ratio tests, we employ a measure of relative efficiency due to Pitman, cf. [7,1]. Given two tests of the same size of the same statistical hypothesis, the relative efficiency of the second test with respect to the first is given by the ratio $n_1/n_2$, where $n_2$ is the sample size of the second test required to achieve the same power for a given alternative as is achieved by the first test with respect to the same alternative when using a sample of size $n_1$. In general, the relative efficiency will depend on $n_1$ as well as on the particular alternative chosen. Thus, we use the relative asymptotic efficiency, taking a sequence of alternatives changing with $n$ in such a way that as $n \to \infty$ the power of the corresponding sequence of tests converges to some number less than 1. Andrews [2] has shown that if under the same sequence of alternatives, two test statistics have noncentral chi-square limit distributions with the same degrees of freedom, then their relative asymptotic efficiency is given by the ratio of their noncentrality parameters. Considering the suggested test and the classical likelihood ratio test for the hypothesis $H_2$ and a sequence of alternatives $K_2: B = b/\sqrt{N}$ ($B_2 = b_2/\sqrt{N}$), $||b|| < c$, $||b|| < ||b_2|| < c$, the asymptotic relative efficiency of $M_N(\hat{M}_N)$ yields to

$$\varepsilon = \sigma^2(F)/B^2(F)/A^2(\hat{\psi}).$$

Herein, $F$ denotes the distribution function of the components of $Y$ with finite variance $\sigma^2(F)$ and $B^2(F) = \int \psi'(F(u)) \, dF(u)$. Expression (7) was shown never to fall beneath 0.956 by Hodges and Lehmann [6]. In the normal distribution case, we have $\varepsilon = 2/\pi \approx 0.655$. This result shows that use of quadratic rank tests can never entail a serious loss of efficiency.

In practice, the simultaneous consideration of some frequencies of interest often arises. Especially in knock detection, we are interested in a few number of resonance frequencies that are excited by knock [5]. Because of dampings due to the physical process, we also consider neighboring frequencies. Thus, we test

$$H_0 := \bigcap_{p=1}^{P} \bigcap_{k=-m}^{m} H_{p,k}, \quad H_{p,k} : B_{p,k} = 0$$

against the two-sided alternative $K_0: B_{p,k} \neq 0$ for at least one pair $(p,k)$. Herein, $B_{p,k} = B(\omega_{p,k})$, $k = -m, \ldots, m$, where $\omega_{p,0} = \omega_p$, $p = 1, \ldots, P$ denote resonance frequencies. Under the assumption that the $M_N$ statistics at different frequencies are independent which
holds for large $T$, we may test the irrelevancy of a sensor by considering
$M_N = \sum_{p=1}^{P} \sum_{m=-m}^{m} M_N(\omega, \delta)$. Under
the hypothesis no regression possible, i.e. the sensor is
irrelevant, this sum is approximately chi-square dis-
tributed with $(2m + 1)^P$ degrees of freedom. We reject
$H_0$ if, for a given level of significance $\alpha$, the statistic
$M_N$ is too large. The boundary of the rejection region
can be determined from a chi-square table. Similarly
we construct $M_N = \sum_{p=1}^{P} \sum_{m=-m}^{m} M_N(\omega, \delta)$ to test
$H_0 : \cap_{p=1}^{P} \cap_{m=-m}^{m} H_{2, \delta, \delta}, B_{2, \delta, \delta} = 0$ against
the two-sided alternative.

In order to find the optimally placed sensor group with
dimension lower than $q/2$, we propose a stepwise pro-
dure. First, we test whether a single sensor is suitable to
predict the reference signal. For this purpose, we com-
pute $M_N$. Sensors for which the hypothesis $H_0$ is not
rejected are removed from the group. Among the re-
mainig sensors, the one concerning the highest $M_N$ is
chosen. In the following steps, the gain reached by
adding a further sensor to the already determined one
is tested. Using $M_N$, we remove sensors for which the
hypothesis $H_0$ is not rejected and select the one with
the highest $M_N$. This procedure is repeated until enough
sensors are determined or no sensor is left, i.e. all sensors
are either chosen or removed.

4. Simulations

To investigate the proposed selection procedure, we have
simulated a pressure signal and various sensor signals.
The model
\[
S(t) = \sum_{p=1}^{P} A_p e^{-\delta t} \cos(\omega_p t + \Phi_p) + Z(t)
\]
is used for the reference signal. Herein, $A_p$ and $\Phi_p$ are
mutually independent random amplitudes and phases, re-
spectively, $\delta_p$ dampings, $\omega_p$ the cavity resonance fre-
dquencies for $p = 1, \ldots, P$ and $Z(t)$ is a white noise
process. To simulate the pressure signal, we have fixed
$P = 4$ and generated 200 records. We have then simu-
lated eight sensor signals. For this, some of the bands
centered at the four resonance frequencies have been fil-
tered out. These are numerated in increasing order and
given in Table 1. Noise has been added to generate a
sensor signal. The estimated SNRs of the sensor signals
are given in Table 1.

In the first step, the procedure yields to the expected
sensor $S_1$ with the highest SNR. In the next step, sen-

Table 1: $M_N$ and $\hat{M}_N$ of the simulated signals. Removed sensors appear slanted, selected ones boldfaced.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>S1</th>
<th>S2</th>
<th>S3</th>
<th>S4</th>
<th>S5</th>
<th>S6</th>
<th>S7</th>
<th>S8</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNR (dB)</td>
<td>-28</td>
<td>-28</td>
<td>-28</td>
<td>-28</td>
<td>-28</td>
<td>-28</td>
<td>-28</td>
<td>-28</td>
</tr>
<tr>
<td>Reson.</td>
<td>1.2</td>
<td>1.5</td>
<td>1.4</td>
<td>2.3</td>
<td>2.4</td>
<td>3.4</td>
<td>4.2</td>
<td>3.2</td>
</tr>
<tr>
<td>1. Step</td>
<td>2.38</td>
<td>194</td>
<td>196</td>
<td>151</td>
<td>153</td>
<td>178</td>
<td>130</td>
<td>80</td>
</tr>
<tr>
<td>2. Step</td>
<td>143</td>
<td>157</td>
<td>138</td>
<td>152</td>
<td>165</td>
<td>127</td>
<td>70</td>
<td></td>
</tr>
<tr>
<td>3. Step</td>
<td>111</td>
<td>130</td>
<td>112</td>
<td>126</td>
<td>91</td>
<td>50</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4. Step</td>
<td>97</td>
<td>108</td>
<td>117</td>
<td>75</td>
<td>50</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The engine was running at 5250 rpm and at full load.
Strong knock intensity was adjusted. Three thousand
cycles were digitized with respect to cylinder 3 for our
analyses. In order to identify knocking cycles, we com-
pared the difference between successive samples of the
cylinder pressure signal at an interval ranging from 10 to
40° crank angle with a suitably chosen threshold. This
method exploits the fact that knock results in a steep
pressure rise. The decision was verified by comparing
the estimated pressure signal power of the selected cy-

Figure 1: Sensor positions on the engine.

The engine was running at 5250 rpm and at full load.
Strong knock intensity was adjusted. Three thousand
cycles were digitized with respect to cylinder 3 for our
analyses. In order to identify knocking cycles, we com-
pared the difference between successive samples of the
cylinder pressure signal at an interval ranging from 10 to
40° crank angle with a suitably chosen threshold. This
method exploits the fact that knock results in a steep
pressure rise. The decision was verified by comparing
the estimated pressure signal power of the selected cy-

5. Experimental Results

Experiments were performed using a 1.8 L, 4 cylinder
engine with production knock control system. The high-
pass filtered pressure signals of the cylinders 1 and 3 (D1
and D3 in Figure 1) was recorded by an instrumentation
tape recorder together with the output of eight accelera-
tion sensors. Two of these sensors (sensor SD and S2 in
Figure 1) had been mounted at the place of the standard
sensor, the other ones at heuristically chosen positions
on the engine wall and on the cylinder head (sensor SK1).
The sensors SA2 and SA23 had been mounted at the out-
let side of the engine block, all others at the inlet side.
Figure 1 schematically shows these positions. With the
exception of the standard sensor SD, all the sensors were
of the same type.
Resonance frequencies were estimated by averaging periodograms of the knocking cycles of $S(t)$. Spectral peaks were found at 8.7, 13.1, 17.4 and 21.8 kHz. Two neighboring frequency bins to each resonance frequency have also been considered and the tests were performed as described above.

<table>
<thead>
<tr>
<th>Sensor</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Step</td>
<td>S2</td>
<td>S23</td>
<td>S3</td>
<td>S34</td>
<td>S42</td>
<td>S423</td>
</tr>
<tr>
<td>1.</td>
<td>141</td>
<td>201</td>
<td>127</td>
<td>122</td>
<td>170</td>
<td>223</td>
</tr>
<tr>
<td>2.</td>
<td>101</td>
<td>118</td>
<td>103</td>
<td>154</td>
<td>197</td>
<td>110</td>
</tr>
<tr>
<td>3.</td>
<td>45</td>
<td>65</td>
<td>57</td>
<td>74</td>
<td>-</td>
<td>48</td>
</tr>
<tr>
<td>4.</td>
<td>38</td>
<td>48</td>
<td>39</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 2: $\mathcal{M}_N$ and $\hat{\mathcal{M}}_N$ of the real signals. Removed sensors appear slanted, selected ones boldfaced.

The experimental results are given in Table 2. The proposed procedure yields to sensor S423 of which signal gives the highest $\hat{\mathcal{M}}_N$. The hypothesis $H_0$ for testing sensor SK1 in the first step is not rejected. Thus, we remove this sensor from the group and proceed in the second step with the test of the hypothesis $H_2$ in order to add a further sensor to S423. It should be noted that sensor SK1 was rather intended to detect noise generated by valve train than to detect knock. Sensor S34 which is placed between cylinder 3 and 4 is determined in the second step. Together with sensor S423, it constitutes the optimally placed group of two sensors. This result is realistic because we have only considered cylinder 3 and the two sensors found are close to it. Nevertheless, Table 2 shows that sensor S3 performs poorly and is removed in the fourth step.

From this experiment, it is seen that the position proposed by the engine manufacturer (sensors SD and S2) is not optimum. However, it should be noted that this proposal was based on heuristic considerations. The approach presented here has also been applied to further cylinders and rotation speeds. In some cases where the engine speed is low and cylinder 1 is considered, good results have been registered for sensor S2 or SD. However, a selection procedure developed in [4] shows that both sensors perform poorly when considering simultaneously three rotation speeds and two cylinders.

6. Conclusions

In the preceding sections, we have presented a method to test the relevancy of sensors in an arbitrary array that is used to detect knock in spark ignition engines. Based on the linear prediction of a reference signal from vibration signals, we have derived a linear regression model in the frequency domain. We have then discussed quadratic rank based tests for hypotheses with respect to the model parameters. We have claimed that the relative asymptotic efficiency of the rank tests relative to the corresponding likelihood ratio tests is close to one. The latter tests assume normality of the regression error that is not necessarily guaranteed in our case. We have proposed a stepwise procedure for identifying an optimally placed single sensor or a sensor group under consideration of multiple resonance frequencies. The method has been investigated by simulations. We have shown that in the very noisy case, the procedure leads to the expected sensors. Results of an experiment performed on a 4 cylinder engine suggest a sensor position that differs from the one proposed by the engine manufacturer.

REFERENCES


PATTERN RECONSTRUCTION USING PARAMETRIC TIME-FREQUENCY METHOD

J. MARS
CEPHAG- ENSIEG
Domaine Universitaire BP 46, 38402 Saint-Martin d'Hères Cedex, France

Non stationary signal processing methods are often applied to signals obtained from mechanics and rotating machines. This paper concerns the study of signals whose energetic time frequency representation exhibits several patterns. Firstly, non related Fourier time-frequency methods named Sliding Lagunas methods is used in order to characterize patterns. Then a new filtering applied to the time-frequency domain is proposed in order to extract the individual objects of this representation. The validation of this processing is shown on synthetic data.

1. INTRODUCTION

Time-frequency signal analysis has recently becomes the focus research as more and more analysts are confronted with the widely reported insufficiencies of classical signal analysis tools, based either on time domain or frequency domain representations of the signal. In this paper, several non stationary methods are proposed : Short Time Fourier Transform (STFT), Sliding Lagunas method issued from Capon's method (Filter bank). Opposite to Short Time FT, Lagunas and Capon's methods do not need Fourier analysis, but they require a knowledge on the signal structure. The original method is issued from classical spectral analysis. A sliding memory of the filter along the total signal duration allows the formulation of an adaptive spectral density. A local stationarity of the signal is necessary. A time-frequency representation allows us to localize different patterns, and we apply a filtering in time-frequency on this result to select each pattern [3].

2. TIME-FREQUENCY REPRESENTATIONS

2.1. SHORT-TIME TRANSFORM FOURIER

Given a finite energy signal \( x(t) \) and a sliding window \( h(t) \), a classical linear time-frequency representation can be obtained by computing the Short Time Fourier Transform (STFT) [1]:

\[
\text{STFT}(t;\nu) = \int_{-\infty}^{+\infty} x(k) h^{*}(k-t) e^{-2\pi i \nu k} \, dk
\]

The spectrogram defined as the energy distribution associated to the STFT, ie \( |\text{STFT}(t;\nu)|^2 \), has been widely used for many signal processing tasks. The local stationarity is needed on successive windows. A classical time and frequency resolution trade off underlies the structure of the spectrogram : the choice of an analyzing window of short duration ensures a good time localization but, at expenses of a poor frequency resolution by Fourier duality.

2.2. CAPON'S METHOD IN TIME FREQUENCY REPRESENTATION

The objective of Capon's method [2],[4] is to build a band-pass filter around a frequency \( \nu_k \) while ensuring the following conditions:

a) The spectrum at the frequency \( \nu_k \) is not altered.

b) The influence of interferences due to other frequencies is minimized.

For this, Capon uses a moving average filter (MA) whose input-output relation is given by:

\[
Y(n) = A^T \cdot S(n)
\]

where \( T \) denotes the transpose, with \( S^T(n) = [S(n), ..., S(n-P)] \) the filter input,

\[
A^T = [A_0, ..., A_P]
\]

the impulse response of the filter,

\( Y(n) \) the filter output.

The first condition implies that, at frequency \( \nu_k \), the transfer function of the filter equals one. To satisfy the second condition, the interferences due to other frequencies present in the signal are minimized at each time. The two conditions lead to:

a) \( A^H Z_k = 1 \),

b) \( A^H R_s A \) minimum,

with \( A^H \) is the conjugate transpose.
The solution to this problem of minimization under constraint using Lagrange multipliers is:

\[ \Delta = \left( \frac{R_{z}^{-1} \cdot z_k}{Z_k} \right) \]

and gives the impulse response of the Capon filter. The output power of the filter is an estimator of the power of the signal at frequency \( \nu_k \). By generalizing to all frequencies, the power estimator can be written:

\[ P_{\text{cap}} = \frac{1}{\Delta^T R_z \Delta} \]

With this method, the estimated filter is adapted to the signal at each frequency, whereas, in the Fourier analysis, the estimated filter is independent of the signal. It is clear that Capon's method estimates the power from the estimation of the signal correlation matrix and its resolution is better than spectrogram resolution. In a nonstationary context, a sliding memory of the filter along the total signal duration allows the formulation of an adaptive spectral density [7].

### 2.3. PRINCIPLES OF SLIDING LAGUNAS'S METHOD

To obtain a power spectrum density estimator called Lag(n,v) from Capon's method, we normalize the power \( P_{\text{cap}} \) by the effective bandwidth \( B_{\text{eq}} \):

\[ P_{\text{cap}} = \int_{-B_{\text{eq}}/2}^{B_{\text{eq}}/2} \text{Lag}(n,v) \left| A(v,v') \right|^2 dv' \]

If \( B_{\text{eq}} \) is very tight, we can consider that the spectral density is flat over the band \(|A(v,v')|=1\). Then

\[ P_{\text{cap}} = \int_{-B_{\text{eq}}/2}^{B_{\text{eq}}/2} \left| A(v,v') \right|^2 dv' \]

The final spectral density is:

\[ \text{Lag}(n,v) = \frac{P_{\text{cap}}}{A^2} = \frac{Z_k^T R_{z}^{-1} Z_k}{Z_k^T R_{z}^{-1} R_{z}^T R_{z}^{-1} Z_k} \]

Spectral information is available at the output of the filter. This original method is issued from classical spectral analysis, and a sliding memory of the filter along the total signal duration permits the formulation of an adaptive spectral density.

### 3. PRINCIPLE OF FILTERING IN TIME-FREQUENCY DOMAIN

The result of time frequency methods on a nonstationary signal gives several patterns in time-frequency domain. Generally, direct separation of patterns in time or in frequency is not easy. By a filtering process in the time frequency domain, we propose to isolate each pattern.

Consider a signal \( S(t) = S_1(t) + S_2(t-t) \), where \( S_1(t) \) is a chirp. We choose a basis of projection as:

\[
\begin{align*}
\mathbf{M}(t) &= F_T(t-T/2) \cdot \cos(\Phi(t)) \\
\mathbf{MQ}(t) &= F_T(t-T/2) \cdot \sin(\Phi(t))
\end{align*}
\]

with \( F_T(t) = 1 \) if \( t \geq 0 \), and \( 0 \) if \( t < 0 \), and \( \Phi(t) \) is the instantaneous phase term.

In this basis, each component can be written as:

\[
\begin{align*}
S_1(t) &= P_1(t) \cdot M(t) - Q_1(t) \cdot MQ(t) \\
S_2(t) &= (P_2(t) \cdot M(t) - Q_2(t) \cdot MQ(t)) \\
&+ (P_2(t-t) \cdot M(t-t) - Q_2(t-t) \cdot MQ(t-t))
\end{align*}
\]

The principle of filtering is to separate the component \( P_1(t) \) and \( Q_1(t) \) of \( P_2(t-t) \) and \( Q_2(t-t) \). For this, we multiply \( S(t) \) by each model \( M(t) \) and \( MQ(t) \). In this product, we obtain an expression with low frequency and high frequency terms. The spectrum of \( P_1(t) \) is centered on the 0 frequency with a bandwidth \( \Delta v_1 \) (\( \Delta v_1 \) is a maximal bandwidth around a pattern). With a low pass filter on low frequency term, it is possible to select \( P_1(t) \) (figure 1).

![Figure 1: Process Filter](image)

The main difficulty is to construct the model. For this, we plot on the time-frequency representation, the points of the selected pattern in order to compute its instantaneous frequency \( f(t) \), where \( f(t) = \frac{1}{2\pi} \frac{d\Phi(t)}{dt} \)

with \( \Phi(t) = \arctan \left( \frac{M(t)}{\dot{M}(t)} \right) \)
\( \tilde{M}(t) \) is the Hilbert transform of \( M(t) \). Generally, the instantaneous frequency presents several wrong points. In order to obtain a correct model without this points, it is necessary to apply an interpolation. Finally, we can consider that the filtering method is a variable frequency demodulation whose development makes up of three steps:
- Elaboration of models.
- Multiplication by the models.
- Low-pass filter.

4. APPLICATION

A synthetic model composed by a summation of a sinusoidal frequency modulation (Signal A) and a parabolic frequency modulation (Signal B) was used to test these processes. The used parameters for the signal A are \( v_{\text{min}} = 0.3 \) Hz; \( v_{\text{max}} = 0.4 \) Hz; and for the signal B these parameters are: \( v_{\text{min}} = 0.05 \) Hz; \( v_{\text{max}} = 0.4 \) Hz. The sampling frequency is 1 Hz. The signal to noise ratio is 20 dB. Temporal representation of its signal recorded on 2048 sampling is given on figure 2.

![Figure 2: Synthetic data](image)

A time frequency representation \( F-t-p \) results from the STFT method with an analyzing window of 63 samples is presented on figure 3. The frequency \( F \) and the time \( t \) are respectively plotted on the abscissa and on the ordinates. The module \( p \) of each frequency represents the third dimension. We can observe an estimation of energy distribution which gives an information on the number of frequencies presents at each time on this signal (Two frequencies in this case).

With a sliding Laguna's method used with an order equal to 14 (7 real frequencies are sought: two for the signal and five for the noise) on a window length equal to 51 samples (memory), we can observe clearly each component in a time frequency representation (figure 4). In this case, the module \( p \) represents the four maxima of each spectrum. At each time, we have two frequencies associated to the signal and two others frequencies at random position which characterizes noise. Using this representation, we can constructs two models for filtering. For this application, only one model is plotted (figure 5). Time frequency representation of sliding Laguna's method applied on the result of the filtering with different bandwidth \( \Delta f \) is presented. If the \( \Delta f \) is short (0.01 Hz), the selection of the pattern is efficient (figure 6a), but if the bandwidth of the low pass filter increases, outline of the filter appears in the time frequency representation, where the module \( p \) represents the \( n \)th maxima of each spectrum. To have an information on the power of the frequencies obtained, it is necessary to applied Capon's method on the data (see also 2.2). Therefore, by associating the Laguna's method and Capon's method, a hybrid method is obtained [7]. The result of this association on the signal filtered with a large bandwidth \( \Delta f \) (0.1 Hz) shows the signal A and a part of signal B (figure 6b). The signal B is characterized by the difference between initial data and the result of filtering (figure 7).

5. CONCLUSION

The use of parametric time frequency methods not based on Fourier analysis allows us to detect clearly dispersive patterns. Thus the construction of models is easier. The application of filtering which is a variable frequency demodulation by choosing models permits to characterize and separate different patterns. This technique which has been validated on simulation, has been efficient to separate two dispersive subsurfaces waves coming from an earthquake located in Aleoutian Islands [6].

6. REFERENCES


Maximum Likelihood Source Location Estimation via EM Algorithm

D. Kraus, D. Maiwald and J.F. Böhme

Department of Electrical Engineering, Ruhr University Bochum, 4630 Bochum, FRG

Approximate maximum likelihood source location estimation is investigated. An extended model of the spectral density matrix of the sensor array output for coherent sources (multipath propagation) is introduced. We show that the EM algorithm can be successfully applied for this special case of practical importance. In contrast to [1], the EM iteration scheme is derived by exploiting that the distribution type of the finite Fourier transformed sensor outputs belongs to the exponential family. Finally, we investigate the loss of information due to the missing data.

1. Introduction

Approximate maximum likelihood estimates (AMLEs) for source locating of wide band sources in the presence of partly unknown noise fields are developed. In comparison to other estimation methods, a significant disadvantage is the cumbersome computation of AMLEs. An efficient alternative to a brute-force algorithm, i.e. global search and a local optimization technique, is the use of the expectation maximization (EM) algorithm, cf. [8], [2], [3] and [1]. In continuation of [1], we introduce an extended model of the sensor array output for coherent sources (multipath propagation) and derive a modified EM iteration scheme by exploiting the distributional properties (exponential family) of the finite Fourier-transformed sensor output of successive data pieces, cf. [8]. This approach requires neither the additional sensor noise parameters $\sigma_0^2$ used in [1] nor an alternating step by step optimization of spectral and source location parameters, but in increase in variance and mention how it can be reduced by spatial filtering if instead of the complete data only the incomplete data can be measured.

An outline of the paper follows. In section 2, the data model and the parameter structure are introduced. The EM algorithm for incomplete data from an exponential family is developed in section 3. In section 4, the EM algorithm is applied for AMLE of source location. In section 5, the information loss is derived. We conclude with some remarks.

2. Data Model

A conventional propagation-reception model is used. Sources generate signals which are transmitted by wavefields. At an array of sensors, the signals are measured additively disturbed by noise. The outputs of the sensors $n = 1, \ldots, N$ are finite Fourier transformed with a smooth, normalized window of length $T$. For every frequency $\omega$ of interest, we get data $X^k(\omega) = \{X_1^k(\omega), \ldots, X_N^k(\omega)\}^T$ of $k = 1, \ldots, K$ successive pieces of sensor outputs.

Suppose there are $M'$ groups of uncorrelated signals. Each group consists of $L_i$ ($i = 1, \ldots, M'$) correlated signals with $\sum_{i=1}^{M'} L_i = M$. Then, the $(N \times N)$ spectral density matrix of the array output can be expressed by

$$C_{\mathbf{X}}(\omega) = \sum_{i=1}^{M'} C_{\mathbf{X}i}(\omega)$$

where

$$C_{\mathbf{X}i}(\omega) = \mathbf{H}_i(\omega)C_{\mathbf{S}}(\omega)\mathbf{H}_i(\omega)^* + \sigma_i(\omega)\mathbf{I}$$

and

$$C_{\mathbf{X}i}(\omega) = \nu_i(\omega)\mathbf{I}$$

with $\sigma_i(\omega)$ known, $\sigma_i(\omega) \geq 0$ and $\sum_{i=1}^{M'} \sigma_i(\omega) \ll \nu_i(\omega)$. The columns of the $(N \times L_i)$ matrix $\mathbf{H}_i(\omega)$ are known as steering vectors $b_{ij}(\omega)$ ($i = 1, \ldots, L_i$) that contain the unknown wave parameter vector $\xi_i$ ($i = 1, \ldots, M'$), e.g. bearings $b_{ij}$ and ranges $r_{ij}$ ($i = 1, \ldots, L_i$) of the signals. The signal spectral parameter vectors $\xi_i(\omega)$ ($i = 1, \ldots, M'$) are given by the entries of the $(L_i \times L_i)$ matrix $C_{\mathbf{S}}(\omega)$ itself. The noise spectral parameters are collected in the vector $\nu(\omega) = (\nu_1(\omega), \ldots, \nu_M(\omega))^T$ and the matrices $\mathbf{J}_i(\omega)$ ($i = 1, \ldots, I$) are supposed to be known nonnegative hermitian matrices. For example, they describe angle bandlimited noise of different directions and widths.
Furthermore, let the groups of signals with \( L_i > 1 \) (\( i = 1, \ldots, M' \)) to be due to multipath propagation (perfect coherence), \( C_{\mathbf{Y}_k}(\omega) \) can be rewritten to

\[
C_{\mathbf{Y}_k}(\omega) = \sigma^2(\omega) d_0(\omega) + \epsilon(\omega) I \quad (i = 1, \ldots, M')
\]

where \( d_0(\omega) \) denotes for \( L_i = 1 \) or \( L_i > 1 \) (\( i = 1, \ldots, M' \)) the usual or a generalized steering vector, respectively.

For an arbitrary array geometry the generalized steering vector is given by

\[
d_0(\omega) = \sum_{i=1}^{L_i} \kappa_i(\omega) d_i(\omega), \quad \kappa_i, \kappa_i \equiv 1, \kappa_i \in \mathbb{C},
\]

where \( d_i(\omega) \) denotes the usual steering vector for the \( i \)-th propagation path of the \( i \)-th signal group (source).

If a line array is employed and plane wave propagation is assumed \( d_0(\omega) \) can be expressed by

\[
d_0(\omega) = \left( \sum_{i=1}^{L_i} \tau_i(\omega) D_i(\omega) \right) d_i(\omega),
\]

\[
\tau_i, \kappa_i \equiv 1, \kappa_i \in \mathbb{R},
\]

with

\[
D_i(\omega) = \text{diag} \left( 1, e^{-j\omega\tau_{i1}}, \ldots, e^{-j\omega(N-1)\tau_{i1}} \right),
\]

\[
\tau_{i0} = 0, \tau_{i1} \in \mathbb{R}.
\]

The \( C_{\mathbf{Y}_k}(\omega) \) can be understood as the spectral density matrix corresponding to a virtual array output \( \mathbf{Y}_k(\omega) = \left( \mathbf{Y}_{1k}(\omega), \ldots, \mathbf{Y}_{nk}(\omega) \right)^T \) (\( k = 1, \ldots, K \)) generated either by a group of correlated signals or by a noise component.

The incomplete data \( \mathbf{X}_k(\omega) \) are related to the complete one, \( \mathbf{Y}_k(\omega) = \left( \mathbf{Y}_{0k}(\omega)^T, \ldots, \mathbf{Y}_{Mk}(\omega)^T \right)^T \) by the linear many-to-one mapping

\[
\mathbf{X}_k(\omega) = \mathbf{I} \mathbf{Y}_k(\omega).
\]

Under the conditions leading to (4), the number of unknown parameters using model (1), model (2) with (3) and model (2) with (4) are \( P = L_0 + L_1 \), and \( P = 3L_1 - 1 \), respectively. The parameter vector is given by \( \theta(\omega) = \left( \xi_{1k}, \ldots, \xi_{Mk}, \xi_{0k}(\omega)^T, \ldots, \xi_{Mk}(\omega)^T, \epsilon(\omega)^T \right)^T \), where \( \xi_{ik} \) the wave parameters and \( \xi_{0k}(\omega) \) the signal parameters of the \( i \)-th group of signals corresponding to the chosen model are suitable collected. Generally, the numbers \( M' \) and \( L_i (i = 1, \ldots, M') \) are unknown. In this paper we assume to know \( M' \) and \( L_i (i = 1, \ldots, M') \) with \( \sum_{i=1}^{M'} L_i = M < N \).

In order to define likelihood functions, the distributional properties of the finite Fourier transformed complete data \( \mathbf{Y}_k(\omega) \) are required. Let us state the well known asymptotic properties of \( \mathbf{Y}_k(\omega) \) if the window length \( T \) is large and \( 0 < \omega_1 < \ldots < \omega_J < \pi \):

i) \( \mathbf{Y}_k(\omega_1), \ldots, \mathbf{Y}_k(\omega_J) \) are independent, complex normally distributed random vectors with zero mean and block diagonal covariance matrices \( C_{\mathbf{Y}_k}(\omega) = \text{diag}(C_{\mathbf{Y}_1}(\omega), \ldots, C_{\mathbf{Y}_{M'+1}}(\omega)) \) (\( j = 1, \ldots, J \)), respectively.

ii) \( \mathbf{Y}_k(\omega), \ldots, \mathbf{Y}_k(\omega) \) are independent and identically complex normally distributed random vectors with zero mean and block diagonal covariance matrix \( C_{\mathbf{Y}_k}(\omega) = \text{diag} \left( C_{\mathbf{Y}_1}(\omega), \ldots, C_{\mathbf{Y}_{M'+1}}(\omega) \right) \).

In the sequel, we shall use the nonparametric spectral density estimates

\[
\hat{C}_{\mathbf{Y}_k}(\omega) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{Y}_k(\omega)\mathbf{Y}_k(\omega)^*,
\]

\[
\hat{C}_{\mathbf{Y}_k}(\omega) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{Y}_k(\omega)\mathbf{Y}_k(\omega)^*,
\]

and

\[
\hat{C}_{\mathbf{X}_k}(\omega) = \frac{1}{K} \sum_{k=1}^{K} \mathbf{X}_k(\omega)\mathbf{X}_k(\omega)^*,
\]

for notational convenience.

3. EM-Algorithm

In contrast to the application of the EM algorithm in [1], we exploit that the density function of the complete data belongs to the (curved) exponential family, cf. [5] and [2]. Hence, we can write

\[
f_{\mathbf{Y}(\mathbf{y}; \theta)} = a(\theta)^{-1} \exp \{ a(\theta)^{-1} t(\mathbf{y}) \},
\]

with respect to a \( \sigma \)-finite measure \( \mu_Y \). After some calculations, when the notation

\[
\alpha_X(\alpha(\theta)) = \int \exp \{ a(\theta)^{-1} t(\mathbf{y}) \} d\mu_Y(\mathbf{y})
\]

is introduced, the conditional density function

\[
f_{\mathbf{Y}(\mathbf{y}; \theta)} = a(\theta)^{-1} \exp \{ a(\theta)^{-1} t(\mathbf{y}) \},
\]

and the density function

\[
f_{\mathbf{X}(\mathbf{x}; \theta)} = \frac{a_X(\alpha(\theta))}{a(\theta)}
\]

can be deduced, with respect to the measures

\[
\mu_X(\mathbf{x}) = \begin{cases} \mu_Y & : \mathbf{X} = \mathbf{g}(\mathbf{Y}) \\ 0 & : \text{else} \end{cases}
\]

and \( \mu_X \), respectively. The log-likelihood function of the incomplete data can be represented by

\[
L_X(\theta) = \log f_{\mathbf{X}(\mathbf{x}; \theta)} = \log \frac{a_X(\alpha(\theta))}{a(\theta)}.
\]
Using some results well known for the exponential family and $\mathcal{T} = \ell(Y)$, the necessary conditions for an optimum $\tilde{q}$ provide after some algebra
\[
\begin{bmatrix}
\frac{\partial \alpha(\tilde{q})}{\partial \tilde{q}}^T (E_{\tilde{q} \tilde{q}}(\mathcal{T}) - E_{\tilde{q} \tilde{q}}(\mathcal{T}X)) = 0.
\end{bmatrix}
\] (7)

This nonlinear equation system motivates the iterations of the EM algorithm that can be described as follows.

E-step: calculating $\tilde{q}^* = E_{\tilde{q} \tilde{q}}(\mathcal{T}X)$,

M-step: solving $\frac{\partial \alpha(\tilde{q})}{\partial \tilde{q}}^T (E_{\tilde{q} \tilde{q}}(\mathcal{T}) - \tilde{q}^*) = 0 \Rightarrow \tilde{q}^{n+1}$.

Because this iteration scheme is a special case of that developed in [2], the same convergence properties as indicated in [6] and [7] can be claimed.

4. Approximate MLE

Let us assume that the sensor outputs can be sufficiently well described in the frequency domain by a number $J$ of narrow bands or by a spectrum, varying slowly with frequency $\omega$, such that it can be reconstructed satisfactorily by a finite number $J$ of interpolation points. These assumptions in conjunction with the asymptotic distributional properties i) and ii) suggest the approximate likelihood functions for the complete data

\[
f_{Y}(y, \theta) = \prod_{j=1}^{J} \frac{1}{\prod_{j=1}^{J} \det(C_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j))^{-1} \cdot 
\exp(- \text{tr}[C_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j))^{-1} \mathcal{C}_{\mathcal{L}}(\omega_j)])}
\]

where the $\mathcal{L}(\omega_j)$ ($j = 1, \ldots, J$) are appropriately collected in $\mathcal{L}$. Exploiting the block diagonal structure of $C_{\mathcal{L}}(\omega)$, we obtain

\[
f_{Y}(y, \theta) = \prod_{j=1}^{J} \prod_{j=1}^{M+1} \frac{1}{\det(C_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j))^{-1})} \cdot 
\exp(- \text{tr}[C_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j))^{-1} \mathcal{C}_{\mathcal{L}}(\omega_j)])
\]

where $\mathcal{L}(\omega_j) = (\mathcal{L}(\omega_j), \mathcal{L}(\omega_j))^T$ ($i = 1, \ldots, M'$) and $\mathcal{L}(\omega_j, \mathcal{L}(\omega_j)) = \mathcal{L}(\omega_j, \mathcal{L}(\omega_j))$ ($i = 1, \ldots, I$), Comparing (5) with (8), the statistic $\mathcal{T}$ and the vector-valued function $\alpha(\tilde{q})$ can be identified by

\[
\mathcal{T} = \begin{pmatrix}
\text{vec}(\mathcal{C}_{\mathcal{L}}(\omega_1))^T, \\
\vdots \\
\text{vec}(\mathcal{C}_{\mathcal{L}}(\omega_{M+1}))^T
\end{pmatrix}^T,
\]

and

\[
\alpha(\tilde{q}) = -(\text{vec}(\mathcal{C}_{\mathcal{L}}(\omega_1, \mathcal{L}(\omega_1)))^T, \\
\vdots \\
\text{vec}(\mathcal{C}_{\mathcal{L}}(\omega_{M+1}, \mathcal{L}(\omega_{M+1})))^T)
\]

After rearranging the nonlinear equation system (7), where the nonlinear equation system can be decoupled in $(M' + I)$ systems of lower dimensions, and with

\[
C_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j)) = \frac{1}{\sigma(\omega_j)} 
\left[
\begin{bmatrix}
\mathcal{I} - \frac{\sigma(\omega_j)}{N\sigma(\omega_j) + \epsilon(\omega_j)} \mathcal{L}(\omega_j) \mathcal{L}(\omega_j)^T
\end{bmatrix}
\right]
\]

the EM algorithm provides the following iteration scheme

E-step: for $i = 1, \ldots, M'$, $\mathcal{T} + I$, cf. [1]

\[
\mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j)) = E_{\mathcal{L} \mathcal{L}}(\mathcal{C}_{\mathcal{L}}(\omega_j)|X(\omega_j)) = 
\mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j)) - 
\mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j)) \mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j))^{-1} \mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j)) + 
\mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j)) \mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j))^{-1}
\]

\[
\mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j))^{-1} \mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j))
\]

M-step: for $i = 1, \ldots, M$ and $k = 1, \ldots, I$, respectively

\[
\sigma_{i+1}(\omega_j) = 
\frac{1}{N} \left(\frac{1}{N} \mathcal{I}_{\mathcal{L}}(\omega_j)^{-1} \mathcal{C}_{\mathcal{L}, \mathcal{L}, \mathcal{L}}(\omega_j, \mathcal{L}(\omega_j), \mathcal{L}(\omega_j))^T \mathcal{L}(\omega_j) - \mathcal{L}(\omega_j)\right)
\]

\[
\mathcal{L}(\omega_j)^{-1} \mathcal{C}_{\mathcal{L}, \mathcal{L}, \mathcal{L}}(\omega_j, \mathcal{L}(\omega_j), \mathcal{L}(\omega_j))^T \mathcal{L}(\omega_j) - \mathcal{L}(\omega_j)
\]

and $\mathcal{L}(\omega_j)$ is obtained by solving the nonlinear equation system

\[
\sum_{j=1}^{J} \frac{\sigma_{i+1}(\omega_j)}{N\sigma(\omega_j) + \epsilon(\omega_j)} \mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j)) \mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j))^{-1} \mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j)) - \mathcal{L}(\omega_j)
\]

Observing that (9) is the necessary condition of the weighted classical beamformer criterion obtained in [1], this approach and that in [1] seem to be equivalent. Nevertheless, there are two major distinctions.

In [1], the log-likelihood function could only be simplified to the weighted classical beamformer criterion if the spectral parameters would be kept fixed and became more complicated if the spectral parameters would be replaced by their explicit solutions. The nonlinear equation system (9) could be derived without such considerations. Moreover, for $J = 1$ or the $\epsilon(\omega_j)$ appropriately chosen such that

\[
\mathcal{L}(\omega_j)^{-1} \mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j)) \mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j))^{-1} \mathcal{C}_{\mathcal{L}}(\omega_j, \mathcal{L}(\omega_j)) - \mathcal{L}(\omega_j)
\]

the equations of the signal spectral parameters and the wave parameters can be decoupled completely. Herein, the sensor noise parameter and the ambient noise parameters $\nu_0(\omega_j)$ used in [1], we have introduced the known auxiliary parameters $\epsilon(\omega_j)$ to guarantee the positive definiteness of $C_{\mathcal{L}}(\omega_j)$. 
5. Information Loss and Increase in Variance

The log-likelihood function for the complete data is given by

$$L_Y(\theta) = L_Y(X) + L_X(\theta),$$  \hspace{1cm} (10)

where $L_Y(X)$ denotes the logarithm of (6). Taking the derivative of (10) with respect to $\theta_p$ ($p = 1, \ldots, P$) and then the conditional expectation given $X = x$, we obtain

$$E(T_{X,p}|X) = T_{X,p},$$ \hspace{1cm} (11)

where $E(T_{XY,p}|X) = 0$ is exploited and $T_{X,p}$, $T_{X,p}$ and $T_{XY,p}$ denote $\partial L_Y(\theta)/\partial \theta_p$, $\partial L_X(\theta)/\partial \theta_p$, and $\partial L_Y(X)/\partial \theta_p$, respectively. Using the well known identity

$$Cov(T_{X,p}, T_{X,q}) = Cov(E(T_{X,p}|X), E(T_{X,q}|X)) + E(Cov(T_{X,p}, T_{X,q}|X)),$$

together with (11), we can derive

$$E(T_{X,p}, T_{X,q}) = E(T_{X,p}T_{X,q}) + E(Cov(T_{X,p}, T_{X,q}|X)).$$

Interpretation in terms of information matrices yields

$$I_Y(\theta) = I_X(\theta) + I_{XY}(\theta),$$

where the last quantity on the right denotes the so-called lost information. Applying the equation

$$I_Y(\theta)(I_X(\theta)^{-1} - I_Y(\theta)^{-1}) = I_{XY}(\theta)I_X(\theta)^{-1},$$

the increase in variance and the information loss can be expressed by

$$I_X(\theta)^{-1} - I_Y(\theta)^{-1} = I_Y(\theta)^{-1}I_{XY}(\theta)I_X(\theta)^{-1}$$

and

$$I_{XY}(\theta) = I_Y(\theta)(I_X(\theta)^{-1} - I_Y(\theta)^{-1})I_Y(\theta),$$

respectively.

Therefore, in order to get more accurate estimates techniques recovering missing data should be employed. One possibility might be spatial filtering as e.g. spherical beamforming, cf. [4]. Another way seems to consist in the following two steps procedure

i) Computing the AMLEs via EM algorithm as described in section 4.

ii) The (initial) estimates obtained from i) are used to recover some missing data by removing certained signals from the data, alternatingly. Then, with less missing data the EM algorithm is applied again.

6. Concluding Remarks

We have introduced an extended model of the spectral density matrix of the sensor array output for sources in a multipath propagating environment. In comparison with [1], we have derived a modified EM iteration scheme and AMLEs. Furthermore, we have investigated the loss of information and the increase in variance due to the missing data, and mentioned two alternatives how it can be reduced.

Numerical experiments with the real data applying the two steps procedure indicated at the end of the last section are under work. The links between EM algorithm, the Kullback-Leibler divergence and the maximum entropy methods are currently investigated.

REFERENCES


An Improved Algorithm for the Spatial Analysis of Correlated Sources

Dominic Grenier, Gilles Y. Delisle
Electrical Engineering Department
Laval University
Quebec (QO), Canada G1K 7P4
tel: (418) 656-2869 (418) 656-2851
e-mail: dgrenier@gel.ulaval.ca gdelisle@gel.ulaval.ca

Abstract

A high resolving power algorithm adapted to narrow band correlated sources detection without the use of a scanning analysis vector in the area of interest is presented. Source correlation and computer time requirements are the major obstacles to a more important use of superresollute methods. The proposed new algorithm used the translation and rotation invariances of the covariance matrix eigenvectors in both directions (forward and reverse) to generate the estimated sources subspace. The source subspace is therefore created again as it is the case for source vectors such as in DEESE. However, the required computations are limited since the vectorial multiplications associated with the scanning using parallel network techniques are cancelled and therefore allows a simple way of finding and saving the positions of the transmitting sources.

1 Introduction

Spatial analysis using second order signal processing techniques permits to achieve a better resolving power for an array of sensors. Even if this characteristic is desirable, the use of this technique is still to be generalized, even if second order signal processing is now more frequently used in specialized applications. This is due to some difficulties associated with the algorithms hypothesis and additional considerations such as:

- second order characteristics that can be used on the received data only for stationary or quasi-stationary system.
- zero or close to zero correlation between sources that does not apply to active systems or those with interferences.
- mathematical operations on complex matrices that generates large computing time.

Over the past few years, many research works have tried to avoid these difficulties. Methods using array or source properties have recently been used in particular situations. The majority of these methods addresses the case of correlated sources using spatial or frequency diversity since it is the most frequent cases [1-7]. Some others methods are however looking at a greater tolerance with respect to the low number of samples [8-10] or simplify the computations or the sensitivity of the material [11-13] using the geometry.

Following this trend, the algorithm presented here combined in an advantageous way all the different manners to use the geometry of a linear array of identical equally spaced elements as well as the invariance in rotation and in translation. This algorithm resolved not only correlated sources but it also avoids the multiple vectorial products required for scanning with an analysis vector and obtains directly the positions of the sources with therefore simplifies the data storage problem.

The rank increase of the source matrix comes from the modified space diversity (translation invariance) to which the partition of the array in subarrays in both directions like in DEESE [6] is added. A change in the partitioning direction is equivalent to a phase reserval of 180° about the center, i.e. the array's sensor numbering is interchanged and the signal's angles of arrival as well. The removal of scanning, follows algorithms such as ESPRIT [12] and TAM [11], namely the parallel way between two array or subarrays do not modify sources and noise subspaces in an independent sources environment (no interference figures).

2 Sources Subspace

An array of "N" identical, aligned and equally spaced (by a distance "d") elements is considered upon which "m" plane wavefronts (n < N) issued from narrow band sources in the directions [θ_1, θ_2, ..., θ_m] as in figure (1).

![Figure (1): Array geometry.](image-url)
The noise free received signals vector, at the central frequency, can be expressed as

$$s = U \alpha$$  \hspace{1cm} (1)

where $U$ is the sources matrix ($N \times n$) formed by the juxtaposition of the sources vectors $u_i$

$$u_i = \frac{1}{\sqrt{N}} [1, e^{i\phi_1}, \ldots, e^{i(N-1)\phi_i}]^T$$  \hspace{1cm} (2)

and $\alpha$, the sources complex envelopes vector $\alpha_i$. The covariance matrix of the noiseless received signals $R_s$ ($N \times N$) writes as

$$R_s = E\{ss^H\} = U S U^T$$  \hspace{1cm} (4)

$$S = E\{\alpha \alpha^H\}$$  \hspace{1cm} (5)

when $S$ is the sources ($n \times n$) covariance matrix. It can then be written as

$$R_s = V A V^T$$  \hspace{1cm} (6)

$$V = [v_1, v_2, \ldots, v_N]$$

$$A = \text{diag} \{\lambda_1, \lambda_2, \ldots, \lambda_p, \lambda_{p+1}, \ldots, \lambda_N\}$$  \hspace{1cm} (7)

where the $\lambda_i$ are the eigenvalues ranked in decreasing order of $R_s$ and $v_i$ are the associated eigenvectors.

If an additive noise is added to $s$, yielding $x$, a noise only covariance matrix $R_n$ is added to $R_s$, making all eigenvalues of $R$ non-zero such as

$$R = R_s + R_n.$$  \hspace{1cm} (9)

The problem created by the correlation between sources comes from the progressive ill-conditioning of the sources covariance matrix $S$ and, consequently, of the covariance matrix $R_s$. Therefore, the image of $R_s$ becomes a subspace with dimensions equal to the number of groups $"g"$ and is not anymore equal to the number of sources $"n"$. This subspace, noted the estimated source subspace is located within the real sources subspace, i.e. the subspace created by the source vectors. Then, the eigenvectors associated with the non-zero eigenvalues of $R_s$ constitute now an orthonormal base of the estimated sources subspace. The source-vectors are not located inside this smaller subspace i.e. the subspace generated by the eigenvectors associated with the non-zero eigenvalues of $R_s$ (these two subspaces are identical in the uncorrelated case). The source subspace must therefore be regenerated.

The spatial smoothing [1] is a known technique to increase the rank of $R_s$ up to "$n" in a coherent sources environment. Smoothing is a preprocessing issued from the split of the main array into subarrays and it uses translation invariance. As presented however, this solution does not sufficiently decorrelate the closely spaced sources and consequently maintaining the superresolution property, this neglecting that the effective number of elements is decreasing. The DEESE [8, 14] method offers a better alternative to the use of translation invariance by decomposing in $"n"$

subvectors of "$m"$ elements ($r = N - m + 1$) each vector of the estimated source subspace:

$$v(r) = [v_{1,r}, v_{2,r+1}, \ldots, v_{k,r+m-1}]^T$$  \hspace{1cm} (10)

The property of rotation invariance is also used to decompose the vectors in the other direction, i.e. by inverting the elements. Those new subvectors must however be conjugated since only the reference is reversed, not the source's position:

$$w(r) = [v_{1,N-r-k}, v_{2,N-r-k+1}, \ldots, v_{N-r-k+m-1}]^T$$  \hspace{1cm} (11)

It can be shown that the $v(r)$ et $w(r)$ ($1 \leq i \leq g, 1 \leq k \leq r$) subvectors generate an incoherent subspace which include all the source-vectors if

- $m > n$;
- $2r \geq p$

where "$p$" is the number of sources in the larger group. The coherent subspace converges toward the real sources subspace even under additive noise and can therefore be used as a good estimation of this subspace. However, since the eigenvectors do not constitute an orthonormal base, the euclidian distance between the analysis vector and the incoherent subspace cannot be considered as a measure of the projection norm of this vector on the base. The solution envisaged makes use of eigenvectors $a_j$ ($1 \leq j \leq n$) orthonormal to the noiseless image of $Q_s$ matrix, constituting the incoherent subspace,

$$Q_s(r) = V(r)W(r)W^T(r)V^T(r);$$  \hspace{1cm} (12)

with

$$V(r) = [v_{1,(r)}, v_{1,(r+1)}, \ldots, v_{r,(r)}, \ldots, v_{N,(r)}]$$  \hspace{1cm} (13)

$$W(r) = [w_{1,(r)}, w_{1,(r+1)}, \ldots, w_{r,(r)}, \ldots, w_{N,(r)}]$$  \hspace{1cm} (14)

The advantages associated with the operation are

- uses of a full covariance matrix ($N \times N$)
- optimal use of the array geometry to increase the number of effective elements
- noise reduction by the exclusive selection of the eigenvectors associated, with a greater confidence, to the sources prior to the decomposition.

3 Rotation Matrix

The non-scanning techniques are using two or more parallel arrays [6, 11, 12] to obtain the rotation matrix $A$ ($N \times N$). For a linear equispaces arrays, the translation invariance matrix is obtained in a similar way (also called rotation invariance by some authors because of the structure of the $A$ matrix).

A structure of the $U$ matrix and the operator $\dagger$ considered in such a way that it produces a shift of one line up, namely
\[ U = \left[ r^1, r^1 \Delta, \ldots, r^1 \Delta^{N-1} \right] \]  
(16)

with
\[ r = \frac{1}{\sqrt{N}} \begin{bmatrix} 1, 1, \ldots, 1 \end{bmatrix} \]  
(16)
\[ \Delta = \text{diag} \left\{ e^{j\phi_1}, e^{j\phi_2}, \ldots, e^{j\phi_N} \right\} \]  
(17)

which yields:
\[ U^\dagger = U \Delta. \]  
(18)

The \( a_j \) vectors, similarly to the source vectors \( u_i \), form a base for the source space and a linear dependence exists between these vectors expressed by the regular matrix \( T \):
\[ UT = A \quad \text{and} \quad U^\dagger T = A^\dagger, \]  
(19)

where \( A \) (\( m \times n \)) is formed by the vectors \( a_j \) placed side-by-side. Using the pseudo-inverse of \( A \), noted \( A^{-1} \), it is possible to find back the diagonal elements of \( \Delta \) and the directives \( \theta_i \):
\[ A^\dagger T^{-1} = A T^{-1} \Delta \]  
(20)
\[ A^{-1} = \left( (A^\dagger A)^{-1} A^\dagger A \right)^{-1} \]  
(21)

The eigenvalues of \( A^{-1} \), \( \lambda_i^\dagger \), are the \( e^{j\phi_i} \)-type elements looked for.

The closer to unity that the modules at the eigenvectors of \( A^{-1} \) are (the eigenvalues are complex since the matrix is not of hermitian symmetry), the stronger will be the hypothesis that the eigenvalue associated to a source and his argument will correspond to the electrical phase shift that it produces between adjacent elements.

4 Simulation Results

The results presented here were obtained on a SUN-SPARC station using Monte-Carlo simulations. For each trial, 200 snapshots were obtained on a 8 elements \( d = \lambda/2 \) array \((N = 8)\), using phase quadrature synchronous demodulation. The theoretical Rayleigh limit is 15° \((20° \text{ in practice})\). The results obtained are plotted in polar coordinates plane and give the position of the eigenvalues of the \( A^{-1} \) in the complex plane.

The first simulation is using 2 sources, one with a signal-to-noise ratio of 6dB located at 5° and the second one of 2dB at -2°. The correlation factor between the two sources as compared to the reference element is 1/53°. Figures (2) and (3) shows the results obtained after 100 trials when the number of sources was overestimated to 3 \((\hat{n} = 3)\) with the first version of DEESE and with the improved algorithm presented in this paper. It is clearly recognize that two clusters of points along the unit circle identifies the two sources (superresolution factor of 3) and the points spreaded in all direction within this circle correspond to the non-existing third source. It is also noted that the more concentrated cluster of points corresponds to the stronger source, showing a variance smaller the position estimation error, the results computed for each source are \( \sigma_{\hat{\phi}_3}^2 = 0.17 \), \( \mu_{\hat{\phi}_2} = 5.06 \) and \( \sigma_{\hat{\phi}_2}^2 = 0.35 \), \( \mu_{\hat{\phi}_2} = -2.04 \).

Finally, the same system has been simulated with an

Figure (2): Spectra given by DEESE for simulations \((10 \text{ trials})\) \( p = 2, q = 1 \).

Figure (3): Pole positions by new algorithm for exactly the same simulations as (2).
independant source of 2dB located at -11°. This increase the number of group to two (g = 2) preserving however the size of the highest group to two (p = 2). Here, the correlation factor between the first and second sources at the reference element is taken equal to 1/22° while this one between the first and third sources equal 14-21° in figure (4) or 1431° in figure (5). It is noted that the separation of each source is less evident because of the interference created by this supplementary source on the others and also that the quality of separation depends of the phase relation between sources.

5 Conclusion

This paper reports initial results obtained up to now for correlated sources signal processing. The simulation and experimental results obtained so far show, however, the applicability and the efficiency of the method for strongly correlated sources. The absence of scanning decreases the required computer time and makes easier to store the position estimates since it is done using only the eigenvalues \( \lambda_i \) and the resultant the \( \theta_i \) angles instead of a family of curves. Otherwise, the possible functions of the sources would have to be search for by an algorithm.

The proposed approach is promising; the problem still to be solved however is the interference between sources of a same group or from one group to the other, a problem that increases with the number of sources. Many solutions are envisaged for implementation; one of these would be the use of frequency diversity. It must be taken into account however that the computation time is already important and such a scheme should not increased if significantly.

References


A Complex Adaptive Eigensubspace Algorithm for DOE or Frequency Estimation and Tracking

J.P. DELMAS

Département Electronique et Communications, Institut National des Télécommunications,
9 rue Charles Fourier, 91011 Evry cédex, France

Abstract

Many authors have proposed adaptive estimation methods of complete noise (or signal) subspace and have suggested different implementation methods. Among them, Regalia [R90] presented an adaptive unit norm filter based on planar rotors which offers interesting properties. We first reformulate this idea and extend it to complex processes, and then prove the convergence of the stochastic coupled algorithms with the help of the associated Ordinary Differential Equation.

1) Introduction

Signal-subspace techniques such as MUSIC [S79], minimum norm [KT83], root Music [B83] and reduced order MUSIC [O86] have become classical superresolution methods for the estimation of frequencies of sinusoids or the directions of arrival of plane waves by an array of sensors. However, implementations of these techniques have been mainly based on batch estimation of eigenvectors derived from the estimated covariance matrix, making them unsuitable for adaptive processing that is needed when tracking nonstationary signal parameters.

Owesley [O78] was the first to introduce an adaptive procedure to estimate the eigenvectors. Thompson [T80] proposed a gradient based adaptive algorithm for the estimation of the eigenvector corresponding to the minimum eigenvalue. Reddy et al. [REK82] developed a least squares algorithm for the same problem. Yang and Kaveh [TK88] studied several gradient based search algorithms and Sarkar and Yang [SY89] proposed a conjugate gradient algorithm for interactively finding the eigenvectors corresponding to complete noise or signal subspace estimation.

Recently, an adaptive unit norm filter was proposed by P. Regalia [R90]. It performs the Karhunen-Loève transformation of the input real signal by an orthogonal-based triangular array.

This paper is organized as follows. After introducing some notations and formulating the problem in Section 2, our solution, based on global parametrization of orthonormal eigenvectors of correlation matrix of complex signal, is described in Section 3. Coupled minimizations are solved by coupled stochastic gradient algorithms in Section 4 and finally the convergence of these coupled stochastic algorithms are proved using the ODE method in Section 5.

2) Notation and formulation of the problem

Let us introduce some notations: for an $N$ element array and $P$ incoherent sources with $P < N$, the algorithm studied adaptively estimates the eigenvectors corresponding to either the $N-P$ smallest or the $P$ largest eigenvalues of a continuously updated sample covariance matrix of order $N$:

$$R_x = E[x(k)x^T(k)]$$

with $x(k) = x(1,k),...,x(N,k)$.

Let $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_P \leq \lambda_{P+1} \leq \cdots \leq \lambda_N$ and corresponding normalized orthogonal eigenvectors. ($*,t,+,$ stand respectively for conjugate, transpose and conjugate transpose).

We shall describe the estimation of the noise subspace, since the estimation of the signal subspace problem can be formulated similarly (e.g., replace the minimizations [1] by maximizations, $L$ defined below by $L = P$ and the sign − by + in the gradient algorithms).
$h_1$ is the solution of the minimization problem

$$\min_{||h||=1} h^* R_x h$$

and each $h_i$ for $i=2,..,L=N-P$ is the solution of the minimization problem

$$\min_{||h||=1, h \perp h_1,..,h_{i-1}} h^* R_x h$$

We note that the solutions of these minimizations are defined up to a multiplicative unit modulus complex constant.

3) Global parametrization of $h_1,..,h_L$

The orthonormal conditions on $h_1,..,h_L$ imply $L^2$ real constants. Therefore these vectors must be parametrized by a set of $2NL-L^2$ real parameters.

One easily proves that the vectors $h_1,..,h_1,..,h_L$ defined up to a multiplicative unit modulus complex constant, may be parametrized by a unique set of $\frac{1}{2}([2NL-L^2-L])$ couples of rotation angles ($\theta_{ij},\phi_{ij}$) $i=1,..,L$ and $j=1,..,N-i$ satisfying $\phi_{ij} \in [\frac{\pi}{2}, \frac{3\pi}{2}]$ and $\theta_{ij} \in [\frac{\pi}{2}, \frac{\pi}{2}]$ as follows. (The factor $L$ comes from an extra condition on each of the $L$ vectors that fixes the multiplicative unit modulus complex constant).

$h_1$ is the last column of a unitary matrix $Q_1$:

$$h_1 = Q_1 \left[ \begin{array}{c} 0 \\ 1 \end{array} \right]$$

and $h_2 = Q_2 \left[ \begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right]$.

$$.., h_L = Q_L \left[ \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right]$$

with each $Q_i$ a unitary matrix of order $N-i+1$:

$$Q_i = U_{ij} \cdot U_{ij}^*$$

with $U_{ij} = \left[ \begin{array}{cccc} 1_{N-1} & 0 & 0 & 0 \\ 0 & \cos \theta_{ij} & \sin \theta_{ij} & 0 \\ e^{i\phi_{ij}} \cos \theta_{ij} & e^{i\phi_{ij}} \sin \theta_{ij} & 0 \\ 0 & 0 & 0 & I_{N-1} \end{array} \right]$.

Proof: Suppose that $h_{1,1}$ is real nonnegative. Since $h^*_1$ is the bottom row $Q_1^*$ (with $Q_1$ defined by [3]), the equation:

$$h_1 = [h_1,..,h_{1,N}] = Q_1 \left[ \begin{array}{c} 1 \\ 0 \end{array} \right]$$

is equivalent to:

$$h_{1,1} = \prod_{i=1}^{N-1} \cos \theta_{1,i}$$

$$h_{1,k} = \prod_{i=1}^{k-1} \cos \theta_{1,i} \sin \theta_{1,k-1} e^{i\phi_{1,k-1}}$$

for $k=2,..,N-1$

$$h_{1,N} = \sin \theta_{1,N-1} e^{i\phi_{1,N-1}}$$

which in turn is equivalent to:

$$\frac{h_{1,2}}{h_{1,1}} = \sin \theta_{1,1} e^{i\phi_{1,1}}$$

$$\frac{h_{1,k+1}}{h_{1,k}} = \sin \theta_{1,k} e^{i\phi_{1,k}}$$

for $k=2,..,N-1$

$$h_{1,1} = \prod_{i=1}^{N-1} \cos \theta_{1,i}$$

Step by step, starting from $i=1$, the couples ($\theta_{1,i},\phi_{1,i}) \in [\frac{\pi}{2}, \frac{\pi}{2}]$ are uniquely determined and the relation $h_{1,1} = \prod_{i=1}^{N-1} \cos \theta_{1,i}$ is automatically satisfied from the $N-1$ preceding relations since $h_{1,1}$ is assumed real nonnegative and $||h_1|| = 1$.

Then $h_2 \Leftrightarrow h_1 = Q_1 \left[ \begin{array}{c} 0 \\ 1 \\ 0 \end{array} \right]$ with $h \in \mathbb{C}^{N-1}$ and $||h|| = 1$. Using the same argument as for $h_1$, we get:

$$h = Q_2 \left[ \begin{array}{c} 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{array} \right].$$

With similar arguments we prove iteratively the existence and uniqueness of [3] if no component of $h_1$ nor of $h$ is zero. We note that if a component of $h_1$ or of $h$ is zero, existence is proved but uniqueness is no longer guaranteed.

4) Coupled minimizations

The minimization [1a] can be performed with the help of a stochastic gradient algorithm, in which the parameter is:

$$p = \left[ \theta_{1,1}, \phi_{1,1},..,\theta_{1,N-1}, \phi_{1,N-1}, \theta_{2,1},..,\theta_{L,1}, \phi_{L,1} \right]^T.$$

With $y_{ij}(k) = h_1^* x(k) = [0^T, 1] Q \left[ \begin{array}{c} 0^T \\ x(k) \end{array} \right] + \zeta_0(k)$, we must minimize $\mathbb{E}[V_1(p, x(k))]$ for $p$. (Here we denote $V_1(p, x(k)) = \frac{1}{2} h_1^* x(k)^2 = h^* x^2(k) x(k)^2 h$).

Since the minimization [1b] in $h$ contains parameters $h_1,..,h_{L-1}$, it can also be performed according to a stochastic gradient algorithm in which the parameters $p_n = [\theta_{n,1}, \phi_{n,1},..,\theta_{n,N-1}, \phi_{n,N-1}]^T$, for $n=1,..,i-1$, are injected from the $i-1$ previous algorithms.
With \( y_i(k) = h_k^i x(k) = [\psi^i_1(k), \psi^i_2(k), \ldots, \psi^i_{2N-L-1}(k)] \), the gradient of the energy function is given by:

\[
\frac{\partial E}{\partial y_i(k)} = \frac{1}{2} \sum_{l=1}^{2N-L-1} \left( \frac{\partial y_i(k)}{\partial x_l} \right)^2 = h_k^i \left( \sum_{l=1}^{2N-L-1} \left| \psi^i_l(k) \right|^2 \right)
\]

Thus, we obtain \( L \) coupled stochastic gradient algorithms, derived from the stochastic approximation algorithms associated with the \( L \) minimizations of \([1]\).

Now we can show, using the following figure, that for \( i = 1, \ldots, L \), the gradient of \( y_i(k) \) with respect to the parameters \( p_i \) is given by:

\[
\frac{\partial y_i(k)}{\partial p_i} = y_i(k) \prod_{l=1}^{N-L-1} \cos \theta_{i,l}
\]

\[
\frac{\partial y_i(k)}{\partial \phi_{i,n-i}} = \left( \prod_{l=1}^{N-L-1} \cos \theta_{i,l} \right) \sin \theta_{i,n-i} \prod_{l=n-i}^{N-L-1} \cos \theta_{i,l}
\]

\[
\frac{\partial y_i(k)}{\partial \psi_{i,j}^l} = y_i(k) \prod_{l=1}^{N-L-1} \cos \theta_{i,l} \sin \theta_{i,n-i} \prod_{l=n-i}^{N-L-1} \cos \theta_{i,l} \prod_{l=1}^{j-1} \sin \theta_{i,l} \prod_{l=j+1}^{N-L-1} \cos \theta_{i,l} \sin \theta_{i,n-i} \prod_{l=n-i}^{N-L-1} \cos \theta_{i,l}
\]

The \( (2N-L-1) \) rotation angles are updated according to:

\[
\begin{align*}
\theta_{i}(k) &= \theta_{i}(k-1) - \gamma \Re \left( \frac{\partial y_i(k)}{\partial \theta_{i}} \right) y_i(k) \\
\phi_{i,n-i}(k) &= \phi_{i,n-i}(k-1) - \gamma \Re \left( \frac{\partial y_i(k)}{\partial \phi_{i,n-i}} \right) y_i(k)
\end{align*}
\]

This leads to a truncated triangular array comprising \( \frac{1}{2}(2N-L-1) \) rotator sections, as in [R90], but using now complex rotation cells.

For example, in case \( N=5 \) and \( L=2 \), we obtain the following flowgraph:

5) Convergence of the stochastic coupled algorithms with the help of the Ordinary Differential Equation.

We note that in the algorithms \([4]\), \( p_n(k) = [\theta_{n,1}(k), \phi_{n,1}(k), \ldots, \theta_{n,N-n}(k), \phi_{n,N-n}(k)] \), for \( n=1, \ldots, L \), are injected from the \( n-1 \) previous algorithms.

The problem of the convergence of the \( L \) coupled stochastic gradient algorithms described previously is stated. In the stationary context, it can be studied by the ODE method [LS83].

The stochastic gradient algorithms can be globally written as:

\[
\begin{bmatrix}
P_1 \\
P_L
\end{bmatrix}
= 
\begin{bmatrix}
P_1 \\
P_L
\end{bmatrix}
- \gamma_k
\begin{bmatrix}
f_1(p_1(k), x(k)) \\
f_L(p_L(k), x(k))
\end{bmatrix}
\]

with \( f_j(p_1(k), \ldots, p_L(k), x(k)) \) the gradient of the energy function.

\[\text{grad}_{p_i} V_i(p_1(k), \ldots, p_L(k), x(k))\]
If the gain sequence $\gamma_k$ satisfies the conditions
$$\sum_{k=1}^{\infty} \gamma_k = +\infty \quad \text{and} \quad \lim_{k \to \infty} \gamma_k = 0 \quad (\text{e.g.,} \quad \gamma_k = \frac{1}{k})$$
the associate ODE is:
$$\frac{dp_1(t)}{dt} = -E[\nabla p_1 V_1(p_1(t),x(t))] \quad \text{(6a)}$$
$$\vdots$$
$$\frac{dp_L(t)}{dt} = -E[\nabla p_L V_L(p_1(t),\ldots,p_L(t),x(t))] \quad \text{(6b)}$$

Studying the convergence of $[3]$ is thus equivalent to studying the stability of the system $[6]$. We show that the parameters that minimize the expressions $[1]$ are globally asymptotically stable for the equations $[6]$.

**Proof:**

Thanks to the uniqueness of the parametrization of $h_1,\ldots,h_L$, the coupled estimation problem has a unique solution in $\{\theta_1,\ldots,\theta_L\}$ $i=1,\ldots,L$ and $j=1,\ldots,N-i$ satisfying $\theta_1(0) = \frac{-1}{2}, \frac{1}{2}$ and $\theta_{N-i}(0) = \frac{-1}{2}, \frac{1}{2}$. We shall show that this solution is globally asymptotically stable for the equations $[6]$. Then the convergence of the coupled stochastic algorithms is ensured.

We suppose that $L=2$, as the extension to $L>2$ is straightforward.

Since $\nabla p_1 V_1(p_1(t),x(t))$ is the derivative of a positive gradient field, the stationary point (denoted $p_1(\rightarrow)$) of $[6a]$ is globally asymptotically stable for that equation.

Similarly for $\nabla p_2 V_2(p_1(t),p_2(t))$, the stationary point $p_2(t)(\rightarrow)$ of $[6b]$ (in which $p_1(t)$ is replaced by $p_1(0)$) is globally asymptotically stable for that equation.

Since $E[\nabla p_2 V_2(p_2(t))]$ is an analytic function in $p_1(0)$ and $p_2$, the stationary point $p_2(t)(\rightarrow)$ is a continuous function of $p_1(0)$.

The solution $p_2(t)(\rightarrow)$ of the first differential equation
$$\frac{dp_2(t)}{dt} = -E[\nabla p_2 V_2(p_1(t),p_2(t))]$$
converges uniformly with respect to $p_1(0)$ to $p_2(t)(\rightarrow)$ when $t \to \infty$, provided we choose $T > T_1$ such that $\|p_1(t) - p_1(\rightarrow)\| \leq \epsilon$.

Then, since
$$\|p_2(t) - p_2(t)(\rightarrow)\| \leq \|p_2(t) - p_2(0)\| + \|p_2(0) - p_2(t)(\rightarrow)\|,$$
the equations $[6]$ have an asymptotically stable solution in which the stationary point is $p_1(t)(\rightarrow),p_2(t)(\rightarrow)$.

Consequently, the convergence of the 2 coupled stochastic algorithms is ensured.

**References**


EXPLOITING MULTIVARIATE AUTOREgressive MODEL IN THE SEPARATION OF SOURCES

Seppo Rantala and Risto Suoranta

Machine Automation Laboratory, Technical Research Centre of Finland
P.O. Box 192, SP-33101 Tampere, Finland, Tel +358-31-163616, Fax +358-31-163694,
Internet: Seppo.Rantala@tut.fi, Risto.Suoranta@tut.fi

In this paper we propose a method to separate independent sources from the signals obtained by using an array of sensors. The method is based on a novel interpretation of results derived from a multivariate autoregressive model, MAR-model. Exploiting the model expressions related to source separation are derived. Due to the fact that MAR-model is black-box model and it can describe systems with feedback-loops, it is very general and useful tool to analyse linear systems. Particularly its strength is in cases, where there is no exact a priori knowledge of internal structures of the system.

1. INTRODUCTION

The problem of extracting independent sources based on measurements obtained from an array of sensors is an essential problem in signal processing. The signal received by a sensor is the sum of elementary contributions that can be called as sources. Sources as well as their mixtures are generally unknown. In this case this task is called blind separation of sources [2]. For instance, antenna or array of acoustic sensors receives a superimposed signal from all the sources which are in its receptive field.

Numerous approaches have been developed to solve the problem of separating different sources. Most of the works related to this problem use the information included in the received signals together with other information. Therefore the usage of these methods is limited to special cases. For instance, one possibility is to make an assumption that one of the sources produces deterministic signal.

In this work we construct a model of the system under study. Using the model it is possible to obtain insight into dynamics and internal relations of the system. If the chosen model is appropriate we are able to reveal the sources by utilizing the model.

The proposed method exploits multivariate autoregressive model (MAR-model) as a system analysis tool. The MAR-model describes the closed loop black-box system with internal feedback loops. Consequently it is an appropriate model to describe complicated systems without a need to do strict assumptions about the system.

In this method we identify the MAR-model and based on the model the multichannel spectral matrix is decomposed. Spectral decomposition offers a new possibility to analyse signals measured from the array of sensors. Utilizing the spectral decomposition we can efficiently estimate which signals can be regarded as independent variables (sources) and which signals are linear combinations of the other signals. Also the sources and their propagation paths inside the observed system can be analyzed applying proposed approach.

2. MULTIVARIATE AR-MODEL

MAR-model is a model for multivariate stochastic process where relationships between internal processes are described. MAR-model can be expressed using matrix notation; that is,

\[ \sum_{i=0}^{p} a(i) X(k-i) = \epsilon(k), \]  

(1)

where \( X(k) \) is \( m \times 1 \) column vector containing \( m \) univariate stochastic processes included in the multivariate process, \( a(i), (i = 1 \ldots p) \) are \( m \times m \) MAR-coefficient matrices and \( \epsilon(k) \) is \( m \times 1 \) column vector containing \( m \) zero mean white noise processes \( e(k) \); \( p \) is the order of AR-model.

As we can see from the Eq. 1, the MAR-model describes closed loop system with linear feedback connections between all signals. The MAR-model has no predetermined inputs and outputs. The driving force of the system is a set of noise sources \( e(k) \) added to each node in the system. In Fig. 1 the structure of a bivariate AR-model is shown; \( N1 \) and \( N2 \) are internal noise sources as well as \( X1 \) and \( X2 \) are measured signals.

The existence of the straightforward recursive estimation algorithm for MAR-model parameters, presented by Wiggins and Robinson, makes the MAR-model method applicable and attractive to the analysis of multichannel data in practice. The fast algorithm can be found e.g. in [3, pp. 400-402].

Spectral matrix for multivariate stochastic process can be derived from the Eq. 1. The spectral matrix
for multivariate process \( \mathbf{X} \) is expressed as

\[
P_x(\omega) = (A(e^{j\omega})^{-1}) \Sigma_x (A(e^{j\omega})^*)^{-1}.
\]  

\( A(e^{j\omega}) \) is the Fourier transform of MAR-coefficient matrices \( a(i) \) and \( \Sigma_x \) is the covariance matrix of white noise processes \( \epsilon_i(k) \), \( i = 1, \ldots, m \).

3. NOISE SOURCE ANALYSIS

Noise processes \( \epsilon_i(k) \) are the only excitations of the modelled system. To understand the dynamics of the system it would be useful to know the properties of each noise source and how the effects of each noise source spread in the system. To analyze the effects of noise sources we present a concept of a noise conditioned spectral matrix (NCSM).

The noise conditioned spectral matrix (NCSM) is a spectral matrix, where the effects of selected noise sources are removed. The notation for the residual spectral matrix is \( P_{x|\epsilon_1, \epsilon_2}(\omega) \), where \( \epsilon_1 \) and \( \epsilon_2 \) are the noise sources to be excluded. An alternative notation for NCSM is \( P_{x|\epsilon_1, \epsilon_2}(\omega) \), where \( \epsilon_1 \) and \( \epsilon_2 \) are noise sources to be included in. In other words "\( \setminus \)" denotes exclusion of variables and accordingly "\( \cap \)" denotes inclusion of variables.

To remove the effects of the noise process \( \epsilon_i(k) \) we modify \( \Sigma_x \) by resetting those values in the covariance matrix to zero which correspond to the removed noise processes. This means that the removed noise process is regarded as zero and consequently the variance and corresponding cross-covariances will also be zero. The modification of the noise covariance matrix can be done by multiplying \( \Sigma_x \) with matrix \( M \), which is derived from the identity matrix \( I \) by zeroing those 1's in diagonal which correspond to the noise sources to be removed. Using modification matrix \( M \) we define noise conditioned spectral matrix, NCSM,

\[
P_{x|\epsilon}(\omega) = (A(e^{j\omega})^{-1}) M \Sigma_x M (A(e^{j\omega})^*)^{-1},
\]

where \( \epsilon \) is a list of excluded noise sources. If noise sources in \( \epsilon(k) \) are independent of each other we can write,

\[
P_x(\omega) = \sum_{i=1}^{m} P_{x\epsilon_i}(\omega).
\]

By exploiting the noise conditioned spectral matrix we can define a multiple coherence function, which is called noise conditioned multiple coherence, \( (NCMCH) \). The NCMCH is a measure of linear dependence between one variable and a set of selected noise sources. The NCMCH between variable \( X_j(k) \) and single noise source \( \epsilon_i(k) \) is defined as

\[
\gamma_{j\epsilon_i}^2(\omega) = \frac{P_{x\epsilon_i}(jj)}{P_{x}(jj)}.
\]

A subscript \((ij)\) denotes the element in row \( i \) and column \( j \) in corresponding matrix. If the noise sources in \( \epsilon(k) \) are independent of each other, then

\[
\sum_{i=1}^{m} \gamma_{j\epsilon_i}^2(\omega) = 1.
\]

The NCMCH describes the linear relationship between noise sources and variable in one frequency \( \omega \). In many cases it would be valuable to know what is the noise power contribution of the noise source to the variables of the system in the selected frequency band. By utilizing the concept of NCMCH we can build up a new measure that describes how much power in selected frequency band is originated from the noise source \( \epsilon_i(k) \). New noise contribution ratio (NCR) is defined as

\[
q_{ij}(\omega_1, \omega_2) = \frac{\int_{\omega_1}^{\omega_2} P_{x\epsilon_i}(jj)(\omega) d\omega}{\int_{\omega_1}^{\omega_2} P_{x}(jj)(\omega) d\omega},
\]

where \( \omega_1 \) is the lower bound and \( \omega_2 \) is the higher bound of the defined frequency band. By combining all values \( q_{ij}(\omega_1, \omega_2) \) to one \( m \times m \) matrix we obtain a system noise contribution matrix, \( (SNCM) \), that is

\[
Q(\omega_1, \omega_2) = \begin{bmatrix}
q_{11} & q_{12} & \cdots & q_{1m} \\
qu_{21} & q_{22} & \cdots & q_{2m} \\
\vdots & \ddots & \ddots & \vdots \\
qu_{1m} & \cdots & q_{mm}
\end{bmatrix}.
\]

The system noise contribution matrix describes the dynamics of the whole system. If the \( Q(0, 0.5) \) matrix is a identity matrix, it means that there are no cross-effects between variables, i.e., all signals are described only by their own noise sources. By studying the matrix \( Q \) we can resolve which signals are sources of the system and which are dependent signals.
4. NUMERICAL EXAMPLE

To illustrate possibilities of the new method we have simulated data obtained from the system depicted in Fig. 2. System includes three variables (sensors) and two sources.

![Diagram of simulated array of three sensors](image)

**Figure 2.** Simulated array of three sensors

There is two sources connected to the system (source1 and source2) and we have measured three signals $m_1, m_2$ and $m_3$. In Fig. 3 the spectra of sources are shown. The system has a marked correspondence to acoustic system, where transfer functions consists of time delays and dampings.

![Graph of spectra of sources](image)

**Figure 3.** Spectra of sources: source1 (solid line) and source2 (dashed line)

In this case 4000 samples were simulated from each channel of the system. The reason for the length was the comparison to the classical Fourier spectral analysis, which needs longer data records.

Due to the cross effect between sources measured signals are all very similar and near white noise. Classical multivariate Fourier spectral analysis [1] offers mainly coherence based estimates to the problem of separating the sources. In this work we have tried separate sources by using traditional ordinary, conditional(partial) and multiple coherence, but the sources were impossible to determine. As an example ordinary and conditional coherence function estimates are shown in Fig. 4.

![Graphs of coherence estimates](image)

**Figure 4.** Coherence estimates between $m_1 m_2$ and $m_3 m_1$

First step in applying MAR-model is to identify the model. The model order $p$ arouses some problems: Akaike information criterion(AIC) suggests order 29, but it must be too high, because the simulation system has maximum delay of 3 samples and used filters are low order. Too high model order may cause artifacts to the spectral quantities. If the AIC criterion and its difference are examined further (Fig. 5), the better model order can be chosen from the point, where the convergence of the difference is clear. Therefore the model order 10 was selected.

![Graph of model order by AIC and its difference](image)

**Figure 5.** Model order by AIC(solid line) and its difference (dashed line)

Utilizing the identified MAR-model the system noise contribution matrix is computed. In Fig. 6 the $Q-$matrix computed over the whole frequency band
can been seen. The Q - matrix is visualized with bars. The maximum height of the bar is 1, and the sum of bars along one row corresponding to one variable is 1. From the matrix it can be remarked that the system has two sources: N1 represents the noise source of the variable m_1 (X1) and correspondingly N3 noise source of the variable m_3 (X3). There is no energy coming to the system from the noise source N2, which corresponds to the variable m_2 (X2). This can be seen from the column of N2: there is no visible bars at all.

Figure 6. System noise contribution plot

Although N1 has affect to all variables, it has most influence to the variable m_1. So we can draw a conclusion that the first source is near sensor m_1. And correspondingly, since the bar of N3 is highest on row X3, we can conclude that the second source is near sensor m_3.

Using the model we have possibility to examine the spectrum of estimated source and compare it to the original source and measurement (Fig. 7 and Fig. 8). From the plots we can see that the spectra of estimated sources are extremely good: in both cases the shape of estimated spectrum corresponds rather well to the true one.

Figure 7. Spectra of source1 (solid line), measured signal (dashed line) and estimated source (dashdotted line)

5. SUMMARY

In this paper a novel method for the separation of sources is presented. In presented method autoregres-

Figure 8. Spectra of source2 (solid line), measured signal (dashed line) and estimated source (dashdotted line)

ate autoregressive model is exploited to construct conditioned spectral matrix NCSM. Conditioned spectral matrix is utilized to construct new measures which are efficient to extract independent sources.

REFERENCES


A Data-Based Enumeration Technique Of Coherent Signals

H. Krim and J. G. Proakis
Laboratoire des Signaux et Systèmes,
Plateau du Moulin, 91192, Gif-Sur-Yvette, France,
1Communications and Digital Signal Processing Center
Northeastern University, Boston, MA 02115

The combination of a smoothing transformation on an array covariance matrix with signal enumeration criteria is shown to be insufficient when dealing with fully correlated signals impinging on a Uniform Linear Array (ULA). We account for the "vestigial" correlation following a smoothing transformation of a matrix appropriately constructed with the received data, and derive a Minimum Descriptions Length (MDL)-based detection criterion. The additional constraint which is introduced, is shown to achieve a significant performance improvement, particularly with short data records and closely spaced signals.

1 Introduction

Eigenstructure-based parameter estimation techniques have been the focus of a great research activity because of their wide applications in a number of fields. In a Direction Of Arrival estimation problem, for example, determining the number of signals impinging on an array, is the first step critical to the success of any of these techniques, selected for the solution. Many algorithms for order estimation of an observed process which have appeared in the literature [2, 8, 10, 9, 11] most often simplify to a test of equality of the smallest eigenvalues of the array covariance matrix. The consistency of these detection or enumeration techniques has been very useful as a comparison measure among them. It is in fact, on account of this asymptotic behavior, that a method based on the MDL principle, proposed by Wax et al.[9, 10], has become very popular. In the presence of fully correlated signals, it requires, however, a Multidimensional (MD) search, since the signal coherence causes a failure of all the classical methods.

A computationally more attractive solution consists in first recovering the rank through a smoothing transformation of the covariance matrix [4, 8] and then applying a Smoothed Rank Profile (SRP) test. This consists in tracking the increase and eventual stabilization of the rank as the rank of the smoothing matrix varies. The stable rank corresponds to the number of signals present in the observed process with probability 1 (w.p.1) [4]. These techniques, however, share a common performance threshold problem which is due to the remaining correlation among the averaged diagonal submatrices after the smoothing takes place. We will show here that a significant performance gain is achieved if this "vestigial" correlation is accounted for.

We shall assume throughout, a uniform linear array composed of $L$ identical, omnidirectional sensors with equal spacing $d = \lambda/2$ where $\lambda$ is the wavelength. We assume that $M < L$ narrowband plane waves (centered about the known frequency $f_0$) impinge from (distinct) directions $\theta_i$, $i = 1, 2, \ldots, M$ and that $M - (L - 1)$ of them are fully correlated. The effective rank of $R = E\{xx^T\}$, is thus denoted by $L'$. The received data vector is denoted by $x(t) = (x_1(t), \ldots, x_L(t))^T$, where $x_i(t)$ is the data received by the $i$th sensor and where $"^T"$ denotes transposition and $"^H"$ conjugate transposition. We also assume that the array outputs and the corresponding noise, $x(t)$ and $n(t)$, respectively, are stationary and ergodic complex-valued random processes having zero mean and that the noise is uncorrelated with the signals; the noise terms are also mutually uncorrelated with unknown but identical variances, $\sigma^2$.

The problem is to directly estimate the number of signals impinging on the above array when given a short data record of length $N < 2L$. We recall that some or all of these signals may be coherent. For this purpose, we choose to use a data matrix $X$ shown below. The $i$th column of $X$ corresponds to the data vector recorded at the $(L - i + 1)^{th}$ sensor:

$$X = \begin{pmatrix} x_L(1) & x_{L-1}(1) & \ldots & x_1(1) \\ \vdots & \vdots & \ddots & \vdots \\ x_L(N) & x_{L-1}(N) & \ldots & x_1(N) \end{pmatrix}.$$ (1)

2 A New Data-Domain Variation

It is well known that full correlation among two or more signals folds the subspace spanned by the eigenvectors with the largest eigenvalues [6]. It was also shown in [6] that an increase in a column dimension of a smoothing transformation applied to the data matrix in Eq. (1), induced a rank increase of the resulting matrix (this is equivalent to decreasing the subarray size). The smoothing was shown to be achieved by first defining the following windowing matrix $F = [F_1, F_2, \ldots, F_m]$, where $F_m = \{0_{m(L-i-1)}^{(i-1)}F_{px}(L-i-1, m), i = 1, 2, \ldots, L - 1\}$, and then carrying out the following formal compact
product
\[ \widehat{X} = \{I_n \otimes X\} F^T, \]
where \( \otimes \) denotes the Kronecker product, and \( I_n \) is the \( n \times n \) identity matrix, with \( n = L - p + 1 \).

Realizing the key role noise plays in model identification, we use Eq. (2) to fit a linear model to the observed process and thus to unravel the underlying signal structure. It is thus reasonable to assume that errors will be incurred when a linear model is fitted to the observed data,
\[ \widehat{X}d + e = \tilde{x}_p, \]
where \( d \) is the \( p \times 1 \) linear prediction vector, and \( e = [e(p), e(p+1), \ldots, e(L)]^T \) is the \( (L-p+1)N \times 1 \) error vector. We shall assume that \( (L-p+1)N \) is large, so that the errors can be taken to be normally distributed with a covariance matrix \( \Gamma = E[ee^H] \), and have minimal end effects. The components of an error subvector (e.g., \( e(p) \)) are clearly uncorrelated, since time samples are assumed independent, and thus, uncorrelated. There exists, however, some correlation between the error subvectors since a subblock of data of any block subarray is used in subsequent ones as can be seen from Eqs. (2) and (3). To simplify the density needed to estimate the model, we whiten the model error. This, in turn requires explicit knowledge of \( \Gamma \). For that purpose, Eq. (3) is first rewritten as
\[ d^T \Gamma d = \varepsilon^T, \]
where \( d^T = [1, d]^T \). The Vec operator with its properties implies the following,
\[ \text{Vec}(d^T \Gamma d) = (I \otimes d^T) \text{Vec}(\widehat{X}d), \]
Using the latter expression in \( \Gamma \) and after a simple manipulation, one can write,
\[ \Gamma = (I \otimes d^T)E[\text{Vec}(\widehat{X}d)\text{Vec}(\widehat{X}d)^H] \times (I \otimes d^T)^H. \]
Recall that each data point has a signal component contaminated with noise (which is not to be confused with the signal subspace component). Denoting the noise components of \( \tilde{x} \) by \( \tilde{N} \) implies that \( \text{Vec}(\widehat{X}d) = \text{Vec}(\tilde{x}_p) + \text{Vec}(\tilde{N}), \) from which
\[ e \approx 0 + (I \otimes d^T) \text{Vec}(\tilde{N}), \]
since the signal components (cissoids) theoretically fit the linear model perfectly. The expression for \( \Gamma \) is now simply deduced as
\[ \Gamma = (I \otimes d^T)E[\text{Vec}(\tilde{N})\text{Vec}(\tilde{N})^H] \times (I \otimes d^T)^H \]
\[ = \sigma^2(I \otimes d^T)I(I \otimes d^T)^H = \sigma^2I \]
where \( I \) is an indicator matrix with 1 and 0 components which result from the covariance of the noise components, and \( \sigma^2 \) is the noise variance. By premultiplying Eq. (4) by \( \Gamma^{-1} \), we obtain,
\[ X' \tilde{x} = \tilde{x}_p, \]
where \( E[ee^H] = \sigma^2I \).

Using the fact that \( e \) is Gaussian, we obtain the conditional probability density function of \( \tilde{x}_p \), and derive an MDL-based criterion after obtaining the Maximum Likelihood (ML) parameter vector \( \hat{\beta} \) (by maximizing the log-likelihood function \( L(\beta) = -\log f(\tilde{x}_p|d^T, \sigma^2) \)).

We first obtain,
\[ \hat{\gamma}^2 = \frac{1}{\alpha} \| \tilde{x}_p - X' d \|^2, \]
and
\[ \hat{d} = (X' X')^H X' \tilde{x}_p, \]
where \# is a symbol used for the pseudo-inverse. Using \( \hat{\beta} = [\hat{d}, \hat{\gamma}^2], \) we can rewrite \( L(\beta) \)
\[ L(\hat{\beta}) = \frac{\alpha}{2} \log 2\pi + \frac{\alpha}{2}, \]
\[ + \frac{\alpha}{2} \log \left( \frac{1}{\alpha} \| \tilde{x}_p - X' (X' X')^H X' \tilde{x}_p \|^2 \right) \]
where \( \alpha = N \times n_p. \) Further simplification results when the Singular Value Decomposition (SVD) of \( x' = \mathbf{Y} \mathbf{U}^H \) is used to obtain its rank-\( k \) approximating matrix. This results in,
\[ \tilde{x}_p' = \mathbf{Y} \left[ \begin{array}{cc} \Sigma_k & 0 \\ 0 & 0 \end{array} \right] \mathbf{U}^H, \]
where the vectors \( \mathbf{Y} \) and \( \mathbf{U} \) are respectively the left and the right singular vectors of \( x' \), and \( \Sigma_k \) is a \( (k \times k) \) diagonal matrix of the largest singular values (assuming that \( k \) signals impinge on the receiving array). Using \( \tilde{x}_p' \), the argument of the second term in Eq. (9) simplifies to,
\[ \varepsilon = \left\{ x_p - x'_p (x' X')^H X' \tilde{x}_p \right\} \]
\[ = x_p - \mathbf{Y} \left[ \begin{array}{cc} I_k & 0 \\ 0 & 0 \end{array} \right] \mathbf{Y}^H x_p. \]
\[ \text{The invariance of } \| \varepsilon \| \text{ under a unitary transformation } \mathbf{Y}^H \text{ further simplifies [3] as,} \]
\[ \| \varepsilon_k \|^2 = \sum_{i=k+1}^{(L-p+1)N} \| \xi_i \|^2, \]
where \( \xi = \mathbf{Y}^H x_p \) and the superscript on \( \varepsilon \) indicates the assumed effective rank.

For an assumed number \( k \) of impinging signals, the optimal linear prediction vector \( d \) is achieved by minimizing \( \| \varepsilon \|^2 \). Since only \( k \) entries of the vector \( e \) are used in the minimization, and, given that complex signals are considered herein, the number of adjustable parameters are the \( 2k \) entries of \( e \). It is now possible to write down an MDL-based test which, when minimized over all possible \( k \), will provide the model order for the chosen subarray size \( p \).

\[ \hat{k} = \arg \left\{ \min_{k \in (1, \ldots, p-1)} \text{MDL}(k) \right\}, \]
where
\[ \text{MDL}(k) = \frac{\alpha}{2} + \frac{\alpha}{2} \log 2\pi + \frac{\alpha}{2} \log \left( \frac{1}{\alpha} \| \varepsilon_k \|^2 \right) \]
\[ + \frac{1}{2} (2k + 1) \log (N(L-p+1)), \]
where \( \alpha \) as previously defined, represents the length of the vector \( \hat{x}_p \) used in estimating the model. An order \( \hat{k} \) (one which minimizes the MDL(k)) is obtained, for several subarray sizes \( p \). If the number of degrees of freedom is sufficient, this order \( \hat{k} \) (effective rank of \( \hat{X} \)) will stabilize w.p.1 [4]. This value of \( \hat{k} \) for which the stabilization takes place, is as described next to be a consistent estimate of the number of signals [6].

The algorithm can now be summarized below:

<table>
<thead>
<tr>
<th>Summary of Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. For a subarray size ( p ) (usually chosen to be ( (L-1) )), obtain an estimate of the linear prediction vector ( \hat{d} ) as in equation (3);</td>
</tr>
<tr>
<td>2. Obtain an estimate of the error covariance matrix ( \hat{\Gamma} ) as in (6);</td>
</tr>
<tr>
<td>3. Whiten the noise via the inverse hermitian square root of ( \hat{\Gamma} ) as described in equation (7);</td>
</tr>
<tr>
<td>4. Use the norm of ( \hat{x} ) for different model orders as they are tested to minimize the MDL in equation (12) to obtain a model order estimate (or effective rank estimate of ( \hat{X} )) ( \hat{k} );</td>
</tr>
<tr>
<td>5. Increase (or decrease) ( p ) and repeat steps 2-4;</td>
</tr>
<tr>
<td>6. If ( \hat{k} ) stabilizes for two or more consecutive subarray sizes (as ( p ) varies), it should be picked as the number of signals; if the rank does not stabilize the number of degrees of freedom is not sufficient to obtain a solution (and there is no solution).</td>
</tr>
</tbody>
</table>

2.1 Consistency of the Detection Scheme

The power of a detection procedure is evaluated by the bias in its estimate of the model order (or number of signals), as the length of the data vector used for prediction grows without bound (or is extremely large for practical purposes).

Lemma 1 The order estimate \( \hat{k} \) obtained via the above algorithm, is a consistent estimate of the number of signals incident on a uniform linear array, i.e.,

\[
\arg \left( \lim_{n \to \infty} \text{MDL}(\hat{k}) \right) = M \quad \text{a.s. (almost surely).} \tag{13}
\]

Proof: We sketch the proof for the above lemma. We first use the criterion developed by Cozzens, et al. [4] to say that as \( p \) varies, the rank will stabilize w.p.1 to the number of signals if the number of degrees of freedom (number of sensors) is sufficient. It is then necessary to prove that the effective rank (of \( \hat{X} \)) determined with the above procedure is asymptotically consistent. We thus want to prove that the length MDL(k) is always greater than the length given by the optimal or true order \( M \) (or MDL(M)). By using the following two results [2],

\[
\lim_{N \to \infty} \frac{C_N}{N} \to 0, \quad \text{and} \quad \lim_{N \to \infty} \frac{C_N}{N} \to \infty,
\]

then

\[
\log \frac{\| \varepsilon^k \|}{\| \varepsilon^M \|} \geq -(k - M)C_N/N, \quad \text{when } k > M ;
\]

one then proceeds to prove that,

\[
\log \frac{\| \varepsilon^k \|}{\| \varepsilon^M \|} > (M - k)C_N/N, \quad \text{when } k < M \tag{14}
\]

and for \( k > M \)

\[
\text{MDL}(k) - \text{MDL}(M) \geq 0 \tag{15}
\]

and for \( k < M \)

\[
\text{MDL}(k) - \text{MDL}(M) \geq 0 , \tag{16}
\]

proving the consistency since any length is greater than the one given by the true number \( M \).

3 Simulations

Example: A 10 element array with equal spacing of \( \lambda/2 \) is considered. Two coherent narrowband signals with a common normalized frequency of \( f_{1(2)} = .25 \) Hz are assumed to impinge on the array from distinct directions \( (\theta_{1(2)} = \pm 2.5 \deg) \). The data record length is \( N=15 \) snapshots. A set of 100 random trials is used for each SNR to evaluate the detection performance of the proposed method.

When we perform additional processing to whiten the model error, the detection performance threshold as encountered in the previously described methods is improved. The performance of this DDDDET method, as noted earlier, will be commensurate with the goodness of estimation of the noise covariance matrix and the data record length. The performance improvement of the DDDDET is illustrated in Fig. 1 and confirms that the vestigial correlation between the overlapping subarrays affects the detection criterion. We should mention that the rank stabilization criterion described earlier is applied throughout this evaluation. The overall detection performance is summarized in Table 3.1. A performance comparison with some of the previously described methods favors the present approach [6].

4 Conclusion

It was shown that the vestigial correlation of the signals following a smoothing transformation needs to be accounted for if a unidimensional signal enumeration technique is to be used. The performance improvement was shown to be significant, as a consequence, with a modest computational requirement.

Footnotes

†A detection algorithm is said to be consistent if the estimated number of signals tends to the true number w.p.1, as the number of snapshots \( N \) (data record length) grows without bounds.

‡The rank of a sequence of smoothed covariance matrices is referred to as a smoothed rank profile.

§The effective rank is the rank coinciding with the number of the largest eigenvalues of the covariance matrix. It is different from the mathematical rank which is full.

∥The approximation for \( X' \) proceeds by first assuming that the process model order is \( k \), and then by selecting the \( k \) largest singular values of \( \Sigma' \) to form \( \Sigma_k' \).
Acknowledgements
One of us wishes to thank Dr. J. H. Cozzens for helpful comments and suggestions.

References


On the equivalence of SVD and TLS based Yule–Walker approaches to frequency estimation

Petre Stoica  
Department of Automatic Control  
Polytechnic Institute of Bucharest  
Splaiul Independentei 313  
R-77206 Bucharest, Romania

Torsten Söderström  
Automatic Control and  
Systems Analysis Group  
Department of Technology,  
Uppsala University, Box 27  
S-751 03 Uppsala, Sweden

Sabine Van Heffel  
ESAT Laboratory  
Department of Electrical Engineering  
Katholieke Universiteit Leuven  
Kardinaal Mercierlaan 94  
B-3000 Heverlee, Belgium

The total least-squares (TLS) technique for solving noise-corrupted linear systems of equations is receiving considerable attention in the signal processing literature. Here, we study the use of TLS within a high-order Yule–Walker (HOYW) procedure for estimating the frequencies of sinusoidal signals, and show that the TLS-based HOYW method has the same large-sample properties as the commonly used singular value decomposition (SVD)-based HOYW procedure.

1 Introduction

The TLS technique formally introduced in [1] is currently receiving a significant attention in the signal processing and system identification literature. See [2] for a recent and comprehensive overview. Applications of TLS to signal processing problems include the TLS–ESPRIT and TLS–Linear Prediction (LP) methods for estimating the frequencies of sinusoidal signals or the bearings of plane waves, from noisy measurements. See [3], [21], and [4], respectively. The two aforementioned estimation problems are closely related and can be treated similarly. However, for the sake of clarity, we focus here on the sinusoidal frequency estimation application.

TLS–ESPRIT has been analysed in [5] (see also [21]), where it was shown to have the same large-sample properties as the computationally-simpler LS–ESPRIT. This result of [5] can be obtained as a special case of a more general result proved in the following, as explained in Section 4.

The TLS–LP method [4] has been studied recently in [6] where it was shown that in the high signal-to-noise ratio (SNR) case, the TLS–LP has the same statistical properties as the commonly-used SVD-based LP technique of [7]. Concerning the TLS–LP, it is interesting to note that this technique is identical to the Min–Norm method of [8]. This fact, overlooked in [4], essentially follows from the original work [8]. It has been stated explicitly in [9], [17] and recently rediscovered in [10].

Here, we study the use of TLS in the HOYW procedure of sinusoidal frequency estimation, and show that the TLS-based HOYW frequency estimates ([11]–[13]). This type of first-order equivalence result is similar to the previously quoted result of [6]. Note, however, that the two results concern different estimation methods, and are derived by different analysis techniques. Furthermore, the result in [6] holds under the assumption of high SNR, whereas our results require that the number of data samples is large (which appears to be a more natural assumption).

2 SVD–based and TLS–based HOYW methods

Consider the noisy sinusoidal signal

\[ y(t) = \sum_{k=1}^{n} \alpha_k e^{(\omega_k t + \phi_k)} + \epsilon(t) \quad (2.1) \]

where \( \alpha_k > 0 \), the frequencies \( \{\omega_k\} \) are distinct, the initial phases \( \{\phi_k\} \) are independently and uniformly distributed over \([0, 2\pi])\), and \( \epsilon(t) \) is complex white noise with zero mean and the following variances

\[ E\epsilon(t)\epsilon^*(s) = \sigma^2 \delta_{ts} \quad E\epsilon(t)\epsilon(s) = 0 \quad (2.2) \]

Here \( E \) is the expectation operator, the superscript "*" denotes the conjugate transpose, and \( \delta_{ts} \) is the Kronecker delta. It is also assumed that \( \{\epsilon(t)\} \) and \( \{\phi_k\} \) are independent random variables. The problem is to estimate the sinusoidal frequencies \( \{\omega_k\} \). Once estimates of \( \{\omega_k\} \) are available, the estimation of the other signal parameters in (2.1) reduces to a simple LS fit (see, for example, [18], [22]). The number of sine waves is assumed to be known (given or estimated).

Define
\[ R = E \begin{pmatrix} y(t - L - 1) \\ \vdots \\ y(t - L - M) \end{pmatrix} (y^*(t - 1) \ldots y^*(t - L)) \]
\[ r = E \begin{pmatrix} y(t - L - 1) \\ \vdots \\ y(t - L - M) \end{pmatrix} y^*(t) \]
where \( L \geq n, M \geq n \). Also, define
\[ \Theta(z) = z^L + \theta_1 z^{L-1} + \ldots + \theta_L \]
\[ \theta = (\theta_1 \ldots \theta_L)^T \]
is the minimum norm solution to the following system of so-called Yule–Walker equations, (see, for example, [11]–[13], [22]),
\[ R \theta = -r \]

It is well known that \( n \) zeros of the polynomial \( \Theta(z) \) are located on the unit circle at \( \{e^{j\omega}\}_{\omega=1}^{\pi} \), and the remaining \( (L-n) \) zeros are situated strictly inside the unit circle [11]–[14]. This property of \( \Theta(z) \) can be used to obtain consistent estimates of the frequencies \( \omega_k \), in the following steps.

(i) Compute the sample covariances from the available samples \( y(1), y(2), \ldots, y(N) \) and form \( \hat{R} \) and \( \hat{r} \).

(ii) Determine a consistent estimate \( \hat{\theta} \) of \( \theta \) from the (sample) \( \text{HOYW} \) system of equations,
\[ \hat{R} \hat{\theta} \simeq \hat{r} \]

(the symbol "\( \simeq \)" used above indicates that (2.8) will in general have no exact solution).

(iii) (Root form). Estimate the frequencies as the angular positions \( \{\hat{\omega}_k\} \) of the \( n \) largest-modulus zeros of \( \hat{\Theta}(z) \) (the polynomial \( \hat{\Theta}(z) \) is made from \( \hat{\theta} \) determined in step (ii)); or, alternatively,

(iii) (Spectral form). Estimate the frequencies as the locations \( \{\hat{\omega}_k\} \) of the \( n \) largest peaks of the "pseudospectrum" \( \left| \hat{\Theta}(e^{j\omega}) \right|^2, \omega \in (0, \pi) \).

Steps (i) and (iii) above are standard. The only critical problem is performing Step (ii). Some possibilities for approaching the problem of Step (ii) are briefly reviewed in the following.

A. LS–based \( \text{HOYW} \)

The LS solution to the linear system of (approximate) equations (2.8) is given by
\[ \hat{\theta}_{LS} = -\hat{R}^T \hat{r} \]

where \( \hat{R}^T \) denotes the Moore–Penrose pseudo-inverse of \( \hat{R} \) (see, for example, [2], [18]). Now, if \( L > n \), then \( \hat{\theta}_{LS} \) in (2.9) is a very poor estimate of \( \theta \). This is so since rank \( (\hat{R}) = n \) (see [12]–[14]), whereas \( \hat{R} \) has full rank \( (w p 1) \), which implies that \( \hat{R}^T \) does not approach \( R^T \) in such a case, even when \( N \) increases without bound. The a priori knowledge that rank \( (R) = n \) is used by the ensuing approaches \( B \) and \( C \), to refine the LS–based estimate (which is never used in the form of (2.9)).

B. SVD–based \( \text{HOYW} \)

Let \( \hat{R}_n \) denote the best (in the Frobenius norm sense) rank-
\( n \) approximation of \( \hat{R} \), see [2], [18]. Replace \( \hat{R} \) in (2.8) by \( \hat{R}_n \) to obtain the following modified system of equations
\[ \hat{R}_n \hat{\theta} \simeq \hat{r} \]

Note that, in general, (2.10) has no exact solution. The minimum norm LS solution of (2.10) is given by
\[ \hat{\theta}_{\text{SVD}} = (\hat{R}_n)^+ \hat{r} \]
The HOYW method which estimates \( \theta \) as in (2.11), is said to be SVD–based since the computation of \( \hat{R}_n \) from \( \hat{R} \) requires a SVD step (as is well known). This method has been found to possess excellent statistical performance, especially for reasonably large values of \( M \) and \( L \) (see [11]–[14]).

C. TLS–based \( \text{HOYW} \)

The SVD–based approach only corrects \( \hat{R} \) in (2.8) using the a priori information that rank \( (R) = n \), and the corrected system remains (generically) incompatible. A more natural way to proceed is to make the system of equations (2.8) compatible by correcting both \( \hat{R} \) and \( \hat{r} \) using the information that
\[ \text{rank}(R) = n \]

This is the basic idea of the TLS–based approach.

Let \( \{\gamma, \lambda\} \) denote the best rank-
\( n \) approximation of the (generically full-rank) matrix \( [\hat{R} \quad \hat{r}] \). Then, the TLS–estimate of \( \theta \) is given by the solution to the following system of equations
\[ \begin{pmatrix} \gamma \\ \lambda \end{pmatrix} \begin{pmatrix} 1 \\ \phi \end{pmatrix} = 0 \]

Any solution to \( \hat{A} \theta = -\gamma \) is a TLS solution. The minimum norm solution, which we call the TLS estimate of \( \theta \), is given by
\[ \hat{\theta}_{\text{TLS}} = -\hat{\lambda}^T \gamma \]

3 Theoretical analysis

The TLS–based estimate of \( \{\omega_k\} \) is more appealing, from an intuitive standpoint, than the SVD–based estimate (as discussed in the previous section). However, these two estimates can be shown to be asymptotically (i.e. for \( N \gg 1 \)) equivalent.

Theorem 3.1. The SVD–HOYW estimate of \( \{\omega_k\} \) is asymptotically equivalent to the TLS–HOYW estimate in the sense that the two estimates coincide (to within a first–order approximation), for sufficiently large values of \( N \). This is true, regardless of the form ("root" or "spectral") used in Step (iii).

Proof. See [23].
The above theorem implies that the SVD-based and TLS-based HOYW estimates of $\{w_k\}$ have the same asymptotic distribution. This distribution has been shown, in [12], [14], to be multivariate Gaussian. An explicit formula for the covariance matrix of the estimation errors has also been derived in these references. The theoretical variances shown in the numerical simulations in the next section were computed using the quoted formula of [14].

4 Numerical study

This section provides numerical evidence leading support to the fact that the asymptotic equivalence shown in the previous section may hold in practice for reasonably-sized sample lengths. See also [23].

Consider a signal consisting of two sine waves with

$$\begin{align*}
\omega_1 &= 1 \\
\omega_2 &= 0.2 \times 2\pi \\
\alpha_1 &= 1 \\
\sigma^2 &= 1
\end{align*}$$

(hence SNR = 0 dB)

Two sample lengths have been considered (as also shown in Figure 1). In each case, 50 independent realizations of the noisy signal have been generated. The root-form SVD-based and TLS-based HOYW frequency estimates were then computed using the procedures described in [14] and [2], [9], respectively. The values of $L$ and $M$ considered are shown in Table 1, along with an attached code used in order to ease the plotting.

<table>
<thead>
<tr>
<th>$L$</th>
<th>2</th>
<th>2</th>
<th>5</th>
<th>5</th>
<th>8</th>
<th>8</th>
<th>15</th>
<th>15</th>
<th>25</th>
</tr>
</thead>
<tbody>
<tr>
<td>$M$</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
<td>8</td>
<td>8</td>
<td>15</td>
<td>15</td>
<td>25</td>
</tr>
</tbody>
</table>

Table 1: The considered combination of $L$ and $M$.

It should be noted from Figure 1 that the SVD-based and TLS-based procedures behave quite similarly in the cases considered. For small estimation errors, this behaviour was predicted by the theory developed in the previous sections. However, it is interesting that the two procedures have similar statistical performance also in the case of fairly large estimation errors. This behaviour, which is not explained by the previously developed theory, can be observed for small values of $(L, M)$, and has also been noted in several other simulations not shown here. Some comments on it can be found in the next section.

5 Some extensions

The previous analysis has also a more general implication discussed in the following.

Consider a general system of equations of the form (2.7) and its noise-corrupted version in (2.8). The matrix $[\mathbf{r} \ \mathbf{R}]$ is assumed to satisfy the rank constraint (2.12), but otherwise it may be arbitrary (i.e. $\mathbf{R}$ and $\mathbf{r}$ are not necessarily covariance matrices as in (2.5), (2.4) etc). Then, under the assumption

![Figure 1: Empirical and theoretical RMS values versus $(L, M)$ (see Table 1). Solid line = empirical RMS for SVD-based HOYW; Dashed line = empirical RMS for TLS-based HOYW; Dotted line = theoretical (asymptotic) RMS for both SVD-based and TLS-based HOYW; Upper two curves $N = 200$, lower two curves $N = 2000$.](image)

of small errors $(\mathbf{R} - \mathbf{R})$ and $(\mathbf{r} - \mathbf{r})$, it follows from the proof of Theorem 3.1 in [23] that

$$R(\hat{\theta}_{\text{SVD}} - \hat{\theta}_{\text{TLS}}) \approx 0$$

(5.1)

(to within a first-order approximation).

If $L = n$ and $R$ has full (column) rank, then (5.1) implies $\hat{\theta}_{\text{SVD}} \approx \hat{\theta}_{\text{TLS}}$, which shows that our results encompass the result in [5] on the equivalence of LS-based and TLS-based ESPRIT procedures.

If $L > n$, then (5.1) no longer implies $\hat{\theta}_{\text{SVD}} \approx \hat{\theta}_{\text{TLS}}$. However, (5.1) is still useful in establishing the equivalence of some sinusoidal frequency or bearing estimation procedures, others than HOYW, based on SVD or TLS steps. For instance, the result in [8] on the first-order equivalence, in high SNR scenarios, of SVD-based and TLS-based LP methods can be derived at once from (5.1).
It should be noted that the first-order equality $\hat{\theta}_{\text{SVD}} \approx \hat{\theta}_{\text{TLS}}$ holds in the case of $L > n$ as well. This follows from the more elaborated (but also more involved) analysis in [20], where it was shown that (in the notation of this paper)

$$\hat{\theta}_{\text{SVD}} - \hat{\theta}_{\text{TLS}} = \text{const}_1 \times \| [r \hat{R}] - [r \hat{R}] \|$$ (5.2)

whereas it is well known that

$$\hat{\theta} = \hat{\theta} - \theta = \text{const}_2 \times \| [r \hat{R}] - [r \hat{R}] \|$$ (5.3)

for both the SVD-based and TLS-based approaches. Note from (5.2) and (5.3) that the difference $\hat{\theta}_{\text{SVD}} - \hat{\theta}_{\text{TLS}}$ may be small even if neither $\hat{\theta}_{\text{SVD}}$ nor $\hat{\theta}_{\text{TLS}}$ is so. This shows that the behaviour observed in the numerical example, reported in Section 4, is indeed possible. A theoretical study of this behaviour would, however, require a detailed analysis of const1 and const2, which is yet to be done.

References


Computationally Efficient Multiple Parameter Estimation of Composite Signals in Noise with Periodic Components

A. Martin, N. Fliege
Telecommunications Group, Technical University of Hamburg, FRG

The problem of estimating amplitudes and arrival times of composite signals in additive noise is addressed. The noise process considered is a composition of Gaussian noise and periodic components with known frequency. The proposed new method is based on a simultaneous estimation of the required signal parameters and some parameters that describe the periodic components. The advantage is that the true amplitudes of the periodic components will have no influence on the estimates of the required signal parameters. It will be shown that for an increasing number of basic signals the variances of the estimates decrease to a minimum which is given by the signal to noise ratio (without periodic components). A decomposition of the estimation algorithm that requires only a linearly increasing computation effort according to the number of basic signals will be presented. This allows to handle large data records and use their advantages while keeping the computation effort relatively low.

1 Introduction

Composite signals appear in many important applications such as data communication systems or measurement problems where estimates of signal amplitudes and time delays with high precision are required.

In practice, the receiver input signals are often not corrupted by additive Gaussian noise only. An offset and further periodic components may occur due to several physical phenomena. The noise process considered is the sum of a white Gaussian noise process and some periodic components (including offset). The assumption of white noise is no general restriction, because the noise process can be prewhitened [3]. Another solution to the coloured-noise-problem can be found by introducing the covariance matrix of the noise process to the optimization criterion [4], but this method would not allow the decomposition of the problem to a very low number of operations.

Standard prefiltering techniques such as highpass or notch filtering cannot always help to eliminate the influence of the periodic components on the estimates. For example, eliminating an offset by highpass filtering is only possible if the signal itself has zero mean value.

The signals considered are linear combinations of basic signals of the form

\[ r(kT) = n(kT) + \sum_{i=1}^{T} a_i \cdot s(kT - (i-1)\tau - D). \]

Throughout this paper the basic signal \( s(t) \) and the delay \( \tau \) are assumed to be known, while the amplitudes \( a_1, \ldots, a_T \) and the delay \( D \) are unknown parameters that have to be estimated in real time. In applications like the analysis of mass spectrometer signals the delay \( \tau \) is also not known a priori, but it can be estimated via a least-squares error minimization in a "warming-up phase".

For the estimation algorithm presented in this paper, the delay \( \tau \) does not have to be an integer multiple of the sampling rate \( T \), so that we do not require any sampling rate synchronization.

We can write (1) in vector notation as

\[ r(kT) = S(kT-D) \cdot a + n(kT) \]

(2)

where \( r(kT) \) includes \( m \) samples of the received signal, \( a \) contains the amplitudes, and the \( m \times T \) dimensional signal matrix \( S \) contains the samples of the delayed versions of the signal \( s(t) \).

In order to estimate the delay \( D \) from the sampled input signal we use the decomposition

\[ D = \Delta + \delta, \quad |\delta| \leq T/2, \]

(3)

where \( \Delta \) is an integer multiple of \( T \). For \( kT = \Delta \) we can rewrite (2) as

\[ r(\Delta) = S(\delta) \cdot a + n(\Delta). \]

(4)

The noise process can be written as

\[ n(kT) = n_W(kT) + n_P(kT) = n_W(kT) + N \cdot c(kT) \]

(5)

where \( n_W(kT) \) and \( n_P(kT) \) denote the white Gaussian noise process and the periodic components respectively. The matrix \( N \) contains samples of the periodic base functions and the vector \( c \) represents the periodic components. By combining the matrices and vectors as follows

\[ B = [S, N], \]

(6)

\[ b^T = [a_T, c_T]^T, \]

(7)
we can rewrite (4) in the short form
\[ r(\Delta) = B(\delta) \cdot b + n_v(\Delta). \]  
(8)

In order to get short and distinct formulae, the time dependence of vectors and matrices will not be written explicitly in the following sections.

2 Maximum Likelihood Estimation

The maximum likelihood estimates \( \hat{b}, \hat{D} \) in additive white Gaussian noise \([2], [4], [5]\) have to be deduced from
\[ \| r - Bb \| = \min \left( b = \hat{b}, D = \hat{D} \right) \]  
(9)

where \( \| \cdot \| \) denotes the Euclidean norm \( \| x \| = x^T x \).

A straightforward derivation of the corresponding likelihood equations
\[ \frac{\partial}{\partial \hat{b}} \| r - Bb \|^2 \bigg|_{b = \hat{b}, D = \hat{D}} = 0 \]  
(10)
\[ \frac{\partial}{\partial \hat{D}} \| r - Bb \|^2 \bigg|_{b = \hat{b}, D = \hat{D}} = 0 \]  
(11)

leads to the orthogonality relations
\[ B^T \cdot [r - Bb] = 0, \]  
(12)
\[ b^T \hat{B}^T \cdot [r - Bb] = 0. \]  
(13)

Equations (12) and (13) say that the estimated noise \( r - Bb \) has to be orthogonal to all signals in \( B \) and to \( Bb \). For building up the matrix \( B \) see Appendix.

It is clear that we can only find unbiased amplitude estimates, if \( B \) contains linear independent rows. Assuming an existing matrix \( [B^T B]^{-1} \), the solution of (12) is given by
\[ \hat{b} = [B^T B]^{-1} B^T r \]  
(14)
\[ \hat{D} = \hat{D} \]

By substituting (14) into (13), we find the scalar target function
\[ Q(\hat{D}) = b^T \hat{B}^T r - b^T \hat{B}^T \hat{B} \hat{b} \bigg|_{\hat{D} = \hat{D}} = 0 \]  
(15)
for the time delay estimation problem. This target function is a nonlinear function of the delay \( D = \Delta + \delta \) and of the input samples. For only one basic signal it can be shown that a target function corresponding to (15) can be produced at the output of a linear filter (see Appendix).

Decomposition

We will now present decompositions of the equations (14) and (15) which require only a low number of operations. Since \( N \) is independent of \( \delta \), the matrix \( \hat{B} \) has the structure (see Appendix)
\[ \hat{B} = [\hat{S}, 0]. \]  
(16)

By re substitution of (6), (7) and using (16) we can rewrite (15) as
\[ Q(D) = \hat{a}^T \hat{S}^T r - \hat{a}^T \hat{S}^T \hat{N} \hat{c}. \]  
(17)

The method of calculating the amplitude estimates is most important for the computation effort. Resubstitution of (6) and (7) into (14), partitioned matrix inversion and some straightforward numerical operations lead to
\[ \hat{c} = C r, \]  
(18)
\[ \hat{a} = M r - M N \hat{c}, \]
where \( M \) and \( C \) are given by
\[ C = \left[ N^T N - N^T S \left[ S^T S \right]^{-1} S^T N \right]^{-1} \]  
\[ \cdot N^T \left[ I - S \left[ S^T S \right]^{-1} S^T \right], \]  
(19)
\[ M = \left[ S^T S \right]^{-1} S^T. \]  
(20)

It should be pointed out that \( C \) has only a few rows and that \( M \) contains only a low number of non-zero elements if the basic signals are mutually orthogonal, especially if they are not overlapping. For a matrix \( S \) that contains orthonormal rows, the pseudo-inverse is given by \( M = S^T \). In general, \( M \) should be calculated from a singular value decomposition (SVD) of \( S \) for achieving a high numerical stability. The matrix \( C \) can be found to be the pseudo-inverse of \( (I - S \left[ S^T S \right]^{-1} S^T) N \) and can also be calculated from a SVD.

3 Application

A typical application is the estimation of the concentration of mass from mass spectrometer signals. Fig. 1 shows a typical signal where the concentration of mass is proportional to the amplitudes of the impulses. In this application, a typical number of basic signals is \( I = 100 \). For parameters \( T = 0.1 \text{msec}, \tau = 10 \cdot T \) and ten non-zero samples of each signal, we receive a total number of \( m = 10 \cdot I = 1000 \) samples. An offset and a sinusoidal disturbance with the frequency \( \omega = 2\pi 50 \text{Hz} \)
are typical, so that the matrix $N$ and the vector $c$ can be written as

$$N = \begin{bmatrix}
\cos(0T\omega_1) & \sin(0T\omega_1) \\
\vdots & \vdots \\
\cos((m-1)T\omega_1) & \sin((m-1)T\omega_1)
\end{bmatrix},
$$

and

$$c = [c_0, c_1, c_2]^T.
$$

It is clear that an extension to further periodic components can simply be made.

## 4 Computation Effort

In order to minimize the computation effort we can calculate all matrices a priori for a number of $p+1$ equally spaced interpolation times

$$\delta^0 = -T/2 + i \cdot T/p, \quad i = 0, 1, \ldots, p
$$

so that we only have to choose the best matrices in a real time application. The number $p$ depends on the desired accuracy.

The time delay has to be found from the zero of $Q(D)$. An example is shown in figure 2, where the true delay is $D = 157T$, and its estimate is $\hat{D} = 148T$.

Close to its zero the target function $Q(D)$ is relatively linear. Therefore we can estimate $\delta$ without great error by calculating the zero of the linear interpolation of $Q(D-\delta)$ and $Q(D+\delta)$. For higher precision nonlinear search methods like a Newton-Raphson search or a modified Newton-Raphson search can be used [1].

We will compare the computation effort of the different solutions while using the example of analysing mass spectrometer signals with the parameters from section 3.

Both equations (14) and (18), respectively, for amplitude estimation can be interpreted as different implementations of the same multidimensional linear filter. We want to show the computation effort for non-overlapping orthonormal base signals in $S$, where each basic signal in $S$ has ten non-zero samples. This means: $M$ is given by $M = S^T$, contains $m = 10 \cdot I$ non-zero samples, and $S^T r$ requires $10 \cdot I$ multiplications. Under these conditions, we need $10 \cdot I^2$ multiplications for the direct solution $(B^T B)^{-1} \cdot r$, or $40 \cdot I + I^2$ multiplications if we first compute $B^T \cdot r$ and multiply the result with $(B^T B)^{-1}$.

The implementation (18) only leads to $43 \cdot I$ multiplications (see Table 1). The important point is that the effort of (18) increases linearly with the number $I$ of basic signals. This means that the relative effort for analysing one basic signal is independent of the total number $I$.

<table>
<thead>
<tr>
<th>Tab. 1: Computation Effort for eq. (18)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$c = Cr$</td>
</tr>
<tr>
<td>30 $\cdot$ $I$</td>
</tr>
</tbody>
</table>

As is shown in table 2, the computation of the nonlinear target function $Q(D)$ requires a linearly increasing effort if it is computed from (17).

<table>
<thead>
<tr>
<th>Tab. 2: Computation Effort for $Q(D)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$r^T S$</td>
</tr>
<tr>
<td>10 $\cdot$ $I$</td>
</tr>
</tbody>
</table>

## 5 Computer Simulation Results

Simulations have been made while using the parameters from section 3. The energy of the basic signals is one. The variance of the white noise process is $\sigma_wn^2 = 1$, and the expected energy of the signal amplitudes is

$$E \{ a_i^2 \} = 10, \quad i = 1, \ldots, I.
$$

Figure 3 shows the variances $\sigma_1^2$ and $\sigma_2^2$ in relation to the number of basic signals. The actual value of the sinusoidal disturbances has no influence on the estimates because they are estimated simultaneously. It can be seen that both variances decrease rapidly if $I$ increases. For a high number $I$ we can reach a value for $\sigma_2^2$ which is close to the minimum $\sigma_{\text{min}}^2 = 1$ given by the signal-to-noise ratio in the white noise case $r = S \hat{a} + n_w$. There are two reasons for the reduction of variances: 1) the periodic noise-to-white noise ratio increases if $I$ increases, 2) under condition (24), we can find $\sigma_2^2 \sim 1/I$ [6].
6 Conclusion

A new method for estimating amplitudes and arrival times of composite signals in additive noise with periodic components has been presented. The method is based on a simultaneous estimation of the required signal parameters and some parameters that describe the periodic components, so that the true amplitudes of the periodic components will have no influence on the estimates of the wanted signal parameters. The variances of the estimates decrease to a minimum which is given by the signal to noise ratio for an increasing number of basic signals. A new decomposition of the estimation algorithm that requires only a linearly increasing computation effort according to the number of basic signals has been presented. This allows to handle large data records and use their advantages without having an effectively increased computation effort.

A Appendix

In order to find unbiased estimates, the matrices $B, \bar{B}$ should achieve the condition

$$\bar{B}^T(\delta) \cdot B(\delta) = 0$$  \hspace{1cm} (25)

for any $\delta$.

We can build up a matrix $\tilde{S}$, analogous to $S$, from the samples of $\frac{\partial}{\partial x} s(t - [t - 1]T - \Delta - \delta)$ and use an orthogonal projection to achieve (25)

$$\bar{B} = \left[ I - B \left[ B^T B \right]^{-1} B^T \right] \left[ \tilde{S}, 0 \right].$$  \hspace{1cm} (26)

The disadvantage of (26) is that $\bar{B}$ will contain a large number of non-zero elements. Since the orthogonal projection from (26) is included in the target function (15) which can be written as

$$Q(D) = \bar{B}^T \left[ I - B \left[ B^T B \right]^{-1} B^T \right] \cdot r,$$  \hspace{1cm} (27)

we can use (16) directly.

By using (18) and (19) the target function (17) can be written as

$$Q(D) = \hat{\delta}^T \left[ \tilde{S}^T \left( \tilde{S}^T \left[ N^T N - N^T S \left[ S^T S \right]^{-1} S^T N \right]^{-1} \cdot N^T \left[ I - S \left[ S^T S \right]^{-1} S^T \right] \right) \right] \cdot \hat{\delta}.$$  \hspace{1cm} (28)

If we only have one basic signal, the expression in round brackets in (28) can be interpreted as an impulse response of a linear filter, and by ignoring the amplitude estimate, we could produce a corresponding target function at the output of a linear filter.

References


THRESHOLD-FREE BAYESIAN ESTIMATION USING CENSORED MARGINAL INFERENCE

Anthony P. Quinn

Signal Processing and Communications Laboratory, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, England.

We contrast the strong Bayesian policies of joint and marginal inference for the deterministic parameters of a general signal model. The marginal inference uniquely trades data support for the model against the Occam Parameter Inference (OPI) which seeks a reduction in model complexity. In stressful regimes, the latter dominates, offering an objective criterion for inference rejection. We present the new Censored Marginal (CMaAP) Estimator, which is threshold-free and optimal. High accuracy estimates of the resolution frequency between two censored is obtained in regimes where the joint and ML estimators fail.

The CMaAP estimator embodies an objective rule for rejection of the Estimation Problem when ill-posed.

1 Strong and Weak Bayesian Inference

The Orthodox School rejects the extended role of Probability Theory as the logic of belief in hypotheses, and admits probabilities only on a random variable Θ, for which a (conceptual) frequency distribution of outcomes may be postulated. Orthodox and Bayesian inference for random parameters are equivalent since, in the former case, a frequency-based prior is proposed on the space $\tilde{\Theta}$ of $\Theta$, which modifies the likelihood function $l(\theta | d, z)$, via Bayes' Theorem, to yield the a posteriori (AP) inference:

$$p(\theta | d, z) \propto l(\theta | d, z) p(\theta | z)$$  \hspace{1cm} (1)

Probabilistic random parameter inference is 'weakly' Bayesian [1] in the sense that it does not require an acceptance of the belief basis of Probability Theory. The distinction between Bayesian and orthodox estimation has been clouded by repeated assertions that Bayesian and 'random' parameter estimation are synonymous [2]. We stress that the inference policies which are being developed here are both Bayesian and deterministic. We classify this inference as 'strongly' Bayesian [1] since it requires the admission of a prior distribution on the deterministic parameter space $\tilde{\Theta}$, something which is excluded by the orthodox philosophy. The currency of the term 'random variable', to refer to the arguments of a probability function, prejudices the Bayesian philosophy and is misleading. We shall prefer the term 'probabilistic parameter' (p.p.) [1] which embraces the diverse nature of such parameters, both fixed and random.

2 Joint (JAP) and Marginal (MaAP) Bayesian Point Inference

Since the prior, $p(\theta | z)$, for deterministic $\theta$, is excluded by the orthodox school, the likelihood function remains as the frequentist 'inference machine'. This is a point function \[3\] whose mode is chosen heuristically as an 'appropriate' estimate of $\theta$, being the one for which the observed data are most probable. In contrast, the AP inference (1) is a measure function \[8\] over the space $\tilde{\Theta}$ of $\Theta$, from which a variety of point inferences may be drawn in order to estimate $\Theta$, using decision theoretic arguments (e.g. see \[2\] [4]). The Maximum a Posteriori (MAP) estimator is revealed as the minimum risk estimator under a uniform loss hypothesis.

Consider the regression of $N$ data samples $d_i$ onto the General Non-Linear Signal (GNLS) model $g(t)$ \[1\] \[4\], formed as the linear combination of $m$ basis functions, $G_k(t)$, each parameterized non-linearly in unknown vector $\omega$:

$$d_i = \sum_{k=1}^{m} h_k g_k(t_i, \omega) + \epsilon_i$$  \hspace{1cm} (2)

where $\epsilon_i$ is an additive noise process. In vector-matrix form, $d = G b + e$, which is the parametric decomposition of the data set into systematic and random components. In the Estimation Problem, the model order $m$ and the basis function type, $G_k(t)$, are known a priori. The deterministic parameters $\omega$ and $b$ are unknown, and are therefore interpreted as p.p.s in the Bayesian Paradigm. The Maximum Entropy Method (MEM), with second order moment constraints, assigns conservative Gaussian priors on the random p.p. vector $e$ and on the deterministic p.p. vectors $\omega$ and $b$. Assuming complex data:

$$p(e | \Sigma, T) \propto \exp(-\frac{e^H \Sigma^{-1} e}{2}) \quad e \in \mathbb{C}^N$$

$$p(b | B, T) \propto \exp(-\frac{b^H B^{-1} b}{2}) \quad b \in \mathbb{R}^m$$

$$p(\omega | W, T) \propto \exp(-\frac{1}{2} \omega^H W^{-1} \omega) \quad \omega \in \mathbb{C}^m$$  \hspace{1cm} (3)

The hyperparameters, $\Sigma$, $B$ and $W$, are testable information in the MEM, and are assumed known. These densities may be employed in Bayes' Theorem to reveal the AP inference $p(\omega, b | d, \Sigma, B, W, T)$ for the parameters of $g(t)$, as described in \[1\] \[4\].

2.1 JAP Estimation of $\omega$ and $b$

If estimates of all $\theta = (\omega^T, b^T)^T$ are required, then the complete inference must be maximized jointly in all its arguments.
The JAP sufficient functions for estimating $\omega$ and $b$ are revealed as [4]

$$
S_{\text{JAP}}(\omega) = d^H \Sigma^{-1} G B_{\text{JAP}}(\omega) - \frac{1}{2} \omega^T W^{-1} \omega \quad (4)
$$

$$
B_{\text{JAP}}(\omega) = (X^H X)^{-1} G^H \Sigma^{-1} d \quad (5)
$$

where $X^H X = G^H \Sigma^{-1} G + B^{-1}$. (4) is maximized at $\omega_{\text{JAP}}$ and the accompanying JAP estimate of $b$ is found by substituting $\omega_{\text{JAP}}$ into $B_{\text{JAP}}$.

2.2 MaAP Estimation of $\omega$

Since $p(b \mid d, \Sigma, B, W, T)$ is a measure function, Probability Theory permits any of its parameters to be marginalized out by integration. Performing $m$ integrations over $\omega$, the sufficient function for marginal estimation of $\omega$ is found to be [4]

$$
S_{\text{MaAP}}(\omega) = S_{\text{JAP}}(\omega) - \ln |G^H \Sigma^{-1} G + B^{-1}| = S_{\text{JAP}}(\omega) + D(\omega) \quad (6)
$$

where $S_{\text{MaAP}}(\omega)$ is also known as the marginal support function for $\omega$ [1]. The term $D(\omega)$ has been ignored in the literature on Bayesian parameter estimation, e.g. [5], principally because the performance of the marginal estimator has only been investigated in unstressful regimes, where $D(\omega)$ is inactive. We shall now show that it is dominant in stressful regimes and confers robustness on the point inference endeavour.

3 Ockham Parameter Inference (OPI)

If the data $d$ are dominantly random, then the fitting of a systematic model, $s = Gb$, violates Ockham’s Desideratum of Simplicity. This requires that observed variations be treated as random unless there is support for the systematic hypothesis [1]. Consider the null hypothesis $H_0 \equiv \{Gb = c\}$ which asserts that the model has been fitted erroneously to noise. The inference for $\omega$, conditioned on $H_0$, is given by [1]

$$
\ln P(\omega \mid \Sigma, B, H_0) = -\ln |G^H \Sigma^{-1} G + B^{-1}| = D(\omega)
$$

assuming a diffuse prior for $\omega$. Thus, $D(\omega)$ measures the hypothesis that we have fitted randomness with determinism in accepting the model (2), and, as such, it is a direct metric of the degree to which the assumed model contravenes Ockham’s Rule. We refer to it as the Ockham Parameter Inference (OPI). If rank($G$) = $q < m$ at some $\omega = \omega_0$, then linear dependence has set in among $m - q$ of the basis functions, and the complexity of the model has diminished. Since $D(\omega) \rightarrow \infty$ in this case (assuming a dominant likelihood [4]), we conclude that the OPI overwhelmingly supports a reduction in model complexity.

3.1 An Absolute Criterion for Inference Rejection

If $\omega_0 \in \mathbb{W}$, the space of $\omega$, then the OPI dominates $S_{\text{JAP}}(\omega)$, and $\omega_0$—being the redundancy inference—will be returned in all data regimes. However, it is inconsistent to test for the redundancy of an accepted prior model hypothesis $I$. A regularizing parameter, $\omega_I > 0$, which expresses our belief in the closest approach to redundancy, must be proposed as a hyperparameter of the inference. The following features of (6) establish the Ockham-based behaviour of the marginal inference policy for $\omega > 0$:

(i) The data enter the inference only via $S_{\text{JAP}}(\omega)$;

(ii) $S_{\text{JAP}}(\omega)$ dominates the inference when the data strongly support the model, and estimates of $\omega$ may therefore be inferred;

(iii) In the absence of data support, $D(\omega)$ dominates, returning an inference of model redundancy. The estimation objective is rejected.

Data support is strong when (i) the basis functions are well resolved in the data and (ii) the SNR is large. The marginal inference inherently ‘calibrates’ this support against the OPI. Estimation is only acceptable when $S_{\text{JAP}}(\omega)$ dominates $D(\omega)$. This is a quantitative statement of Ockham’s Rule, and furnishes an absolute criterion for rejection of parameter inference from $d$, conditioned on $I$. The criterion is regularized by the hyperparameters $\Sigma, B, W$ and $\omega_I$ of the hypothesis. Attempts to inspect $S_{\text{JAP}}(\omega)$ in isolation, in order to determine model redundancy, are invariably heuristic.

3.2 The Necessity of Marginalization

JAP inference is conditioned upon an absolute acceptance of the model $I$, and may therefore incur the overfitting of the data. Thus, JAP (and ML) estimation are ill-posed in stressful regimes. They suffer a threshold effect with increasing noise, due to the chaotic proliferation of viable point estimates, as $d$ recedes from the signal subspace $\mathbb{G}$ [1].

Marginalization over $\mathbb{B}$ (6) measures our belief, not in particular points of the parameter space, but in the entire subspace spanned by the model basis functions at a particular $\omega$. Consequently, $p(\omega \mid d, \Sigma, B, W, I)$ employs a volume displacement metric of the signal subspace $\mathbb{G}$ from $d$ [1]. This metric is functional on (i) the closest approach of $\mathbb{G}$ to $d$ (measured by $S_{\text{JAP}}(\omega)$) and (ii) the volume element entrapped by the basis functions of $G$ (measured by $D(\omega)$). Smaller volume elements imply increased basis function redundancy and receive greater support via (6). Thus, $D(\omega)$ behaves as an “intra-hypothesis Ockham factor” [4].

We conclude that joint inference is inadequate as a policy for admitting the Simplicity Desideratum, which should guide all attempts at model-based inference. The Desideratum should always be active in stressful regimes, necessitating the adoption of the marginal policy. We reject a number of fallacies which often accompany the interpretation of marginalization in the Bayesian Paradigm [3]:

(i) It does not ‘get rid of’ $b$, since $p(\omega \mid d, \Sigma, B, W, I)$ remains conditioned on $I$, which implies both $b$ and $\omega$, via (2);

(ii) The space in which numerical optimization is required is not reduced by marginalizing over $\mathbb{B}$. In contrast, the numerical load is increased because of the requirement to calculate $D(\omega)$ in (6);
(iii) Marginal estimates cannot be combined to form a joint estimate of the complete parameter set. A policy which is consistent with the inference question should be adopted.

4 Bayesian Superresolution of Two Complex Tones

Consider the problem of estimating the resolution frequency \( \omega \) (scalar) in the following model:

\[
g(t) = e^{j\omega t} \left[ b_1 + b_2 e^{j\omega t} \right]
\]

Assuming \( \omega_1 \) known, and adopting the uncorrelated priors \( \Sigma = \sigma^2 I_N \) and \( B = \sigma^2 I_m \), and a diffuse prior for \( \omega \), then, from (3), (4) and (5):

\[
S_{\text{JAP}}(\omega) = f^H B_{\text{JAP}} f(\omega)
\]

where \( \omega \geq \omega_1 > 0 \), \( B_{\text{JAP}}(\cdot) = \left( G^H G + \rho B_0 \right)^{-1} f \) and \( \rho = \frac{\sigma^2}{\sigma^2 + \text{SNR}} \).

\[
f = \{ F(\omega_1), \ldots, F[\omega_1 + \omega] \}
\]

being samples of the DFT of \( d \) at candidate frequencies \( \omega_1 \) and \( \omega_1 + \omega \). The OPI is given by

\[
D(\omega) = -\sigma^2 \ln \left( \frac{(N + \rho)^2 - \sin^2 \left( \frac{\rho\omega}{2} \right)}{\sin^2 \left( \frac{\rho\omega}{2} \right)} \right)
\]

which depends only on the difference frequency \( \omega \), as is appropriate for a function which measures the ‘degree of redundancy’ in the model. It is plotted in Fig. 1 for \( \omega \geq \frac{1}{N} \) DFT bins, where a DFT bin-width is defined to be \( \frac{\sigma^2}{\sigma^2 + \text{SNR}} \) the sampling frequency. The OPI exhibits the characteristic redundancy peak below the first DFT bin, manifesting the Ockham support for the coexistence of the cisdal basis functions in (7), which occurs at \( \omega = 0 \). The OPI is minimized ‘on-bin’, where the cisdal basis functions are orthogonal.

5 Robust Sub-Threshold Inference using the Censored Marginal (CMaAP) Estimator

A point inference, \( \omega_{\text{CMaAP}} \), drawn from \( S_{\text{CMaAP}}(\omega) \), is classified as follows:

(i) If the data-based support exceeds the support for redundancy, \( \omega_{\text{CMaAP}} \) should be used to estimate \( \omega_1 \);

(ii) If the OPI support for redundancy exceeds the data support, \( \omega_{\text{CMaAP}} \) should be used to reject the model.

In a worst case scenario, a point inference, \( \omega_{\text{CMaAP}} \), is obtained when

\[
S_{\text{CMaAP}}(\omega_{\text{CMaAP}}) = [S_{\text{CMaAP}}(\omega_1)]^T
\]

i.e. the overall support for \( \omega_{\text{CMaAP}} \) just exceeds the support for redundancy at \( \omega_1 \). We define the censoring threshold, \( \omega_c \), for estimation, to be that frequency whose redundancy support is equal to its data support in this worst case scenario:

\[
S_{\text{JAP}}(\omega_1) = S_{\text{JAP}}(\omega_c) = D(\omega_c) - D(\frac{\omega_1}{N})
\]

Figure 1: The 2-Cis OPI (\( \sigma^2 = 1 \); \( N = 32 \)).

Eliminating \( S_{\text{JAP}}(\cdot) \) from (6), (10) and (11), we obtain

\[
D(\omega_c) = \frac{1}{2} \left[ D(\omega_1) + D(\frac{\omega_1}{N}) \right]
\]

in the CMaAP estimation policy, we reject an estimate, and the model \( I \), if \( \omega_{\text{CMaAP}} < \omega_1 \). An estimate \( \omega_{\text{CMaAP}} \geq \omega_1 \) is accepted, being data-dominated.

(i) \( \omega_c \) depends on the fixed parameters of the hypothesis \( I \), namely \( m \), \( N \), \( \sigma^2 \), \( \rho_1 \), \( \omega_1 \) and \( \omega_0 \), but not on the parameters of \( d \), i.e. \( \omega \) and SNR. It therefore constitutes an objective criterion for censoring an estimate;

(ii) Since \( \omega_{\text{CMaAP}} \geq \omega_0 \), then \( \omega_c \) is the smallest estimable frequency resolution using this policy;

(iii) The rule is conservative. If \( S_{\text{CMaAP}}(\omega_{\text{CMaAP}}) > S_{\text{CMaAP}}(\omega) \) for \( \omega_{\text{CMaAP}} < \omega_1 \), then its data support may still be dominant. The ‘hard’ threshold ensures that all CMaAP estimates are data-dominated.

5.1 Performance of the CMaAP Estimator

We shall take \( N = 32 \), \( f_s = 32 \) Hz, \( \sigma^2 = 1 \) and \( \omega_1 = 2.5 \) DFT bins. We take \( \rho = 0 \), so that JAP and ML estimation are synonymous. The censoring threshold \( \omega_c \) is determined by substituting (9) into (12) yielding the following relation:

\[
\frac{\sin^2 \left( \frac{\omega_1}{2N} \right)}{\sin^2 \left( \frac{\omega_1}{2N} \right)} = N^2 \left[ 1 - \frac{1 - \sin^2 \left( \frac{\omega_1}{2N} \right)}{N^2 \sin^2 \left( \frac{\omega_1}{2N} \right)} \right]
\]

The solution is found at \( \omega_c = \frac{10}{\text{dof}} \) DFT bins. 2000 Monte Carlo trials were undertaken at each tested value of SNR and \( \omega \). We note the following (Fig. 2):

(i) The JAP/ML and MaAP policies exhibit thresholds. The Threshold SNRs increase as \( \omega \) diminishes;

(ii) The deleterious consequences of ignoring Ockham's Rule are manifested in the proliferation of outlier estimates, and failure of the JAP/ML policy, below threshold;

(iii) The MaAP performances bifurcate from the JAP/ML curves, and the MaAP bias falls to \(-100\%\), as the OPI becomes more dominant with decreasing SNR. The variance exhibits a turning point and decreases in very noisy regimes;
The CMaAP estimator behaves as an 'outlier detector' [1], and is threshold-free. It is virtually unbiased below the JAP/ML threshold, and the rate of increase of the CMaAP variance with falling SNR is retarded;

(v) The danger—which attends the JAP/ML policy—of estimation below the noise threshold is eliminated in the CMaAP policy. Estimates which exceed \( \omega_k \) are highly confident;

(vi) As the SNR decreases, so does the probability of being able to make an estimate under the CMaAP policy. The probability also diminishes with decreasing \( \omega \), as indicated in Fig. 3. There were no data-dominant estimates of \( \omega = 0.3 \) for SNR \( \leq -40 \) dB over the 2000 trials. This sets an upper bound on the noise level in the data for which estimation is possible at \( \omega = 0.3 \). The bound increases as \( \omega \) increases;

(vii) The odds in favour of being able to make a CMaAP estimate below the JAP/ML threshold are good: e.g. there is a 50% chance of being able to estimate \( \omega = 0.3 \) bias at \(-15\) dB, with zero bias and an estimator variance of \(-28\) dB. This stressful regime incurs a JAP/ML bias of \(+34\)% with a variance of \(-9\) dB.

6 Asking the Right Question

The inference policy which we adopt must be consistent with the nature of the question being asked. Questions about the values of the global parameter set are ill-posed in stressful regimes and must be avoided. This encourages the adoption of marginal inference questions which admit Ockham's Rule.

Marginalization over deterministic parameter spaces is the exclusive territory of the strong Bayesian. Marginal inference uniquely enforces the stresses inherent in model-based signal processing, trading the belief in the model as 'truth' against the belief in the model as 'error'. The embodiment of the Simplicity Desideratum, and the ability to undertake robust estimation below threshold, are powerful incentives for the adoption of the strong Bayesian philosophy. Marginalization deserves a pre-eminence in Inference, since it confers a reducible aptitude on the inductive task.

References


EFFICIENT RECURSIVE "ML" BEARING ESTIMATION IN NON STATIONARY CONTEXT

P.LARZABAL* and H.CLERGEOT†

* THOMSON-CSF, Division RGS, Service traitement du signal, 66 rue du fossé blanc 92231 Génnevilliers FRANCE—Tel (1)-46-13-20-00—Fax 46-13-25-55
†LBSIR,URA CNRS n° 1375, Ecole Normale Supérieure, 61 avenue du président Wilson, 94235 CACHAN CEDEX, FRANCE—Tel (1)-47-40-21-19—Fax 47-40-20-74

ML or approximate ML may be considered as the upper state of the art in high resolution methods, but they suffer from an extensive multidimensional search. Starting from a crude initialization with a robust low resolution method, the "recursive" ML presented in [5] avoids this extensive multidimensional search and provides an opportunity for extension to moving sources tracking. Emphasis in this paper is on the non stationary case.

1. INTRODUCTION

1.1 The problem addressed

Maximum likelihood is presently recognized as the most promising approach in array processing for high resolution bearing estimation, including such important topics as array perturbations. In a stationary context, three very similar algorithms, Maximum likelihood, Approximate Maximum Likelihood or Weighted Subspace Fitting methods provide very similar practical implementations of the same ML principle [1,2,3]. We have recently proposed in [4] a recursive formulation with moving sources, assuming a known, fixed number of sources and a good initialization. In [5] our interest was focused on these two last remaining fundamental topics (sources detection and initialization), but it was restricted to stationary sources. We propose here to extend this concept to the case of maneuvering targets which can appear or disappear.

1.2 Existing solutions

Presently existing algorithms amount to application of stationary methods on adjacent intervals short enough to neglect the sources displacement. There is a compromise to find between an interval long enough to resolve closely spaced sources but short enough to minimize non stationarity effects.

1.3 Main features of our algorithm

We propose a time recursive algorithm with an update on a small number of snapshots (even with one snapshot). In [4] the source's number was known, hence it is not, so that the algorithm is conceived for simultaneous estimation and detection with the ability to detect a new source and update M at each iteration. This enables the use of a low resolution method for initialization with a value M<MIo. We present an implementation for moving sources with occasional on/off power switching.

2. METHOD DESCRIPTION IN A STATIONARY CONTEXT

2.1. Bearing estimation algorithm

2.1.1. Notations and Model

No assumption is made about array's geometry. We suppose an N dimensional observation vector in the form:

\[ \mathbf{X}(t) = \mathbf{Y}(t) + \mathbf{N}(t) \]  

(1)

N is an additive noise, which for simplicity, is assumed to be a circular complex white noise with covariance \( \sigma^2 \mathbf{I} \). The steering vector associated to a source with azimuth \( \theta \) is \( \mathbf{s}(\theta) \). Let's denote \( \mathbf{G}(t) \) the deterministic unknown sources amplitude vector. Expression (1) becomes:

\[ \mathbf{X}(t) = \mathbf{S} \cdot \mathbf{G}(t) + \mathbf{N}(t) \]  

(2)

where

\[ \mathbf{S} = [\mathbf{s}(\theta_1), \mathbf{s}(\theta_2), \ldots, \mathbf{s}(\theta_M)] \]  

(3)

2.1.2 Maximum Likelihood Method

The loglikelihood for this model can be easily calculated from a set of T independent snapshots:

\[ \ln L(\Delta, \mathbf{G}) = -\frac{1}{2} \ln (2\pi \sigma^2)T - \frac{1}{2} \sum_{t=1}^{T} \mathbf{X}(t) \cdot \mathbf{G}(t)^\dagger \]  

(4)
where $\delta^\Delta$ are the candidate values for vectors $\Theta$ and $G$. Separate maximisation with respect to the amplitude vectors $\Omega(t)$ yields:

$$L_0(\delta) = \text{Tr}[\Pi_N \hat{R}_X]$$  \hspace{1cm} (5)

with $\hat{R}_X$ the estimated covariance matrix and $\Pi_N$ the projector onto noise subspace. If $P$ is the sample covariance matrix of the amplitudes, we have:

$$\mathbb{E}[\hat{R}_X] = S.P.S^\dagger + \sigma^2(I) = R_Y + \sigma^2(I)$$  \hspace{1cm} (6)

where $R_Y$ stands for the noiseless covariance. Note that

$$R_Y = S.P.S^\dagger = \Pi_S \mathbb{E}[\hat{R}_X - \sigma^2(I)] \Pi_S$$  \hspace{1cm} (7)

2.1.3. Subspace Fitting Methods

According to standard EVD techniques, the signal subspace is identified as the space spanned by the eigenvectors associated to the largest eigenvalues of $R_X$. Let $\Pi_S = \hat{E}_S \hat{E}_S^\dagger$ be the corresponding signal projector. Then $R_Y$ may be estimated by:

$$\hat{R}_Y = \Pi_S [\hat{R}_X - \sigma^2(I)] \Pi_S$$  \hspace{1cm} (8)

But for the statistical deviation of the noise eigenvalues from $\sigma^2$, we obtain:

$$\hat{R}_X \approx \hat{R}_Y + \sigma^2(I)$$  \hspace{1cm} (9)

Substitution in (5) results in a first loglikelihood approximation

$$L_0(\delta) = \text{Tr}[\Pi_N \hat{R}_Y + (N-M)\sigma^2]$$  \hspace{1cm} (10)

This expression fits an unifying presentation of bearing estimators recently proposed by Viberg[3], by minimization of

$$\text{Tr}[\Pi_N \hat{E}_S \hat{W} \hat{E}_S^\dagger]$$, where, in (10), $\hat{W} = \hat{A}_S - \sigma^2$. For high SNR, this is very close to the "optimal" weighting matrix derived by Viberg:

$$\hat{W}_{\text{opt}} = \left(\hat{A}_S - \sigma^2\right)(1 - \sigma^2 \hat{A}_S^4)$$  \hspace{1cm} (11)

2.1.4 Approximate Maximum Likelihood Method (AMMLM)

In [2], we demonstrate that (10) could again be locally approximated by:

$$\text{Tr}[\Pi_N \hat{R}_Y] = \text{Tr}[\Pi_N \hat{R} Y] = \text{Tr}[\Pi_N \hat{S} P \hat{S}^\dagger]$$  \hspace{1cm} (12)

In this equation $P$ will be replaced by an estimate obtained recursively from the last $\hat{S}$ and from observations, according to:

$$\hat{P} = (S^\dagger S)^{-1} S^\dagger \hat{R} Y S (S^\dagger S)^{-1}$$  \hspace{1cm} (13)

2.1.5 Corresponding Time Recursive Algorithms

The three previous methods are asymptotically efficient, with the Hessian at the convergence point equal to the Fisher matrix. From simulations it appears that algorithms taking advantage of EVD may be more robust at low SNR than those without EVD. They are very similar concerning gradient and Hessian computation.

We address the problem of real time localization with sources position update every T snapshots. In a stationary context, one could update the value of $R_X$ integrated over all the past and perform the corresponding ML estimation using a Newton Gauss algorithm with starting positions provided by the previous estimation. This was done in [5]. We have also proposed a recursive update, the advantage of which is to be extendible to the moving sources case [4, 6]. It may be interpreted as a Kalman filter generalisation [6].

The remaining problem is in initialization and sources number selection.

2.2. First Bearing initialization

Roughly speaking sources should be initialized within one beamwidth from their true location. As pointed out in the introduction they don't need being resolved at this step. For decorrelated sources, a good initialization point could be obtained by MUSIC. In our simulations, we have preferred conventional beamforming, which proves to be very robust, even for correlated sources and for a very small snapshots number.

We also need an initial guess for the bearings covariance. We use a diagonal matrix with standard deviation equal to about one quarter beamwidth. This value is not critical.

2.3 Recursive sources detection

2.3.1. Principle

Along the same idea than in [7], when M sources have been identified with there estimated position $\hat{G}$, we construct the corresponding projector $\Pi_N(\hat{G})$ on the noise subspace, so that the projection

$$\tilde{Z}(t) = \Pi_N(\hat{G}) . \tilde{X}(t)$$  \hspace{1cm} (14)

should be residual noise only. The first point is then to
perform a "sphericity test " on $\Sigma_{n}$ based on $\Pi_{n}(\theta) \cdot R_{x} \cdot \Pi_{n}(\theta)$ eigenvalues. If the noise only hypothesis is false, $M$ is incremented.

When a new source is detected, the second point will be to provide a correct initialization for the sources (position and covariance): this is obtained by a modified beamforming method in the noise subspace.

### 2.3.2. Projection Sphericity Test

This is reduced to a test of "equality" of the non zero eigenvalues of $\Pi_{n}(\theta) \cdot R_{x} \cdot \Pi_{n}(\theta)$. This is a very classical topic. We use a new criterion, giving a better insight to the problem.

Due to statistical fluctuations, even under the noise only hypothesis the eigenvalues are not equal. When sorted in order of non increasing values, they approximately fit an exponential law, with a decay rate which can be evaluated, using the first and second order of the distribution [8]. The next figure illustrates this point. The natural eigenvalues logarithm is reported on the graph. Of course, one source can be detected in the upper graph.

![Graph showing eigenvalues logarithm for detection]

Our test is based on detection of any decay rate departure to the theoretical law [8]. When detection occurs, we also record the eigenvectors $U_{i}$ corresponding to the maximum eigenvalue of $R_{x}$, which will be used for source bearing initialization.

### 2.3.3 New Source Initialization

Let $U(\theta) = \Pi_{n}(\theta) \cdot \text{norm}(\Pi_{n}(\theta))$ be our modified steering vector. The new source bearing is located by a "modified beamforming" method as the maximum of $f(\theta) = |U(\theta) \cdot U_{i}|^{2}$. See [5] for more detail. The recursive estimation performance is now illustrated. The following simulations show the good behavior of our algorithm in difficult cases which, to our knowledge can't be resolved by existing methods. An uniform linear array of five elements with half wavelength is used. The beamwidth is $25^\circ$. In fig (a), 3 non equipowered signals with 100% correlation arrive from $-30^\circ, 55^\circ, 10^\circ$ relative to the array broadside. At each iteration only 5 snapshots are used.

![Graph showing initialization and sources number update]

### 3 TRACKING OF MANOEUVERING TARGETS

#### 3.1 Introduction

In many practical problems of tracking a maneuvering target, a simple kinematic model can fairly accurately describe the target dynamic characteristics for a wide class of maneuvers. However, since the target can exhibit a wide range of dynamics characteristics, no fixed Kalman filter can be matched to estimate, to the required accuracy, the states of the target for every specific maneuver. For a complete solution of the problem it is necessary to detect the instant of initiation or a change of the maneuver and estimate the proper value of the Kalman filter process noise covariance matrix. A simplified target tracking scheme based on an adaptive Kalman filter is presented, it includes an estimation on line of sources speed and an original adaptive estimation of the modelling noise. The scheme is suitable for tracking after targets which may have substantial different dynamic characteristics. The effectiveness of the proposed filter is demonstrated by extensive computer simulation.

Dealing with maneuvering targets implies a Kalman like recursive algorithm with adaptive estimation of the targets dynamics (see 3.2). Possible source disappearing will be detected by estimated power monitoring (see 3.3). Contrary to stationary case, new source detection will use residual noise correlation smoothing on a finite sliding time interval and not on all the past.

#### 3.2 Recursive algorithm

Assume that evolution model is given by:

$$\bar{\Theta}_{t+1} = \bar{\Theta}_{t} + \bar{U}_{t} + \bar{E}_{t} \quad (15)$$

$\bar{E}_{t}$ account for "fast" decorrelated position increments (random walk) and $\bar{U}_{t}$ for long term average motion (see [5] for more details).

$\bar{U}_{t}$ will be estimated by prediction from past observations by smoothing $\bar{\Theta}_{t} - \bar{\Theta}_{t-1}$. The estimation of $R = E(\epsilon \epsilon^{T})$ needs some particular attention as we'll see now.

The loglikelihood is approximatively quadratic in $\Theta$

$$L(\Theta) = \frac{1}{2} \left( \Theta - \Theta_{ML} \right)^{T} \cdot H \cdot (\Theta - \Theta_{ML}) + C^{T} \epsilon \quad (16)$$

$H$ is the Hessian. The gradient in $\Theta$ is given by:

$$\nabla = H \cdot (\Theta - \Theta_{ML}) \quad (17)$$

At iteration $t$, the a priori estimate is $\bar{\Theta}_{t}$.
\[ \text{cov}(H^t \nabla t) = \text{cov}(\Theta - \Theta_{ML}) = \left( \text{cov}(\Theta) + \text{cov}(\Theta_{ML}) \right) = (R_t + \Gamma_t + H^{-1}) \] (18)

\[ \hat{R}_t = R_{t-1} + \mu H^t \nabla t \nabla^t_t H_t \Gamma_t H^t (19) \]

\[ \text{fig(5)} \]

5 CONCLUSIONS

We have developed a new detection/bearing estimation algorithm which gives good results in difficult non-stationary contexts. Sources can exhibit a wide range of dynamic characteristics.

REFERENCES

EFFICIENT IMPLEMENTATION OF NONLINEAR VOLterra SYSTEMS VIA LAGUERRE ORTHOGONAL FUNCTIONS

S. A. Alshebili, B. G. Mertzios, and A. N. Venetsanopoulos

1Department of Electrical Engineering, King Saud University, Riyadh, Saudi Arabia
Department of Electrical Engineering, University of Toronto, Toronto, Canada M5S 1A4

ABSTRACT: In this paper we present an efficient implementation structure for nonlinear Volterra-type systems. The implementation structure that we propose is based on orthogonal expansion of kernels of the nonlinear system using discrete Laguerre orthogonal functions. The resulting expansion permits the use of current implementation structures available in the literature, with a substantial reduction in the dimensionality of kernels. To illustrate the idea, two implementation structures are discussed, namely the direct form implementation and the matrix decomposition-based implementation.

I. Introduction

In recognizing the important role played by the discrete nonlinear Volterra-type systems in solving many problems of digital signal processing and the pervasive advances in microelectronic technology, several authors [1-7] proposed different realizations/implementations for the discrete Volterra series taking into account the great opportunities offered by VLSI technology.

The appearance of VLSI has resulted in enormous possibilities for the implementation of sophisticated algorithms of high complexity, in a variety of important applications, by using low-cost, very efficient, high-speed special purpose hardware. The emphasis on minimizing the number of dynamic elements (registers, adders, multipliers, etc.) of a single device has been replaced by a new set of criteria based on considering modularity and regularity of structures and reducing the data throughput delay.

In this paper we study the implementation problem of nonlinear systems represented by Volterra-type models. The implementation structure that we propose is based on orthogonal expansion of the kernels of the nonlinear system using discrete Laguerre orthogonal functions (DLOF) given by Goulil [8]. This type of expansion, however, resembles Wiener’s expansion [9] proposed in the continuous case.

The main features motivated the use of DLOF over other orthogonal functions are their time and frequency domain properties. In the time domain, the p-order Laguerre function \( L_p(n) \) is zero for \( n < 0 \), and hence it is possible to construct a causal linear time-invariant (LTI) system with unit impulse response equal to \( L_p(n) \). In the frequency domain the transfer function of \( L_p(n) \) takes the form of a cascade connection of a first-order low-pass system with \( p \) identical first-order all-pass systems [10]. The impulse responses produced at the output of the cascade chain are simply the set of DOLF \( L_0(n), L_1(n), \ldots, L_p(n) \).

In the next section we introduce Laguerre representation of Volterra kernels, and in section III we use this representation to model an \( s \)-th-order Volterra operator. The direct and matrix decomposition implementation structures of the expanded operator are also presented and discussed.

II. Laguerre Representation of Volterra Kernels

The input-output relationship of a discrete \( s \)-th-order Volterra system is given by

\[
y(n) = h_0 + \sum_{i} h_i(\tau_1, \ldots, \tau_i)z(n - \tau_1) \cdots z(n - \tau_i) = h_0 + \sum_{i} H_i[z(n)] \tag{1}
\]

where

\[
y_i(n) = H_i[x(n)] = \sum_{\tau_1, \ldots, \tau_i} h_i(\tau_1, \ldots, \tau_i)z(n - \tau_1) \cdots z(n - \tau_i) \tag{2}
\]

The symbol \( H_i \) is called an \( s \)-th-order Volterra operator,
while the functions $h(\tau_1, \tau_2, \ldots, \tau_i)$ are referred to as the Volterra kernels of the system. The kernels are bounded and discrete in each $\tau_i$, symmetric functions of their arguments, and, for causal systems, $h(\tau_1, \tau_2, \ldots, \tau_i) = 0$ for any $\tau_i < 0$.

In this section we introduce Laguerre representation of Volterra kernels. Generally, any arbitrary kernel of finite energy can be expanded into an infinite series of orthogonal functions defined over a suitable domain. The coefficients of this expansion then become information bearing quantities. Since we are considering discrete nonlinear systems with finite energy kernels, we expand the kernels of the system in a series of DLOF.

The discrete Laguerre function of order $p$ is given by [8,11]

$$I_p(n) = (-1)^p \left( \frac{1-\beta}{\beta-e^{-\beta}} \right)^{1/2} \phi_p(n)$$  \hspace{1cm} (3)

In (3) $\phi_p(n), p = 0, 1, 2, \ldots$ are the discrete orthogonal Laguerre polynomials (DOLP) defined by

$$\phi_p(n) = \beta^p \sum_{r=0}^{n} \binom{p}{r} \binom{n}{r}$$ \hspace{1cm} (4)

where $\beta = \exp(-\mu)$ $(\mu$ is a positive real number), and $\binom{p}{r}$ is the binomial coefficient.

The functions $I_p(n)$ are orthogonal and satisfy the relation

$$\sum_{n=0}^{\infty} I_i(n) I_j(n) = \delta(i,j)$$  \hspace{1cm} (5)

where $\delta$ is the Kronecker delta. The parameter $\beta \in (0,1)$ is called the discount factor. The orthonormal functions (3) form a complete set of basis functions in a space which has an infinite number of dimensions. The elements $a(p_1, p_2, \ldots, p_i)$, which constitute the 'Laguerre spectrum' are obtained by the Laguerre series as follows

$$a(p_1, p_2, \ldots, p_i) = \sum_{n=0}^{\infty} h(\tau_1, \ldots, \tau_i) l_p(\tau_i) \cdot \cdots \cdot l_p(\tau_1)$$  \hspace{1cm} (6)

The element of a given causal kernel $\{h(\tau_1, \tau_2, \ldots, \tau_i)\}$ can in turn be expressed in terms of the discrete functions $I_p(n)$ by way of inverse Laguerre series

$$h(\tau_1, \tau_2, \ldots, \tau_i) = \sum_{p_1, p_2, \ldots, p_i=0}^{\infty} a(p_1, p_2, \ldots, p_i) I_p(\tau_i) \cdots \cdots I_p(\tau_2) \cdot I_p(\tau_1)$$ \hspace{1cm} (7)

By analogy with the classical Fourier transform (6) and (7) can be viewed as a transform pair.

In practice the first few terms of the series of orthogonal functions often suffice to yield a very good approximation to the desired response. That is,

$$h(\tau_1, \tau_2, \ldots, \tau_i) = \sum_{p_1, p_2, \ldots, p_i=0}^{N-1} a(p_1, p_2, \ldots, p_i) I_p(\tau_i) \cdots \cdots I_p(\tau_2) \cdot I_p(\tau_1)$$ \hspace{1cm} (8)

where we have truncated the infinite sum of (7) to the first 'N' terms. The error introduced by the above truncation can be reduced by optimally selecting the value of the free parameter 'N' of the Laguerre function.

III. Laguerre Representation of Volterra Operators

In this section we introduce Laguerre representation of Volterra operators. Based on the results obtained in the previous section we can express the output $y(n)$ of an $i$th-order Volterra operator $H_i[x(n)]$ in terms of the DLOF as follows

$$y(n) = H_i[x(n)] = \sum_{r_1, r_2, \ldots, r_i=0}^{N-1} \sum_{p_1, p_2, \ldots, p_i=0}^{N-1} a(p_1, p_2, \ldots, p_i) I_p(\tau_i) \cdots \cdots I_p(\tau_2) \cdot I_p(\tau_1) x(n-r_i) \cdots x(n-r_2) x(n-r_1)$$ \hspace{1cm} (9)

where '*' denotes convolution operation. If we set $s_k(n) = l_k(\tau) \ast x(n)$, (9) can be written in the following form

$$y(n) = \sum_{p_1, p_2, \ldots, p_i=0}^{N-1} a(p_1, p_2, \ldots, p_i) s_{p_1}(n) \cdots s_{p_i}(n)$$ \hspace{1cm} (10)

Remark: Equation (10) resembles (2). Therefore any technique used for the implementation of (2) can be also used for the implementation of (10). The main advantage of using (10) over (2) is that by expanding the Volterra kernels using orthogonal functions the dimensionality of kernels is considerably reduced.

Two different implementation structures which are based on Laguerre series expansion of the Volterra kernels are now in order.

Direct Form Implementation

Based on the above remark we expect that the direct implementation of (10) will result in a substantial saving in the number of multipliers compared to the direct implementation of unexpanded kernels, (2). Figure 1 shows
the direct implementation structure of (10). We have used the fact that the z-transform \( L_p(z) \) of the Laguerre function \( I_p(n) \) is given by

\[
L_p(z) = \sqrt{1 - \beta} \left( \frac{z}{z - \sqrt{\beta}} \right)^{1 - \beta} \left( \frac{1}{z - \sqrt{\beta}} \right)^p \quad \text{and} \quad LP(z) \left[ AP(z) \right]^T
\]

(11)

Note that the function \( I_p(n) \) takes the form of a cascade connection of a first-order low-pass system with \( p \) identical first-order all-pass systems. When the input to the system is \( x(n) \) the responses produced at the outputs of the cascade chain are simply the set: \( y_0(n), y_1(n), \ldots, y_p(n) \).

Matrix Decomposition Based Implementation

For a second-order Volterra operator (10) can be written in a matrix form as shown below

\[
y_2(n) = s^T(n)A\sigma(n)
\]

(12)

where \( s^T = (s_0(n), s_1(n), \ldots, s_{N-1}(n)) \) and \( A = \{a[p_1, p_2]\} \) is \( N \times N \) symmetric matrix. Applying the SVD to the symmetric matrix \( A \) may be decomposed exactly into a finite sum of rank one matrices as follows

\[
A = \sum_{j=1}^{q} \lambda_j R_j R_j^T
\]

(13)

where \( q \) is the rank of the matrix, \( \lambda_j \) are real scalars, and \( R_j = (r_j(0), \ldots, r_j(N-1))^T \) are \( M \times 1 \) vectors. Substituting (13) into (12) one can easily obtain

\[
y_2(n) = \sum_{j=1}^{q} \lambda_j s^T(n)R_j[R_j^Ts(n)]
\]

(14)

where

\[
d_j(n) = s^T(n)R_j = \sum_{k=0}^{N-1} r_j(k)s_k(n), \quad j = 1, 2, \ldots, q
\]

(15)

Equation (15) describes \( q \) independent, linear, time-invariant, parallel digital systems, and (14) represents a set of \( q \) parallel square-in-add-out type of operations. The implementation structure of (14) is shown in Figure II.

References


Figure I.

Figure II.
The "Z-slice": an Identification Method for Noisy MA Systems using Higher Order Statistics

A. G. Constantinides, Dept. of Elec. Eng., Imperial College, Exhibition Road, London, UK
E. Fleuret, Sponsored by Electricité de France, DSR/TAI, 6 quai Watier, 78400 Chatou, France

In this paper, we present a new approach for the blind identification of MA systems. The algorithm is based on slices of the Z-transform of the cumulants, and it uses all the information provided by the cumulants. The extraction of the signal samples, at some stage of the algorithm, relies on the solution of a set of non-linear equations, and this is achieved through the use of some novel techniques. The results appear superior to other methods.

Introduction

The identification of systems only from output measurements is an important task in many areas. And it is normally the case that the unknown systems are considered as AR, MA or ARMA models. The parameters of these models are almost invariably determined by employing some estimate of the covariance matrix of the output signal. In [1], Mendel stressed that the estimation of an ARMA model could be divided in two steps: an AR estimation and a MA estimation. Thus, the subset of methods for blind estimation of MA models covers a broad area of applications. Several methods using Higher Order Statistics (HOS) have already been proposed to solve the issue. Some techniques in [1, 2, 3] are based on recursion, but their main drawback is the propagation of errors. Others such as those in [4, 5, 6] are based on linear algebra, and provide more robust solutions. However they depend on the estimation of second order moments which are of course sensitive to additive noise.

This paper presents a new approach, "the Z Slice Method", to determine the parameters of a MA-system from the observed noise degraded signal. The algorithm makes use of HOS in order to benefit from the following properties: 1/ to preserve the true phase of the signal spectrum, 2/ to reduce the influence of additive Gaussian noise. From these, a good estimate of the MA parameters is obtained. In this paper the problem and its resolution are presented and compared with previous methods.

1 Presentation of the Z-slice method

In this paper third order statistics are used but the method can be easily generalised to higher orders. The signals are generated according to figure (fig. 1). The output $y$ of a finite-dimensional MA model satisfies the equation (1) where $u$ is assumed to be non-Gaussian, i.i.d., zero-mean with $E(u^2(k)) = \sigma^2$, $E(u^3(k)) = \beta$. The orders $p$ (causal), $q$ (anticausal) are assumed to be known.

$$y(n) = \sum_{k=0}^q h_k u(n-k), \quad v(n) = y(n) + w(n) \quad (1)$$

The signal $y$ is contaminated by additive Gaussian noise $w$. Our aim is to find the parameters $h_k$ of the MA-system from the observed zero-mean process $v$.

Input $u$ | MA model | $+$ noise | Output $v$

![Figure 1: General schema of the signal generation](image)

The Gaussian noise is removed or, at any rate, its influence reduced by taking the third order cumulant $cum_3(k, l)$ of $v$, in which case, in theory, $cum_3(k, l)$ are also the cumulants of $y$. Reference [7] gives all the appropriate definitions of the cumulants, in our case the third order cumulants of $v$ is $cum_3(k, l) = E[v(n)y(n+l)v(n+k)]$ [8]. But the cumulants can also be expressed by using the impulse response (or the parameters) of MA models such as in (2) [1]. From the parameters point of view the cumulants contain considerable redundancies.

$$cum_3(k, l) = \sum_{n=0}^N h_n h_{n+k} h_{n+l} \quad (2)$$

$N = Max(q, k - q, q - l)$ ; $M = Min(p, p - k, p - l)$

That is the reason why earlier methods to identify MA model with HOS require only the computation of the cumulant along specific lines. In so doing, the Swami and Mendel recursion (SWA) in [2] or the $C(k, q)$ method in [1] are not exploiting the redundancy of information in third order cumulant plane. The GM algorithm [4] and its adaptations T1 [5], T2 [6] partly incorporate redundant information for increased robustness, but they still employ autocorrelation values.

Our aim is to build an approach which guarantees...
good robustness to the estimation error, and which also makes good use of all the cumulant values. To achieve that, we take the Z-transform \( C_3 \) of \( \text{cum}_3(k,l) \) which can be expressed in terms of the Z-transform \( H \) of the impulse response of the MA process as [1] (the Z-space version of equation (2)).

\[
C_3(x_1, x_2) = \beta H(x_1)H(x_2)H(x_1^{-1}x_2^{-1})
\]

(3)

Consider the slice \( S_3 \) of \( C_3 \) along \( z = x_1 = x_2 \). Then equation (3) leads to (4).

\[
S_3(z) = \sum_{j=-2(p-q)}^{2(p-q)} P_j(\tilde{h}) z^{-j} = \sum_{j=-2(q-n)}^{2(q-n)} g_j z^{-j}
\]

(4)

\( P_j() \) are \((p-q+1)\)-variable polynomials of order 3, and \( g_j \) correspond to the summation of the cumulants along the lines \( k + l = j \). By a simple process of coefficient identification between each of \( P_j() \) and \( g_j \), we obtain \((p-q+1)\) non-linear equations. Now the blind identification issue consists of finding an accurate solution for \( \tilde{h} \) from the set of these non-linear equations.

2 New Recursive Solution

Despite the lack of accuracy of recursive solutions due to the error propagation, our first method uses alternatively equations (5) and (6) below to obtain a recursive solution \( \tilde{h}_R \). The recursion can be initialized by assuming the value \( h_0 \) to be known a priori or by calculating \( h_0 \) and \( h_p \) from (5) and (6) with \( n = 0 \). Equations (5) and (6) are obtained by multiplying \( S_3(z) \) and \( S_3(z^{-1}) \) by \( z^{-2\theta-n} \), and taking the \( n^{th} \) derivative at \( z = 0 \).

\[2h_p h_{p+k} = g_{2(p-q)-n} - \sum_{i+j=2k} h_{p-i}h_{p-j}h_{p+k} \quad \text{(5)}\]

\[2h_p h_{p+k} = g_{2(p-q)-n} - \sum_{i+j=2k} h_{p-i}h_{p-j}h_{p+k} \quad \text{(6)}\]

\( i, j \geq 0 \) \( k > 0 \)

3 Optimisation Solutions

A second approach to the solution is to obtain a solution for \( \tilde{h}_M \) as that vector that minimises \( \psi(\tilde{h}) = \sum (P_j(\tilde{h}) - g_j)^2 \). This can conceivably be achieved by using classical mathematical tools. But in view of the third order nonlinearity one cannot guarantee, in this case, that \( \tilde{h}_M \) provides the global minimum. This is so, even if we start the solution with an initial guess from the recursive solution \( \tilde{h}_R \) to make the procedure more robust. As there is still an uncertainty about the confidence one can attach to the solution \( \tilde{h}_M \), an alternative method has been deployed relying on “The Lagrange Programming Neural Network” (LPNN) [9]. The problem is now formulated as “to minimise \( f(\tilde{h}) \) under some constraints \( P_j(\tilde{h}) - g_j = 0 \).”

This could be any appropriate function but for our problem here, we have taken \( f(\tilde{h}) = \sum (P_j(\tilde{h}) - g_j)^2 \). Thus, the Lagrange function \( L \) and the set of dynamic equations below are obtained

\[
L(\tilde{h}, \tilde{\lambda}) = f(\tilde{h}) + \sum_i \lambda_i (P_i(\tilde{h}) - g_i) \quad \text{(7)}
\]

\[
\frac{d\tilde{h}}{dt} = -\nabla_h L(\tilde{h}, \tilde{\lambda}) \quad \text{and} \quad \frac{d\tilde{\lambda}}{dt} = \nabla_\lambda L(\tilde{h}, \tilde{\lambda}) \quad \text{(8)}
\]

One can distribute \((P_i(\tilde{h}) - g_i)\) either in \( f(\tilde{h}) \), in the set of constraints or both. According to [9], if we consider the Lagrange function (7), the equilibrium of (8) is reached for the global solution \( \tilde{h}_L \) when the Hessian of \( L \) is always positive definite.

4 Results

All the presented tests have been performed for length of signal of 4096 (32*128) samples. The cumulants have been computed according to the algorithm given by Nikias in [10]. Two example models are used model 1 (mixed phased) \( 1.0 - 0.8z^{-1} + 0.2z^{-2} - 0.12z^{-3} + 0.2z^{-4} \) and model 2 (minimum phase) \( 1.0 - 0.85z^{-1} + 0.175z^{-2} \). The phase rather than the amplitude has been plotted on figure 2 because its variations are more visible.

If fast computation is required, the recursive solution \( \tilde{h}_R \) offers a good estimate of \( \tilde{h} \) when the orders \( (p,q) \) of the model are small or the length of the data is large. At any rate, \( \tilde{h}_R \) is a good initial guess for the optimisation based methods which then obtain better estimates of \( \tilde{h}_M \), \( \tilde{h}_L \) is computed by a Quasi-Newton Method. It gives accurate results as those in figure 2 and in tables 1, 2. Moreover, one has several ways for evaluating the accuracy of \( \tilde{h}_M \). As \( \tilde{h}_R \) is not fixed, its closeness to 1.0 (this value is assumed by others methods) indicates the goodness of \( \tilde{h}_M \). The value of \( \psi(\tilde{h}_M) \) and the cumulant error (the distance between the estimated cumulants and the reconstructed cumulants from \( \tilde{h}_M \)) also indicate the confidence one can attach \( \tilde{h}_M \).

In practice, (8) is computed by the Runge-Kutta algorithm. In the LPNN method, the step-size of this algorithm slows down considerably the computation of \( \tilde{h}_L \). The differences between \( \tilde{h}_M \) and \( \tilde{h}_L \) are due to the introduction of constraints in the LPNN. It has an adverse effect on \( \tilde{h}_L \) in table 2 but a good one in figure 2. It depends mainly on the goodness of the \( g_i \) involved in the constraints.

\( C(k,q) \), SWA and \( \tilde{h}_R \) are fast solutions to our problem. But they need an accurate estimation of the cumulants to achieve useful results. GM, SWA are sensitive to additive Gaussian noise. GM, T1, T2 are based on different set of linear equations involving specific cumulants. When these cumulants are badly estimated \( \tilde{h}_M \), \( \tilde{h}_L \) always give better results specially with additive noise.
<table>
<thead>
<tr>
<th>h</th>
<th>1.000</th>
<th>0.85</th>
<th>0.175</th>
</tr>
</thead>
<tbody>
<tr>
<td>h.R</td>
<td>1.000</td>
<td>0.755</td>
<td>0.224</td>
</tr>
<tr>
<td>3dB</td>
<td>1.000</td>
<td>0.842</td>
<td>0.255</td>
</tr>
<tr>
<td>0dB</td>
<td>1.000</td>
<td>0.907</td>
<td>0.275</td>
</tr>
<tr>
<td>h.M</td>
<td>1.032</td>
<td>0.892</td>
<td>0.191</td>
</tr>
<tr>
<td>3dB</td>
<td>1.050</td>
<td>0.879</td>
<td>0.215</td>
</tr>
<tr>
<td>0dB</td>
<td>1.061</td>
<td>0.850</td>
<td>0.234</td>
</tr>
<tr>
<td>h.L</td>
<td>1.090</td>
<td>0.861</td>
<td>0.174</td>
</tr>
<tr>
<td>3dB</td>
<td>1.046</td>
<td>0.881</td>
<td>0.214</td>
</tr>
<tr>
<td>0dB</td>
<td>1.058</td>
<td>0.852</td>
<td>0.234</td>
</tr>
<tr>
<td>GM</td>
<td>1.000</td>
<td>0.755</td>
<td>0.042</td>
</tr>
<tr>
<td>3dB</td>
<td>1.000</td>
<td>0.881</td>
<td>0.214</td>
</tr>
<tr>
<td>0dB</td>
<td>1.000</td>
<td>0.907</td>
<td>0.160</td>
</tr>
<tr>
<td>T1</td>
<td>1.000</td>
<td>0.562</td>
<td>0.136</td>
</tr>
<tr>
<td>3dB</td>
<td>1.000</td>
<td>0.669</td>
<td>0.108</td>
</tr>
<tr>
<td>0dB</td>
<td>1.000</td>
<td>0.590</td>
<td>0.095</td>
</tr>
<tr>
<td>T2</td>
<td>1.000</td>
<td>0.883</td>
<td>0.241</td>
</tr>
<tr>
<td>3dB</td>
<td>1.000</td>
<td>1.149</td>
<td>0.349</td>
</tr>
<tr>
<td>0dB</td>
<td>1.000</td>
<td>1.306</td>
<td>0.359</td>
</tr>
</tbody>
</table>

Table 1: Sensitivity to additive Gaussian noise of impulse response identified by different approaches

Conclusion

Methods are presented for the identification of MA systems through Higher Order Cumulants using the new approach named "The Z slice". They provide a good solution for the estimation of the parameters of an MA model even with small data lengths, and are robust in the presence of additive gaussian noise. The entire set of cumulants of a specific order are involved in the computation of the solution, which permits the use of all the information known on the model and the solution is therefore less sensitive to estimation errors.

References


Results for SWA

![Results for SWA](image-url)
Figure 2: Phase of model 1 identified by different methods without noise and with SNR = 3dB, 0dB and from the same set of 4096 samples.
THE DISCRETE-TIME RUNNING CROSS-CORRELATION FUNCTION
ESTIMATED FROM THE LAGUERRE SPECTRA

A. C. DEN BRINKER

EE-Department, Eindhoven University of Technology, Eindhoven
P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands

We propose a simple and computationally efficient mechanism to calculate a running or local cross-correlation function of two discrete-time signals. In order to obtain a running cross-correlation function, the signals must be windowed. It is argued that an appropriate window for a local cross-correlation is an exponential sequence. To obtain a computationally efficient mechanism, the windowed functions are decomposed in discrete-time Laguerre series. Cross-correlation of the windowed functions is identical to cross-correlation of the Laguerre coefficients and subsequent multiplication by cross-correlation pattern functions.

1. INTRODUCTION

In many applications of signal processing, running or local measures of certain characteristics of the signal are required, instead of global measures. We consider the problem of constructing a simple mechanism to calculate a running or local cross-correlation function. Such a mechanism can be used in a large number of applications, for instance in adaptive filtering or system identification.

First, we will consider which window is appropriate to construct a running cross-correlation function. Next, we represent the windowed signals as a series of orthogonal functions. In our case this will be a Laguerre series. The coefficients of this series are obtained by filtering the input signals. The local cross-correlation function is subsequently expressed in terms of the orthogonal set of functions. We show that the running cross-correlation function is a weighted summation of cross-correlation pattern functions where the weights are determined as the cross-correlation of the Laguerre coefficients. This method is computationally efficient if the number of coefficients of the Laguerre series that is required to represent the signal accurately is small. To this end the window function and the set of orthonormal functions are matched.

2. THE WINDOW FUNCTION

In the case of time-domain signals, taking a running or local measure as a meaningful quantity is based on the following two assumptions. First, running measures prohibit the use of future signal values, i.e., we only know the signal at the current moment \( n \) and its past. Second, the signal in the distant past of \( n \) should not, or at most only weakly, determine our running measure at \( n \). Thus, in order to obtain a local cross-correlation, we have to window the signals, i.e., we multiply the signals, say \( x(m) \) and \( y(m) \), by \( w(n - m) \). The function \( w(m) \) is called the window function.

Since we may not use information of future signal values, we set \( w(m) = 0 \) for \( m < 0 \). Furthermore, we require that the shape of the windowed function \( x(m)w(n - m) \) is independent of the `current moment' \( n \). This is a necessary requirement since cross-correlation involves a comparison of the shape of the signals. The only window satisfying this requirement of constancy of the shape of the windowed function is given by an exponential sequence: \( w(m) = w(0)\xi^m \). Without loss of generality we can take \( w(0) = 1 \). Since we want to consider real-valued windows and to take past events less into account we take a real scaling parameter and \( 0 < \xi < 1 \). To summarize, we use the window function \( w(m) \) given by \( w(m) = \xi^m c(m) \), where \( c(m) \) is the unit step function.

The suitable range of the scaling parameter \( \xi \) involves a compromise. On the one hand we want to take \( \xi \) as close as possible to unity in order to get an accurate estimate of the cross-correlation function, while on the other hand for \( \xi \to 1 \) the effect of windowing is lost.

Consider now the windowed function \( \hat{x} \) according to \( \hat{x}(n; m) = x(m)w(n - m) \). The local cross-correlation function \( R_{xy}(n; l) \) is simply defined as the cross-correlation of the windowed signals \( \hat{x} \) and \( \hat{y} \)

\[
R_{xy}(n; l) = \sum_{m=-\infty}^{\infty} \hat{x}(n; m)\hat{y}(n; m + l).
\] (1)
3. THE LAGUERRE TRANSFORM

Instead of directly calculating the cross-correlation of the windowed signals, these signals are transformed first to the Laguerre domain. Next, we will show that the cross-correlation results in a simple expression with respect to the Laguerre coefficients. Our approach links up with the renewed interest in the possibilities of signal descriptions by orthogonal polynomials in both system identification (e.g., [1]-[6]) and coding [7].

We write a windowed function as a discrete-time Laguerre series:

\[ u(n; m) = \sum_{k=0}^{\infty} g_{uk}(n) \phi_k(n - m). \]  

(2)

\( u \) stands for both \( x \) and \( y \), \( g_{uk} \) are the Laguerre coefficients or the Laguerre spectrum, and \( \phi_k(m) \) are the pattern functions. One may consider this as a transform: \( u(n; m) \to g_{uk}(n) \) which is called a forward polynomial transform. The Laguerre spectrum depends on \( n \) and thus constitutes a running decomposition. Since the transform involves Laguerre polynomials, we speak about a Laguerre transform. The pattern functions are defined by

\[ \phi_k(m) = (-1)^k \nu_k \gamma_k(m) \theta^{m/2}, \]  

(3)

where \( \nu_k = \sqrt{(1 - \theta)} / \theta^k \) and \( \gamma_k(m) \) is the discrete Laguerre polynomial [8] defined by

\[ \gamma_k(m) = \theta^{-m} \Delta^k \left( \begin{array}{c} m \\ k \end{array} \right) \theta^m \epsilon(m), \]  

(4)

with \( k \) the order of the polynomial, \( \Delta \) the forward difference operator, and \( 0 < \theta < 1 \). The pattern functions \( \phi_k(m) \) constitute a complete orthonormal set of functions on the interval \([0, \infty)\):

\[ \sum_{m=0}^{\infty} \phi_k(m) \phi_{\nu}(m) = \delta_{k,\nu}, \]  

(5)

where \( \delta_{k,\nu} \) is the Kronecker delta. Assuming that \( u(n) \) is a bounded signal, the windowed signal \( u(n; m) \) is square-summable and the proposed decomposition (2) exists.

The first three pattern functions for \( \theta = 0.9 \) are shown in Fig. 1. The number of zero-crossings in the pattern functions is equal to the index \( k \), and so, higher-order pattern functions are increasingly oscillatory.

Taking \( \theta = \xi^2 \) amounts to matching the signal decomposition on the window function. The Laguerre coefficients can be easily derived from the original signal \( u(n) \) via convolution \( g_{uk}(n) = u(n) * d_k(n) \), where \( d_k(n) \) are the impulse responses of the decomposition filters according to \( d_k(n) = \phi_k(n) u(n) \).

We denote the Z-transforms of decomposition filters and pattern functions by \( \Phi_k(z) \) and \( D_k(z) \), respectively. We find

\[ \Phi_k(z) = \sqrt{1 - \xi^2} \frac{z}{z - \xi} \left( \frac{1 - \xi z}{z - \xi} \right)^k, \]  

(6)

Figure 1: The pattern functions \( \phi_k(m) \) for \( \theta = 0.9 \) and \( k = 0, 1, 2 \) (as indicated in the plot).

\[ D_k(z) = \sqrt{1 - \theta} \frac{z}{z - \theta} \left( \frac{\sqrt{\theta}(1 - z)}{z - \theta} \right)^k. \]  

(7)

In Fig. 2 an efficient filtering scheme to obtain the coefficients \( g_{uk} \) from the original signal is shown. The set of decomposition filters encompasses the effect of the windowing operation as well as the Laguerre transform.

\[ u(n) \xrightarrow{\text{filter bank}} g_{ul}(n) \xrightarrow{g_{ul}(n)} g_{ul}(n) \xrightarrow{g_{ul}(n)} \cdots \]  

Figure 2: Filter bank to derive the running Laguerre coefficients \( g_{ul}(n) \) from the original signal \( u(n) \).

4. THE LOCAL CROSS-CORRELATION FUNCTION

We now proceed to express the local cross-correlation function \( R_{xy}(n; m) \) in terms of the Laguerre coefficients of the windowed signals. By substituting the Laguerre series for the windowed signals in (1) we obtain

\[ R_{xy}(n; l) = \sum_{k=0}^{\infty} \sum_{m=0}^{\infty} g_{uk}(n) g_{yl}(m) \hat{g}_{uk; m}(l), \]  

(8)

where \( \hat{g}_{uk; m}(l) = Z^{-1} \{ \hat{g}_{uk; m}(z) \} \) (\( Z^{-1} \) denotes the inverse Z-transform) and \( \hat{Q}_{ly}(z) = \Phi_k(z) \Phi_m(z)^{-1} \). Substituting the Z-transforms (6), we find that the function \( \hat{Q}_{ly}(z) \) depends only on the difference \( k - m \)

\[ \hat{Q}_{ly}(z) = \hat{Q}_{ly}(z; k-m) = \frac{1 - \theta}{(z - \xi)(z^{-1} - \xi)} \left( \frac{1 - \xi z}{z - \xi} \right)^{k-m}. \]  

(9)

We define \( \hat{g}_k(l) \) as the inverse Z-transform of \( \hat{Q}_k(z) \). The local cross-correlation function can now be expressed as
a summation of weighted local cross-correlation pattern functions $q_k$ according to

$$R_{xy}(n; l) = \sum_{k=-\infty}^{\infty} c_k(n) q_k(l). \quad (10)$$

The weighting factors $c_k$ which are called the cross-correlation coefficients are given by

$$c_k(n) = \sum_{m=-\infty}^{\infty} g_{x,k+m}(n) g_{y,m}(n), \quad (11)$$

where the Laguerre spectra $g_{x,m}$, $g_{y,m}$ for negative $m$ are defined as $g_{x,m} = g_{y,m} = 0$. We find that the local cross-correlation function is a weighted summation of cross-correlation pattern functions (10). The weights are calculated by cross-correlating the Laguerre spectra of the windowed signals (11). The advantage of calculating $R_{xy}(n; l)$ by (10) instead of a direct cross-correlation of the windowed functions, is its computational simplicity in the case the Laguerre series has a fast convergence. To that end, we already matched the scale parameter of the window function and that of the Laguerre series.

From the Z-transforms of the cross-correlation pattern functions we see that $Q_x(z) = Q_{-a}(1/z)$ and thus that $q_x(l) = q_x(-l)$. The function $q_x(l)$ has a simple explicit expression by $q_0(l) = \xi^l$, and for positive $k$ the cross-correlation pattern function can be expressed as the sum of two pattern functions $\phi$:

$$q_x(l) = \frac{1}{\sqrt{1-\xi^2}} [\phi_k(l) + \xi \phi_{k-1}(l)]. \quad (12)$$

Three cross-correlation pattern functions are shown in Fig. 3 for $\theta = 0.9$ and $k = 0, 1, 2$.

5. DISCUSSION

We have given a formal definition of the discrete-time local cross-correlation function. We argued that the best choice for a window function is an exponential window, since cross-correlation involves a comparison of the shape of two signals and the shape of a windowed signal is independent of its location in the case of an exponential window. Furthermore, we have shown that the local cross-correlation function can be simply computed from the Laguerre coefficients of the windowed signals.

The definition of the local cross-correlation function and its method of calculation immediately raises two questions. First, in which cases is the proposed local cross-correlation function a meaningful quantity? Second, when is the proposed method of calculation efficient?

The local character ensures that the given definition is meaningful not only for stationary processes, but also for quasi-stationary processes. However, in that case the time scale associated with changes in the stochastics must be much larger than that associated with the signals themselves. In practical situations this will not be a very restrictive requirement.

The proposed method of calculation is efficient if the number of Laguerre coefficients needed to represent the signals accurately is small. This is the case if the time scale $1/(1-\xi)$ associated with the parameter $\xi$ is matched to that of the signal content of $x$ and $y$ (or more precisely, to the parts of $x$ and $y$ that are correlated). The choice for $\xi$ is in essence the question on which time scale we want to view the signals, i.e., which scale provides the best representation of the characteristics that are of interest. At the same time, it is a question of computational simplicity: the best scale is that which provides a minimum of significant coefficients of the Laguerre transform.

A restriction that follows from a specific choice of $\xi$ is that the proposed local cross-correlation function is only meaningful if the time scale of the cross-correlation $E[x(n)y(n-l)]$ is also within the scale $1/(1-\xi)$. This is obvious from the fact that all cross-correlation pattern functions are on this scale. For instance, if $x$ and $y$ are delayed replicas with delay $L$ and if $\xi^L \ll 1$, we will not be able to estimate this delay reliably from the local cross-correlation function.

To summarize, there are three time scales, associated with the change in stochastics, the signals $x$ and $y$, and the cross-correlation function, respectively. On the other hand we have only one free parameter ($\xi$) in the definition of the local cross-correlation function. The proposed definition and its method of calculation is meaningful and efficient under the following three requirements. The time scale associated with the change in stochastics is much larger than that associated with the signals. The time scale associated with the cross-correlation function is smaller than or equal to that associated with the signals. The parameter $\xi$ is matched to the time scale of the signals.
We feel that in many areas a formal definition of a local cross-correlation function and an easy assessment thereof may be a useful tool. In particular we are thinking of applications in the field of system identification and adaptive filtering [9].

REFERENCES

Some Robust Stability Results Concerning Sparse Predictors

D. Docampo*, F. Pérez,* J.R. Fernández** and C. Abdallah†

* BSP Group, ** Chronobiology Group
Departamento de Tecnología de las Comunicaciones
ETSI Telecomunicación, Universidad de Vigo, 36200-Vigo, Spain

† BSP Group, Department of Electrical and Computer Engineering
University of New Mexico, Albuquerque, NM 87131, USA

This paper applies new extreme-point robust stability results, to the study of the stationarity (stability) of sparse AR processes. Point estimates of the coefficients of those sparse predictors are uncertain if the inaccuracies introduced by the variance of the estimators are considered. Making use of bootstrap techniques, non-parametric confidence intervals are obtained around the nominal coefficients (point estimators) and then robust stability properties of the All-Pole Filters associated with the sparse AR models are discussed.

1 Introduction

Since the seminal work of Kharitonov [1] concerning the stability of continuous-time uncertain systems, the stability of polynomials with structured perturbations has become a very active area of research. In recent years, the discrete-time counterpart has also commended a great deal of research effort. It is now known that a strictly strong Kharitonov-like result does not exist in discrete time. However, weak or partial results are available in some special cases, see for example [2] and [3]. Such cases allow us to handle perturbations affecting only half of the coefficients and the case of symmetric or antisymmetric polynomials.

The first objective of this paper is to present the results obtained previously by some of the authors, see [4] and [5] in the discrete-time domain. By expressing a discrete polynomial using its Barycentric coordinates (BC) and using standard results, we were able to generalize some existing discrete-time stability results but more importantly, to introduce and prove more general results. In fact, we were able to show that a Kharitonov-like result holds even though the upper-half of the polynomial coefficients are uncertain. This result will simplify the stability test of a family of uncertain polynomials to that of some combinations of the extremes of the family. Moreover, when the uncertainty under consideration is symmetric or antisymmetric (see section 2), all the coefficients are allowed to vary independently while an extreme-point check for stability is still valid. This implies that a generalised linear-phase type of perturbations may be analyzed with the so called weak-Kharitonov tests.

Once those results have been reviewed, we show how sparse predictors (namely, those having a number of non-zero coefficients lower than the actual length of the predictor) fit into the cases of uncertain filters as previously studied. Examples of such predictors can be AR processes [6] where only a short number of coefficients are non-zero, or long-term predictors used in Linear Predictive Speech Coding. We will use this last example, where assuming stability for the predictor is required, as a showcase of our results. With our robust stability test available, only a small number of polynomials need to be considered in order to study the stationarity of all the possible filters obtained by combining all the possible parameter values falling in some precalculated confidence intervals.

2 Robust Stability Results

An uncertain polynomial may be described as a member of a set of possible polynomials. Then, the stability of the polynomial may be deduced from the stability of the underlying set. The original problem of Kharitonov deals with uncertainties in the continuous-
time case. Specifically, consider the family of polynomials \( P \), where each polynomial can be written as

\[
p(s, q) = q_0 + q_1 s + q_2 s^2 + \cdots + q_n s^n,
\]

where each coefficient \( q_i \) takes values in the range \([q_i^- , q_i^+]\). The variation in each coefficient is assumed independent from all others. The question answered by Kharitonov [1] is to find necessary and sufficient conditions such that all polynomials in \( P \) are strictly Hurwitz (i.e., all the roots lie in the strict left half plane). His theorem solves this problem with the following simple test. Given the four polynomials

\[
P_1(s) = q_0^- s + q_1^- s^2 + q_2^- s^3 + q_3^- s^4 + \cdots
\]

\[
P_2(s) = q_0^+ s + q_1^+ s^2 + q_2^+ s^3 + q_3^+ s^4 + \cdots
\]

\[
P_3(s) = q_0^- s^2 + q_1^- s^3 + q_2^- s^4 + \cdots
\]

\[
P_4(s) = q_0^+ s^2 + q_1^+ s^3 + q_2^+ s^4 + \cdots
\]

then the stability of these four polynomials is necessary and sufficient to guarantee the stability of the family \( P \). Moreover, when the degree of the polynomials involved is less than 4, then a reduced stability test is possible, where fewer than four polynomials are needed. Therefore, Kharitonov's theorem provides necessary and sufficient conditions for the analysis of the stability of a family of continuous time polynomials subject to independent perturbations. Note the important property of the test, that the number of polynomials to check does not depend on the degree of the polynomials in \( P \).

After the results of Kharitonov became widely known, a great effort was expanded trying to adapt them to the discrete-time case. Unfortunately, the extension of Kharitonov's work to discrete-time systems is not straightforward. Actually, it is known that a four-polynomials test does not hold for the discrete-time case, and neither does a test in which the stability of all extreme polynomials (not just a special set of four) is checked. Clearly, this means that for a \( n \)-th order interval polynomial family, the \( 2^n \) combinations of corners can be stiff, but some member of the family is not. This fact was established in some well-known counterexamples; e.g., see [7]. Some partial results are however available for discrete-time polynomials, some of which are discussed below.

In the following, we will deal with discrete-time polynomials with real coefficients. Such polynomials can be written as

\[
P(s) = a_0 s^n + a_1 s^{n-1} + \cdots + a_n.
\]

We will consider first the case pointed out in [2] where only the upper half of the coefficients are allowed to vary. Concretely, we have that

\[
a_i \in [a_i^- , a_i^+], \quad i = \left\lfloor \frac{n+1}{2} \right\rfloor, \ldots, n.
\]

where \( \left\lfloor \frac{n+1}{2} \right\rfloor \) denotes the integer part of \((n + 1)/2\). Then, the stability of the entire family is equivalent to that of the extreme polynomials, obtained by combining all the possible extremes of the coefficients defined before. This result is, of course, weaker than Kharitonov's theorem but, as we will see later, it will be useful in studying the stability of some specific kinds of uncertainties.

Another interesting partial result applies to symmetric or antisymmetric perturbations [3]. In this case, the variation in the coefficients is such that

\[
a_i - a_i^- = a_{n-i} - a_{n-i}^-
\]

or

\[
a_i^+ - a_i = a_{n-i} - a_{n-i}^+
\]

Again, the complete family of polynomials is stable if and only if all the combinations of extreme polynomials are stable. It should be noted that this result contains two important constraints:

1. The coefficients must be coupled as \((a_i, a_{n-i})\). This means that different types of coupling are not allowed nor is a single coefficient allowed to vary on its own.

2. The edges allowed in the pairwise variation defined in 1) have \( \pi/4 \) and \( 3\pi/4 \) slopes in the parameter space.

In [4] and [5], we showed that the previous results can be generalized by removing those restrictions. This was done by using the barycentric coordinates i.e. the bilinear transformation, and exploiting some symmetries found in the transformation matrix. The following theorem was then obtained:

**Theorem 1** Consider a polytope in the coefficient space where each pair \((a_i, a_k)\), for \( 0 \leq i \leq n \) and \( n - i \leq k \leq n \) is varying inside a polygon with edges sloped in the closed interval \([\pi/4, 3\pi/4]\) and where the pairwise variations \((a_i, a_k), k \neq i\) are not allowed simultaneously. Then, every polynomial in the polytope will be stable if and only if all the polynomials obtained by combining all the polygon corners are stable.
Clearly, if \( i = k \) for \( i = \lceil \frac{n+1}{2} \rceil, \cdots n \) we obtain the result of [2]. When \( k = n - i \) for \( i = \lceil \frac{n+1}{2} \rceil, \cdots n \) and the slope is set to \( \pi/4 \) and \( 3\pi/4 \) the result of [3] is recovered.

3 Stationarity in AR Processes

In a number of applications of Digital Signal Processing we are interested in fitting an all-pole model to a signal, usually based on short data records. If stationarity for the time series (stability for the all-pole model) is desired, we have to deal not only with the nominal coefficients of the model (obtained through point estimations taken from the data), but also with all the possible values for the parameters that lie in the confidence intervals that can be placed around the actual values of the estimators.

Once these intervals are obtained, we are in a position to study the stability of the whole family of possible values of the parameters (robust stability), where now the perturbations of the nominal coefficients are precisely the ranges of variation inside the confidence intervals. Due to the lack of a priori information about the underlying probabilistic properties of the estimators, it is not possible to get theoretical bounds for those intervals. Nevertheless, that problem can be overcome using non-parametric bootstrap techniques [8] as it is usual in fitting linear regression models [9].

In order to summarize the bootstrap analysis, we assume the validity of a model structure used to fit the data, and find residuals around the nominal model over the actual data record; then, resampling those residuals we can generate artificial data that provides new estimations for the parameters of the model. Repeating the process as many times as necessary the bootstrap distribution of the parameters is obtained; such an empirical distribution can then be taken as an approximation of the true distribution of the parameters estimates. Finally, using the bootstrap distribution, empirical confidence intervals for the parameters can be computed, and a value for the error in the estimation can also be obtained. The great advantage of this technique resides in its ability to handle problems where no statistical information is available, and no assumptions are made about the underlying statistics of the model, improving the classical theory of confidence intervals construction.

In the next section, those intervals are taken as uncertainties (or structured perturbations) of the parameters, and a case is made for the study of the robust stability properties of such models.

4 Numerical Example

As an example for testing the results and techniques developed in the paper, we have made simulations with a zero-mean AR process having only three non-zero coefficients, specifically the last three. This is actually a good example of a long term predictor taken from those we find in Linear Predictive Speech Coders, as the CELP or Multipulse among others.

Bootstrap confidence intervals were computed from a data record containing 256 samples. A modified version (assuming that only the last three coefficients are different from zero) of the classical LMS algorithm was used to estimate the parameters of the predictor. The results are shown in Table 1, where values for the point-estimations and confidence intervals can be found. It should be pointed out that these three intervals do not contain zero, so that we can assure that the true parameters of the model are not null.

The predictor under consideration can then be written as

\[
A(z) = 1 + a_{24}z^{-24} + a_{25}z^{-25} + a_{26}z^{-26}. \tag{10}
\]

Therefore, since the variation is only on a small part of the upper half of the coefficients, i.e. in \( a_i \); \( i \geq 13 \), our results can be applied to the study of the stationarity of this AR process. As we have three free coefficients, we will need to check all the possible combinations of extremes, that is, \( 2^3 \) according to Section 2. In Table 2 we present the eight extreme combinations and the maximum value of the modulus of the roots for each corresponding polynomial. Since this maximum is less than 1 for all the extremes, it can be concluded that for every possible value of the parameters inside the confidence intervals, stationarity is guaranteed.

<table>
<thead>
<tr>
<th>Coefficient</th>
<th>Point Estimates</th>
<th>Confidence Intervals</th>
</tr>
</thead>
<tbody>
<tr>
<td>24</td>
<td>-0.19</td>
<td>-0.31 ( \leq a_{24} ) ( \leq -0.039 )</td>
</tr>
<tr>
<td>25</td>
<td>0.23</td>
<td>0.15 ( \leq a_{25} ) ( \leq 0.27 )</td>
</tr>
<tr>
<td>26</td>
<td>-0.33</td>
<td>-0.38 ( \leq a_{26} ) ( \leq -0.065 )</td>
</tr>
</tbody>
</table>

Table 1: Estimates and Confidence Intervals

<table>
<thead>
<tr>
<th>( a_{24} )</th>
<th>( a_{25} )</th>
<th>( a_{26} )</th>
<th>Max. Modulus</th>
</tr>
</thead>
<tbody>
<tr>
<td>-0.32</td>
<td>0.15</td>
<td>-0.38</td>
<td>0.99399</td>
</tr>
<tr>
<td>-0.039</td>
<td>0.15</td>
<td>-0.38</td>
<td>0.97821</td>
</tr>
<tr>
<td>-0.33</td>
<td>0.27</td>
<td>-0.38</td>
<td>0.98919</td>
</tr>
<tr>
<td>-0.039</td>
<td>0.27</td>
<td>-0.38</td>
<td>0.98549</td>
</tr>
<tr>
<td>-0.33</td>
<td>0.15</td>
<td>-0.065</td>
<td>0.97754</td>
</tr>
<tr>
<td>-0.039</td>
<td>0.15</td>
<td>-0.065</td>
<td>0.94689</td>
</tr>
<tr>
<td>-0.33</td>
<td>0.27</td>
<td>-0.065</td>
<td>0.98355</td>
</tr>
<tr>
<td>-0.039</td>
<td>0.27</td>
<td>-0.065</td>
<td>0.96513</td>
</tr>
</tbody>
</table>

Table 2: Extreme Polynomials and Roots
5 Conclusions

The stationarity of sparse AR processes has been analyzed when the uncertainties introduced by the point estimators are considered. The main contribution of the paper resides in the use of techniques from robust stability and bootstrap analysis to study classical signal processing problems. Our future research will study the stability of generalized linear-phase filters under perturbations in the filter coefficients and the study of the complex filters case as it applies to multipath communications problems.

References


The Scale-Angle Representation in Image Analysis with 2D Wavelet Transform

Jean-Pierre ANTOINE, Pierre CARRETTE and Romain MURENZI

Institut de Physique Théorique, Université Catholique de Louvain
B - 1348 Louvain-la-Neuve, Belgium

Wavelet analysis [1,2] is by now recognized as a powerful tool for the analysis and reconstruction of signals. Since it is a scale-space decomposition, it is particularly efficient for detecting features which are well-localized both in scale and in space. In 1 dimension, this means basically discontinuities in the signal or one of its derivatives: sharp peaks (spectral lines, e.g. in NMR spectroscopy), onset of a signal (e.g. in speech or music analysis), etc. In 2 dimensions, i.e. in images, typical features are edges, tips, sharp luminance changes, coherent structures in 2D turbulent flows, etc. In this contribution we will focus on the 2D case. The use of 2D wavelets in image processing has been advocated by Mallat [3], using a method based on multiresolution analysis. This is an instance of a discrete wavelet transform (WT), and it is essentially confined to Cartesian geometry: the 2D analysis is obtained by combining (tensor product) 1D analyses for the x- and y-directions separately. Thus the method is not well-adapted to the detection of directional features of an image. For that purpose, one has to use the continuous WT, that we now briefly describe [4]. A 2D signal is represented by a bounded nonnegative function s, square-integrable over the plane $\mathbb{R}^2$. An analyzing wavelet is a complex-valued, square-integrable function $\psi \in L^2(\mathbb{R}^2,d^2x)$, satisfying an admissibility condition, which, for $\psi$ regular enough, amounts to a zero mean condition:

\[ \int d^2x \psi(x) = 0. \]  

The WT of the signal $s$, with respect to the wavelet $\psi$ (suitably normalized) is the function:

\[ S(a,\theta,\vec{b}) = \langle \psi_{a,\theta,\vec{b}} | s \rangle \]
\[ = a^{-1} \int d^2x \overline{\psi}(a^{-1}r_{-\theta}(x - \vec{b}))s(x). \]  

Here $\langle \cdot | \cdot \rangle$ is the usual scalar product in $L^2(\mathbb{R}^2,d^2x)$, and $\psi_{a,\theta,\vec{b}}$ is obtained from $\psi$ through dilation by $a > 0$, translation by $\vec{b} \in \mathbb{R}^2$ and rotation by $\theta \in [0,2\pi]$:

\[ \psi_{a,\theta,\vec{b}}(x) = a^{-1} \psi(a^{-1}r_{-\theta}(x - \vec{b})), \]
\[ r_\theta(x) = (x \cos \theta - y \sin \theta, x \sin \theta + y \cos \theta), \]
\[ 0 \leq \theta < 2\pi. \]  

These transformations generate the 2D Euclidean group with dilations, and indeed the whole WT formalism stems from a suitable representation of that group [4]. The linear transform $W_\psi : s \mapsto S$ can be inverted on its range by the transposed map, and thus one gets integral reconstruction formulas, such as:

\[ s(x) = \int \int a^{-2} \, da \, d\theta \, d^2\vec{b} \, \psi_{a,\theta,\vec{b}}(x) \, S(a,\theta,\vec{b}). \]  

Of course, for practical purposes, both the direct WT (2) and the inverse WT (4) have to be discretized, which poses the question of determining the optimal sampling rate in all variables (see below). Because of the mean zero condition (1), the wavelet transform (2) acts as a very efficient filter in position ($\vec{b}$), in scale ($a$) and in direction ($\theta$) : the WT $S(a,\theta,\vec{b})$ is appreciable only in those regions of parameter space $a,\theta,\vec{b}$ where the signal is appreciable, that is, where the wavelet $\psi_{a,\theta,\vec{b}}$ 'matches' the features of the signal $s$. This explains why the WT may be interpreted as a 'mathematical microscope', with optics $\psi$, global
magnification $1/a$ and orientation tuning parameter $\theta$. Now the WT $S(a, \theta, \vec{b})$ is a function of 4 variables: $a > 0$, $\theta \in [0, 2\pi)$, $\vec{b} \in \mathbb{R}^2$. Clearly, in order to obtain a manageable tool, one must limit oneself to special sections of the parameter space $\{a, \theta, \vec{b}\}$. It turns out that two such sections, both 2-dimensional, are very natural: the position representation, where $(a, \theta)$ are fixed and $S(a, \theta, \vec{b}) \equiv S_{a, \theta}(\vec{b})$ is considered as a function of $(b_x, b_y)$, and the scale-angle representation, where $(b_x, b_y)$ are fixed and $S(\alpha, \theta, \vec{b}) \equiv S_{\alpha}(\alpha, \theta)$ is considered as a function of $(a, \theta)$. Actually the pair $(a^{-1}, \theta)$ plays the role of spatial frequency, expressed in polar coordinates; this interpretation is supported both by mathematical analysis (phase space geometry) and by some physiological evidence [6].

The position representation is standard and has been used in all applications so far [2-5]. The aim of this contribution is to demonstrate the usefulness of the scale-angle representation. Here the WT is examined through a key-hole located at $\vec{b}$, and examined at all scales and in all directions:

$$S_{\vec{b}}(a, \theta) = a^{-1} \int dx dy \tilde{\psi}(a^{-1} r - \theta(\vec{x})) s_{\vec{b}}(x, y),$$

where $s_{\vec{b}}(x, y) = s(x + b_x, y + b_y)$ represents the signal seen in a Cartesian frame located in $\vec{b}$. The expression (5) is evaluated with the FFT algorithm, after transcription in polar coordinates and use of a Mellin transform.

Whereas the position representation is used for the general purposes of image processing (detection of specific features, such as contours, edges, etc., noise filtering, data compression), the scale-angle representation is useful for several applications [2,4,5]:

- determination of the sampling grid in the variables $(a, \theta)$ for the numerical evaluation of the reconstruction integral (4), using the reproducing kernel of the continuous WT;
- (multi) fractal recognition, since the signal is analysed at all scales;
- determination of the local regularity of the function $s$ at the point $\vec{b}$;
- disentangling of a superposition of plane waves.

Let us sketch this last application. The basic problem is to decompose a complex signal into single components which have a simple form, more or less known a priori, and to isolate their individual contribution to the WT; this is the key for noise filtering or elimination of unwanted components. A typical example in 1D is the estimation of spectral lines (e.g., NMR spectroscopy, geophysical time series). The problem is to identify with great precision the position of the peaks (local maxima) and to measure their parameters [7]. Another 1D example is the analysis of refracted waves in underwater acoustics [8], although this particular example is in fact a 3D situation. We shall consider here the 2D version of the problem. The signal is a superposition of plane waves, damped in a given direction, and we want to measure the various parameters of each component. We shall take the signal in the following form:

$$f(\vec{x}) = \sum_{n=1}^{N} c_n e^{i \vec{k}_n \cdot \vec{x}} e^{-i \vec{\tau}_n \cdot \vec{x}},$$

where $c_n = A_n e^{i \phi_n}$, with $A_n$ and $\phi_n$ respectively the modulus and phase of the $n$th component at the origin; $\vec{k}_n$ is the wave vector and the vector $\vec{\tau}_n$ in a (oriented) damping factor.

The choice of an exponential damping factor $e^{-(\vec{\tau}_n \cdot \vec{x})}$ makes of course the computation especially simple, but it is by no means essential. For instance, similar results would be obtained with a damping factor acting in one direction only, such as $\vartheta(\vec{k}_n \cdot \vec{x}) e^{-(\vec{\tau}_n \cdot \vec{x})}$, where $\vartheta$ is the jump function or a smoothed version thereof. The important point is that the perturbation be local: the method still allows the determination of the various parameters sufficiently far away from the perturbation (as in 1D).

A straightforward evaluation, using a Morlet wavelet (Gabor function) with isotropic modulus yields the WT of the signal (6):

$$F(a, \theta, \vec{b}) = 2\pi e^{\frac{1}{2} k^2} \sum_{n=1}^{N} c_n e^{i \vec{k}_n \cdot \vec{b}} e^{-i \vec{\tau}_n \cdot \vec{b}} P_{\vec{k}_n}(a, \theta).$$

Of course, the linearity of the WT implies that the result is the linear superposition of the contributions of the various components. In a first
step, we fix \( \vec{b} \) and analyze the \((a, \theta)\) dependence of \( F(a, \theta, \vec{b}) \). More precisely, writing

\[
F_{\theta}(a, \theta) = \sum_{n=1}^{N} c_{\theta,n} F_{\theta,n}(a, \theta),
\]

we observe that \( a^{-1} |F_{\theta,n}(a, \theta)| \) has a unique maximum localized at \((a_n^{\text{max}}, \theta_n^{\text{max}})\). Now, in the full transform (7), each term has its own local maximum, but these need not be well separated: one maximum may hide another one, totally or partially. This masking effect will happen, for instance, when:

- one component has a much bigger amplitude, \( |c_{\theta,n}| >> |c_{\theta,m}| \), for all \( m \) not equal to \( n \) (total masking);
- two wave vectors are close to each other, \( \vec{k}_n \simeq \vec{k}_m \), but with different amplitudes, \( |c_{\theta,n}| > |c_{\theta,m}| \) (partial masking).

In both cases, the two waves can be separated, by increasing the selectivity of the wavelet (for instance, using a Morlet wavelet with a more anisotropic modulus [5]). If the two waves have close wave vectors \( (\vec{k}_n \simeq \vec{k}_m) \) with similar amplitudes \( (|c_{\theta,n}| \simeq |c_{\theta,m}|) \), but different damping vectors \( (\vec{l}_n \neq \vec{l}_m) \), then they can still be separated, by changing the observation point \( \vec{b} \). Otherwise the method will fail, the two waves interfere inextricably, none of them dominates the other one.

When the masking effect is not too important, the maxima will be sufficiently prominent that the interferences between the different components will become negligible (in the modulus) and one may write

\[
a^{-1} |F_{\theta,n}(a, \theta)| \simeq \sum_{n=1}^{N} |c_{\theta,n}| a^{-1} |F_{\theta,n}(a, \theta)|.
\]

This is the formula used in practice. Now we come back to the full transform (7) and evaluate it successively at every local maximum \((a_n^{\text{max}}, \theta_n^{\text{max}})\), thus reverting to the position representation. This amounts to filter out all components except the \( n \)th:

\[
F_{a_n^{\text{max}}, \theta_n^{\text{max}}}(x, y) \simeq d_n e^{i \vec{k}_n \cdot \vec{x}} e^{-i \vec{l}_n \cdot \vec{y}}.
\]

This expression, which we call \( F(\vec{b}) \) for short, represents a single damped plane wave. For evaluating the wave parameters \( \vec{k}_n = \vec{k}_n (\cos \alpha_n, \sin \alpha_n), \vec{l}_n = \vec{l}_n (\cos \beta_n, \sin \beta_n) \), it suffices now to consider the lines of constant modulus and constant phase of \( F(\vec{b}) \): their orientation yields the angles \( \alpha_n \) and \( \beta_n \), their spacing the norms \( k_n \) and \( l_n \), respectively. To get rid of noise, one may average \( \log |F(\vec{b})| \) and \arg F(\vec{b}) \) on the corresponding line of constancy passing through the point \( \vec{b} \) considered. Finally the original coefficients \( c_n = A_n e^{i \phi_n} \) are evaluated by measuring \( F(\vec{b}) \) itself at a particular point \( \vec{b}_0 \), typically 0.

In conclusion, the feasibility of this procedure rests, as in 1D, on two key features:

(i) the linearity of the CWT, which allows to treat each component separately;

(ii) the filtering effect of the CWT, which essentially eliminates all components except one.

We may remark that, in case of a masking, one of the components hides the other one. In that case, one must evaluate first the dominant component and then subtract it from the full transform. This is the 2D analogue of the subtraction of unwanted spectral lines in 1D, one of the spectacular achievements in the analysis of NMR spectra [7]. The comparison with the 1D case also confirms that the parameters \((a^{-1}, \theta)\) together play the role of a wave vector: localizing the peaks (local maxima) in the \((a, \theta)\) plot corresponds to the first step in the iterative procedure of localization of spectral lines.

The method has been tested on an explicit example, consisting of a superposition of four waves, with the parameters chosen so as to exhibit a partial masking \((a_1 \simeq \alpha_0)\). This signal is analyzed by the Morlet wavelet \( \psi (x, y) = \exp(i \phi y) \exp[-\frac{1}{2}(\epsilon^{-1} x^2 + y^2)] \). Fig.1 shows the level curves of the renormalized modulus of the WT, seen from the origin, \( a^{-1} |F_{\theta,\phi}(a, \theta)| \). Only the prominent features survive, the function is negligible except in the neighborhood of the local maxima. Fig.1(a) is obtained with the isotropic wavelet \((\epsilon = 1)\); it clearly shows the presence of four waves (four maxima), but those numbered 1 and 4 are not well separated. Thus the analysis is redone with \( \epsilon = 5 \), and the result plotted
in Fig.1(b) clearly shows the improved selectivity, the two maxima are now clearly identifiable. The position of the four local maxima is measured by a standard algorithm, then the full WT is evaluated at each of the maxima, which allows to measure the parameters of each plane wave.

The result is that all parameters are recovered with an excellent precision (relative error of the order of $10^{-3}$), except for the amplitude $A_4$. This is, of course, a consequence of the masking of wave 4 by wave 1. If one plots the phase of each individual wave, as given by (10), each plot indeed represents the phase of a pure plane wave, except for side effects and the interference between waves 1 and 4, mostly visible in the plot of wave 4. For improving the result, we could subtract the (reconstructed) wave 1 from the signal and repeat the analysis. Also a better precision could be obtained by a finer discretization of the signals (the grid used throughout is $128 \times 128$ points).

References


[8] G. Saracco, A. Grossmann, Ph. Tchamitchian, Use of wavelet transforms in the study of propagation of transient acoustic signals across a plane interface between two homogeneous media, in [1], pp. 139-146.
Lowering the Threshold SNR of Fourier Signal Subspace

Jyrki Joutsensalo and Juha Karhunen

Helsinki University of Technology,
Laboratory of Computer and Information Sciences,
Rakentajankatu 2 C, SF-02150 Espoo, Finland
Fax: +358-0-4513277

Abstract

In sinusoidal frequency estimation and array processing subspace type methods are currently studied extensively. In these methods (e.g. MUSIC and ESPRIT) the signal subspace which defines the unknown frequencies or directions of arrival is approximated by computing the eigenvectors of the sample correlation matrix. We have earlier proposed two methods which use the Discrete Fourier Transform (DFT) for estimating the signal subspace instead of eigenvectors. These methods can make use of prior information provided by classical spectral estimators. They are also more robust against noise and overestimation of the number of signals. In this work we present two new methods in which the signal subspace is first estimated crudely using DFT. This estimate is then postprocessed, leading to a lowered threshold SNR, and improved performance compared to our earlier methods. We consider sinusoidal frequency estimation, but with small restrictions and modifications the methods are applicable to array processing, too.

1 Data model

The data are assumed to consist of M complex sinusoids in additive white noise. The N available data samples \( x(0), x(1), \ldots, x(N-1) \) are thus modeled as

\[
x(n) = \sum_{m=1}^{M} a_m e^{j(\omega_m n + \theta_m)} + w(n),
\]

where the amplitudes \( a_m \), angular frequencies \( \omega_m \), and phases \( \theta_m \) are unknown constants. The complex white noise term \( w(n) \) has zero mean and variance \( \sigma^2 \). In practical estimation a data matrix

\[
X = \begin{bmatrix}
x(0) & x(1) & \cdots & x(N-L) \\
x(1) & x(2) & \cdots & x(N-L+1) \\
\vdots & \vdots & \ddots & \vdots \\
x(L-1) & x(L) & \cdots & x(N-1)
\end{bmatrix}
\]

\[
X = X_* + W = E_{LM}A(E_{KM})^T + W
\]

is used, where

\[
E_{LM} = \begin{bmatrix} e_L(\omega_1) & e_L(\omega_2) & \cdots & e_L(\omega_M) \end{bmatrix},
\]

\[
e_L(\omega_m) = \begin{bmatrix} 1 & e^{j\omega_m} & \cdots & e^{j(N-1)\omega_m} \end{bmatrix}^T,
\]

\[
A = \text{diag} \{ a_1 e^{j\theta_1}, a_2 e^{j\theta_2}, \ldots, a_M e^{j\theta_M} \},
\]

and \( W \) is a noise matrix having similar structure as \( E_{LM} \). Superscripts \( T \) and \( H \) denote transpose and complex conjugate transpose, respectively. \( E_{LM} \) is an \( L \times M \) matrix and it consists of frequency information or direction vectors \( e_L(\omega_m) \). \( E_{KM} \) has the same structure as \( E_{LM} \), but with the size \( K \times M \), where \( K = N - L + 1 \) is the number of columns (data vectors) in \( X \). We use this subscript convention also later to denote the size of a matrix. The first term \( X_* \) in Eq. (3) is deterministic signal part, and its rank is \( M \). The number \( M \) of signals is assumed to be known, at least roughly.

2 Estimation of signal subspace using DFT

In the subspace type methods a vector-valued DFT can be used for estimating the signal subspace [2,3]. We have used two different approaches for forming the DFT operator matrix. The first approach is to use the periodogram; the second approach is to use the Bartlett estimate. Before presenting the new methods we define these operators.

In the periodogram approach, the vectorial DFT operator matrix is constructed using frequency estimates given by the periodogram. The form of the DFT matrix depends on the form of the matrix to be transformed. In the case of \( X \) one tries to get the DFT matrix to be as close as possible to the complex conjugate of \( (E_{KM})^T \) in Eq. (3). Then the DFT matrix is

\[
\hat{E}_{KM} = \begin{bmatrix} e_K(-\omega_1) & e_K(-\omega_2) & \cdots & e_K(-\omega_M) \end{bmatrix},
\]

where \( \omega_m, m = 1, 2, \ldots, M \) yield the \( M \) highest peaks of the periodogram of data sequence which is assumed to
be of the form (1). The reason to this type of choice is that the Frobenius norm of the signal part is improved with respect to the Frobenius norm of the noise part:

$$\|X\hat{E}_{KM}\|_F > \|X\|_F.$$

In general there may exist other frequency bins than those closest to the correct ones yielding high peaks, too. In the new methods we take this into account by choosing $H$ ($H > M$) frequencies in the DFT matrix: $\hat{E}_{KM}$ in Eq. (4) is replaced by $\hat{E}_{KH}$. Multiplying the data matrix $X$ by $\hat{E}_{KH}$ corresponds to a filtering where one tries to retain all the relevant signal information and remove as much noise as possible.

When the data matrix $X$ is Fourier transformed – e.g. in DFT-MUSIC – one can alternatively use the Bartlett estimate. In this case one first constructs $K$ vectors $\hat{q}_k$ corresponding to the uniformly spaced frequencies $\Omega_k$:

$$\hat{q}_k = X e_{KH}(\Omega_k), \quad \Omega_k = \frac{2\pi(k-1)}{K}, \quad k = 1, 2, \ldots, K.$$  

Then one computes the squared norms

$$\|\hat{q}_k\|^2 = \hat{q}_k^H \hat{q}_k = e_{KH}^H(\Omega_k)X^H X e_{KH}(\Omega_k) = e_{KH}(\Omega_k) R_{KH}(\Omega_k), \quad k = 1, 2, \ldots, K,$$  

and chooses the vectors $\hat{q}_k$ corresponding to the $H$ largest norms. These $H$ vectors are taken as the columns of the matrix $\hat{Q}$. Eq. (6) is clearly the Bartlett estimate.

3 New methods

From the fact that the rank of $X$ is $M$, we know that the data vectors lie in the $M$-dimensional signal subspace if there exist no noise. The idea in the new methods is to first prefilter the data by forming $H$ vectors $\hat{q}_k$, $k = 1, 2, \ldots, H$ using either of the methods explained in Section 2. After this one estimates the true signal subspace by searching for a smaller $M$-dimensional subspace in which this vector set approximately lies. The search is done by applying again the vectorial DFT to the correlation matrix constructed of the $\hat{q}_k$-vectors. The eigenvector approach can be used, alternatively. The algorithms are as follows:

1. (Only in the periodogram version.) Compute the periodogram of the samples $x(0), x(1), \ldots, x(N - 1)$. Select the negative counterparts $-\omega_m$ of the frequencies $\omega_m := \frac{2\pi m}{N}$, $m = 1, \ldots, M$. $H > M$ corresponding to the $H$ highest peaks of the periodogram. Using these frequencies, construct $\hat{E}_{KH}$ as in Eq. (4).

2. Construct a data matrix $X$ from the samples using Eq. (2).

3. (Periodogram version) Take the DFT of $X$ at selected frequencies:

$$\hat{Q} = X \hat{E}_{KH}.$$  

(Bartlett estimate version) Take the DFT of $X$ in $K$ frequency bins using Eq. (5). Select the $H$ vectors with largest norms from the set $\hat{q}_k$, $k = 1, 2, \ldots, K$ (Eq. (6)); form the matrix $\hat{Q}$ from these vectors.

4. Algorithm 1: Construct a new DFT operator $\hat{E}_{LM}$ corresponding to the $M$ highest peaks $\hat{\Omega}_M$ or $\hat{\Omega}_L$ yielding $M$ largest norms and take the Fourier transform of $Q \hat{Q}^H$:

$$\hat{V} = \hat{Q} \hat{Q}^H \hat{E}_{LM}.$$  

Algorithm 2: Compute the $M$ principal eigenvectors of $Q \hat{Q}^H$ and form the matrix $\hat{V}$ from these vectors.

5. Orthonormalize $\hat{V}$ and insert the result to the MUSIC or ESPRIT estimate [1].

In these methods one can use the FB estimates: matrix product $Q \hat{Q}^H$ is replaced by the product $Q \hat{Q}^H + \hat{S}_{HH}$ in which the first row of $\hat{S}$ is the last row of $\hat{Q}$ complex conjugated, the second row of $\hat{S}$ is the second last row of $Q$ complex conjugated, and so on. Both of these methods are computationally less complex than the basic version of MUSIC.

Illustrative example: Consider the simple model

$$x(n) = e^{2\pi \sigma_1 n} + e^{2\pi \sigma_2 n} + w(n), \quad n = 0, \ldots, 24.$$  

Let us construct a $10 \times 16$ matrix $X$. By taking the periodogram of 25 point sequence with FFT we get 32 values corresponding to 32 frequency bins. Assume that the four ($= H$) highest peaks in decreasing order are obtained at frequency bins 3/32, 6/32, 4/32, and 7/32. Then 3/32 and 6/32 yield the two $(= M)$ highest peaks, and $\hat{E}_{16, 4}$ in (7) and $\hat{E}_{10, 2}$ in (8) are

$$\hat{E}_{16, 4} = \begin{bmatrix} e_{16}(\frac{\pi}{32}) & e_{16}(\frac{\pi}{16}) & e_{16}(\frac{\pi}{8}) & e_{16}(\frac{\pi}{4}) \\ e_{16}(\frac{\pi}{32}) & e_{16}(\frac{\pi}{16}) & e_{16}(\frac{\pi}{8}) & e_{16}(\frac{\pi}{4}) \end{bmatrix},$$  

$$\hat{E}_{10, 2} = \begin{bmatrix} e_{10}(\frac{\pi}{32}) & e_{10}(\frac{\pi}{16}) \\ e_{10}(\frac{\pi}{32}) & e_{10}(\frac{\pi}{16}) \end{bmatrix}.$$  

4 Analysis of the algorithms

For simplicity, we analyze only the periodogram and backward version. Our goal is to show that the methods yield the matrices spanning the same subspace as Eq. (4). With this condition MUSIC and ESPRIT can extract the frequencies. In Step 1 one forms a matrix $P$ with

$$P = \mathcal{E}(\hat{P}) = \mathcal{E}(\hat{Q} \hat{Q}^H),$$  

where $\mathcal{E}(\cdot)$ denotes the expectation operator. From Eq. (3) and Step 3 we get

$$\hat{Q} = X \hat{E}_{KH} = E_{LM} K + W \hat{E}_{KH},$$  

where $K = A (E_{KM})^T \hat{E}_{KH}$. Using Eqs. (9), (10), and
the facts that the absolute values of the elements of the DFT matrices are 1, and that the noise is white with zero mean, we get

\[
P = P_{LM}K^H(P_{LM})^H + P_{LM}K(P_{KH})^H(EW^H) + \varepsilon(W^H)E_{KH}^H(P_{KH})^H + \varepsilon(W^H)E_{KH}^H(P_{KH})^H W^H
\]

\[
= P_{LM}B(P_{LM})^H + \sigma^2 I,
\]

(11)

\[
B = KK^H = A(E_{KM})^2 E_{KH}(E_{KH})^H(E_{KM})^H A^H,
\]

where \( \varepsilon \) denotes the complex conjugate. The rank of \( B \) is \( M' \) due to the \( E_{KM} \)'s Vandermonde property so that the signal part of the matrix \( P \) is of full rank \( M' \). This is a necessary condition for subspace type methods. In algorithm 2 one computes the \( M' \) principal eigenvectors of \( P \) which span theoretically the same subspace as the columns of \( E_{LM} \), as we see from Eq. (11). This is a sufficient condition, so the algorithm 2 works theoretically. In algorithm 1 one takes the DFT in Step 4:

\[
V = P\hat{E}_{LM} = P_{LM}H + \sigma^2 \hat{E}_{LM},
\]

(12)

\[
H = B(P_{LM})^H \hat{E}_{LM}.
\]

Denoting the \( m \)-th row and the \( k \)-th column element of \( H \) by \( h_{mk} \), we get

\[
h_{mk} = \hat{a}_m(0) \sum_{n=1}^{M} \hat{a}_n(0) \sum_{l=1}^{N} g_n(-\omega_l + \omega_k)g_n(\omega_l - \omega_k)
\]

\[
\times g_l(\omega_m - \omega_k), \quad m, k = 1, 2, \ldots, M,
\]

\[
g_K(\omega) = \sin(K\pi/2) / \sin(\omega/2) ^{+K+1/2}.
\]

\( g_K(\omega) \) is a sinc function. The signal part \( E_{LM}H \) in Eq. (12) spans the same subspace as \( E_{LM} \). In general \( \omega_m \approx \omega_m \) and there exists (usually negligible) bias in the MUSIC spectrum due to the matrix \( K\sigma^2 \hat{E}_{LM} \).

For studying the ability of Fourier transform in improving the SNR, we must make certain simplifications. Assume that the estimates corresponding to the highest peaks of periodogram are exactly the same as the true frequencies: \( \omega_m \approx \omega_m \), \( m = 1, 2, \ldots, M \). Then \( V \) in Eq. (12) spans exactly the desired subspace. Assume also that \( H = 2M \), and that there exists to each frequency a pair close to it: \( \omega_m \approx \omega_m \). To make more simplifications, let the frequencies be orthogonal to each other: \( \delta_{K}(\omega_1)\delta_{K}(\omega_2) = 0 \), \( \delta_{K}(\omega_1)\delta_{K}(\omega_2) = K\delta_{ij} \), where \( \delta_{ij} \) is the Kronecker delta. Then

\[
(E_{KM})^2 E_{KH} \approx K [ I \ I \ I ], \quad (E_{KM})^H \hat{E}_{LM} = LI.
\]

And an approximation to \( V \) is:

\[
V \approx 2K^2 I L E_{LM} A A^H + K\sigma^2 E_{LM}.
\]

Magnification factor \( 2K^2 I L / K = 2K L \) and \( A A^H \) (diagonal matrix containing the powers of amplitudes in its elements) imply that SNR is clearly improved.

5 Generalization

The new algorithms, as well as the methods introduced in [2, 3], belong to a more general category. Matrices \( X \) and \( P \) are of to special type; Fourier transform enhances the energy of a signal part with respect to the noise part. We define a general matrix \( \hat{M} \) having this property. It has the form

\[
\hat{M} = \hat{M}_s + \hat{M}_n,
\]

(13)

\[
\varepsilon(\hat{M}) = M,
\]

\[
\varepsilon(\hat{M}_s) = E_{LM}C(G_{KM})^H,
\]

\[
\varepsilon(\hat{M}_n) = \alpha J,
\]

\[
\Rightarrow \quad \hat{M} = E_{LM}C(G_{KM})^H + \alpha J,
\]

(14)

\[
G_{KM} = \begin{bmatrix} e_k(\nu_1) & e_k(\nu_2) & \cdots & e_k(\nu_M) \end{bmatrix},
\]

(15)

\[
J = \begin{bmatrix} 1 & 0 \end{bmatrix},
\]

(16)

\[
\begin{cases} \alpha \in C, \quad \text{when } \nu_m = \nu_m, \quad m = 1, 2, \ldots, M, \\
\alpha = 0, \quad \text{otherwise} \end{cases}
\]

(17)

Here \( C \) is arbitrary full rank \( M \times M \) matrix. It is the complex number set, and the parameters \( \omega_m \) and \( \nu_m \), \( m = 1, 2, \ldots, M \) are unknown. The \( \omega_m \)'s are the frequencies or directions of arrival (DOAs) that are to be estimated. The \( \nu_m \)'s are frequencies which are used in the prefiltering. The DFT operations corresponding to \( \nu_m \)'s yield large norm linear combinations of \( e_k(\omega_m) \) vectors:

\[
V = MG_{KM} = E_{LM}C(G_{KM})^H \hat{G}_{KM} + \alpha J \hat{G}_{KM} = E_{LM}C(G_{KM})^H \hat{G}_{KM} + \alpha \hat{G}_{LM}.
\]

(18)

The condition (17) on parameter \( \alpha \) is explained as follows. If \( \alpha \neq 0 \), then \( \alpha \hat{G}_{LM} \) in Eq. (18) may cause peaks in the locations \( \nu_m \). If \( \omega_m \neq \nu_m \), they are undesired spurious peaks; otherwise they are near the correct frequency locations.

There are several ways of constructing the matrix \( \hat{M} \) fulfilling the conditions (13)–(17). For example, with mild assumptions, correlation and crosscorrelation matrices belong to this category. The matrix \( \hat{M} \) arises in its nearly most general form when two sensor arrays are used:

\[
\hat{M} = \frac{1}{N} \sum_{n=0}^{N-1} x(n) y^H(n),
\]

where \( x(n) \)'s are snapshots of the first array, and \( y(n) \)'s are snapshots of the second array. If the arrays are not in parallel direction, then they "see" different DOAs: \( \nu_m \)'s and \( \nu_m \)'s are different. In the data matrix \( X \) they also are different: \( \nu_m \neq \nu_m \). Then it must hold: \( \alpha = 0 \). This holds when noise is zero mean. The advantage of the generalized matrix is that it is possible to resolve the frequencies or directions from the crosscorrelation matrix that suppresses noise or in general from a matrix that is indefinite so that eigenvector analysis is not possible. We must only know the frequencies \( \nu_m \) yielding the highest peaks of periodogram or largest norm vec-
tors of the Bartlett estimate; then we can estimate the signal subspace.

In the array signal processing applications, the theory can be extended from linear arrays to deal with arbitrary array geometries

\[ x(n) = D_{LM}a(n) + w(n), \quad n = 0, \ldots, N - 1. \]

Here the general matrix \( D_{LM} \) has taken the place of \( B_{LM} \), and it describes the structure of the array. \( a(n) \) contains the amplitudes of the signals, and \( x(n) \) and \( w(n) \) are \( L \)-dimensional data and noise vectors, respectively. The estimated matrix is \( D_{LM} \), and the linear matrix transform is no more of Fourier type, but reflects the array structure.

6 Experiments

In the experiments we compare the new methods to forward-backward MUSIC. MUSIC estimate

\[ \hat{P}(\omega) = \frac{1}{N} L^1 e^{j/\omega} \hat{y}(\omega) \hat{y}^H(\omega) \]

was evaluated at 2048 frequencies. The compared properties are probability of resolution (Table 1), and the number of spurious peaks when the dimensionality of the signal subspace was overestimated by one (Table 2). The SNR varied from −4 dB to 16 dB (first row in the tables). 100 simulations were made at every SNR. Noise process, amplitudes \( a(\nu) \), frequencies \( f = \omega/2\pi \), and phases \( \theta(\nu) \) were chosen randomly in each realization so that they satisfied the conditions

\[ a_1, a_2 \in (1, 4), \quad |f_1 - f_2| = 0.02, \quad \theta_1, \theta_2 \in [0, 2\pi). \]

Each realization consisted of \( N = 40 \) samples so that \( |f_1 - f_2| < 1/N \). \( L = 17 \). The frequencies were considered to be resolved if

\[ \sqrt{\frac{1}{2}((f_1 - \hat{f}_1)^2 + (f_2 - \hat{f}_2)^2)} \leq 0.02. \]

In determining the average number of peaks, peaks at most 30 dB below the maximum value of MUSIC spectrum were counted. The results in Tables 1 and 2 show that the algorithms 1 and 2 are clearly more reliable than MUSIC in noisy circumstances, and at high SNRs they yield spurious peaks rather seldom.

7 Conclusions

In this paper we presented new methods for estimating sinusoidal frequencies. They are combinations of classical Fourier methods and modern subspace type methods. Experiments show that these algorithms are clearly more robust than MUSIC, and they perform somewhat better than those proposed in [2,3]. A heuristic explanation is that the use of DFT makes the estimated signal subspace more robust, while the subspace approach yields high resolution. In this way one can combine beneficially classical Fourier-based spectral estimation with the high resolution subspace techniques.

References


<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SNR →</th>
<th>-4</th>
<th>-2</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUSIC</td>
<td></td>
<td>4</td>
<td>8</td>
<td>13</td>
<td>19</td>
<td>37</td>
<td>48</td>
<td>67</td>
<td>80</td>
<td>89</td>
<td>95</td>
<td>101</td>
</tr>
<tr>
<td>Alg. 1, Periodogram ( (H = 3) )</td>
<td></td>
<td>40</td>
<td>45</td>
<td>54</td>
<td>62</td>
<td>68</td>
<td>65</td>
<td>93</td>
<td>94</td>
<td>98</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>Alg. 1, Norm ( (H = 3) )</td>
<td></td>
<td>10</td>
<td>12</td>
<td>18</td>
<td>22</td>
<td>28</td>
<td>48</td>
<td>64</td>
<td>76</td>
<td>95</td>
<td>92</td>
<td>98</td>
</tr>
<tr>
<td>Alg. 2, Periodogram ( (H = 3) )</td>
<td></td>
<td>35</td>
<td>48</td>
<td>64</td>
<td>71</td>
<td>76</td>
<td>89</td>
<td>93</td>
<td>96</td>
<td>98</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>Alg. 2, Norm ( (H = 3) )</td>
<td></td>
<td>6</td>
<td>12</td>
<td>20</td>
<td>31</td>
<td>42</td>
<td>52</td>
<td>74</td>
<td>86</td>
<td>93</td>
<td>97</td>
<td>99</td>
</tr>
</tbody>
</table>

Table 1: Probability of resolution.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>SNR →</th>
<th>-4</th>
<th>-2</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>6</th>
<th>8</th>
<th>10</th>
<th>12</th>
<th>14</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>MUSIC</td>
<td></td>
<td>2.89</td>
<td>2.83</td>
<td>2.83</td>
<td>2.77</td>
<td>2.76</td>
<td>2.75</td>
<td>2.66</td>
<td>2.60</td>
<td>2.57</td>
<td>2.59</td>
<td>2.49</td>
</tr>
<tr>
<td>Alg. 1, Per ( (H = 4) )</td>
<td></td>
<td>2.77</td>
<td>2.68</td>
<td>2.56</td>
<td>2.39</td>
<td>2.16</td>
<td>2.07</td>
<td>1.99</td>
<td>1.86</td>
<td>2.01</td>
<td>2.02</td>
<td>2.01</td>
</tr>
<tr>
<td>Alg. 1, Norm ( (H = 4) )</td>
<td></td>
<td>2.77</td>
<td>2.74</td>
<td>2.68</td>
<td>2.64</td>
<td>2.39</td>
<td>2.32</td>
<td>2.24</td>
<td>2.20</td>
<td>2.14</td>
<td>2.08</td>
<td>2.02</td>
</tr>
<tr>
<td>Alg. 2, Per ( (H = 4) )</td>
<td></td>
<td>2.82</td>
<td>2.72</td>
<td>2.66</td>
<td>2.38</td>
<td>2.12</td>
<td>2.06</td>
<td>1.99</td>
<td>1.95</td>
<td>2.01</td>
<td>2.02</td>
<td>2.02</td>
</tr>
<tr>
<td>Alg. 2, Norm ( (H = 4) )</td>
<td></td>
<td>2.91</td>
<td>2.91</td>
<td>2.01</td>
<td>2.84</td>
<td>2.76</td>
<td>2.58</td>
<td>2.60</td>
<td>2.43</td>
<td>2.39</td>
<td>2.29</td>
<td>2.25</td>
</tr>
</tbody>
</table>

Table 2: Average number of peaks when the dimensionality of the signal subspace was overestimated by one \( (M = 3) \).
ESTIMATION OF THE FUNDAMENTAL FREQUENCY OF A NOISY SUM OF CISOIDS WITH HARMONIC RELATED FREQUENCIES - APPLICATION TO SONAR MULTIPATHS DELAY ESTIMATION *

A. Ferrari, G. Alengrin and C. Theys

LASSY Université de Nice-Sophia Antipolis, Département de l’I3S
41 Bd Napoléon III, 06041 CEDEX FRANCE
tel.: +33-93 217 653, e-mail: ferrari@imnoos.unice.fr

We tackle in this paper the problem of the estimation of the fundamental frequency of a noisy complex sinusoidal signal with \( p \) harmonic related frequencies. This problem is studied considering an extension of the MUSIC algorithm based on the minimisation of a function of the \( p \) principal angles between two \( p \)-dimensional subspaces. This one-dimension minimisation is achieved via a quasi-Newton algorithm. The method is then applied to the multipaths delay estimation problem. Results on a SONAR signal are presented.

1. INTRODUCTION

This paper deals with the problem of estimating the fundamental frequency \( \omega_0 \) of a complex sinusoidal signal with \( p \) harmonic related frequencies \( k \omega_0, k = 1, \ldots, p \), corrupted by a white Gaussian noise. We propose in this paper to take into account the constraints between the frequencies to reduce the problem to the estimation of \( \omega_0 \).

\( k = 1, \ldots, p \), can be modeled in the classical way:

\[
z = W(\Omega^0)x_0 + n
\]  

(1)

where, the \( M \times p \) Vandermonde matrix \( W(\Omega^0) \) is defined as:

\[
W(\Omega^0) = \begin{pmatrix} d(\omega_1^0) & \cdots & d(\omega_p^0) \end{pmatrix}
\]  

(2)

with the notations:

\[
d(\omega) = (1 \exp j\omega \cdots \exp j(M-1)\omega)^t
\]  

(3)

\[
\Omega^0 = (\omega_1^0 \cdots \omega_p^0)^t
\]  

(4)

According to the classical result on the eigen-decomposition of the autocorrelation matrix of the process \( z \), we have:

\[
\mathcal{R}W(\Omega^0) = \mathcal{R}E_s
\]  

(5)

where:

- \( E_s \) denotes the \( M \times p \) matrix associated to the signal subspace
- \( \mathcal{R}Q \) denotes the range of the matrix \( Q \)

So, when \( \Omega \) varies, \( \mathcal{R}W(\Omega) \) is a continuum that has an intersection with \( \mathcal{R}E_s \) for \( \Omega = \Omega^0 \). Hence,

* Work partly supported by DCN St. Tropez.
\[ \Omega^0 \] is the global minimizer of the distance between \( \mathcal{R}W(\Omega) \) and \( \mathcal{R}E_s \). In the general case, this approach leads to a \( p \)-dimensional minimization problem. This complex problem is solved in a sub-optimal way in the MUSIC algorithm replacing the global \( p \)-dimensional minimization by the 1-dimensional minimization of the distance between \( \mathcal{R}E_s \) and a particular vector of \( \mathcal{R}W(\Omega) \):

\[
W(\Omega).(1 \ 0 \ \cdots \ 0)^t = d(\omega)
\]  \(6\)

We will in the sequel investigate the global optimization problem. This method is related to the minimization of the distance \( D \) between two linear spaces of dimension \( p \). This distance is related to the notion of principal angles [2]. If we denote \( \theta_k, k = 1, \ldots, p \) the \( p \) principal angles between two linear spaces \( A \) and \( B \), \( 0 \leq \theta_1 \leq \cdots \leq \theta_p \leq \pi/2 \), we have:

\[
D(A, B) = \sin \theta_p
\]  \(7\)

The angles between \( A \) and \( B \) are usually computed using the singular value decomposition of the matrix \( Q_A^H \cdot Q_B \) where \( Q_A \) is a matrix having as columns an orthogonal basis of \( A \). The singular values of the above matrix are \( \cos \theta_k, k = 1, \ldots, p \). This method has a very high computational cost, in fact it requires for every \( \Omega \), an orthogonalization of \( W(\Omega) \) and the computation of the singular value decomposition of an order \( p \) matrix.

A less expensive method has then been investigated. It is based on the property that the matrix \( E_s \) being orthonormal, the matrix \( T(\Omega) \) defined as:

\[
T(\Omega) = E_s^H \cdot W(\Omega) \cdot (W(\Omega)^H \cdot W(\Omega))^{-1} \cdot W(\Omega)^H \cdot E_s
\]  \(8\)

has \( \cos^2 \theta_k, k = 1, \ldots, p \) as eigenvalues, [1]. Even if the computational cost has been considerably reduced, this method still requires the maximisation of the largest eigenvalue of the \( p \times p \) matrix \( T(\Omega) \).

To cope with this problem, we will slightly change the estimation criterion. We will replace the minimisation of \( \theta_p \) by the minimisation of a convex function of all the principal angles. Moreover, we will impose the constraints:

- The function is convex on the domain \([0, \pi/2]^p\).
- The global minimum of the function is at the origin.
- The function is symmetric over its \( p \) variables.

This choice is also motivated by the fact that, according to the inequality between the \( \theta_k, \mathcal{R}W(\omega) = \mathcal{R}E_s \) is equivalent to \( \theta_1 = \cdots = \theta_p = 0 \). So, in the ideal case, the two criteria are identical. According to these remarks, we decide to minimize the new cost function \( \nu(\Omega) \) defined as:

\[
\nu(\Omega) = -\prod_{k=1}^{p} \cos^2 \theta_k
\]  \(9\)

\[
= -|T(\Omega)|
\]  \(10\)

\[
= -\frac{\text{abs}|W(\Omega)^H \cdot E_s|^2}{|W(\Omega)^H \cdot W(\Omega)|}
\]  \(11\)

We can note that the general approach described above includes the Weighted Subspace Fitting criterion of M. Viber and al., [4]. In fact, this last criterion corresponds to the maximization of the function \( \sum_{k=1}^{p} \cos^2 \theta_k \).

2.2. Harmonic related frequencies case

In this particular case, the cost function \( \nu(\Omega) \) is a one dimensional function of the fundamental frequency \( \omega \). We hence have to perform a one dimensional optimization of \( \nu(\nu) \). Moreover a fundamental property of \( W(\omega) \) is that, due to the particular relation between the frequencies, this matrix has a double Vandermonde struc-
ture, vertical and horizontal. So, $W(\omega).W(\omega)^H$ and $W(\omega)^H.W(\omega)$ have both a Toeplitz structure. This leads to the efficient formula for the criterion $v(\omega)$:

$$v(\omega) = -\frac{\text{abs}|W(\omega)^H.E_\omega|^2}{\prod_{k=1}^p \epsilon_k^2(\omega)}$$ (12)

where $\epsilon_k^2(\omega), k = 1, \ldots, p$ are the variances of prediction errors obtained applying the Levinson recursion to $W(\omega)^H.W(\omega)$.

2.3. Optimization algorithm

The cost function $v(\omega)$ is minimized using a Newton method [3]. Using the expression of the differential of the determinant of a matrix, we obtain the following expression of the first derivative for $v(\omega)$:

$$v'(\omega) = 2.v(\omega).\Re\text{tr}(W'(\omega)^H.E_\omega.(W(\omega)^H.E_\omega)^{-1} -$$

$$W'(\omega)^H.W(\omega).(W(\omega)^H.W(\omega))^{-1})$$ (13)

where $\Re\text{tr}Q$ is the real part of the trace of $Q$. This expression can be efficiently evaluated using the structure of $W(\omega)^H.W(\omega)$. The formula of the second derivative of $v(\omega)$ being rather complicated, this quantity is computed using finite difference (quasi-Newton). An initialisation scheme corresponding to the studied problem is presented in the simulation section.

3. MULTIPATHS DELAY ESTIMATION

This method will be applied to the multipaths delay estimation in SONAR signals.

3.1. Problem formulation

The SONAR signal envelope is composed of a direct path returned from reflection on the target additioned with multipaths returned from reflection on sea surface. When the sea surface is smooth, the intensity of sound reflected from the sea is nearly equal to that incident upon it, and so this reflection can falsify and even prevent detection and location of the target. Consequently we have to separate the instants of reflection on sea surface from the instant of reflection on the target, we hence have to estimate the multipath delays. Figure 1 represents the first three successive reflections.

(i) is the direct path: SONAR-Target-SONAR.

(ii) is the first reflection on the sea surface: SONAR-Target-Sea Surface-SONAR.

(iii) is the second reflection on the sea surface: SONAR-Sea Surface-Target-Sea Surface-SONAR.

If we note the delays of these successive reflections respectively $t_d$, $t_1$ and $t_2$, we notice that:

$$t_2 - t_1 = t_1 - t_d$$ (14)
$$t_2 - t_d = 2.(t_1 - t_d)$$ (15)

and more generally:

$$t_k - t_d = k.(t_1 - td)$$ (16)

where $t_k$ denotes the $k^{th}$ instant of arrival.

Taking into account the fact that the delays of the reflections are regularly spaced, (16), this signal can be modeled as:

$$x(t) = \sum_{k=1}^p A_k.s(t - k.\tau)$$ (17)
where \( s(t) \) is the direct received signal, \( A_k \) and \( k\tau \) are respectively the amplitude and the delay of the \( k^{th} \) reflected path, \( (\tau = t_1 - td) \). We assume there is no noise and the propagation affects linearly the amplitude of the emitted signal. If we note \( X(f) \) the Fourier transform of \( x(t) \) and \( S(f) \) the Fourier transform of \( s(t) \), we obtain:

\[
Y(f) = \frac{X(f)}{S(f)} = \sum_{k=1}^{p} A_k \exp(-2\pi j f k \tau)
\]  

(18)

If we consider now the \( N \) points signal \( Y_k \) obtained sampling uniformly \( Y(f) \) in the frequency domain where \( S(f) \neq 0 \), we are reduced to the estimation of \( p \) harmonic related frequencies.

3.2. Simulation results

Figure (2) represents 512 data points of a band-pass filtered SONAR signal with multipath. The signal \( s_n \) is estimated from the 100 first samples of this signal. The signal \( Y_k \) is obtained using a Fast Fourier Transform of \( s_n \) and \( x_n \). Figure (3) represents a Fast Fourier Transform of \( Y_k \). The attenuation due to the successive reflections are clearly visible on this plot. The optimization algorithm is initialized with the frequency associated to the most powerful spectrum line (the first reflection). Moreover, we have chosen \( p = 5 \) and \( M = 8 \). The estimated fundamental frequencies of the seven first iterations given by the algorithm and converted in time samples are: 500, 745, 526, 523, 521, 522, 522. The first and last one of these values are represented on the figure (2) by dashed and solid lines. The corresponding value calculated according to the geometry of the system measured during the experiment is 530 samples. In regard to these results, the proposed method is a good solution to the multipaths delay estimation problem.

References

WHAT IS A GOOD ESTIMATE?

P. L. Combettes and W. W. Edmonson

1Department of Electrical Engineering, City College and Graduate School, City University of New York, New York, NY 10031, USA.

2Computational Science and Engineering Research Center, Howard University, Washington, DC 20059, USA.

Numerous point and set theoretic estimation procedures have been used in signal processing. One can however question the adequacy of standard estimation objectives and their subsequent performance evaluation criteria in practical applications. In this paper, we address these questions and analyze the important factors that should be involved in the rational quest for meaningful estimates.

1. INTRODUCTION

The goal of most signal processing and systems theory problems is to produce estimates of objects relevant to specific analysis or synthesis purposes. Thus, in system identification, one estimates the parameters involved in the model of a system; in signal recovery, the goal is to estimate the original form of a signal; in spectral estimation, the objective is to estimate the spectral distribution of a stochastic process.

One tends to select an estimation scheme on grounds which are seldom related to rational and practical goals reflecting the specificities of the problem at hand. In general, one favors standard estimation procedures that provide, without too much effort, a solution. This point is illustrated by the widespread use of the maximum likelihood estimator, whose practical virtues have yet to be rationed out but which yields a solution in a fairly straightforward manner. In most cases, little or no attention is paid to critical factors such as the incorporation of a priori information and the relevance of the estimation objective. Thus, procedures which yield “optimal” solutions have been very popular despite the fact that the optimality criterion is often arbitrary and that the resulting optimization problem is not easily modifiable to incorporate whatever a priori information may be available about the problem. An alternative approach, which has proven effective in a steadily increasing number of systems theory and signal processing problems, is that of set theoretic estimation. In that framework, more emphasis is placed on the uncertainty that surrounds the problem. The requirement for a single estimate is relaxed and the solution is a set of equally valid solutions.

Various definitions for “good” estimates have been proposed. Thus, some will argue that a maximum likelihood estimate is desirable while others will discount it on account of its many pathologies. In fact, such pathologies exist for almost every type of estimation procedure and have given rise to comparable controversies [6], [7], [8], [9], [14].

The purpose of this paper is not to propose a formal and universal definition of what a good estimate should be but, rather, to discuss the various factors that should be involved in the selection of an estimation method in order to obtain physically meaningful estimates in an objective manner. In the next section, we briefly describe the most common estimation procedures. In Section 3, we proceed to analyze the tools available to evaluate the performance of these procedures. We discuss various estimation procedures in Section 4 and conclude the paper in Section 5.

2. ESTIMATION OBJECTIVES

2.1. General Model

Two key elements in an estimation procedure are the solution space and the observation space. The solution space $\Xi$ is the space to which the true object $h$ (the state of nature) belongs. Depending on how $h$ is modeled, this space can take various forms, e.g., a field of scalars, a matrix space, a functional or distributional space. On the other hand, the observation space $\Delta$ is the space to which the observed data $x$ belong. An estimation procedure is a mapping $T$ which assigns to the observed data $x$ a subset $T(x)$ of $\Xi$, that is,

$$T: \Delta \to 2^\Xi$$

$$x \mapsto T(x). \tag{1}$$

The set of points $T(x)$ represents our guess of the value of the true state of nature $h$ given the data $x$ and some $a$ priori knowledge.

In statistics, the data model is probabilistic. If $(\Omega, \mathcal{F}, \mathbb{P})$ denotes the underlying probability space, the observed data are regarded as a realization $x = X(\omega)$ of a stochastic process $X : \Omega \to \Delta$ and a set theoretic estimator is a measurable map $\mathcal{M} = T \circ X : \Omega \to \Xi$, where $\Xi$ is a $\sigma$-algebras of subsets of $\Xi$ [1].
2.2. Point Estimates

The objective of a point estimation procedure is to provide a single estimate, i.e., $T(x) = \{a(x)\}$. The estimate $a(x)$ is usually obtained by optimizing an objective function over $\Xi$. Various optimality criteria have been proposed for point estimates, e.g., maximum likelihood, minimax, maximum entropy, maximum a posteriori and other Bayesian criteria [1], [10], [12]. We describe the main approaches here.

Let $(\Omega, F, (P_h)_{h \in \Xi})$ be the underlying statistical model, assumed to be dominated by $P$. Then a maximum likelihood estimator is a point $a(x)$ which maximizes the likelihood function $L(a) = dP_a/dP$ over $\Xi$. In the Bayesian approach, the true object $h$ is regarded as a realization of a random variable whose distribution (the prior) is known. Moreover, one selects a function $L$, where $L(a, h)$ is the loss associated with estimating $h$ by $a$. The associated Bayesian estimate $a(x)$ is the point which minimizes the posterior expected loss, i.e., the expected loss conditioned on the data. The posterior distribution involved in the computation of a Bayesian estimate comprises a prior distribution, which reflects the a priori information on $h$, and a likelihood function, which reflects sample information. In the minimax approach, one seeks to minimize the maximum expected loss, which is usually regarded as a conservative procedure.

2.3. Set Estimates

The result of a set theoretic estimation procedure is not a single point but a subset $T(x)$ of $\Xi$, which usually does not reduce to a singleton. Several approaches have been proposed to define $T(x)$.

In statistics, confidence regions constitute a well-established set theoretic method of estimation [1], [12]. Let $\omega$ be the elementary event giving rise to the observation $X(\omega) = x$ of the data process. A confidence region is a subset $T(x) = T(X(\omega))$ of $\Xi$ such that

$$\forall h \in \Xi \quad P_h(\omega \in \Omega | h \in T(X(\omega))) \geq \alpha,$$

where $(\Omega, F, (P_h)_{h \in \Xi})$ is the underlying statistical model.

Another set theoretic approach is the so-called Bayesian framework proposed in [5] and further discussed in [13]. In Bayesian point estimation, the specification of a prior distribution is critical for the estimator often lacks robustness, particularly if the data record is not extensive. One is required to select a single prior distribution that will properly model a priori knowledge and, at the same time, yield a tractable optimization problem. A way to relax this requirement while remaining faithful to the Bayesian philosophy is to consider a set of prior distributions, each of which is an equally acceptable candidate to model the uncertainty surrounding the true state of nature. An estimate is obtained for each prior distribution and $T(x)$ is therefore a set.

In [4], a general set theoretic estimation framework is presented in which the emphasis is placed on consistency with respect to all the information available about the problem, i.e., the a priori knowledge and the data. Let $\Psi : \Xi \rightarrow [0, 1]$ be the fuzzy logic proposition associated with a particular piece of information. If $(\Psi_i)_{i \in I}$ is the family of all such propositions, given a real number $\psi$ in $[0, 1]$, a family $(S_i)_{i \in I}$ of property sets in the solution space $\Xi$ can be constructed as follows

$$\forall i \in I \quad S_i = \{a \in \Xi | \Psi_i(a) \geq \psi\}.$$  (3)

$S_i$ is the set of all estimates which are consistent with the information carried by $\Psi_i$ at level $\psi$. The pair $(\Xi, (S_i)_{i \in I})$ is called a $\psi$-level set theoretic formulation. The solution is the feasibility set $T(x) = \bigcap_{i \in I} S_i$. Any point in $S$ is called a set theoretic estimate. Although it is non-statistical, this set theoretic framework encompasses the two above set theoretic procedures if appropriate statistical information is present. Indeed, the sets produced by these procedures can be incorporated into the set theoretic formulation as specific property sets, i.e., $S_i = T(x)$.

3. EVALUATION TOOLS

It is generally accepted that an estimation procedure should be accompanied by an estimate of its accuracy. However, as noted by Savage [12], there are conceptual problems associated with evaluation procedures. First of all, it creates an endless regression since an estimate of accuracy should be accompanied by an estimate of its own accuracy, etc. Second, one can question the purpose of an evaluation procedure. Except for satisfying our curiosity, there is little practical use to know how accurate the estimate is once the estimation process is completed. One could argue that evaluation tools may help in deciding that a given estimate is better than another, but such a decision is subjective since it is based on the choice of the accuracy criterion.

3.1. Point Estimates

There exist a plethora of tools for evaluating point estimates and we mention only the most important here.

A popular index to evaluate the accuracy of a point estimate is the mean-square error, $E[a - \hat{a}]^2$, which reduces to the variance in the unbiased case. In connection with this index, the so-called Cramér-Rao\footnote{Historical note: the name of the bound refers to the work of Rao (1945) and Cramér (1946) but, is not justified. Indeed, the bound was derived by Fréchet in 1943 and, for the multi-parameter case, by Darmois in 1943 [12].} bound (CRB) has become the most often used criterion in determining the accuracy of an estimator. The CRB represents the minimum variance that can be achieved by an unbiased estimator. Its popularity is mainly due to the fact that it is usually easy to compute. Another way to evaluate the accuracy of an estimation procedure is to compute the divergence between the actual probability structure and the probability structure resulting from the estimate. These criteria fall into three basic
categories: the generalized Kolmogorov variational distance, the $f$-divergence, and the Chernoff distance (e.g., see [2], [11] and the references therein). The Kullback-Leibler and Bhattacharyya distances are two of the more popular criteria of this class.

The CRB and the information and distance criteria are based on two assumptions, infinite sample and known probability distributions, which prove to be unrealistic in practice. The number of samples is always finite where these criteria are assuming that we have infinite data. In most instances, knowledge of the probability distributions is either incomplete or nonexistent. Under these conditions, estimates of the probability distributions should be used, but they do not perform as well as predicted from theory. Finally, let us stress that the CRB is useful only for unbiased estimates. This is a severe restriction since there is no sound theoretical or practical reason to prefer unbiased estimates to biased ones.

Utility [12] is a statistical concept that has received little attention in the evaluation of estimators. Utility is used along with the above criteria. It is a function of the intended use of the estimate and of the risk and benefits associated with the criterion. As an example, suppose that we are concerned with estimating the spectral content of a time series only within a specific frequency band. One estimator is more accurate over the complete spectrum while the second is more accurate within the desired frequency band. Then the second estimator has more utility than the first.

Finally, let us mention that a straightforward way to evaluate the accuracy of an estimation procedure is to run Monte Carlo computer simulations to obtain a qualitative indication of the dispersion of the estimates over realizations. This brute-force approach does not rely on a particular analytical criterion but provides a global qualitative description of the accuracy of the estimation procedure.

3.2. Set Estimates

The accuracy of a set theoretic estimation procedure can be assessed by the dispersion of the estimates over the solution set. In [3], various monotone set functions, called measurments, were proposed for this purpose. Assuming that the solution space $\mathcal{S}$ is a metric space with distance $d$, examples of measurements of a solution set $S$ are the diameter of $S$; the thickness of $S$ (i.e., the diameter of the largest open ball contained in $S$); $\mu(S)$, $\int_0^\infty \delta d\eta$, and $\mu$-ess sup $\{\delta(a) \mid a \in \mathcal{S}\}$, where $\mu$ is a measure and $\delta$ a proper deviation function.

4. DISCUSSION

We have seen that two main types of estimation procedures exist. One, in which there is a single "good" estimate which is optimal in some predetermined sense and one in which there is a set of equally "good" estimates, usually determined in terms of some feasibility argument.

Following Zadeh [14], we can can question the rationality of the quest for optimal estimates when different users, through their own interpretation of the optimal way to solve the problem, may obtain different solutions. In fact, our insistence on optimal solutions often leads to arbitrary decisions because the selection of a criterion of performance is inherently subjective and solving the problem may require oversimplifications in its formulation. For some particular point estimation procedures, more specific reservations can be expressed. For instance, the maximum likelihood (ML) estimator has enjoyed widespread use on account of its "desirable" properties such as normality and efficiency. However, the practical validity of these properties is quite debatable and, moreover, they are only asymptotic. Theoretically, the ML estimator may not be defined or lead to irrelevant results. Finally, computational tractability generally imposes standard assumptions, such as normality of the random processes involved, and leaves almost no room for the incorporation of a priori knowledge. On that last point, the Bayesian framework is better suited. However, it requires to model the object $h$ to be estimated as the realization of a random variable to which a distribution function is assigned (in most situations, there is however nothing random about $h$, which may for instance be a fixed but unknown physical constant). In addition, not all a priori information can be conveniently described in probabilistic terms and the resulting prior distribution is usually too complex to yield a tractable minimization of the resulting conditional expectation. At all events, a Bayesian technique is fundamentally subjective since the specification of the prior distribution and of the loss function is left to the user. In the extreme, the user could reach any conclusion he desires through biased choices of these specifications.

The central principle governing set theoretic estimation is that of feasibility. An acceptable solution is any object which is consistent with all the information available about the problem, namely the data and the a priori knowledge. By recasting this concept in a solution space, the estimation problem is to find a point in the intersection of property sets, each of which represents the class of objects consistent with a particular piece of information. It is clear that set theoretic estimation departs radically from the conventional approach which relies on optimal decision theory and in which feasibility is either ignored or of secondary importance. In this regard, it has been charged that the validity of set theoretic methods could be questioned on the grounds that they do not yield a unique solution but a set of solutions and are therefore suboptimal. Let us first remark that, although it may be gratifying to have obtained the "best" solution, optimality claims were seen to have little practical value. At best, their result can be regarded as a qualitative selection of a feasible solution. Moreover, from a conceptual standpoint, demanding that only one solution be acceptable for problems which are notoriously affected by uncontrollable factors (e.g., noise, uncertain data formation model) seems
somewhat unwise. Let us also note that methods which yield unique solutions are usually iterative and their solution depends on the stopping rule. Certainly, such a criterion does not define a unique solution.

The basic objective of the set theoretic approach is to provide a flexible framework for the incorporation of a wide range of information. This is contingent on the availability of methods to generate feasible solutions, i.e., methods to find a point in the intersection of a collection of property sets. If all the sets are closed and convex and \( \Xi \) is hilbertian, several methods are available to solve this problem; in the nonconvex case, methods have been developed but they do not apply to all cases [4]. Hence, the lack of algorithm for the nonconvex case constitutes the main limitation of the set theoretic framework as there are many problems for which all the information cannot be associated with closed and convex subsets of a Hilbert solution space.

The selection of set theoretic or conventional estimation theory to solve a problem depends on several factors. First of all, it should be noted that even if it is at the expense of rationally positing the estimation problem, an advantage of some conventional estimation methods is to yield simple problem formulations and, in some cases, closed-form solutions. Indeed, one can always have recourse to standard cost functions and assumptions to simplify the problem. This approach has at least the merit of leading to a solution. On the other hand, although there are methods for computing set theoretic estimates for a wide class of problems, setting up a tractable set theoretic estimation formulation may not always be possible. Even if it is, the use of whatever conventional estimation method seems appropriate should not be precluded, especially if a solution can be computed efficiently. This solution can then be tested for feasibility with respect to available information. If it is feasible, it must be accepted; if not, one should expect a set theoretic solution to bring improvement. Of course, an ideal estimation procedure would be one in which a fully relevant objective function is rationally selected and minimized over the feasibility set. Unfortunately, given the state-of-the-art in constrained optimization theory, this approach seldom viable.

5. EPilogue

The title of this paper is similar to Alice in Wonderland asking the Cheshire Cat, “Would you tell me, please, which way I ought to go from here?” - “That depends a good deal on where you want to go,” said the Cat. Then the Cat began to vanish, first the tail and then the body of conventional statistical methods. The last visible part was the set theorist’s grin, “which remained some time after the rest of it had gone”. But that, too, disappeared.

REFERENCES

OPTIMAL RECONSTRUCTION OF TRANSIENT SIGNALS USING DECONVOLUTION ALGORITHMS

T. Dabóczi and I. Kollár

Department of Measurement and Instrument Engineering, Technical University of Budapest
H-1521 Budapest, Hungary, Phone, fax: +36 1 166-4938
c-mail: h4427dab@illa.uwcp

Estimation of the transient input signal of a system from its measured output is an often arising problem, because a dynamical system usually significantly distorts the waveform. There are several inverse filtering methods for this purpose. However, the optimization method is generally heuristic, and dedicated for a certain inverse filtering algorithm.

In this paper a new approach is proposed to calculate the error energy in the estimation of the input signal as a tool to find the optimal reconstruction in least squares sense. The proposed algorithm requires a rough approximation only of the absolute value of the input spectrum and noise variance. It is shown that the quality of the optimal reconstruction is robust with respect to slight errors in input spectrum and noise variance. The proposed method can be applied for any inverse filtering algorithm to get the optimal reconstruction.

Keywords: inverse filtering, deconvolution, signal reconstruction, optimal filtering.

1. Introduction

The measurement of fast transient signals is usually a difficult task, since the bandwidth of the measuring instrument is often lower than that of the measured signal, that is, the measuring instrument distorts the waveform. When this distortion is not acceptable, the effect of the system should be compensated for. Considering a linear time invariant measuring system, the relation between the input and output signals can be described by convolution:

\[ y(t) = x(t) * h(t), \]  

where \( x(t) \) is the input signal, \( y(t) \) is the output signal, and \( h(t) \) is the impulse response of the system. The exact output waveform is usually not known, because the measured signal is always corrupted by noise (electromagnetic interferences, quantization noise etc.): 

\[ y_n(t) = x(t) * h(t) + n(t), \]  

where \( y_n(t) \) is the observed (noisy) signal and \( n(t) \) is the noise component. In the frequency domain this convolution corresponds to multiplication:

\[ Y_n(f) = X(f)H(f) + N(f), \]  

where the capital letters correspond to the frequency domain signals. For the input signal the following estimator eliminates the effect of the measuring system:

\[ X_{\text{est}}(f) = \frac{Y(f)}{H(f)} = \frac{X(f) + N(f)}{H(f)}, \]  

where \( X_{\text{est}} \) is the estimated input spectrum. From Eq. (4) it can be seen that the noise will be amplified at the frequencies where \( H(f) \) is small, usually in the high frequency band. The effect of the noise increases rapidly with the order of the transfer function of the system, and makes the estimation result unusable. It will be called rough deconvolution. Therefore, the noise should be suppressed to get a smooth result. Since the noise and the useful signal component usually cannot be separated, the noise reduction will lead to a bias in the estimation. A compromise has to be found between variance and input signal distortion.

There are several factors which influence the performance of the deconvolution:

1. input signal shape (or, equivalently, its spectrum),
2. transfer function of the measurement system,
3. distribution and level of the measurement noise,
4. errors in the identification of the measurement system,
5. deconvolution algorithms,
6. degree of noise suppression vs. bias.

Throughout this paper the first three factors will be supposed to be given.
In the deconvolution process the transfer function of the system is assumed to be exactly given. In practice this is not true, since non-parametric identification is also a deconvolution problem, and the noisy measurement causes similar problems as above. Fortunately, in many cases the input signal estimation is not too sensitive with respect to slight identification errors.

The deconvolution algorithm can be chosen by the user. There are several methods described in the literature. The different error criteria lead to different deconvolution algorithms [1]-[7]. The obtained results are close to each other, though they are not the same.

The degree of noise reduction vs. bias can be adjusted by a certain parameter in every algorithm, which can be either manually or automatically chosen. However, in the case of transient deconvolution the optimal setting of deconvolution parameters is often heuristic. A systematic approach, deduced from a mathematical model, is usually lacking.

In this paper the tradeoff between bias and noise variance is studied. A new method is proposed to calculate the approximate error energy of the input signal estimation as use a tool to find the optimal deconvolution in least squares (LS) sense. Since energy estimation is approximated, the given result is suboptimal, but it will be shown that the obtained optimum is very close to the exact LS solution.

2. MODEL FOR THE APPROXIMATION OF THE ERROR ENERGY

The error energy of the input signal estimation can be calculated for a simplified model of the measurement setup. The model is shown in Fig. 1.

\[
EE = \sum_{n=-\infty}^{\infty} (x(n) - y(n))^2
\]

(5)

where \( EE \) is the error energy, \( T_s \) is the sampling period, \( x(n) \) is the measured input signal at time instant \( nT_s \) and \( y(n) \) is the estimated input signal. The energy can be computed in the frequency domain using Eq. (4) and Parseval's theorem:

\[
EE = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left| X(e^{j\omega}) - Y(e^{j\omega}) \right|^2 d\omega
\]

(6)

where the capital letters correspond to the Fourier transforms of the signal sequences, \( C(e^{j\omega}) \) is the transfer function of the discrete signal and noise shaping filter, and \( \phi \) is the angle of the two absolute valued terms in the last integral. The error energy is split into three terms, where \( EE_{bias,noise} \) is due to the cross connection of the bias and noise term. The theoretically correct error energy can be calculated from Eq. (6). In practice, however some approximations have to be used because of the following reasons:

1. The signal sequences can only be measured in a finite time interval.
2. The noise and input signal spectra are not exactly known.

The finite time record causes edge effects, but these can be neglected if the measurement time is long enough, compared to the transient duration.
Another problem is that the noise is a stochastic signal, and the longer is the measurement time, it has the more energy. In order to avoid this uncertainty, the record length must be standardized.

For the second problem an approximate solution is proposed. In many cases the observation noise can be assumed to have a white spectrum, its variance can be more or less accurately measured. With the above assumptions the output noise spectrum can be approximated by a constant.

The input spectrum can be approximated on the basis of rough deconvolution (Eq. [4]). Considering a lowpass system, the noise will be significantly amplified only at high frequencies, the low frequency band remains useful. The absolute value of the input spectrum can be approximated by fitting an appropriate signal model to the low frequency part of the spectrum of the rough deconvolution. For the signal model a differential equation of appropriate order can be given, which can be well fitted in frequency domain by adjusting the coefficients.

In the term $E_{bias,noise}$ the angle of two terms should be given. Since the phase of the spectra of the noise and the input signal is not known, an upper limit can be given by $\cos(\phi)=1$. From the simulation it was found that usually the first two terms dominate in the energy expression (7), and this last term can be neglected. In the following sections only the bias and noise term will be calculated.

In the next sections the proposed method will be verified by simulations and experimentally.

![Fig. 3. Simulated input and output signals](image)

3. STUDY OF THE PERFORMANCE LIMITATION

A standard test signal [1] was used to investigate the input error energy and to verify the accuracy of the proposed approximation. The simulated signals are shown in Fig. 3. The two ramps form a good model for front chopped high voltage impulses. A second order system was taken into consideration by applying the backward difference transform to $H(f)$. The output signal was corrupted by white Gaussian noise sequence with 40 dB signal-to-noise ratio, where the SNR is defined by:

$$\text{SNR} = 10 \log_{10} \left( \frac{1}{T} \int_{0}^{T} y(t)^2 \, dt}{\sigma^2} \right),$$

where $\sigma^2$ is the noise variance. For the inverse filter the constrained least squares solution was used [2], [4]:

$$K(f) = \frac{|H(f)|^2}{|H(f)|^2 + \lambda},$$

where $\lambda$ is the parameter which has to be optimized to find the optimal compromise between bias and variance. (The measurement setup and deconvolution algorithms given here are only examples for the problem of the inverse filtering.)

![Fig. 4. Deconvolution results with different settings of the optimization parameter](image)
spectrum of the rough deconvolution. The input spectra are to be seen in Fig. 5. The model approximates well the absolute value of the input spectrum. The error energy of the deconvolution process was computed for different λ values, substituting the approximated input spectrum into the first two terms of Eq. (6). The true and the approximated error energies are shown in Fig. 6. The estimation is close to the true one. The relative error is no more than 40%. The most important aspect is the precision of the minimum detection since the main goal is the optimal reconstruction. The approximated minimum was found by a 30% precision. Fortunately, the constrained LS algorithm is not too sensitive with respect to the λ value, the input estimation with approximated λ_{min} differs only at few points from the LS solution (Fig. 7), what means that the result obtained by the approximated parameter can be considered to be optimal.

Fig. 5. The input (---), noisy input (- - -) and approximated input spectra (---)

The precision of the energy estimation depends on the approximations of the input spectrum and noise variance. In the next sections we are going to investigate, how robust is the proposed approximation with respect to errors in the input signal model and the output noise level.

Fig. 7. Optimal deconvolution in LS sense (---), and the approximated optimum (- - -)  

Fig. 8. Badly modeled input signal  

Fig. 9. Estimation of the error energy using a bad input signal model

3.1. Bad input signal model

The time constants of the spectrum estimation were drastically modified, by a factor of about 2. The badly modeled spectrum is drawn in Fig. 8. The error in the EE estimation was increased to 200% at larger λ values, where the bias term dominates (Fig. 9), but the minimum
was obtained with about 60% error only. The LS and the obtained optimal deconvolution are close to each other (Fig. 10). The rising slope is nearly the same, while the peak is about 2% higher than that of the LS solution. Such a large error can usually be avoided at the input spectrum, so the precision of the minimum detection is robust enough with respect to the input signal model.

3.3. Experimental verification

A second order Sallen-Key lowpass filter was excited with the input signal shown in Fig. 3. The time constants of the filter are the same as in the simulations. A Gaussian noise was added to the output of the filter by a noise generator (SNR = 39 dB) and the noisy response was measured. For the input signal a smooth optimum was obtained, depicted in Fig. 11.

4. CONCLUSIONS

A new approximate method was proposed to estimate the error energy in deconvolution, which makes possible to find the optimal solution in LS sense. Since the error energy is an approximation, the obtained optimum is suboptimal, but it was shown that the solution is very close to the true LS one. The calculation of the error energy requires the estimation of the noise level and a rough approximation about the absolute value of the input spectrum. It was shown that the optimum is robust with respect to errors in estimation of the noise level and input spectrum, respectively. Furthermore, the optimal reconstruction can be found for any deconvolution algorithms, it requires only the knowledge of the signal and noise shaping filter characteristics.

References:

UNIFICATION OF SEVERAL APPROACHES TO MEAN FREQUENCY ESTIMATION FOR DOPPLER DETERMINATION

G. D. Cain and A. Yardim

School of Electronic & Manufacturing Systems Engineering
Polytechnic of Central London
115 New Cavendish Street
London W1M 8JS, England

1. ABSTRACT

Doppler tracking may be viewed as a progression of short-term evaluations of the mean of the energy density spectrum of moving-windowed fragments of the time signal. This is particularly appropriate when the spectrum is broadband, failing to exhibit the near-monochromaticity that so significantly aids some frequency estimation techniques. Several methods employing differing mixes of correlation, differentiation and Hilbert transformation to achieve such spectral "centroid tracking" frequently crop up in frequency estimation studies. Each appears in isolation and, since there is no guidance given on alternatives, readers are faced with a number of dissimilar-looking formulations all aiming at a common measurement goal. These individual competitors are shown to be variants of the same basic calculation, and prime candidates for a unified treatment. A "square-root" filter is shown to be a very cost-effective replacement for most of the variants examined.

2. BROADBAND EXPRESSIONS

The measurement of frequency is an instrumentation problem elemental in nature and of seemingly never-ending interest to engineers. Scarcely a month goes by without a leading technical journal carrying a paper examining this issue from yet another perspective.

The wealth of literature which has built up can crudely be partitioned into two segments: that dealing with pure sinusoidal models contaminated by noise and that aimed at broadband spectral characterization resulting in descriptors of "a frequency" somehow representative of the gross shape of the spectrum. Reference [1] gives an authoritative background relating to the first category, while papers in the second category - most often concerned with Doppler frequency determination - are typified by the classic [2]. There have been relatively few attempts to draw together the disparate estimation approaches into quantified comparisons; among these, [3]-[7] attacked subsets (for digital estimation) of that problem. Though the zero-crossing method is simple and extremely popular, a second approach - that of spectral moments - is also very prominent. This paper focuses on deterministic calculation of spectral means, and is cast in continuous-time terms prior to arriving at a digital filter-based formulation of a proposed estimator structure.

We start by assuming a real-valued signal \(s(t)\) possessing an energy density spectrum \(|S(f)|^2\) and a (time) autocorrelation function \(R_s(t)\). Our interest centres on determining the centroid of the positive-frequency portion of the (even-symmetric) energy density spectrum:

\[
fm = \frac{\int_{0}^{\infty} f|S(f)|^2 \, df}{\frac{1}{2} R_s(0)}
\]

(1)

where normalization by half the signal's energy, \(R_s(0)\), is - as in analogous moment calculations for force distributions - required for correct calibration of the energy centroid location, \(f_m\). Our efforts are directed at evading explicit evaluation of the integral in (1), relying instead upon various...
arrangements of simple transform properties to permit our processing - though equivalent to (1) - to be wholly couched in the time domain.

The key is to trade for a double-sided integral so that we tackle

\[ \int_{-\infty}^{\infty} |f| S(f) S^*(f) \, df = R_s(0) f_m \quad (2) \]

The possibilities which open up for processing strategies become obvious when we make just one of the possible re-castings of a term that lies at the heart of the integrand:

\[ |f| = (j2\pi f)(-j\text{sgn}\, f)(\frac{1}{2\pi}) \quad (3) \]

since this can be readily recognized as the spectral effect imparted by the (scaled) cascading of a differentiator and a Hilbert transform filter. This immediately allows us to write

\[ 2\pi R_s(0) f_m = \int_{-\infty}^{\infty} [j2\pi f] S^*(f)(-j\text{sgn}\, f) S(f) \, df \]

or

\[ f_m = \hat{R}_s(0) / 2\pi R_s(0) \quad (4) \]

where \( R_s(\tau) \) is our nomenclature for the cross-correlation of the original signal and its Hilbert transform, \( \hat{S}(t) \), and \( \hat{R}(\tau) \) signifies the \( \tau \)-derivative of a correlation function.

Of course (5) suggests a form of hardware that could be employed for a Doppler-determining system. However, this is but one of a multitude of formulations that can be obtained by rearrangement of the groups of ingredients in our fundamental equation (4). About the simplest change from (5) is to associate the differentiation with the \( S^*(f) \) term in (4) so that

\[ f_m = -R_{ss}(0) / 2\pi R_s(0) \quad (6) \]

Or, we could pull both operations out so that only the autocorrelation function is worked upon:

\[ f_m = \hat{R}_s(0) / 2\pi R_s(0) \quad (7) \]

This type of arrangement is alluded to in [8] and made very explicit in [9]. Clearly there are numerous variations possible that involve mixes of operations on the raw signal and correlation functions derived from it.

3. BASEBAND AND INSTANTANEOUS FREQUENCY ALTERNATIVES

We can move away from the formulation where processing takes place at the original spectral location of \( s(t) \) (which we call the "broadband" processing situation) and down-modulate to a baseband scenario. One particularly popular embodiment involves derivatives of in-phase and quadrature components (e.g. [10] and [11]).

If we take \( f_o \) as a convenient "carrier frequency" to be nearby \( f_c \), then we need to find the offset frequency \( f_o \) that corrects to give our mean:

\[ f_m = f_c + f_o \quad (8) \]

We can do this by a complex envelope description of our given \( s(t) \):

\[ s(t) = \Re [x(t)e^{j2\pi f_c t}] = \Re [(x(t) + jy(t))e^{j2\pi f_c t}] \]

where \( x(t) \) is the in-phase component of \( s(t) \) and \( y(t) \) is the associated quadrature component (for the given \( f_c \) value assumed). Substituting \( z(t) \) to the same kind of energy density centroid calculation as was used in (1), we immediately obtain

\[ f_o = \frac{jR_{ss}(0)}{2\pi R_s(0)} = \frac{\int [x(t)y(t) - x(t)y(t)] \, dt}{2\pi \int [x^2(t) + y^2(t)] \, dt} \quad (10) \]

where over-dots denote time derivatives. This baseband formulation not only enables us to treat low-frequency signals, but also permits an analytically handy doubling-up of integration that would not have been meaningful in the context of (1) (due to the symmetry of \( |S(t)|^2 \) for the real signal \( s(t) \)). Of course, the price is the double processing load associated with a complex envelope signal \( z(t) \).

It is no accident that (10) bears a strong resemblance to \( z(t) \)'s "instantaneous frequency" definable [12] as

\[ f_i(t) = \frac{d}{dt} \left( \tan^{-1} \left[ \frac{y(t)}{x(t)} \right] \right) = \frac{x(t) \dot{y}(t) - y(t) \dot{x}(t)}{x^2(t) + y^2(t)} \quad (11) \]

and, as pointed out by Papoulis [13, p. 321], a
mean frequency based on statistical autocorrelation is a weighted average which has the same form as (10), where the counterpart of time integration is expectation.

Obviously we can again proceed to move the differentiation operation about to modify (10); Angelsen and Kristoffersen prefer

\[ f_o = -j \hat{R}_x(0) \over 2\pi \hat{R}_x(0) \]  

(12)
in their series of useful papers on this topic [4].

4. MORE DIRECT DISCRETE-TIME BROAD-BAND APPROACHES: A VEE FILTER AND A SQUARE-ROOT FILTER

There are five papers ([8], [9], [14] - [16]) known to us which set out the case for time-domain broadband mean frequency calculation in its very essence, without the trappings of differentiation and Hilbert transformation which are unnecessary in the context of this particular problem. Curiously enough, all four authoring teams have done their work in connection with biomedical measurements.

In [8], [9] and [16] what we call the "vee digital filter" seems to have been independently derived. The idea is to view the discrete-time counterpart of (2) (where \( \nu \) is normalized frequency and \( k \) is the discrete-time index),

\[ \nu_m \hat{R}_x(0) = \int_{-0.5}^{0.5} S(\nu)\nu^2 d\nu = \sum_{k=-\infty}^{\infty} h_\nu(k) \hat{R}_x(k) \]  

(13)
as the initial value of a convolution in which a filter impulse response sequence \( h_\nu(k) \) fully engages all samples of the \( \hat{R}_x(k) \) autocorrelation sequence. Of course practical realization restricts \( h_\nu(k) \) to causality and forces departure from the idealized vee-shaped transfer function \( \nu \) for \( \nu \in (-0.5, 0.5) \). Closed-form expressions for a simple FIR approximation to \( h_\nu(k) \) are given in [9], [9] and [16].

We much prefer to avoid autocorrelation calculation altogether and to concentrate instead on direct filtering of the raw signal \( s(k) \). Of course (2) suggests that this can be done by choosing a digital filter with transfer function \( \sqrt{\nu} \) for \( \nu \in (-0.5, 0.5) \). It seems that Gerzberg and Melendi were first to report this approach (e.g. [14]), though (apparently independently) [8] and [15] report the same approach. In [5] Strube points out the advantage of an FIR realization of this "square-root filter" to permit moving-window tracking of a nonstationary spectral mean. The elegance and economy of this approach, which we advocate as the preferred solution for spectral mean calculation, can be seen from the complete processing system's block diagram (where \( \alpha \) is the linear-phase square-root filter's group delay):

![Diagram](image)

Figure 1

and where "NPMA" signifies an N-point moving window averaging filter and \( \tilde{v}_s(k) \) is a time-varying estimate of the centroid of the spectrum of \( s(k) \).

The square-root filter should, in our view, be FIR and may well need to have a different number of coefficients from the NPMA smoothing filter. Aside from a single closed-form expression in [8], we have found no report of how to design the square-root filter apart from our own work in [17] and [18]. We have found this design target to be an interesting challenge, particularly as regards low frequency performance, and are extending our investigation of practical realizations. We have one 512-point square-root filter already in routine real-time use for a vehicular speed measuring testbed, yielding excellent tracking results.

5. REFERENCES


OPTIMAL SVD-BASED DETERMINATION OF NUMBER OF SINUSOIDS

H. Tsuji† and A. Sano‡

†Communications Research Laboratory, M. P. T., 4-2-1 Nukui kitamachi, Koganei 184, Japan
‡Department of Electrical Engineering, Keio University, 3-14-1 Hiyoshi, Kohoku-ku, Yokohama 223, Japan

By introducing multiple adjustable free parameters into the singular value decomposition (SVD) and determining their optimal values, the optimal truncation scheme for the singular values is derived. The number of retained singular values gives the estimated number of sinusoids in white noise. Three criteria are presented in this paper to decide the free parameters; one is the minimum MSE-like criterion, the second one is based on the generalized cross validation method, and the third one is based on the Bayesian information criterion. The new algorithms using the obtained optimal free parameters are given and compared to the AIC and MDL in numerical simulations.

1. INTRODUCTION

The eigenstructure approach has recently become a powerful tool in wide area of digital signal processing [1]. The singular value decomposition (SVD) can attain the immunity of noise effects in spectral estimation by truncating small singular values of a signal data matrix adequately [2]. The number of the retained singular values corresponds to the dimension of the signal subspace, which indicates the number of harmonics embedded in white noise. In actual cases, however, large and small singular values do not separate neatly in the presence of large noises. Therefore theoretical criteria are required to determine the number of principal singular values, which will be useful in the SVD-based signal processing.

In this paper, we introduce multiple free parameters into the SVD of the signal data matrix and determine them so as to minimize various types of criterion. We first clarify the existence of the optimal values in the sense of the minimum mean squares error (MSE). The truncation scheme of the singular values can be given by comparing the optimal free parameters with the corresponding singular values. We propose three types of criterion for deciding the free parameters by using only accessible signal data. The first is an MSE-like criterion which has been investigated by the authors [3] in ill-conditioned parameter estimation. The second one is based on the generalized cross validation (GCV) approach proposed originally in [4]. The third one is based on the Bayesian statistical approach [5]. The proposed algorithms for deciding the number of sinusoids are compared to other criteria such as the AIC and MDL in numerical simulations.

2. PARAMETRIZED INVERSE IN USE OF SVD

Let the observed signal \( \{ y_k \} \) be a complex autoregressive (AR) process described by

\[
y_k = \sum_{i=1}^{L} \gamma_i y_{k-i} + e_k
\]

where \( e_k \) is a Gaussian white noise with zero-mean and the variance \( \sigma_e^2 \). If we adopt a least squares fitting of the AR model to the given data, (1) is used as the linear prediction model as

\[
\hat{y}_k = \sum_{i=1}^{L} \phi_i y_{k-i}
\]

where \( \phi_k \) can be regarded as a prediction error defined by

\[
ed_k = y_k - \hat{y}_k \]

Then (1) can be expressed in the LP method as

\[
\begin{bmatrix}
\{ y_{L-1} \} \\
\{ y_{L+2} \} \\
\vdots \\
\{ y_N \}
\end{bmatrix}
= 
\begin{bmatrix}
\{ y_L \} & \{ y_{L-1} \} & \cdots & \{ y_1 \} & \{ x_1 \} & \{ \phi_{L+1} \} \\
\{ y_{L+1} \} & \{ y_L \} & \cdots & \{ y_2 \} & \{ x_2 \} & \{ \phi_{L+2} \} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\{ y_{N-1} \} & \{ y_{N-2} \} & \cdots & \{ y_{N-L} \} & \{ x_L \} & \{ \phi_N \}
\end{bmatrix}
\begin{bmatrix}
\{ \phi_{L-1} \} \\
\{ \phi_{L+2} \} \\
\vdots \\
\{ \phi_N \}
\end{bmatrix}
\]

or

\[
y = Ax + e
\]

where \( L \) is the order of prediction, \( N \) the data number and \( x_k \) the LP coefficients. The least squares estimate in (2) can be expressed by

\[
\hat{x} = \sum_{i=1}^{L} \frac{1}{\sigma_i} v_i u_i^T y
\]

where \( \sigma_i \) is singular value of \( A \), \( v_i \) and \( u_i \) are the singular vectors. The accuracy of the solution (4) is degraded in a case of low SN ratio. Tuft and Kumaresan [2] suggested a truncated SVD in which the number of retained singular values is chosen the number of sinusoids if it is known.

In this paper, in order to treat with the case when the number of sinusoids is unknown, we introduce multiple adjustable free parameters \( p_i \) into the expression (4) to give a parametrized estimate of the LP coefficients as
\[ \hat{x}(\rho_i) = \sum_{i=1}^{L} \frac{\sigma_i}{\sigma_i^2 + \rho_i} u_i^T y \] (5)

If \( \rho_i = 0 \) for \( i = 1, \ldots, K \) and \( \rho_i = \infty \) for \( i = K+1, \ldots, L \), then (5) is reduced to the truncated SVD-based estimate, where \( K \) is referred to as the truncation number. Generally, the problem is how to determine the optimal values \( \{ \rho_i \} \).

3. OPTIMAL FREE PARAMETERS IN SENSE OF MINIMUM MEAN SQUARES ERROR

As a criterion for determining the free parameters \( \{ \rho_i \} \) in (5), first we consider the mean squares error (MSE) criterion, which can be evaluated approximately, for sufficiently large \( N \), as

\[ MSE(\{ \rho_i \}) = E\left[ \| x - \hat{x}(\{ \rho_i \}) \|^2 \right] = \sum_{i=1}^{L} \frac{\sigma_i^2}{(\sigma_i^2 + \rho_i)^3} \| u_i^T x \|^2 + \sum_{i=1}^{L} \frac{\sigma_i^2 \rho_i^2}{(\sigma_i^2 + \rho_i)^4} \] (6)

The optimal values \( \{ \rho_i^* \} \) that minimize the RHS of (6) can be analytically given by

\[ \rho_i^* = \frac{\sigma_i^2}{\| u_i^T x \|^2}, \quad \text{for } i = 1, \ldots, L \] (7)

The aim is to determine the truncation number \( K \), which also indicates the dimension of the signal subspace. The optimal truncation problem can be formulated as the minimization of (6) subject to that \( \{ \rho_i \} \) takes either 0 or \( \infty \). The result is given as [3]:

\[ \rho_i^{(T)} = \begin{cases} 0, & \text{if } \sigma_i^2 \geq \rho_i^* \\ \infty, & \text{if } \sigma_i^2 < \rho_i^* \end{cases} \] (8)

It is seen from (8) that \( \rho_i^* \) works as threshold values for deciding whether the singular value \( \sigma_i \) should be retained or discarded. Therefore, if \( \rho_i^{(T)} \) given by (8) satisfies that

\[ \rho_i^{(T)} = \begin{cases} 0, & \text{for } i = 1, \ldots, K \\ \infty, & \text{for } i = K+1, \ldots, L \end{cases} \] (9)

(9) gives the optimal truncation law for deciding \( K \).

However, the optimal values \( \{ \rho_i^* \} \) include the unknown true values of \( \sigma_i \) and \( \sigma_i^2 \). In order to overcome this difficulty, we can take a modified MSE criterion using only accessible data [3], as

\[ EMSE(\rho_i) = \sum_{i=1}^{L} \frac{\sigma_i^2 \rho_i}{(\sigma_i^2 + \rho_i)^3} \| u_i^T y \|^2 + \sum_{i=1}^{L} \frac{\sigma_i^2 \rho_i^2}{(\sigma_i^2 + \rho_i)^4} \] (10)

\[ \hat{\sigma}_i^2(\rho_i) = \frac{1}{N-L} \left[ \| y \|^2 - \sum_{i=1}^{L} \frac{\sigma_i^2}{\sigma_i^2 + \rho_i} \| u_i^T y \|^2 \right] \] (10)

By using (8) and (9), we can obtain the truncation number \( K \) by replacing \( \rho_i^* \) with \( \rho_i \) which minimize the EMSE in (10).

4. GCV CRITERION FOR FREE PARAMETERS

We present the generalized cross-validation (GCV) criterion without using the unknown \( \sigma_i \) and \( \sigma_i^2 \) to determine the multiple free parameters. The criterion is derived on a basis of a rotation-invariant version of the ordinary cross-validation [4] as

\[ P(\rho_i) = \frac{1}{n} \sum_{k=1}^{n} \left[ \| (A x^{(k)}(\rho_i))^T y_k \|^2 \left[ \sum_{i=1}^{L} \frac{1}{\lambda_i} \right] - \sum_{i=1}^{L} \lambda_i \right] \] (11)

where \( n = N-L \) and \( x^{(k)}(\rho_i) \) represents the least squares estimate of \( x \) with the \( k \)-th data point \( y_k \) omitted. \( I_k \) denotes the \( k \)-th element of the vector. \( w_k(\rho_i) \) is the weighted function as

\[ w_k(\rho_i) = \frac{1 - q_{kk}(\rho_i)}{1 - \frac{1}{n} \sum_{i=1}^{L} \frac{1}{\lambda_i + \rho_i}} \] (12)

where \( q_{kk}(\rho_i) \) is the \( (k,k) \)-th entry of \( Q(\rho_i) \).

From the SVD point of view, we can modify the GCV criterion in terms of \( \{ \rho_i \} \) explicitly as

\[ P(\rho_i) = \frac{1}{n} \sum_{k=1}^{n} \left[ \left[ \frac{1}{\lambda_i} \right] - \sum_{i=1}^{L} \frac{1}{\lambda_i + \rho_i} \right] \] (13)

where \( \lambda_i \), for \( i = 1, \ldots, n \), are the eigenvalues of \( A A^H \) and \( \lambda_i = \sigma_i^2 \) for \( i = 1, \ldots, L \), and \( \lambda_i = 0 \) for \( i = L+1, \ldots, n \). We can find out the optimal free parameters \( \{ \rho_i \} \) through minimization of (13) by applying nonlinear optimization techniques and determine the number of harmonics on a basis of (8) and (9).

5. BAYESIAN INFORMATION CRITERION FOR FREE PARAMETERS

Finally we take the Bayesian statistical approach to determine the optimal value of the adjustable free parameters in (5) as well as the optimal truncation number \( K \) by using only accessible observed data.

We first assume the prior distribution of \( x \) as

\[ P_0(x | \sigma_e, \rho_i) = \text{det}(G) \pi^{-n/2} \exp\left( -\frac{1}{2} x^H G x \right) \] (14)

where \( G \) is already defined in (12) which includes the free parameters \( \{ \rho_i \} \). Since \( e \) is a zero mean white Gaussian random vector, the conditional probability density of \( y \) is given by

\[ P(y | x, \sigma_e) = \frac{1}{\pi^{N-L} \sigma_e^{2(n-L)}} \exp\left( -\frac{1}{2 \sigma_e^2} y^H A x \right) \] (15)

Using (14) and (15), we can specify the posterior distribution, when the observed signal \( y \) is given, as
\[ P_p(x | y, \sigma^2, \rho_i) = P_p(y | x, \sigma^2) P_p(x | \sigma^2, \rho_i) \]
\[ = \frac{\det(G)}{\pi^N \sigma^2} \exp \left( -\frac{1}{\sigma^2} \left( |y - Ax|^2 + xx^T G x \right) \right) \]
\[ = L(x, \sigma^2, \rho_i | y) \] (16)

where \( L(\cdot | \cdot) \) is called the Bayesian likelihood. We take the Bayesian statistical approach [5] to give the ABIC for determining \( \rho_i \) and \( \sigma^2 \). The Bayesian information criterion (ABIC) is finally given by integrating (16) with respect to \( x \).

Then we finally obtain the form of ABIC [7] as
\[ \text{ABIC}(\rho_i) = \sum_{i=1}^{K} \log \left( \frac{\sigma_i^2}{\rho_i} + 1 \right) \]
\[ + n \log \left( |y|^2 - \sum_{i=1}^{K} \frac{1}{\sigma_i^2 + \rho_i} |u_i^T A^T y|^2 \right) \] (17)

By calculating the optimal values of \( \rho_i \) so as to minimize (17), we can estimate \( K \) from (8) and (9) by replacing \( \rho_i \) with \( \rho_i^* \) which minimize (17). However, it should be remarked that the ABIC is different from the EMSE in (10) and the GCV in (13) in that the free parameters \( \rho_i \) cannot take zero. Therefore, it is not simple to give the ABIC criterion in terms of \( K \) explicitly by putting zero and infinity on \( \rho_i \). Then by using small positive \( \delta \), we can rewrite the ABIC (17) as:
\[ \text{ABIC}(K) = \sum_{i=1}^{K} \log \left( \frac{\sigma_i^2}{\delta} + 1 \right) \]
\[ + n \log \left( |y|^2 - \sum_{i=1}^{K} \frac{1}{\sigma_i^2 + \delta} |u_i^T A^T y|^2 \right) \] (18)

where \( \delta \) should be sufficiently small positive value. Practically we may set up \( \delta \) as \( \varepsilon \sigma_i^2 \), where \( \sigma_i \) is the smallest singular value in the adopted ones and \( \varepsilon \) is chosen small such as 0.01 so that the value of \( \varepsilon \sigma_i^2 \) may not effect the calculation of estimate (5). Therefore we use the following form:
\[ \text{ABIC}(K) = \sum_{i=1}^{K} \log \left( \frac{\sigma_i^2}{\varepsilon \sigma_i^2} + 1 \right) \]
\[ + n \log \left( |y|^2 - \sum_{i=1}^{K} \frac{1}{\sigma_i^2 + \varepsilon \sigma_i^2} |u_i^T A^T y|^2 \right) \] (19)

6. DETECTION OF NUMBER OF SINUSOIDS

The observed signal consists of \( p \) complex sinusoids and white Gaussian noise as
\[ y_k = \sum_{i=1}^{p} a_i \exp(j2\pi f_i/k) + w_k, \quad k = 1, \ldots, N \] (20)

where \( a_i \) is the complex amplitude with unknown magnitude and phase. \( f_i \) is the unknown frequency and \( p \) denotes the number of sinusoids. \( w_k \) is the complex white Gaussian noise with zero mean and the covariance \( \sigma_w^2 \). The aim of this section is to give an algorithm for detecting the number of the sinusoids in (20) by the optimal truncation of the singular values.

The outline of the algorithm is given as follows:

1. Calculate the singular values \( \sigma_i \) and the singular vectors \( u_i, v_i \) of \( A \).
2. Calculate the multiple free parameters \( \rho_i \) so as to minimize (10), (13) or (17) by using the quasi-Newton method.
3. Find the truncation number \( K \) satisfying that \( \sigma_K^2 > \rho_K \) and \( \sigma_{K+1}^2 < \rho_{K+1} \) using (8) and (9).

We consider the first example in which the observed signal consists of three complex sinusoids in white noise with \( f_1 = 0.30, f_2 = 0.33, f_3 = 0.35, \sigma_1 = \sigma_2 = \sigma_3 = 1, N = 64, L = 30 \) for various SNR's. Fig. 1 shows the profiles of the singular values \( \sigma_i^2 \) vs. \( \rho_i^* \) in (7) and the calculated parameters \( \rho_i \) minimizing (13) and (17) for \( i = 1, \ldots, 6 \), in the case of SNR = 3dB. The plotted curves of \( \rho_i \) intersect the profile of \( \sigma_i^2 \) at between \( i = 3 \) and \( i = 4 \), then it is decided that the number of sinusoids is three. In cases of low SNR, it is difficult to determine the optimal truncation by watching the profile of the singular values \( \sigma_i^2 \). The proposed scheme can give the distinctive threshold for the decision by calculating the optimal \( \rho_i \) minimizing (10), (13) or (17).

Table 1 shows the distribution of the detected number of sinusoids for 100 trials using the EMSE, GCV, ABIC, AIC and MDL criterion. The MDL criterion has the following form [6] as
\[ \text{MDL}(K) = -2 \log \left( \frac{\prod_{i=K+1}^{L} \lambda_i^{(L-K)}}{\lambda_i^{(L-K) - 1}} \right)^{n(L-K)} \]
\[ + K(2L-K) \log n \] (21)

where \( \lambda_i \) is the eigenvalue of the sample covariance matrix. In the case of the AIC, the second term in (21) is replaced by \( 2K(2L-K) \). The MDL and AIC criteria depend on only the singular values. In the case of high SNR, the GCV, ABIC and MDL have similar performance, while in the case of low SNR ABIC(\( \rho_i \)) can detect most precisely compared to the MDL and other criteria. The AIC tends to over-estimate the number of harmonics and the MDL tends to underestimate it. The proposed schemes can detect the number of sinusoids more accurately than the AIC or MDL criterion.

Next, we consider the second example with \( f_1 = 0.30, f_2 = 0.34, \sigma_1 = \sigma_2 = 1, N = 16, L = 8 \). The results are sum-
marized in Table 2, which indicates the proposed method based on the ABIC($\rho_l$) performs best even in cases of small number of data and low SNR.

7. CONCLUSIONS

By choosing the adjustable free parameters induced to the SVD, we can give the optimal truncation of the singular values and the algorithm for deciding the number of sinusoids embedded in white noise. Three types of criterion for deciding the free parameters have been presented.

REFERENCES


![Figure 1. Plots of the calculated free parameters and singular values](image)

Table 1. Comparison of the proposed schemes with other methods minimizing AIC and MDL.

<table>
<thead>
<tr>
<th>SNR</th>
<th>3dBD</th>
<th>0dBD</th>
<th>-3dBD</th>
<th>-6dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
</tr>
<tr>
<td>EMSE($\rho_l$)</td>
<td>0</td>
<td>0</td>
<td>89</td>
<td>9</td>
</tr>
<tr>
<td>GCV($\rho_l$)</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>ABIC($\rho_l$)</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>ABIC(K)</td>
<td>0</td>
<td>0</td>
<td>100</td>
<td>0</td>
</tr>
<tr>
<td>AIC(K)</td>
<td>0</td>
<td>0</td>
<td>71</td>
<td>18</td>
</tr>
<tr>
<td>MDL(K)</td>
<td>0</td>
<td>0</td>
<td>98</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 2. The distribution of detected number of sinusoids in example 2.

<table>
<thead>
<tr>
<th>SNR</th>
<th>10dB</th>
<th>6dB</th>
<th>3dB</th>
</tr>
</thead>
<tbody>
<tr>
<td>K</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>EMSE($\rho_l$)</td>
<td>0</td>
<td>0</td>
<td>90</td>
</tr>
<tr>
<td>GCV($\rho_l$)</td>
<td>0</td>
<td>3</td>
<td>97</td>
</tr>
<tr>
<td>ABIC($\rho_l$)</td>
<td>0</td>
<td>0</td>
<td>100</td>
</tr>
<tr>
<td>ABIC(K)</td>
<td>0</td>
<td>0</td>
<td>94</td>
</tr>
<tr>
<td>AIC(K)</td>
<td>0</td>
<td>0</td>
<td>91</td>
</tr>
<tr>
<td>MDL(K)</td>
<td>0</td>
<td>0</td>
<td>91</td>
</tr>
</tbody>
</table>
APPLICATION OF THE BOOTSTRAPPED TOTAL LEAST SQUARES (BTLS) ESTIMATOR IN LINEAR SYSTEM IDENTIFICATION.

H. VAN HAMME and R. PINTERON

Vrije Universiteit Brussel, Dienst ELEC, Pleinlaan 2, B-1050 Brussel, Belgium.
Tel.: 32-2-641 2944; Fax.: 32-2-641 2850; e-mail: hvhamme@vnet3.vub.ac.be

Abstract - The estimation of the parameters of linear time-invariant single-input-single-output continuous-time or discrete-time systems with an errors-in-variables model is examined. The Bootstrapped Total Least Squares Estimator (BTLS; pronounce BeaTes) is shown to produce consistent estimates in this setting. The near-maximum likelihood performance of BTLS is shown through asymptotic analysis and verified through simulation. The BTLS estimates have a smaller mean-square estimation error than the generalized total least squares (GTLS) estimates.

1. Introduction - modeling linear systems.

We consider the problem of estimating the parameters of a linear time-invariant continuous-time or discrete-time single-input-single-output system from noise-corrupted input/output samples. Let \( t \) represent time (integer values for discrete-time systems) and let \( u(t) \) and \( y(t) \) denote the input and output signal respectively. Using the notation \( k \) for the \( k \)-th order derivative, the continuous-time system obeys the differential equation:

\[
a_n y^{(n)}(t) + a_{n-1} y^{(n-1)}(t) + \ldots + a_0 y^{(0)}(t) =
\]

\[
b_m u^{(m)}(t) + b_{m-1} u^{(m-1)}(t) + \ldots + b_0 u^{(0)}(t).
\]

(1.1a)

The discrete system obeys the difference equation:

\[
a_n y(t-n) + a_{n-1} y(t-n+1) + \ldots + a_0 y(t) =
\]

\[
b_m u(t-m) + b_{m-1} u(t-m+1) + \ldots + b_0 u(t)
\]

(1.1b)

Equation (1.1b) can be seen as an algebraic relation among the system parameters \( a_n, \ldots, a_0, b_m, \ldots, b_0 \). This relation can be evaluated at different values of \( t \).

The same is valid for (1.1a), except that it requires the derivatives of the input and the output, which are not known if only an input/output record is available. However, (1.1a) can be transformed into an algebraic relation [1]:

\[
a_n y(t) + a_{n-1} y(t) + \ldots + a_0 y(t) =
\]

\[
b_m u(t) + b_{m-1} u(t) + \ldots + b_0 u(t)
\]

(1.1c)

where \( y(t) \) is obtained by passing \( u(t) \) through a hybrid linear system consisting of an analog anti-alias filter followed by a numeric filter. The whole chain has a frequency response:

\[
\exp(j\omega) \exp((1 + \delta(j\omega)) \exp(j\omega)
\]

where \( G(j\omega) \) is some complex weighting function and \( 16\|a\| G(j\omega) \exp(j\omega) \exp((1 + \delta(j\omega)) \exp(j\omega)
\]

(1.2)

Here, \( p' = (a_n, \ldots, a_0, b_m, \ldots, b_0) \), \( l = n + m + 2 \), and \( h_l \) are the coefficients of the system parameters in (1.1a) and (1.1c). In matrix notation:

\[
H p = 0
\]

(1.1d)

where \( H \in \mathbb{R}^{K \times K} \) is a matrix whose \( t \)-th row is \( h_t \). We exclude the trivial solution \( p = 0 \) by fixing the first entry of \( p \) at 1 (any entry could have been chosen). This is written symbolically as \( p' e_1 = 1 \), where \( e_1 \) is the first unit vector.

In the next sections, we consider the least squares, the generalized total least squares, the (approximate) maximum likelihood and the bootstrapped total least squares estimation method for estimation of the parameters in model (1.1c). Due to space limitations, the exposition will be based on heuristics, rather than on hard mathematical proof. The reader is referred to [1] for more details.

2. Parameter Estimation.

When the input/output record is contaminated by additive zero-mean noise of any joint distribution, all \( h_t \) will be perturbed by noise \( n_t \in \mathbb{R}^{1 \times 1} \), yielding the observations \( g_t \).
\[ g = h + n \quad t = 0 \ldots K-1 \]  

Equivalently, \( H \) will be perturbed by some noise matrix \( N \in \mathbb{R}^{K \times d} \): \( G = H + N \) and generally \( G_p \) will not be zero for any value of \( p \). We then need to find a 'best' solution of an overdetermined set of linear equations, where all columns of \( G \) are subject to noise. One obvious approach is to solve \( G_p = 0 \) in a least squares sense:

\[ p_{LS} = \arg \min_{p, p_0 = 1} \frac{1}{\| p \|^2} \mathbf{G}^T \mathbf{G} p = \left( A^T A + \frac{1}{\| p_0 \|^2} \mathbf{I} \right)^{-1} A^T \mathbf{b} \]  

where \( \mathbf{b} \) is the first column of \( \mathbf{G} \) and \( A \) is the matrix containing the remaining columns: \( G = (b \ A) \). The least squares (LS) method will generally produce asymptotically biased results.

To obtain consistent estimates, the generalized total least squares (GTLS) solution can be used:

\[ p_{GTLS} = \arg \min_{p, p_0 = 1} \frac{1}{\| p \|^2} \mathbf{G}^T \mathbf{G} p + \frac{1}{\| p_0 \|^2} \mathbf{G}^T \mathbf{G} \mathbf{c}_1 \]  

where \( \mathbf{c}_1 \) is the right-singular vector corresponding to the smallest singular value of \( \mathbf{G} \mathbf{T} \) and \( \mathbf{C}_0 = (\mathbf{T}^\top)^{-1} = \mathbf{E} \mathbf{N} \mathbf{N}^\top \) (see Van Huffel and Vanderwaal, 1991).

To obtain the maximum likelihood estimates (MLE) for Gaussian observation noise, we define \( \mathbf{B}_p = \mathbf{I}_K \otimes \mathbf{p}' \) (\( \otimes \) denotes Kronecker product), \( \mathbf{n}_1 = (n_1 \ n_2 \ \ldots \ n_N) \), i.e., \( \mathbf{n} \in \mathbb{R}^{K \times l} \) is obtained by stacking the \( n_k \) on top of each other, and \( \mathbf{C}_n = \mathbf{E} \mathbf{n} \otimes \mathbf{n} \). Then the MLE for Gaussian \( \mathbf{n} \) is given by:

\[ p_{MLE} = \arg \min_{p, p_0 = 1} \frac{1}{\| p \|^2} \mathbf{G} \mathbf{B}_p \mathbf{C}_n \mathbf{B}_p^\top \mathbf{G} p \]  

In contrast to (2.2) and (2.3), an explicit solution of the minimization problem is not given here, because at present, no closed form solution is known to the authors. The minimizer in (2.4a) is calculated using iterative minimization techniques. It requires an initial guess, which can be provided by e.g. GTLS.

Alternatively, an approximate maximum likelihood estimator can be defined, where \( \mathbf{C}_n \) is replaced by \( \hat{\mathbf{D}} \):

\[ \hat{\mathbf{p}}_{MLE} = \arg \min_{p, p_0 = 1} \frac{1}{\| p \|^2} \mathbf{G} \mathbf{B}_p \hat{\mathbf{D}} \mathbf{B}_p^\top \mathbf{G} p \]  

The block-Toeplitz \( \hat{\mathbf{D}} \) is related to \( \mathbf{C}_n \) in a way that will be specified below.

All estimators above are written as the minimizer of a cost function \( I_K(p, G) \):

\[ I_K(p, G) = \frac{1}{2K} \mathbf{p}^\top \mathbf{G} \left( \mathbf{B}_p \hat{\mathbf{D}} \mathbf{B}_p^\top \right)^{-1} \mathbf{G} p \]  

where the index \( K \) denotes the number of rows in \( G \) and \( \hat{\mathbf{D}} \) is a weighting matrix that depends on the estimator. Comparing (2.2), (2.3) and (2.4) with (2.5), we find for the specific estimators:

LS:
\[ \hat{\mathbf{D}} = \mathbf{c}_1 \otimes \mathbf{I}_K \]  

GTLS:
\[ \hat{\mathbf{D}} = \mathbf{C}_n \otimes \mathbf{I}_K \]  

MLE:
\[ \hat{\mathbf{D}} = \mathbf{C}_n \]  

AMLE:
\[ \hat{\mathbf{D}} = \mathbf{\hat{D}}. \]

The cost in (2.5) was scaled by \( 2K \) for later convenience, which doesn't affect its minimizer. Also, \( \log(p, G) \) is invariant for scaling of all entries of \( p \) by the same constant. Hence, the constraint \( \mathbf{p}^\top \mathbf{e}_1 = 1 \) can be omitted.

### 3. Asymptotic Analysis.

Assume that the system excitation is quasi-stationary and that the noise on the input/output record is wide-sense stationary and statistically independent of the excitation, i.e.

\[ C_n = \mathbf{E} \mathbf{n}_k \mathbf{n}_k^\top \in \mathbb{R}^{l \times l} \]

is independent of \( k \) for any \( n \) and

\[ S_n = \lim_{K \to \infty} \frac{1}{K} \sum_{k=1}^{K} \mathbf{E} \mathbf{h}_k \mathbf{h}_k^\top \]

exists. One can define the noise and signal spectra \( \Phi_n(\omega) \) and \( \Phi_s(\omega) \) via the Fourier relations:

\[ C_n = \frac{1}{\pi} \int_0^\pi \Phi_n(\omega) e^{-j\omega n} d\omega \]

and

\[ S_n = \frac{1}{\pi} \int_0^\pi \Phi_s(\omega) e^{-j\omega n} d\omega \]

where (3.2b) is a Stieltjes integral. Likewise, since \( \hat{\mathbf{D}} \) is a block-Toeplitz matrix with \( l \times l \) blocks for LS, GTLS, MLE and AMLE, it is possible to define a weighting spectrum \( \Phi_g(\omega) \) via:

\[ \hat{\mathbf{D}}_n = \frac{1}{\pi} \int_0^\pi \Phi_g(\omega) e^{-j\omega n} d\omega \in \mathbb{R}^{l \times l}, \ n = -\infty \ldots \infty \]

where \( \hat{\mathbf{D}}_n \) is the \( k,l \)-th \( l \times l \) block of \( \hat{\mathbf{D}} \). With (2.6):

LS:
\[ \Phi_n(\omega) = \mathbf{e}_1 \mathbf{e}_1^\top \]

GTLS:
\[ \Phi_n(\omega) = \mathbf{C}_n \]

MLE:
\[ \Phi_n(\omega) = \Phi_s(\omega) \]

In the AMLE approach, we choose:

AMLE:
\[ \Phi_n(\omega) = f(\omega) \Phi_s(\omega) \]

where \( f(\omega) \) is a real-valued positive weighting function. The matrix \( \hat{\mathbf{D}} \) in (2.3b) is defined via (3.2c) with (3.2d). For \( f(\omega) = 1 \), the AMLE reduces to MLE.

It was shown in [1] that under fairly weak restrictions, the cost function (2.5) converges uniformly in \( p \) to:

\[ I(p) = \int_0^\pi \frac{1}{4\pi} \mathbf{p}^\top \Phi_g(\omega) \mathbf{p} + \mathbf{p}^\top \Phi_g(\omega) \mathbf{p} \]

as \( K \) tends to infinity. Due to the uniform convergence of the cost, the asymptotic estimates are given by the minimizer of \( I(p) \). The two positive terms in (3.3) represent the contribution to the asymptotic cost of the signal and the noise respectively.

The quantity \( \mathbf{p}^\top d\Psi_g(\omega) \mathbf{p} \) is the contribution of the
excitation to the power spectrum of the residuals $\mathbf{G}$ and $p'$ \( \Phi_C(\omega) \) is the contribution of the noise to this residual power spectrum. In (3.4), these spectra are both weighted by the estimator-dependent function $p' \Phi_C(\omega) p$. For the MLE (3.5c), this function equals $p' \Phi_C(\omega) p$, i.e., the residuals are weighted according to their noise spectrum. AMLE offers an extra freedom in the choice of $f(\omega)$, which can be used to emphasize some frequencies in (3.4). For LS and GTLS, $p' \Phi_C(\omega) p$ is independent of frequency, which makes them more susceptible to noise perturbation. This will now be illustrated in the context of system identification, where the input (output) signals are perturbed by mutually uncorrelated noise with a power spectral density $\sigma_n^2(\omega)$ ($\sigma_n^2(\omega)$). For discrete-time systems:

$$p' \Phi_C(\omega) p = \sigma_n^2(\omega) 1B(e^{j\omega})^2 + \sigma^2(\omega) 1A(e^{j\omega})^2, \quad (3.5a)$$

where $B(e^{j\omega})/A(e^{j\omega})$ is the system frequency response associated with the parameter vector $p$. Functions (3.5) tend to increase for higher frequencies. Hence, since this dependence on the frequency is missing for LS and GTLS, these estimators over-emphasize high frequencies. Deviations at low frequencies will hardly be penalized, resulting in poor asymptotic estimates for the transfer function at these frequencies.

Now suppose that there exists a true parameter vector $p_e$ that satisfies (1.1) exactly, i.e., $H p_e = 0$. In view of (3.1b), the positive form $p' d^2 \Phi_s(\omega) p$ will be zero for $p = p_e$. Under the persistency condition that rank($S_0) \geq k-1$ (where $S_0$ is defined in (2.6b)), the first term will only be zero when $p$ is proportional to $p_e$. Since this term is positive, it is minimal at $p_e$. With (3.2) and (3.3), the second term equals:

$$\text{LS: } \frac{1}{4\pi} \int_0^{2\pi} \frac{p' \Phi_C(\omega) p}{(p' \Phi_C(\omega) p)^2} \ d\omega = \frac{1}{2}, \quad (3.6a)$$

$$\text{GTLS: } \frac{1}{2\pi} \int_0^{2\pi} \frac{p' \Phi_C(\omega) p}{(p' \Phi_C(\omega) p)^2} \ d\omega = \frac{1}{2}, \quad (3.6b)$$

$$\text{MLE: } \frac{1}{4\pi} \int_0^{2\pi} \frac{p' \Phi_C(\omega) p}{(p' \Phi_C(\omega) p)^2} \ d\omega = \frac{1}{2}, \quad (3.6c)$$

$$\text{AMLE: } \frac{1}{2\pi} \int_0^{2\pi} \frac{p' \Phi_C(\omega) p}{f(\omega)} \ d\omega = \frac{1}{2}, \quad (3.6d)$$

For GTLS, MLE and AMLE, the second term in (3.4) is a constant, so the asymptotic estimates will be given by the minimizer of the first term, which is $p_e$ for all estimators considered. Hence, these estimators will be consistent. For LS, the (3.6a) is not independent of $p$, and the minimizer of (3.4) will not coincide with the true parameters and the estimates are inconsistent. LS will therefore be omitted from further discussion.


The GTLS estimator is attractive because it yields consistent estimates that can be calculated quite efficiently [2]. Its poor frequency-domain weighting can be improved by premultiplying the observation matrix $\mathbf{G}$ with $(B_p \hat{\mathbf{B}}' P_p)^{-1/2}$ to yield $\hat{\mathbf{G}}$ (likewise, $\hat{\mathbf{N}} = (B_p \hat{\mathbf{B}}' P_p)^{-1/2} \mathbf{N}$), and then applying the GTLS technique to the set of equations $\hat{\mathbf{G}} p = 0$. The purpose of incorporating this weight is to approximate the AMLE cost as close as possible, while retaining the computational advantage of GTLS. The transformed GTLS problem is solved by minimizing:

$$p' \hat{\mathbf{G}} (B_p \hat{\mathbf{B}}' P_p)^{-1} \hat{\mathbf{G}} p$$

where $D_{p_e,\text{tls}} = E[H' \hat{\mathbf{N}} E[H' \hat{\mathbf{N}}]^{-1}]$.

The cost (4.1) is minimized by the right-singular vector corresponding to the smallest singular value of $\hat{\mathbf{G}} \mathbf{T}$, where $D_{p_e,\text{tls}} = (\mathbf{T} \mathbf{T})^{-1}$. Comparing (2.5) and (4.1) yields the weighting matrix:

$$\mathbf{D} = (B_p \hat{\mathbf{B}}' P_p) \otimes D_{p_e,\text{tls}},$$

from which the weighting spectrum is found:

$$\phi_D(\omega) = f(\omega) (p_e' \Phi_C(\omega) p_e) D_{p_e,\text{tls}},$$

With (3.4) and knowing that $D_{p_e,\text{tls}}$ converges to 1:

$$\frac{1}{2\pi} \int_0^{2\pi} \frac{\phi_D(\omega)}{f(\omega) p_e' \Phi_C(\omega) p_e} \ d\omega$$

the asymptotic cost is found:

$$\frac{1}{4\pi} \int_0^{2\pi} \frac{p' d^2 \Phi_s(\omega) p}{f(\omega) p_e' \Phi_C(\omega) p_e + \frac{1}{4\pi} \int_0^{2\pi} \frac{1}{f(\omega)} \ d\omega}$$

In (4.4), the contribution of the noise is a constant. As for GTLS, MLE and AMLE, this ensures the consistency of the estimates. Also, when minimizing (4.1), the emphasis at high frequencies is present in the asymptotic cost (4.4). However, the construction of the weights $B_p \hat{\mathbf{B}}' P_p$ requires the knowledge of the true parameters $p_e$. Therefore, $p_e$ is replaced by an estimate $p_{\text{est}}$, which is obtained from a previous step, hence the name of bootstrapped total least squares. The BTLS algorithm is then:

1. choose a $p_{\text{est}}$, set $i = 0$.
2. find $p_{\text{est}}$, the minimizer of (4.1) where $p_e$ is replaced by $p_{\text{est}}$. This is obtained by computing a right-singular vector as explained above.
3. if the stopping criterion is not met, increment $i$ and go to 2, else, the final estimate is $p_{\text{est}}$. 


The initial value \( \mathbf{p}_0 \) is not critical. Possible choices are a vector of ones, or a parameter vector that corresponds to \( B = 0 \) and \( A = 1 \). In the latter case, the weights are those used by the GTLS estimator when the observation noise is white. Reasonable stopping criteria are that \( i \) exceeds a prescribed value (e.g. 5) or that the parameter change is less than some bound.

An important property of the BTLS algorithm is that every step yields consistent estimates and that all \( \mathbf{p}_i \) for \( i \geq 2 \) have the same asymptotic distribution (see [1]). Therefore, limiting the number of iterations is sensible. For Gaussian observation noise \( \mathbf{N} \), its asymptotic covariance differs from the asymptotic covariance of the AMLE only by a small positive definite contribution. In the calculation of the covariance for \( \mathbf{p}_0 \) it is essential that \( \mathbf{p}_{14} \) is a consistent estimate of \( \mathbf{p}_0 \). Simulation studies show that the difference is small, even at high noise levels. For instance, for a continuous-time 5th order plant measured with noise excitation and with observation noise on the input and the output samples with a standard deviation that is nearly 50 \% of the excitation noise standard deviation, the standard deviation on the transfer function estimate increases by no more than 10 \% since this deterioration is proportional to the observation noise level, even better results are obtained at lower noise levels.

The implementation of (4.1) can be accomplished by solving a Toeplitz set of equations. This can be done efficiently using the methods described in [3] and its references, or using filtering [1].

5. Simulations.

The consequences of the bad weighting in GTLS are illustrated by simulating a 5th order continuous-time Chebychev filter with 1 dB pass band ripple and a cutoff frequency of 0.15 Hz, sampled at 1 Hz. The estimation is based on (1.1c) and (1.2), where \( |G(j\omega)| = 1 \) below 0.25 Hz and \( |G(j\omega)| = 0 \) above that frequency with a sharp transition. This weighting function is required to suppress the small deviations in (1.1c) which would otherwise be significant for frequencies above 0.25 Hz. To avoid that the estimator boosts these errors up again, \( f(\omega) \) is chosen equal to \( 10(e^{j\omega})^{-2} \) in AMLE and BTLS. The excitation is noise with a nearly flat spectrum up to the measurement frequency of 0.25 Hz. The observation noise is also white in this frequency range, but it is about 10 times smaller. The experiment is repeated for 100 independent observation noise realizations, where each time \( K = 150 \) rows of \( \mathbf{G} \) are gathered. The mean square error (MSE) of the transfer function estimate is found and compared with the Cramér-Rao lower bound (CRB) on the variance of the transfer function. The ratio of these quantities is depicted in Fig. 1 in dB (\( 10 \log_{10} \text{MSE/CRB} \)). We notice that BTLS and MLE outperform GTLS and that they both nearly attain the CRB.

![Figure 1: ratio of the mean square transfer function estimation error and the Cramér-Rao bound.](image)

It must be stressed that the profit of using BTLS instead of GTLS depends on \( pe \Phi_0(\omega) pe \) and hence on the plant. In [1], examples can be found where BTLS and MLE exhibit a standard deviation that is about 100 times smaller than that of GTLS. In other simulations, where \( pe \Phi_0(\omega) pe \) is nearly constant in the measurement frequency range, the increased computational load of BTLS is not justified by its reduced variance.

6. Conclusion.

We have presented a new estimator that has MLE performance but with reduced computational load. It is based on the GTLS estimator but weights the rows of the observation matrix to improve statistical efficiency. The idea can be applied to other problems that can be formulated as a regression problem.

References.

GAUSSIAN NOISE INSSENSITIVE FIR SYSTEM IDENTIFICATION

Gaetano Scarano*, Fausto Gorina**, Alessandro Neri** and Giovanni Jacovitti**

* Istituto di Elettronica, Universitá di Perugia, S.Lucia Canetola, I-06143 Perugia, Italy
** Dip. INFCOM, University of Rome "La Sapienza", Via Eudossiana 18, I-00184 Roma, Italy

ABSTRACT

A contribution to the problem of FIR system identification based on higher order statistics of the output only is presented. As all equation error methods the parameters estimates are obtained through least-squares solution of a (overdetermined) linear set of equations, in which, in the proposed method, only higher than two cumulants are involved. This fact implies that, in principle, the solution is insensitive to any additive, possibly coloured, Gaussian noise contaminating the FIR system output.

1. INTRODUCTION

In this contribution we address the problem of estimating the parameters of a moving average (MA) model from higher-order statistics of the noisy observations of its output. To be specific, let us consider a FIR system of order q

\[ z(n) = \sum_{i=0}^{q} b(i) \cdot w(n - i) \]  

(1)

where the driving noise \( w(n) \) is a zero mean, independent and identically distributed (i.i.d.) non-Gaussian series. In many application of interest, such as digital communication, geophysical signal processing, astronomical signal processing, etc., the parameters \( b(i) \)'s characterize a nonminimum phase system, which must be identified from \( N \) consecutive samples of the system output (1) contaminated by additive Gaussian noise

\[ y(n) = z(n) + v(n) \quad ; \quad n = 0, 1, \ldots, N - 1. \]

It is well known that second-order (spectral) analysis of the output enables to measure the magnitude of the transfer function of the system, but it does not provide any information about the phase of the transformation. Higher-order analysis (higher-order moments, cumulants and associate polyspectra) allows, in principle, for unambiguous identification.

In this paper, we are concerned with the problem of recovering the coefficients \( b(i) \)'s from the cumulant statistics taken from the available observations. This problem has been addressed by several authors (for instance see [1,2,3,4,5,6,7,8]) and a very interesting tutorial has been recently published [9]. Other applications of higher-order statistics and polyspectra in more general parameter estimation problems can be found in [10,11,12].

Approaches to the FIR system identification are usually classified in three broad classes [4]:

1. Inverse filter error: this class includes algorithms based on minimization of some criterion involving functionals computed from the output of an inverse filter of the FIR transformation, designed to obtain an estimate \( \hat{w}(n) \) of the input \( w(n) \);

2. Fitting error: these solutions are obtained by matching the estimated (measured) statistics to the model (a priori) statistics;

3. Equation error: generally, in this class are collected algorithms where \( b(i) \)'s are directly obtained by solving linear equations ideally satisfied by ensemble moments.

In the first two classes we find algorithms which result into non-linear optimization of suitably defined functionals, whereas in the last class the MA parameters are non-optimally determined by a linear set of equations. For this reason, equation error algorithms can be also used somewhere as coarse estimates for setting the starting point for (non-linear) estimation procedures belonging to the first two classes.

Almost all equation error methods involve both second-order statistics as well as higher-order statistics. For this reason the observation noise \( v(n) \) can be accommodated by modeling it as a MA process of order \( q \) significantly lower than the order \( q \) of the MA process of interest. In this way, by eliminating those equations which involve the non-zero noise autocorrelation lags (see [4]), noise insensitive algorithms are obtained. This elimination allows to include also second-order (autocorrelation) statistics which are more statistically stable than higher-order statistics. Clearly, when the noise cannot be modelled as a MA process, these methods give biased estimates.

Here, we propose a MA parameter estimation equation error based method which makes use of higher-order cumulant statistics only, in order to obtain co-
loured Gaussian noise insensitive estimates without resorting to any equation elimination. Simulation results show that the proposed technique generally performs better than autocorrelation-including methods for medium signal-to-noise ratio values, while resulting in insignificant performance losses for high SNR values.

2. FIR SYSTEM IDENTIFICATION

Let us briefly recall the basic steps of equation error FIR system identification techniques based on relationships between output cumulants and MA parameters. For simplicity, let us refer to the third order cumulant of the noise contaminated observation, defined as

\[ c_2y(k, l) = E \{ y(n) y(n + k) y(n + l) \} \]  

(2)

The assumed independence of the additive noise \( \nu(n) \) allows to decompose (2) in a sum of cumulants

\[ c_2y(k, l) = c_{2x}(k, l) + c_{2\nu}(k, l). \]  

(3)

If the noise is Gaussian, all cumulants of order greater than two vanish and (3) becomes

\[ c_2y(k, l) = c_{2x}(k, l). \]  

(4)

Now, (4) allows to simply relate the noisy output cumulants to the unknown MA parameters. In fact we have

\[ c_{2x}(k, l) = \gamma_{2w} \sum_{i=0}^{q} b(i) \cdot b(i + k) \cdot b(i + l) \]  

(5)

where \( \gamma_{2w} = E \{ w^3(n) \} \) is the third order cumulant of the i.i.d. driving noise \( w(n) \). In the following we suppose that the system order \( q \) is perfectly known. This means that we must resort to system order estimation before MA parameters estimation: this can be done by classical techniques as specified also in [4]. Here, we focus on the MA parameters estimation only.

From the basic equation (5), we obtain the following sets of equations setting the lag \( l = k \) and \( l = q \), respectively:

\[ c_{2x}(k, k) = \gamma_{2w} \sum_{i=0}^{q} b^2(i + k) \cdot b(i); \quad |k| \leq q \]

\[ c_{2x}(k, q) = \gamma_{2w} \cdot b(0) b(q) \cdot b(k); \quad 0 \leq k \leq q \]

Taking \( Z \)-transforms of both sides of the previous equations yields

\[ Z \{ c_{2y}(k, k) \} = \gamma_{2w} \cdot B(z^{-1}) [B(z) \ast B(z)] \]  

(6)

\[ Z \{ c_{2y}(k, q) \} = \gamma_{2w} \cdot b(0) b(q) \cdot B(z) \]  

(7)

Along with these equations, classical methods makes use of the power spectral density also, expressed by

\[ Z \{ c_{2y}(k) \} = \sigma_y^2 \cdot B(z) \cdot B(z^{-1}) + Z \{ c_{2\nu}(k) \} \]  

(8)

where \( c_{2y}(k) = E \{ y(n + k) y(n) \} \) is the noisy output series autocorrelation, \( c_{2x}(k) = E \{ x(n + k) x(n) \} \) is the observation noise autocorrelation and \( \sigma_y^2 \) is the variance of the noise \( w(n) \). Elimination of \( B(z) \) from (7) and (8) and inverse transformation yields the following sets of \( 2q + 1 \) equations [4]

\[ c_{2y}(k) = \frac{\sigma_y^2}{\gamma_{2w} b(0) b(q)} \sum_{i=0}^{q} b(i) c_{2x}(k + i, q) + c_{2\nu}(k) \]  

\[ -q \leq k \leq q. \]  

(9)

Note that these equations contain the unknown correlation of the unobservable noise \( \nu(n) \). If this noise is white, by eliminating the equation for \( k = 0 \) we suppress any noise correlation contribution in (9) and consequently, any biasing effects on MA parameters estimates. We recall that the equation error estimation method of [2] makes use of (6) and (8), solving equations by elimination of \( B(z^{-1}) \). This approach is affected by the noise correlation in different way with respect to (9) and it is not possible to suppress white noise contribution by eliminating suitable equations (as done in (9)). Moreover, the consistency of the estimates is not assured (see [4]).

For non-white noise, it is suggested in [4] that, by modelling it as an MA series of order \( \hat{q} \leq \frac{q}{2} \), it is possible to write \( 2q + 1 - 2\hat{q} \) equations obtained from (9) for \( |k| = \hat{q} + 1, \cdots, q \)

\[ \sum_{i=1}^{q} b(i) c_{2y}(k + i, q - \epsilon) c_{2y}(k) = -c_{2y}(k, q) \]  

(10)

where \( \epsilon = \frac{2\hat{q}}{\sigma_y^2} B(z) \) and \( b(0) \) has been assumed equal to 1 without loss of generality.

Here \( \epsilon \) has been defined as an MA series of order \( \hat{q} \leq \frac{q}{2} \), it is possible to write \( 2q + 1 - 2\hat{q} \) equations obtained from (9) for \( |k| = \hat{q} + 1, \cdots, q \)

\[ \sum_{i=1}^{q} b(i) c_{2y}(k + i, q - \epsilon) c_{2y}(k) = -c_{2y}(k, q) \]  

(10)

where \( \epsilon = \frac{2\hat{q}}{\sigma_y^2} B(z) \) and \( b(0) \) has been assumed equal to 1 without loss of generality.

Now, (10) do not include unobservable quantities but they suffer the fact that the MA observation noise order must be known a priori. Moreover, the number of available equations has been reduced and, as we will see later, this fact impairs the statistical stability of the MA parameters estimates. This statistical instability depends also on the fact that large cumulant lags are involved.

To overcome these problems we have derived a different set of equations, which results in completely noise contribution free system. To this purpose let us take the (complex) convolution of (7) with itself

\[ Z \{ c_{2y}(k, q) \} \ast Z \{ c_{2y}(k, q) \} = [\gamma_{2w} \cdot B(z)]^2 \cdot [B(z) \ast B(z)]. \]  

(11)

By eliminating \( [B(z) \ast B(z)] \) from (6) and (11), we obtain, after inverse transformation, the following equations relating the MA parameters to the cumulant statistics of the output.
\[
\sum_{i=1}^{s} b(i) \cdot c_{pq}(t+k,q) - \varepsilon'' \cdot c_{pq}(k,k) = -\gamma_{pq} \cdot b(q)^2 \] (12)

where \( \varepsilon'' = \gamma_{pq} \cdot b(q)^2 \).

Here, the blind noise rejection relies on the Gaussianity of the noise, and further assumptions on its model has not been made.

Both (10) and (12) are solved in a least squares sense considering \( \varepsilon \) and \( \varepsilon'' \) as unknowns independent of the MA parameters \( b(1), b(2), \ldots, b(q) \). Estimates are obtained by substituting expectations with sample moments. The proof of the consistency of the estimation follows the same guidelines presented in [4]. Note that any additive Gaussian noise affects the performance of the estimation only, not the derivation of the solving equations (12) (see (2)). Extension to fourth-order cumulant statistics is straightforward.

3. SIMULATION RESULTS

To test the effectiveness of the presented additive Gaussian noise insensitive FIR identification method (12) against the method (10), computer simulations have been carried out for non-zero skewness driving noise \( w(n) \) using third-order cumulants, and for zero skewness, non-zero kurtosis driving noise using fourth-order cumulants. A second-order FIR system \( b(1) = -2.05, b(2) = 1 \) already reported in literature, has been considered.

The bias and the root mean square error (RMSE) of the MA parameters estimates obtained using third order cumulants are shown in Fig. 1 for \( b(1) = -2.05 \) and Fig. 2 for \( b(2) = 1 \). Here the driving noise is exponential distributed and the additive coloured noise is a first order MA series obtained filtering white noise through the FIR filter with transfer function \( H(z) = 1 + 0.9 z^{-1} \). The estimates are taken over 2000 observed samples and 100 independent MonteCarlo runs.

Similarly, in Figs. 3 and 4, the driving noise is an independent binary series with symbols \( \pm 1 \) having the same probability while the additive coloured noise has a Gaussian shaped autocorrelation \( c_{pq}(k) = 0.9^{-k^2} \). The estimates are taken over 4000 observed samples and 100 independent MonteCarlo runs.

In both cases we observe that the proposed method reduces the RMSE essentially by reducing the bias of the estimates induced by the additive coloured noise, with a slight increase of the variance. Note that the first order MA model of the additive noise of Fig. 1 cannot be accommodated by (10), resulting this latter in a underdetermined set of equations. To overcome this fact we have used (10) by considering the additive noise as it were white, i.e. by eliminating the equation for \( k = 0 \), as indicated in [4].

Simulations on other classical scenarios considered in literature show that the proposed method generally outperforms existing methods, even though, in some cases, have been observed moderate performance losses. General conclusion should be made on a theoretical basis following the guidelines of [5] and it will be matter of a forthcoming paper.

In conclusion, the proposed Gaussian noise insensitive FIR system identification can be usefully employed in on-line applications, where one-shot solutions methods are required, as well as in initial (rough) estimation, where it provides good starting point for more computationally intensive (iterative) non-linear optimum estimation solutions.

REFERENCES

ITERATIVE TECHNIQUES FOR BLIND SOURCE SEPARATION
USING ONLY FOURTH-ORDER CUMULANTS

Jean-François Cardoso
Telecom Paris. Dept Signal, 46 rue Barrault, 75634 Paris CEDEX 13, FRANCE.
Email: cardose@sig.enst.fr

Abstract. "Blind source separation" is an array processing problem without a priori information (no array manifold). This model can be identified resorting to 4th-order cumulants only via the concept of 4th-order signal subspace (FOSS) which is defined as a matrix space. This idea leads to a "Blind MUSIC" approach where identification is achieved by looking for the (approximate) intersections between the FOSS and the manifold of 1D projection matrices. Practical implementations of these ideas are discussed and illustrated with computer simulations.

1. INTRODUCTION

This paper addresses the problem of blind source separation or independent component analysis in the complex case where it can be seen as a narrow-band array processing problem: the output, denoted $x(t)$, of an array of $n$ sensors listening at $n$ discrete sources takes the form:

$$x(t) = \sum_{p=1}^{n} s_p(t) a_p + n(t)$$

where $s_p(t)$ denotes the complex signal emitted by the $p$-th source; $a_p$ is a fixed (deterministic) vector called the $p$-th source signature; and where $n(t)$ is an independent additive noise assumed to be normally distributed with arbitrary covariance.

In "standard" array processing, the same data model is used but the source signatures depend on very few location parameters and this dependence is assumed to be known via the array manifold. In contrast, we address here the blind problem where no a priori information is available about source signatures. Hence blind qualifies any processing based on the sole observations. "Blindness" is compensated by exploiting the hypothesized assumption of source independence. In the following it is assumed that:

* source signals are statistically independent,
* source signals have non vanishing kurtosis,
* source signatures are linearly independent.

Blind source separation is understood as estimation of the source signals $s_p(t)$, while blind identification refers to the estimation of the source signatures $a_p$. In the following, we focus on identification since separation can be based on signature estimates.

The blind problem is interesting because its solution allows to process narrow band array data without explicit knowledge about array geometry and without assumptions about wavefront shapes. Various solutions relying on the use of both 2nd-order and 4th-order cumulant statistics of the array output have already been reported [1-4]. These approaches make use of 2nd-order information to whiten the data, and 4th-order information is then used to process the resulting orthogonalized problem. The additive noise is assumed to be normally distributed; it has no effect on the (exact) 4th-order cumulants but 2nd-order noise cumulants do not vanish so that the spatial structure of the noise covariance has to be known, modelled or estimated in order to achieve consistent 2nd-order pre-whitening. These limitations can be overcome by giving up the idea of 2nd-order whitening and resorting to 4th-order cumulants only [5-6]. It is the purpose of this communication to show how the concept of fourth-order signal subspace yields simple implementations of 4th-order only blind identification.

2. BLIND IDENTIFICATION

2.1 On identifiability. The blind context does not lead to full identifiability of the model (1) because any complex factor can be exchanged between $s_p$ and $a_p$ without modifying the observation. Hence if no a priori information is available, each signature can be identified only up to a scale factor. We take advantage of this to assume, without any loss of generality, that each signature $a_p$ has unit norm. With this constraint, an unidentifiable phase term is still present in $a_p$. For each source $p$, we denote as $\Pi_p$ the orthogonal projector onto the 1D space where the $p$th component lives. It is the space spanned by the signature $a_p$ and the projector onto it is

$$\Pi_p \triangleq a_p a_p^*$$

This hermitian matrix is unaffected by any phase term in $a_p$, and conversely determines $a_p$ up to a phase term. It follows that the projectors $\Pi_p$ are the algebraic quantities that can be, at best, identified in the blind context. It is
easily seen that knowing these projectors is sufficient to perform blind separation of the source signals \( s_p(t) \) because they allow to construct for each source the linear filter that zeroes all the components but the one specified. We then define blind identification as the problem of estimating the projectors \( \Pi_p \) from sample statistics only.

2.2 Blind MUSIC. The approach to blind identification presented in this contribution can be seen as a blind 4th-order version of the celebrated MUSIC algorithm. The MUSIC technique is based on the concept of signal subspace which is the vector space spanned by the steering vectors. It can be summarized as i) Estimate the signal subspace using the covariance. ii) Search for the steering vectors which are the closest to the signal subspace. The search is, of course, across the array manifold: this is how MUSIC exploits the a priori information contained in the parameterization of the steering vectors.

The fourth-order signal subspace (FOSS) is defined as the real span of the projectors \( \Pi_p \), i.e. as the linear matrix space made of all possible linear combinations with real coefficients of the projectors \( \Pi_p \):

\[
\text{FOSS} = \{ M | M = \sum_{p=1,\ldots,n} \gamma_p a_p a_p^* , \gamma_p \in \mathbb{R} \}.
\]

Let us now consider the following idea for blind identification i.e. estimation of the \( \Pi_p \): i) Estimate the FOSS using 4th-order cumulants. ii) Search for the orthogonal 1D projectors which are the closest to the FOSS. The closest projectors to the FOSS are taken as estimates of the source projectors \( \Pi_p \). Such an idea could be termed "blind MUSIC" because, in spite of its strong analogy with the classical MUSIC, no signature parameterization is assumed here: the search is across the so-called rank one manifold (ROM) which is defined as the set of all rank-one unit-norm hermitian matrices i.e. across all the 1D orthogonal projectors.

The reason why Blind MUSIC works is that a matrix space with structure as in eq. (3) is shown, under mild conditions, to contain no other 1D projectors than the ones used in its construction, i.e. the \( \Pi_p \). This is obviously true as soon as the signatures \( a_p \) are linearly independent since in that case, a matrix \( M \) as in eq. (3) has a rank equal to the number of non-zero coefficients \( \gamma_p \). If \( M \) is a 1D projector, it has rank one, hence all the coefficients \( \gamma_p \) but one are zero and \( M \) is then necessarily one of the \( \Pi_p \).

We now have to discuss i) FOSS estimation from 4th-order cumulants ii) practical implementations of the blind MUSIC search.

3. FOURTH-ORDER SIGNAL SUBSPACE

3.1 Quadrnocovariance. We find convenient to make temporary use of indexed notations to express the cumulants of the vector process \( x \). Let us denote by \( x_i \) the \( i \)-th coordinate of vector \( x \) and by \( x^* \) the \( i \)-th coordinate of its dual \( x^* \). Of course, only orthonormal basis are used: \( x^* \) is just the complex conjugate of \( x \). The covariance classically is the matrix \( R \) whose \((i,j)\)-coordinate denoted \( r_{ij} \) is the 2nd-order cumulant of \( x_i \) and \( x_j \):

\[
r_{ij} \triangleq \text{Cum} \left( x_i, x_j \right) \quad 1 \leq i, j \leq m.
\]

Similarly, we define the quadrnocovariance of \( x \) as the set of \( m^4 \) complex scalars, \( q_{ik}^{jl} \), \( 1 \leq i, j, k, l \leq m \):

\[
q_{ik}^{jl} \triangleq \text{Cum} \left( x_i, x_j, x_k, x_l \right)
\]

Our approach to process 4th-order information is to consider the quadrnocovariance as a matrix mapping denoted \( Q \), which to any matrix \( M \) with coordinates \( m_{ij} \) associates the matrix \( N = Q(M) \) with coordinates \( n_{ij} \) according to:

\[
n_{ij} = \sum_{k,l \in \mathbb{N}} q_{ik}^{jl} m_{kl}
\]

The quadrnocovariance has the two following properties: i) it maps any hermitian matrix to another hermitian matrix. ii) it is itself an hermitian operator in the (usual) sense that for any matrices \( M \) and \( N \) we have \( \langle N \mid Q(M) \rangle = \langle M \mid Q(N) \rangle \) with the Euclidian scalar product \( \langle M \mid N \rangle \triangleq \text{Tr}(NM^H) \). These are trivial consequences of cumulant symmetries. It follows [3] that the quadrnocovariance admits \( m^2 \) real eigenvalues, denoted \( \mu_i, i=1, \ldots, m^2 \) and \( m^2 \) corresponding orthonormal hermitian eigen-matrices, denoted \( E_i, i=1, \ldots, m^2 \), verifying:

\[
\forall i=1, \ldots, m^2, \quad Q(E_i) = \mu_i E_i \quad \text{with} \quad E_i = E_i^H, \quad \mu_i \in \mathbb{R}, \quad \text{Tr}(E_i E_j) = \delta(i,j)
\]

As a simple consequence [3] of cumulant additivity and multilinearity, the quadrnocovariance of a linear mixture (1) of independent components takes the special form:

\[
Q(M) = \sum_{p=1,\ldots,n} k_p a_p^* M a_p
\]

with no contribution from the additive noise (since it has been assumed Gaussian and independent of the signals) and where the kurtosis of the \( p \)-th source is denoted by \( k_p \):

\[
k_p \triangleq \text{Cum} \left( s_p, s_p^*, s_p^*, s_p \right)
\]

Equation (7) evidences that the image space of the quadrnocovariance \( Q \) is spanned by the projectors \( \Pi_p = a_p a_p^* \) (hence the name "FOSS"). It has exactly rank \( n \) if no kurtosis \( k_p \) is zero and if the projectors \( \Pi_p \) are linearly independent. This last condition is fulfilled whenever the signatures \( a_p \) are themselves independent. It follows that quadrnocovariance eigen-decomposition shows only \( n \) non-zero eigenvalues. Let us assume that they are numbered in such a way that the corresponding \( n \) eigen-matrices are \( (E_i)_{i=1,\ldots,n} \). These eigen-matrices form an hermitian orthonormal basis of the FOSS.

3.2 FOSS estimation. When a strongly consistent estimate of the signal covariance is used, the 2nd-order signal subspace estimate obtained via eigen-decomposition also is strongly consistent. The same can be shown to hold for the FOSS estimates obtained from an eigen-decomposition of the sample quadrnocovariance into eigen-matrices. This should be the preferred FOSS estimation method for small arrays but eigen-decomposition of the quadrnocovariance may be too expensive with large arrays.
Note however that only a small number of eigen-matrices need to be estimated ($n$ and not $m^2$) and this fact can lead to large computational savings (see [7]). Even in that case, the whole set of 4th-order cumulants is needed, and quadrivariate estimation cost may be prohibitive. Fortunately, the FOSS can be estimated in a simpler manner. We demonstrate this in the (rather common) case where signals are circularly distributed. Cumulant expression in terms of the moments then reduces to:

$$q_{jk}^{R} = E\{ x_i x_j x_k x_l \} - r_i^{-1} r_j^{-1} r_k^{-1} r_l^{-1}$$

and it readily checked that the quadrivariate image of any matrix $M$ accordingly reduces to:

$$Q(M) = E\{ (M x)(M x)^T \} - RMR - RTR(MR)$$

This expression admits an obvious sample counterpart showing that $Q(M)$ can be estimated at a cost similar to the covariance. This suggests to choose a priori a set of $n$ hermitian matrices $M_i$ and to estimate $Q(M_i)$ according to (10). The result will be a set of $n$ almost surely independent matrices of the FOSS. They can then be orthonormalized (by a Gram-Schmidt procedure for instance) into an orthonormal hermitian basis of the FOSS. Such a procedure obviously yields FOSS estimates with higher variance than those obtained by eigendecomposition of the whole set of 4th-order cumulants. Since it is not possible to ensure in advance that the $Q(M_i)$ actually are independent, a safer solution would be to use a number of $M_i$ larger than $n$.

4. BLIND MUSIC IMPLEMENTATION

From now on, we assume that a FOSS estimate is available in the form of a set of $n$ hermitian orthonormal matrices: $M_i, i=1,n$ forming a basis of the estimated FOSS. The following search implementations do not depend on the particular FOSS estimation technique.

Blind MUSIC can be implemented as searching through the FOSS the closest ROM matrix (see the PQN technique below) or, alternatively, as searching through the ROM the closest FOSS matrix (see IV/3) In both cases, the suggested techniques do not implement the search via a gradient (or similar) approach but expresses the Blind MUSIC estimates as fixed points of an appropriate mapping. In our simulations, we have found that these fixed points were the only stable points: the blind MUSIC search can then be implemented as the iteration of these mappings with arbitrary starting points.

4.1 IV/3 Search. The natural approach to blind MUSIC is to maximize the norm of the projection onto the FOSS of a matrix $A$ under the constraint that it is a 1D projector. Using an orthonormal hermitian basis, the squared norm of this projection, denoted $d$, is the sum of the squared projections onto each basis matrix.

$$d = \sum_{i=1,n} \text{Tr}^2 (A M_i)$$

Blind MUSIC estimates are obtained as the maximizers of $d$ under the constraint that $A = vv^*$ with $v^* v = 1$. Since $\text{Tr}(A M_j) = v^* M_j v$, the variation with respect to $v$ of a Lagrange function $L = 1/2d - \lambda v^* v$ associated to this constrained optimization problem is:

$$\delta L = \sum_{i=1,n} (v^* M_i v) (v^* M_i v) - \lambda (v^* v)$$

Defining the cubic vector mapping $v \rightarrow \phi(v)$ as:

$$\phi(v) = \sum_{i=1,n} (v^* M_i v) M_i v$$

the Lagrange function variation is rewritten in:

$$\delta L = (\phi(v) v) ^* v + \delta v ^* (v^* v)$$

which is zero for any $\delta v$ if $\phi(v) = \lambda v$. This is equivalent to $v$ being a fixed point of the mapping $v \rightarrow \Phi(v)$ where:

$$\Phi(v) = \phi(v) / | \phi(v) |$$

The IV/3 search (where V3 is a reminder for the cubic dependence on the iterated vector) starts with a random vector and then iteratively computes its image through $\Phi$.

4.2 PQN Search. An alternate approach is to search for matrices of the FOSS that are as close as possible to the ROM. The basic idea is to start with an arbitrary matrix of the FOSS and to repeatedly project it onto the ROM and back onto the FOSS. Projection onto the ROM is equivalent to truncating the matrix to its first principal eigen-component, which requires an eigen-decomposition at each step. On the other hand, repeatedly squaring a matrix has the effect of enhancing the dominant eigenvalue. In our experiments, we have found that the projection onto the ROM, being included in the iteration loop, could be replaced by a simple matrix squaring followed by renormalization. The PQN algorithm is just cycling through the three steps of projection, quadrature, and normalization, hence the acronym "PQN". After convergence, the dominating eigen-vector is extracted, providing an estimate of one of the source signatures.

Quadrature and projection can be efficiently implemented in a single step by representing the iterated matrix, say $A$, by its (real) coordinates $a_{ij}, i=1,n$ in the FOSS basis: $A = \sum a_{ij} M_i$. The squared matrix then is $A^2 = \sum a_{ij} a_{jk} M_i M_j$. Since an orthonormal basis is used, the projection of $A^2$ onto the FOSS has coordinates $a_k^2$ given by:

$$a_k^2 = \sum_{i,j} t_{jk} a_i a_j$$

with $t_{jk} = \text{Tr}(M_i M_j M_k)$

Hence, by pre-computing the table $t_{jk}$, each quadrature-projection is computed in the single step (16), involving only $n^3$ real multiplications.

4.3 Simulation results The following simulation results are for a uniform linear half-wavelength array of 4 sensors, two independent PSK modulated sources of unit variance located respectively at 0 and 20 degrees (i.e. under the same lobe). The signal is corrupted by additive Gaussian white noise with covariance $\sigma^2$. We performed 20 Monte-Carlo runs using the PQN algorithm for data lengths of 50, 100, 200, 300, 1000 and for noise levels $\sigma = -10, 0, 10, 20$ dB. For each run, we plot a performance index $\rho$ defined
as

\[ r = \frac{1}{n} \sum_{p=1}^{n} 1 - \text{Tr} \left( \hat{\Pi}_p \Pi_p \right) \]

where each \( \hat{\Pi}_p \) is the estimated \( p \)-th projector. This index also is the squared size of the angle between each signature and its estimate (averaged on the sources).

![Graph showing performance index vs. sample size for different SNR levels.](image)

The graphs above illustrate the performance index for different SNR levels (-10 dB, 0 dB, 10 dB) and sample sizes (50, 100, 200, 500, 1000). The performance index is defined as:

\[ r = \frac{1}{n} \sum_{p=1}^{n} 1 - \text{Tr} \left( \hat{\Pi}_p \Pi_p \right) \]

of the estimates is proportional to the noise power. Another remark is that these simulations are for equipowered sources: the performance would degrade for a source when its power gets weaker. This effect shows at any order but is naturally more severe at fourth-order than at 2nd-order.

Just as in the MUSIC case, asymptotic performance can be obtained in closed form but room is definitely lacking for exposition of the results. They will be presented at the conference.

**CONCLUSION.** The notion of fourth-order signal subspace (FOSS) has been introduced. This matrix space, function of the 4th-order cumulants, is the natural 4th-order counterpart of the classical 2nd-order signal subspace. Exploited in a "blind MUSIC" fashion, it allows for blind identification without resorting to 2nd-order information. This can be done with one of the low cost fixed point techniques presented here.

**REFERENCES**

PHASE RECONSTRUCTION OF AUTOREGRESSIVE PROCESSES USING THIRD ORDER CUMULANTS

Asoke K NANDI\(^1\) and Ralf MEHLAN\(^2\)

\(^1\)Department of Electronic and Electrical Engineering, University of Strathclyde, Royal College, 204 George Street, Glasgow, G1 1XW, UK

\(^2\)Lehrstuhl für Elektrische Regelungstechnik, RWTH Aachen, Templergraben 55, 5200 Aachen, Germany

The knowledge of the phase is very important in many situations. It is well known [1-2] that phase information is, in principle, available from the third order cumulants (except when third order cumulants happen to be zero as in the Gaussian case). Methods for reconstructing phases in autoregressive (AR) processes, using estimated third order cumulant sequences from the time-series data are presented in this paper and they are shown to have the consistency of performance.

1. INTRODUCTION

In many situations the knowledge of the phase is very useful and sometimes extremely important. While the second order statistics (SOS) is widely used and is very useful, it has its limitations. It is well known [1-2] that the phase information is preserved in higher order statistics (HOS) - that is, for order greater than two. In principle then, the phase information is available from the third order cumulants (except when third order cumulants happen to be zero as in the Gaussian case).

The bispectrum, \(B(\omega_1, \omega_2)\), of a time-series, \(x(n)\), is in general a complex quantity and it can therefore be written as

\[ B(\omega_1, \omega_2) = |B(\omega_1, \omega_2)| \exp(j\phi(\omega_1, \omega_2)), \]

where \(|B(\omega_1, \omega_2)|\) is referred to as the bispectal magnitude and \(\phi(\omega_1, \omega_2)\) is called the bispectral phase. If the time-series can be well modelled as the output of an AR filter, with the system function \(H(z)\), driven by ideal, white noise the bispectrum can be written as

\[ B(\omega_1, \omega_2) = \beta H(\omega_1) H^*(\omega_2), \]

where the constant \(\beta\) is often called the skewness or \(C_3(0,0)\) of the ideal, white noise. The phase of the bispectrum is related to the phase of the filter by

\[ \phi(\omega_1, \omega_2) = \phi(\omega_1) + \phi(\omega_2) - \phi(\omega_1 + \omega_2), \]

where \(\phi(\omega)\) is the phase of the model filter and is also referred to as the phase of the observed signal, \(x(n)\). Non-parametric phase reconstruction algorithms [3-8] try to recover the phase samples \(\phi(\omega)\) from the estimates of the bispectrum phase, \(\phi(\omega_1, \omega_2)\).

Parametric phase reconstruction methods are based on the assumption that the observed time-series is the output from a specific type of process, say obeying an AR model. Once such identification of the process is made the transfer function of the filter, \(H(\omega)\), becomes a known function with unknown model parameters, \(a_i\)'s, which are the parameters of the model. Now if these model parameters can be obtained, the \(H(\omega)\) can be numerically computed and subsequently the phase, \(\phi(\omega)\), can be estimated. Thus the problem in parametric phase reconstruction methods can be reduced to how to best estimate the model parameters (and, if need be, the model order) from the observed time-series, \(x(n)\). Usual (dis)advantages of parametric methods over non-parametric methods as in the SOS apply here.

2. TIR EQUATIONS

For AR models a set of linear equations in terms of the model parameters and third order cumulants can be derived [7] such along the lines of Yule-Walker equations for second order cumulants [8]. In [7] Nikias and Raghuveer derived the following set of linear equations for AR models

\[ C_3(-m,-n) + \sum_{i=1}^{p} a_i C_3(i-m,i-n) = \beta \delta(n,n) \]  

(2.1)

where \(p\) is the order of the AR model, \(a_i\)'s are the model parameters, and \(C_3(m,n)\) represents the third order cumulant of lags \(m\) and \(n\) of the observed time-series. These were termed the third order recursion (TIR) equations.

By choosing the diagonal slice, namely setting \(m = n\) and choosing \(m = 0, 1, \ldots, p\), a set of \((p+1)\) linear equations are obtained with \((p+1)\) unknown parameters of the AR model. In matrix notation the set of equations can be written as

\[ \beta A = \beta, \]

(2.2)
where

$$
\mathbf{Z} = \begin{bmatrix}
C_0(0,0) & C_0(1,1) & \cdots & C_0(p,p) \\
C_0(-1,-1) & C_0(0,0) & \cdots & C_0(p-1,p-1) \\
\vdots & \vdots & \ddots & \vdots \\
C_0(-p,-p) & \cdots & \cdots & C_0(0,0)
\end{bmatrix}
$$

\(\mathbf{A} = [a_0, a_1, \ldots, a_p]^T\) of dimension \((p+1)x1\),
\(\mathbf{B} = [\theta_0 \ldots 0]^T\) of dimension \(nx1\),
\(\mathbf{f}\) is an \(nx(p+1)\) dimensional matrix;

Step 3: Obtain the least squares solution of (3.1) from

$$
\mathbf{A} = (\mathbf{f}^T \mathbf{f})^{-1} \mathbf{f}^T \mathbf{B} \quad (3.2)
$$

In this algorithm, it is yet to be specified how many TOR equations should be used (i.e., the value of N) and which specific third order cumulants (values of m and n) should be included in this overetermined set of equations.

3.2. SELECTION OF TOR EQUATIONS

The aim of this section is to outline how the investigation is carried out to discover the set of indices \((m,n)\) that generate TOR equations, from (2.1), in the form of (3.1) such that the matrix \(\mathbf{f}\) is of rank \((p+1)\). Three strategies are adopted with respect to two criteria - 1) the matrix \(\mathbf{f}\) should be of rank \((p+1)\) with the minimum of TOR equations, and 2) the accuracy of phase estimates from (3.2) as measured in terms of the mean square error with respect to the exact phase must be less than that obtained from (2.2).

The following three strategies are used to determine the selection of TOR equations:

A) Here the TOR equations come from a diagonal slice in the \((m,n)\) index-plane but differently from (2.2). First the \((p+1)\) equations are chosen corresponding to \(m=0, 1, \ldots, p\). Then additional equations are chosen along the diagonal of the \((m,n)\) index-plane to satisfy the above two criteria.

B) Now the TOR equations come from a growing triangle in \((m,n)\) index plane with column by column growth. It starts with the following \((m,n)\) values in the order they are stated - \((0,0),(1,0),(1,1),(2,0),(2,1),(2,2)\), and so on.

C) In this strategy diagonal as well as non-diagonal indices are used, as in B) but in the following order of \((m,n)\) values - first the full triangle \(\{(0,0), (1,1), \ldots, (p,p)\}\), \(\{(1,0), (2,1), \ldots, (p-1,p)\}\), \(\{(2,0), (3,1), \ldots, (p-2,p)\}\), \(\{(p,0)\}\), and then the columns \(\{(p+1,p+1), (p+2,p+1), \ldots, (p+1,0)\}\), \(\{(p+2,p+2), (p+2,p+1), \ldots, (p+2,0)\}\), and so on.

Simulations have been carried out to answer which of above three strategies is the best and how many TOR equations are needed.
4. SIMULATION

Twenty different AR models of order up to seven are chosen; some of these have been used in [7,14]. For each of these models, the exact third order cumulants were distorted with the following error model to represent the estimated cumulants

\[ \hat{C}_3(m,n) = C_3(m,n) \cdot [1 + \epsilon] \]

where \( \epsilon \in [-\nu, +\nu] \) and is uniformly distributed. For example, \( \nu=0.1 \) means that no estimated cumulant deviates from the exact value by more than 10%. For each model, ten different values of \( \nu \) are chosen. Thus for each of the twenty AR models and each of the ten values of \( \nu \), a set of \( \hat{C}_3(m,n) \) for various values of \( m,n \) (i.e. there are 200 such sets) is obtained.

Now in each of these 200 sets:

1) samples of the phase are estimated via (2.2) and the mean square error (MSE) is computed as

\[ \text{MSE}(2.2) = \frac{1}{P} \sum_{i=1}^{P} (\phi(i) - \hat{\phi}(i))^2 \]

where \( \phi(i) \) and \( \hat{\phi}(i) \) represent the exact phase and the estimated phase via (2.2), and \( P \) is the number of phase samples;

2) samples of the phase are estimated via (3.2) according to strategies A, B, and C but for various values of \( N \), the number of TQR equations. For each strategy, the MSE depends on \( N \) and is computed from

\[ \text{MSE}(A,N) = \frac{1}{P} \sum_{i=0}^{P} (\phi(i) - \hat{\phi}_{A,N}(i))^2 \]

where \( \phi(i) \) represent the exact phase, while \( \hat{\phi}_{A,N}(i) \) represent the estimated phase via (3.2), and \( P \) is the number of phase samples; correspondingly MSE(B,N) and MSE(C,N) can be defined;

3) for each values of \( N \), calculate the following normalised mean square errors

\[ \text{NMSE}(A,N) = \frac{\text{MSE}(A,N)}{\text{MSE}(2.2)} \]
\[ \text{NMSE}(B,N) = \frac{\text{MSE}(B,N)}{\text{MSE}(2.2)} \]
\[ \text{NMSE}(C,N) = \frac{\text{MSE}(C,N)}{\text{MSE}(2.2)} \]

Note that if NMSE(C,N) is less than 1.0, the proposed method with strategy C with \( N \) TQR equations is able to reconstruct the phase better than using (2.2) which uses cumulants on the specific diagonal slice.

4.1. RESULTS - PART I

The following general points are observed:

1) Selecting the TQR equations following strategy A appears to have large fluctuations in NMSE(A,N) around the value of 1 as \( N \) changes. No clear advantage in the strategy A over (2.2) emerges.

2) The NMSE(B,N) is a decreasing function of \( N \) although the rate of decrease slows down for large \( N \) values. When \( N \) is small (the matrix \( Y \) is not of rank \((p+1)\)), NMSE(B,N) is larger than 1; while NMSE(B,N) is smaller than 1 for larger values of \( N \).

3) In the cases where TQR equations follow the strategy C, the NMSE(C,N) decreasingly fluctuates for smaller values of \( N \) and settles down to small values, much less than 1, as \( N \) increases.

Based on the above observations, which are independent of the values of \( \nu \) and the AR process studied, it is concluded that the strategy A should be discarded and that the strategy C appears to be the best choice. With the strategy C and for an AR(p) process, the value of \( N = (p+1)(p+2)/2 \) tends to be the point beyond which the NMSE(C,N) settles down to the 'lowest error level'. It is suggested that a total of \( N^*=(p+2)(p+3)/2 \) TQR equations following the
strategy C is sufficient to guarantee a considerable reduction in the error of the phase estimates of an AR(p) process.

4.2. RESULTS - PART II

As an example, an AR(3) process is chosen with parameters $a_0=1$, $a_2=0.7$, $a_3=0.2$, and $a_4=0.15$. The value of $\nu$ is set to 0.2 and ten different sets of the third order cumulants, $C_3(m,n)$, are obtained. For each of these ten sets of cumulants, phases are estimated using the diagonal slice with (2.2). The resulting ten sets of phase estimates are displayed in fig. 1.1. In the figures here, estimated phases (in radians) of the AR process under study are plotted versus the 'frequency' (in radians). Using the same cumulants, phase estimates are obtained using (3.2) with N TOR equations following strategy C. These new sets of phase estimates are plotted in fig. 1.2. It is clear that estimates in fig. 1.2 are lot less variable than the ones in fig. 1.1.

As a second example, an AR(4) process is chosen with parameters $a_0=1$, $a_1=0.8$, $a_2=0.7$, $a_3=0.2$, and $a_4=0.1$. Setting the maximum error in any cumulant to be 20% (i.e. $\mu=0.2$), ten different sets of the third order cumulants, $C_3(m,n)$, are obtained. For each of these ten sets of cumulants, phases are estimated using the diagonal slice with (2.2). The resulting ten sets of phase estimates are displayed in fig. 2.1. For each of the same sets of cumulants, phase estimates are obtained from (3.2) with 21 TOR equations following strategy C. These new sets of phase estimates are plotted in fig. 2.2. It is clear that estimates in fig. 2.2 (from the proposed method) are lot less variable than the ones in fig. 2.1.

5. SUMMARY

In this paper a new algorithm based on (3.2) has been proposed to estimate the phase of an AR filter. Although the algorithm based on (2.2) is computationally efficient the proposed algorithm in this paper provides much less variable and more consistent phase estimates than the algorithm based on (2.2).

REFERENCES

DIGITAL CALCULATION OF THE FUNDAMENTAL SINEWAVE OF SIGNALS USING METHODS CONSTRAINED IN FREQUENCY DOMAIN

Tadeusz ŁOBOS Piotr RUCZEWSKI Jarosław M. SZYMŃA

Technical University of Wrocław, PL-50370 Wrocław, pl. Grunwaldzki 13, Poland

Real-time determination of parameters of voltages and currents is very important for measurement, control and protection tasks in electrical power systems. The paper presents a method which enables us to develop fast filter algorithms with auxiliary constraints. The problem of designing is converted to an optimization problem with constraints using standard least-square criterion. The minimization problem is mapped to a simpler problem of solving a set of linear equations, by using the Lagrange multipliers. Using the method we can develop filter algorithms for estimating the phasor of fundamental waveforms of signals optimizing the algorithms for concrete application.

1. INTRODUCTION

Serious problems for the control and protection of electrical power systems arise in estimating the fundamental waveforms of voltages and currents from measured signals with high noise levels. Various algorithms have been suggested to estimate the fundamental waveforms buried in noise and distorted by higher harmonics [1-4]. The methods presented in the literature optimize the estimation in the time domain. The waveforms of voltages and currents in electrical power systems may include harmonics of basic waveforms due to nonlinear elements and random noise due to faults and other disturbances. In some case, e.g. in circuits with thyristor converters, it is possible to predict the frequencies of the higher harmonics. To estimate the phasor of the fundamental waveform in such situations we have developed and investigated filter algorithms which enable us to suppress specific higher harmonics effectively.

To estimate the real part of the phasor the argument of the transfer function of the algorithm for the fundamental frequency should be equal to zero, and for imaginary part equal to $-\pi/2$. In the both cases the magnitude of the transfer function for the fundamental frequency should be equal to one. In the developed method we can prescribe the values of the magnitude and argument, the values of the derivatives of the magnitude and argument at any frequency, adjusting the filter characteristics to special requirements of the application.

The proposed methods are illustrated by some numerical examples.

2. CONSTRAINTS OF THE FREQUENCY RESPONSE

The frequency characteristic of a digital non-recursive filter should approximate the periodic function $G(j\omega)$

$$G(j\omega) = \sum_{k=1}^{m} g_k e^{-jk\omega T}$$  \hspace{1cm} (1)

where

$$g_k = \frac{T}{2\pi} \int_{-\pi/T}^{\pi/T} G(j\omega)e^{-jk\omega T}d\omega$$  \hspace{1cm} (2)

$T$ - sampling interval,  
$\omega$ - angular frequency,  
$g_k$ - coefficients of the filter being the discrete values of the impulse response of an ideal filter.

The function (1) describes the characteristic of a noncausal ideal filter. We are looking for a causal filter with a finite impulse response, which frequency characteristic is possibly closed to the ideal. For a given characteristic $G(j\omega)$ and a given filter order $N$ we have to calculate the coefficients $h_n$ of the real filter by minimizing the objective function

$$Q = \int_{-\pi/T}^{\pi/T} |G(j\omega)-H(j\omega)|^2 d\omega$$  \hspace{1cm} (3)

where

$$H(j\omega) = \sum_{k=0}^{N-1} h_k e^{-jk\omega T}$$

This work was mainly supported by the State Committee for Scientific Research (Poland), grant 306619101
It is required that the partial derivatives of the function with respect to $h_k$ be zero:

$$\frac{\partial Q}{\partial h_k} = 0 \quad \text{for } k=0,1,\ldots,N-1 \quad (4)$$

In this way we obtain the desired coefficients $h_k$, which are the coefficients of a truncated Fourier series ($h_k \neq 0$).

In the proposed method we will prescribe the values of the magnitude and argument, the values of the derivatives of the magnitude and argument at any points, chosen according to the desired application. In this way we obtain a set of linear equations which can be solved for $h_k$.

Introducing the notation

$$H(j\omega) = H(\omega) e^{j\phi(\omega)} \quad (5)$$

and using eqn. 3 we obtain

$$H(\omega) = \sum_{k=0}^{N-1} h_k e^{-j(\omega k T + \phi(\omega))} \quad (6)$$

The complex eqn. 6 can be expressed in a form of two real equations

$$\sum_{k=0}^{N-1} h_k \cos(\omega k T + \phi(\omega)) = H(\omega) \quad (7)$$

$$\sum_{k=0}^{N-1} h_k \sin(\omega k T + \phi(\omega)) = 0 \quad (8)$$

Using (7) and (8) we can also introduce the non-zero constraints.

If the magnitude of a transfer function for assumed frequency should be equal to zero, the argument at the point is not determined and eqns. 7 and 8 take the form

$$\sum_{k=0}^{N-1} h_k \cos(\omega k T) = 0 \quad ; \quad \sum_{k=0}^{N-1} h_k \sin(\omega k T) = 0 \quad (9,10)$$

Now we can obtain equations representing the conditions for derivatives by differentiation of the eqn. 6 with respect to the frequency $\omega$.

In this way we can determine the number of equations we need in order to take into account the assumed constraints. The filter order is equal to the number of equations. As solution of the set of equations we obtain the coefficients $h_k$ ($k=1,0,\ldots,N-1$) of a nonrecursive filter algorithm.

The set of equations can be described as

$$\sum_{k=0}^{N-1} C_{ik} h_k = f_i \quad \text{for } i=1,2,\ldots,D \quad (11)$$

where

$C_{ik}, f_i$ - coefficients depending on the constraints and the sampling frequency,

$D=N$ - number of equations equal to the filter order.

3. APPROXIMATION OF FREQUENCY CHARACTERISTICS WITH CONSTRAINTS

In this chapter we will study design methods which can be classified as optimization methods with constraints. When the filter order $N$ can be higher than the number of design equations (11) we can achieve improved frequency characteristic, adjusting it to requirements of the application. The set of equations cannot be solved explicitly to give the filter coefficients $h_k$. We apply a mathematical optimization procedure that minimizes an error function, subject to the characteristic of an ideal filter and to the constraints. Using the Lagrange multipliers [5] we obtain the objective function.

$$Q_0 = Q_0 + \sum_{i=1}^{D} \sum_{k=0}^{N-1} C_{ik}^* h_k - f_i^* \quad (12)$$

where

$u_i$ - Lagrange multipliers

The optimization procedure requires that the partial derivatives of the function $Q_0$ with respect to coefficients $h_k$ and Lagrange multipliers $u_i$ be zero

$$\frac{\partial Q_0}{\partial h_k} = 0 \quad \text{and} \quad \frac{\partial Q_0}{\partial u_i} = 0 \quad (13)$$

Applying the coefficients of a truncated Fourier series $g_k$ (as in section 2) describing the approximated characteristic of an assumed filter, we obtain the set of linear equations

$$\sum_{k=0}^{N-1} C_{ik} h_k = f_i \quad ; \quad h + \sum_{i=0}^{D} C_{ik} u_i = g_k \quad (14,15)$$

which can be written in a compact matrix notation

$$C H = F \quad ; \quad H + C^T U = G \quad (16,17)$$

where

$F=(f_1,f_2,\ldots,f_D)^T$ - coefficients depending on the constraints and the sampling frequency,
Fig. 1. Magnitude and phase spectrum of transfer function of algorithm for (a) real part (basic sine wave), (b) imaginary part (orthogonal function) of a phasor. The zeros of transfer function at the 5th, 7th and 11th harmonics, \( f_s = 1200 \) Hz, \( N = 8 \).

Fig. 2. Magnitude and phase spectrum of transfer function of algorithm for (a) real part (basic sine wave), (b) imaginary part (orthogonal function) of a phasor. Maximum of transfer function at 50 Hz (basic waveform) Zeros at all odd harmonics, \( f_s = 1000 \) Hz, \( N = 15 \).
$D = (e_0^T, e_1^T, \ldots, e_{N-1}^T)^T$ - coefficients of a truncated Fourier series,

$H = (h_0, h_1, \ldots, h_{N-1})^T$ - the desired coefficients of a filter algorithm,

$U = (u_1^T, u_2^T, \ldots, u_k^T)^T$ - constraints of a filter characteristic,

$C = [C_{i,k}]_{D\times N}$ - coefficients depending on the constraints and the sampling frequency.

Hence it follows the solution of the problem

$$ H = G + C^T (C^T C)^{-1} \left( F - C \Phi \right) \quad (10) $$

The special cases are

1. $H = G$ - truncated Fourier series ($D = 0$)

2. $H = C^{-1} F$ - where $D = N$

4. FILTERS DESIGNING

A computer program for designing of the filter algorithms has been elaborated and frequency characteristics of developed algorithms have been investigated.

Fig. 1 shows frequency magnitude and phase response of algorithms for estimating the phasor of the fundamental sinusoidal current and voltage in three-phase circuits with thyristor converters. As constraints the magnitude of the frequency response of the algorithms is zero at the 5th, 7th and 11th harmonics. The phase response of the algorithms for estimation of the real part of the phasor is, at the fundamental frequency (50 Hz), equal to zero, and for the imaginary part equal to $-\pi/2$. Substituting into eqns. 7 and 8 $\omega = 2\pi \times 50$, $\varphi(\omega) = 0$ and $H(\omega) = 1$ we obtain two equations, and substituting into eq. 8 and 10 successive 2x200, 2x500 and 2x550 we obtain the next six equations.

The set of 8 equations can be solved for $h$ ($N=8$), which are coefficients of the algorithm for determination of the real part of the phasar. To calculate the coefficients of the algorithm for the imaginary part of the phasor, we substitute into eqns. 7 and 8 $\varphi(\omega) = -\pi/2$.

The other six equations remain the same as for the algorithm for the real part of the phasor. By the sampling frequency of $f = 1200$ Hz the filter order is equal to $N=8$. In this way we have minimized the sampling window and the filter order. The transient response time is equal to one third period of the main sinusoid. When estimating the phasor of the fundamental sinusoid, it is important that the values of the magnitude of transfer function do not change considerably by varying the network frequency in a small range. This can be achieved when the maximum of the frequency response is exactly at the fundamental frequency. Thus, the first derivative of the magnitude spectrum at the frequency should be equal to zero. The widely used Fourier algorithms [2] do not have a maximum at the fundamental frequency if the sampling window is not exactly equal to one period. As an example Fig. 2 shows the frequency characteristics of algorithms developed according to optimization method with constraints.

In this case we have assumed zeros of the transfer function at all odd harmonics and the first derivative of the function at the fundamental harmonic equal to zero. The optimization procedure minimizes the error function, subject to the characteristic of an ideal lowpass filter with a cutoff frequency of $f = 100$ Hz, taking into account the constraints.

5. CONCLUSIONS

The widely used algorithms for online estimation of the fundamental sinusoidal currents and voltages in electrical power systems are based on the Fourier technique or on the Kalman filter theory. They optimize the estimation in the time domain approximating the distorted signals by sinusoidal signals. The proposed method enables us to develop filter algorithms for estimation of the phasor of the main component, which frequency responses have specific properties. It is possible to prescribe the values of the transfer function and its derivatives at any frequency. The optimization procedure makes it possible to minimize an error function subject to the characteristic of an ideal filter, taking into account the additive constraints.

Using this method we can optimize the selection of algorithms for concrete application.

REFERENCES

Generalized Wigner-Ville Distributions: Examples and Properties

Boualem Boashash and Branko Ristic

Centre for Signal Processing Research
Queensland University of Technology
GPO Box 2434, Brisbane, Q 4001, Australia

This paper presents the concept of reduced time-varying higher order spectra which is based on generalized Wigner-Ville distributions. The relationship with the higher-order Wigner-Ville distributions and the general properties in the context of the time-frequency distributions are studied. Two members of the class are addressed and their applications illustrated.

1 Introduction

Time-varying higher order spectra have become a very attractive research area recently, due to their potential applications. The earlier work by Gerr [9], defined the third-order Wigner distribution (i.e. Wigner bispectrum) as the two-dimensional Fourier transform of the third order time-varying moment $m_3(t, \tau_1, \tau_2)$:

$$W_3(t, \tau_1, \tau_2) = \mathcal{F} \left\{ m_3(t, \tau_1, \tau_2) \right\}$$

(1)

Subsequently, the work by Gerr was extended and generalized in [12], [8] and [10]. Swami, [13], studied the properties of the Wigner bispectrum and proposed its extension to higher-order time-frequency distributions. In [8], Wigner higher-order spectra are defined as a slightly modified extension of eq.(1). Finally, in [10], a cumulant based higher-order Wigner-Ville Distribution (WVD) was proposed, again following the method of Gerr [9].

Definition of the higher-order WVD. Consider the k-th order 'moment' of a deterministic signal $x(t)$:

$$m_k(t, \tau) = x(t + a_1)x(t + a_2)\ldots x(t + a_k)$$

(2)

where $\tau = [\tau_1 \ldots \tau_{k-1}]$ and the lags $a_1, a_2, \ldots, a_k$ are required to satisfy:

$$a_1 + a_2 + \ldots + a_k = 0 \quad \text{(lag centering)}$$

(3)

$$a_2 - a_1 = \tau_1; a_3 - a_2 = \tau_2; \ldots; a_k - a_{k-1} = \tau_{k-1}$$

(4)

The $k$-th order WVD is defined as:

$$W_k(t, \tau) = \int_{-\infty}^{\infty} \ldots \int_{-\infty}^{\infty} m_k(t, \tau) e^{-j2\pi T \tau} d\tau_1 \ldots d\tau_{k-1}$$

(5)

where $\tau = [\tau_1 \ldots \tau_{k-1}]$ and $T$ denotes a transpose.

If $x(t)$ is a random signal, then $m_k(t, \tau)$ is defined with the expectation value operator $E$. In [8], eq.(2) is slightly modified, having $x^*(t + a_1)$ instead of $x(t + a_1)$.

The constraints (3) and (4) are common to [13], [8] and [10]. They ensure the $k$-th order WVD of a stationary zero mean random signal to be equal to the $k$-th order (time-invariant) moment spectra of the signal. The entire concept of the higher-order WVDs can be expressed in terms of cumulants rather than moments [10].

In parallel with the work on the higher-order WVDs, a generalized WVD (GWVD) was introduced and studied in [9], [4], [6] and [3].

Definition of the generalized WVD. For a deterministic signal $x(t)$, the generalized WVD is defined as a Fourier transform of a $k$-th order time-varying moment reduced to one lag variable, and expressed in a symmetric form as:

$$m_k^g(t, \tau) = \prod_{i=1}^{k} x_i(t + c_i \tau) x_i^*(t + c_{-i} \tau)$$

(6)

that is:

$$W_k^g(t, \tau) = \int_{-\infty}^{\infty} m_k^g(t, \tau) e^{-j2\pi T \tau} d\tau$$

(7)
where \( k = \sum_{i=1}^{p} (|b_i| + |c_{-i}|) \). For zero-mean random signals, the expectation value is used in definition (6).

The choice of the coefficients \( q, b_i \) and \( c_i \) depends on the particular problem/application, as it will be illustrated in section 2. Comparing the two different concepts, the higher-order WVDs and the generalized WVDs, we observe the following:

1. \( W_k^f(t, f) \) would be a two-dimensional slice of the \( k \)-dimensional \( W_k(t, f) \), if the appropriate complex-conjugate operations were used in (2).

2. In the generalized WVD, lag constraints are imposed so that the GWVD of a complex sinusoid \( x(t) = e^{j2\pi f_0 t} \) is:

\[
W_k^f(x(t))(t, f) = \delta(f - f_0) \quad (\forall k \in N)
\]  

(8)

Satisfying the above condition, the GWVD automatically fulfills constraints (3) and (4), that is a \( k \)-th order GWVD of a stationary signal equals the one-dimensional slice of its \( k \)-th order moment spectra.

3. The physical interpretation of the generalized WVD for deterministic or random signals is straightforward for two reasons: (i) it has a single frequency axis; (ii) the energy of a sinusoid appears as a spike at the frequency \( f_0 \), eq.(8). The interpretation of the higher-order WVDs is very difficult.

2 Two members of the class of the GWVD

In this section we illustrate two members of the class of the generalized WVDs. Each of them is dedicated to a specific problem (application) that can not be optimally solved using conventional time-frequency distributions. Notice that the WVD is a member of the class of GWVDs with parameters: \( q = 1, b_1 = b_{-1} = 1, (k = 2) \), and \( c_1 = -c_{-1} = 0.5 \).

2.1 A tool for analysis of polynomial phase signals

Consider a polynomial phase signal in noise:

\[
x(t) = a(t) e^{j\phi(t)} + n(t)
\]

(9)

where

\[
\phi(t) = 2\pi \sum_{n=1}^{p} b_n t^n
\]

(10)

If \( p \leq 2 \) (linear FM) and \( a(t) \) non-random, the conventional WVD would produce the optimal time-frequency representation of the signal \( x(t) \). Moreover, the estimate of the instantaneous frequency from the peak of the WVD, if noise \( n(t) \) is white and Gaussian, would be unbiased and its variance would meet the CR bound [2]. For case \( p > 2 \), we could either use the dechirping approach proposed in [11], or the generalized WVD in order to obtain an optimal time-frequency signal representation. For example, if \( p = 3 \) or \( p = 4 \), the 6-th order reduced time-varying moment has to be used [4, 6], with parameters:

\[
q = 2 \\
b_1 = b_{-1} = 2 \\
b_2 = b_{-2} = 1 \\
c_1 = -c_{-1} = 0.675 \\
c_2 = -c_{-2} = -0.85
\]

(11)

Figures 1(a) and 1(b) illustrate the conventional WVD and the generalized WVD with parameters (11), of the same quadratic FM \( (p = 3) \) signal. The superior behaviour of the latter is indicated by the sharpness of the peaks in the Fig.1(b). From the peaks of the GWVD, the quadratic law can be recovered easily as opposed to the smeared WVD showing oscillations without any physical meaning.

2.2 A tool for analysis of FM signals in multiplicative noise

Consider again the signal given by (9) and (10) and assume that \( a(t) \) is a real white noise. By extending the work by Dwyer [7] and Abeyesekera [1], it can be shown that a 4-th order reduced spectra is a very effective tool for analysis of such signals. This member of the class of the generalized WVD is studied in detail in [12] and is referred to as a reduced Wigner bispectrum (RWT). Its parameters are:

\[
q = 1 \\
b_1 = b_{-1} = 2 \\
c_1 = -c_{-1} = 0.25
\]

(11)

Figure 2(a) and 2(b) show the WVD and the RWT of the same linear FM signal. Notice that the WVD is completely blind in this example, unable to detect the presence of a deterministic component in the noisy signal. The RWT clearly shows the existence of a linear FM.

3 Properties of the generalized WVDs

The generalized WVD has preserved and generalized a great deal of the properties that characterize the WVD. In derivation of the following properties we assumed: \( b_i = b_{-i} \) and \( c_i = -c_{-i} \). The properties are valid for \( \forall k \in N \).

P-1. The GWVD is real for any signal \( x(t) \):

\[
W_k^f(x(t))(t, f) = [W_k^f(x(t))(t, f)]^*
\]

(12)
The GWVD is an even function of frequency if \( x(t) \) is real:

\[
W_{k(x(t))}^\phi(t, -f) = W_{k(x(t))}^\phi(t, f)
\]  

(13)

P-3. A shift in time by \( T \) and in frequency by \( F \) (i.e., modulation by \( e^{j2\pi Ft} \)) of signal \( x(t) \) results in the same shift in time and frequency of the GWVD:

\[
W_{k(x(t) + T)e^{j2\pi Ft}}^\phi(t, f) = W_{k(x(t))}^\phi(t - T, f - F)
\]

(14)

(\( \forall T, F \in \mathbb{R} \))

P-4. The GWVD preserves the convolution and modulation. If \( x(t) = u(t) + z(t) \) and \( y(t) = w(t)z(t) \) then:

\[
W_{k(x(t))}^\phi(t, f) = W_{k(u(t))}^\phi(t, f) *_{1} W_{k(z(t))}^\phi(t, f)
\]

(15)

\[
W_{k(x(t))}^\phi(t, f) = W_{k(u(t))}^\phi(t, f) *_{f} W_{k(z(t))}^\phi(t, f)
\]

(16)

P-5. Projection of the \( W_{k(x(t))}^\phi(t, f) \) to the time axis:

\[
\int_{-\infty}^{\infty} W_{k(x(t))}^\phi(t, f) df = |x(t)|^2
\]

(17)

P-6. The local moment of the GWVD in the frequency gives the instantaneous frequency of the signal \( x(t) \):

\[
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ W_{k(x(t))}^\phi(t, f) \right]^2 df dt = \int_{-\infty}^{\infty} \left| \frac{d}{dt} x(t) \right|^2 dt
\]

(19)

P-7. Moyal's formula:

\[
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left[ W_{k(x(t))}^\phi(t, f) \right]^2 df dt = \int_{-\infty}^{\infty} \left| \frac{d}{dt} x(t) \right|^2 dt
\]

(20)

P-8. Projection of \( W_{k(x(t))}^\phi(t, f) \) to the frequency axis:

\[
\int_{-\infty}^{\infty} W_{k(x(t))}^\phi(t, f) dt = \left| X(f) \right|^2
\]

(21)

P-9. Total volume under the generalized WVD:

\[
\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} W_{k(x(t))}^\phi(t, f) df dt = \int_{-\infty}^{\infty} |x(t)|^2 dt
\]

(22)

The last term in (22) is a cross-term which is indistinguishable from the two auto-terms. This property of the GWVDs (as well as the higher order spectra in general) makes the analysis of multicomponent signals very difficult.

4 Conclusion

This paper compared two different approaches to time-varying higher-order spectra, the higher-order WVDs and the generalized WVDs. Two members of the class of the GWVDs are shown to have practical meaning since they can optimally solve problems that conventional signal processing tools can not. Finally, the properties of such GWVDs are presented. These properties are either identical or generalized versions of the properties of the WVD.

References


LINEAR PREDICTION, MAXIMUM FLATNESS, MAXIMUM HIGHER-ORDER ENTROPY ON AR POLYSPECTRAL ESTIMATION

Chong-Yung Chi
Department of Electrical Engineering, National Taing Hua University, Hsinchu, Taiwan, ROC.

Abstract: It is well-known that linear prediction (LP) spectral estimator is equivalent to maximum entropy spectral estimator [1,2] and they are also equivalent to maximum spectral flatness spectral estimator [3,4] for AR processes of known order. In this paper, we propose a new higher-order statistics (HOS) based linear prediction error (LPE) filter. We also present LP polyspectral estimator, maximum polyspectral flatness polyspectral estimator, maximum higher-order entropy polyspectral estimator and equivalencies on these polyspectral estimators. The results presented in this paper provide some theoretical foundations on the polyspectral estimation and modeling of non-Gaussian AR processes.

1. Introduction

It is well-known on correlation (second-order statistics) based autoregressive (AR) spectral estimation that linear prediction (LP) spectral estimator is equivalent to maximum entropy spectral estimator [1-2] as well as maximum spectral flatness spectral estimator [3,4] as the order of AR processes is known a priori, although they are based on different theoretical foundations. Among the existing cumulant (higher order statistics (HOS)) based AR parameter estimators, many of them (5) and references therein) are developed based on fitting a set of linear equations by least-squares method without resort to the minimization of a cost function of the prediction error as LP spectral estimator is based on minimum mean-squared prediction error, because, to the author's knowledge, a well-defined linear prediction error filter (LPE) based on HOS is never reported in the open literature.

In this paper, in order to provide some theoretical foundations on the polyspectral estimation and modeling of non-Gaussian AR processes, we begin with a brief review of non-Gaussian AR processes in Section 2, and then present a new HOS based LPE filter, maximum polyspectral flatness measure and maximum higher-order entropy criterion for AR polyspectral estimation, in Sections 3, 4 and 5, respectively. We also present equivalencies on these polyspectral estimators. Finally, we provide a discussion and some conclusions.

2. Non-Gaussian AR Processes

Assume that \( x(k) \) is the output of a causal autoregressive (AR) model of order \( p \) as follows:

\[
x(k) = - \sum_{i=1}^{p} a(i) x(k-i) + u(k)
\]

where \( u(k) \) is real, zero-mean, independent identically distributed (i.i.d.), non-Gaussian with \( M \)-th order cumulant \( \gamma_M \). The transfer function

\[
A(z) = 1 + \sum_{i=1}^{p} a(i) z^{-i}
\]

is assumed to be minimum-phase for \( x(k) \) to be stationary. Let \( C_{M,N}(k) \) where \( k = (k_1, k_2, \ldots, k_{M-1}) \) denote the \( M \)-th order cumulant function of \( x(k) \). The \( M \)-th order polyspectrum, denoted \( S_{M,N}(f) \) where \( f = (f_1, f_2, \ldots, f_{M-1}) \) of \( x(k) \), which is the \( (M-1) \)-dimensional Fourier transform of \( C_{M,N}(k) \), is given by

\[
S_{M,N}(f) = \gamma_M \frac{1}{A(f)} \frac{1}{A(-f)} \ldots \frac{1}{A(-f_{M-1})} \cdot \frac{1}{A^*(f_1 + \cdots + f_{M-1})}
\]

where \( A(f) = A(z = e^{2\pi if}) \). Next, let us proceed with the new LPE filter based on the HOS of \( x(k) \).

3. Prediction Error Filter Based on HOS

Let \( \hat{h}(z) \) be the transfer function of the LPE filter of order \( p \) with \( \hat{h}(0) = 1 \). The output, denoted \( e(k) \), of the LPE filter with the input \( x(k) \) is then

\[
e(k) = x(k) + \sum_{i=1}^{p} \hat{h}(i) x(k-i)
\]

The new HOS based LPE filter \( \hat{h}(z) \) is the one which minimizes the sum of absolute squares of \( M \)-th order cumulant function of \( e(k) \) as follows:

\[
J(\hat{A}(z)) = \sum_{k_1=-\infty}^{\infty} \sum_{k_2=-\infty}^{\infty} \ldots \sum_{k_{M-1}=-\infty}^{\infty} |C_{M,e}(k)|^2.
\]

The associated LP polyspectrum can be computed by (3) with \( A(f) \) replaced by the HOS based LPE filter.
\( \hat{A}(\theta) \) associated with minimum \( J(\hat{A}(\theta)) \).

It is well-known that the conventional LPE filter based on the minimization of \( E[\sigma^2(k)] \) is minimum-phase [4]. We show, in Appendix A, the following result.

(R1) The HOS based LPE filter \( \hat{A}(\theta) \) is minimum-phase.

Remark that for \( M=2 \), \( J(\hat{A}(\theta)) \) reduces to

\[
J(\hat{A}(\theta)) = \sum_{k=\infty}^{\infty} \left| r_{ee}(k) \right|^2
\]

where \( r_{ee}(k) \) is the autocorrelation function of \( e(k) \).

It is sufficient that \( J(\hat{A}(\theta)) \) given by (6) is minimum if \( \left| r_{ee}(k) \right| \) is minimum for all \( k \). A set of sufficient conditions for this to occur is that \( r_{ee}(0) = E[e^2(k)] = 0 \) and \( r_{ee}(k) = 0 \) for \( k \neq 0 \) in the meantime. In other words, minimizing \( J(\hat{A}(\theta)) \) given by (6) is equivalent to minimizing \( E[e^2(k)] \) because the latter leads to the well-known fact that \( e(k) \) is white when \( x(k) \) is an AR process of order \( p \). Therefore, we have shown the following result.

(R2) The HOS based LPE filter for \( M=2 \) is identical with the conventional LPE filter.

4. Maximum Flatness of Polyspectra

Let \( \xi_e \) denote the flatness of \( M \)-th order polyspectrum of \( e(k) \) defined as

\[
\xi_e = \frac{\exp\left\{ \int_0^1 \cdots \int_0^1 \ln \left| S_{M,e}(f) \right|^L df \right\}}{\int_0^1 \cdots \int_0^1 \left| S_{M,e}(f) \right|^L df}
\]

where \( L \) is a positive integer and \( df \) denotes \( df_1 \cdot df_2 \cdots df_{M-1} \). The maximum polyspectral flatness (MPF) polyspectrum is obtained by (3) with \( \hat{A}(\theta) \) replaced by the optimum \( \hat{A}(\theta) \) associated with maximum \( \xi_e \). Note that \( \xi_e \) is the geometric mean of \( \left| S_{M,e}(f) \right|^L \) divided by its arithmetic mean and therefore can be shown to satisfy \( 0 \leq \xi_e \leq 1 \). Note that \( \xi_e = 1 \) if \( \left| S_{M,e}(f) \right| \) equals a constant for all \( f \). On the other hand, \( \xi_e = 0 \) if \( \left| S_{M,e}(f) \right| \) is very peaky. Remark that for \( L=1 \) and \( M=2 \), \( \xi_e \) is exactly the same as the well-known Gray and Markel's spectral flatness measure [3,4].

The following results associated with the MPF polyspectrum can be proven.

(R3) The optimum \( \hat{A}(\theta) \) associated with maximum \( \xi_e \) and the HOS based LPE filter are identical as long as \( x(k) \) is a stationary non-Gaussian linear process.

(R4) The optimum \( \hat{A}(\theta) \) associated with maximum \( \xi_e \) is equal to \( A(\theta) \). Hence, the LP polyspectral estimator is equivalent to the MPF polyspectral estimator, and both of them offer the same polyspectrum as \( S_{M,x}(\theta) \).

The proof of (R3) and the proof of (R4) are given in Appendices B and C, respectively. By (R3) and (R4), we conclude that the HOS based LPE filter is an "Mth-order whitening" filter which suggests that \( \gamma_M \) can be estimated as the sample cumulant \( \hat{C}_{M,e}(k=0) \) when data are finite.

5. Maximum Higher-Order Entropy

We define the \( M \)-th order entropy, denoted \( \Gamma(\hat{S}_{M,x}(\theta)) \), of the polyspectrum \( \hat{S}_{M,x}(\theta) \) of non-Gaussian linear process \( x(k) \), as

\[
\Gamma(\hat{S}_{M,x}(\theta)) = -\frac{1}{2} \int_0^1 \cdots \int_0^1 \ln \left| \hat{S}_{M,x}(\theta) \right|^2 df.
\]

The maximum \( M \)-th order entropy (MME) polyspectrum of \( x(k) \) is the one which maximizes \( \Gamma(S_{M,x}(\theta)) \) subject to the constraint

\[
\hat{C}_{M,x}(k) = C_{M,x}(k) \text{ for all } k \in \mathbb{R}(M,p)
\]

where \( \mathbb{R}(M,p) \) denotes the finite domain of support associated with the \( M \)-th order cumulant function of any non-Gaussian moving average (MA) process of order \( p \), and \( C_{M,x}(k) \) for \( k \in \mathbb{R}(M,p) \) are the known cumulant samples associated with \( \hat{S}_{M,x}(\theta) \). In other words, the MME polyspectral estimator attempts to extrapolate the \( M \)-th order cumulant function for all \( k \in \mathbb{R}(M,p) \) from the known \( M \)-th order cumulants \( C_{M,x}(k) \) for all \( k \in \mathbb{R}(M,p) \) such that \( \Gamma(S_{M,x}(\theta)) \) is maximum. Remark that for \( M=2 \), \( \Gamma(S_{M,x}(\theta)) \) reduces to

\[
\Gamma(S_{\hat{S}_{M,x}}(\theta)) = \int_0^1 \ln \hat{S}_{xx}(\theta) df
\]

which is proportional to the entropy rate associated with the well-known Burg's maximum entropy spectral estimator, where \( S_{xx}(\theta) \) denotes power spectrum estimate of \( x(k) \).

We show the following two results associated with the MME polyspectrum in Appendix D.

(R5) The MME polyspectrum is equivalent to the \( M \)-th order polyspectrum of a stationary non-Gaussian AR process of order \( p \).

(R6) \( S_{M,x}(\theta) \) given by (3) is an MME polyspectrum.

Furthermore, we have the following result by (R4) and (R6).

(R7) Both the LP polyspectrum and the MME polyspectrum are an MME polyspectrum.

Remark that if the optimum \( \hat{A}(\theta) \) (minimum-phase) associated with \( \Gamma(\hat{S}_{M,x}(\theta)) \) described in (R5) is unique, then \( \hat{A}(\theta)=A(\theta) \) and the MME polyspectrum is the same as \( S_{M,x}(\theta) \) due to (R6) and (R7). Unfortunately, it is still unknown whether the
solution \( \hat{A}(z) = A(z) \) under the constraint given by (9) is unique or not although this is true for the case of \( M=2 \).

6. Discussion and Conclusions

The previous results described in (R1) through (R7) are based on the assumption that \( \sigma(k) \) is a non-Gaussian AR process of known order \( p \). However, when the order \( p \) of the AR process of interest is not known in advance, the proposed HOS based LPE filter \( \hat{A}(z) \) is then no longer identical with the true \( A(z) \) when the order, \( \hat{p} \), of \( \hat{A}(z) \) is less than \( p \). On the other hand, if \( p \geq \hat{p} \), \( a(i) = a(i) \) for \( 1 \leq i \leq p \) and \( a(i) = 0 \) for \( i > p \).

Assume that \( y(k) \) is the output of an allpass filter with input \( x(k) \). It can be easily shown, by Parseval's theorem, that \( J(\hat{A}(z)) \) associated with the signal \( x(k) \) is the same as that associated with the signal \( y(k) \). Due to this fact as well as (R3) we have the following result:

(R8) The LP polyspectral estimator is allpass factor blind, so is the MPF polydispolar estimator. Corresponding to (R8) for \( M=2 \) is the well-known fact that all power spectral estimators are allpass factor blind.

The results we have presented contribute to theoretical foundations on the polydispolar estimation and modeling of non-Gaussian AR processes.

The performance of the proposed HOS based PEP filter with finite data will be reported in the future.

Appendix A. Proof of (R1)

Assume that \( \hat{A}(z) \) is not minimum-phase with a zero \( z_i \) outside the unit circle (i.e. \( |z_i| > 1 \)). We can then express \( \hat{A}(z) \) as

\[
\hat{A}(z) = (1 - z_i z^{-1}) \ A'(z)
\]

(A1)

where \( A'(z) \) is a \((p-1)\)th-order polynomial of \( z^{-1} \). We need the following inequality in the proof.

\[
|1 - z_i \exp(-j2\pi f)| = |z_i| - 1 - (1/z_i^* \exp(-j2\pi f)| - 1
\]

(A2)

Let

\[
\hat{B}(z) = (1 - (1/z_i^*)z^{-1}) A'(z).
\]

(A3)

One can easily see, from (A1), (A2) and (A3), that

\[
|\hat{A}(f)| \geq |\hat{B}(f)|.
\]

(A4)

Next, we infer, from (5) and (A4), that

\[
J(\hat{A}(f)) = \int_0^1 \int_0^1 \left| S_{M,0}(f) \right|^2 df \]

\[
= \int_0^1 \int_0^1 \left| S_{M,0}(f) \right|^2 \cdot |\hat{A}(f_1) \hat{A}(f_2) \ldots \hat{A}(f_{M-1})|^2 df
\]

\[
> \int_0^1 \int_0^1 \left| S_{M,0}(f) \right|^2 \cdot |\hat{B}(f_1) \hat{B}(f_2) \ldots \hat{B}(f_{M-1})|^2 df = J(\hat{B}(f)) \quad \text{by (A4)}
\]

(A5)

which implies that \( J(\hat{A}(f)) \) can never be minimum unless \( \hat{A}(z) \) is minimum-phase.

Appendix B. Proof of (R3)

If \( \hat{A}(f) \) is minimum-phase then

\[
\int_0^1 \int_0^1 |\hat{A}(f)|^2 df = 0.
\]

(B1)

Then we have

\[
\int_0^1 \int_0^1 \left| S_{M,0}(f) \right|^L df = \frac{1}{2} \int_0^1 \int_0^1 \left| S_{M,0}(f) \right|^2 df
\]

\[
= \frac{1}{2} \int_0^1 \int_0^1 \left| S_{M,x}(f) \right|^2 df
\]

\[
+ \frac{1}{2} \int_0^1 \int_0^1 \left( \ln |\hat{A}(f_1)|^2 + \ldots + \ln |\hat{A}(f_{M-1})|^2 \right) df
\]

\[
= \int_0^1 \int_0^1 \left| S_{M,x}(f) \right|^L df \quad \text{by (B1) (B2)}
\]

which is not a function of \( \hat{A}(z) \). Substituting (B2) into (7) provides

\[
\xi_e = \exp\left( \int_0^1 \int_0^1 \ln |S_{M,0}(f)|^L df \right)
\]

(B3)

where

\[
J'(\hat{A}(z)) = \int_0^1 \int_0^1 \left| S_{M,0}(f) \right|^L df.
\]

(B4)

Thus, maximizing \( \xi_e \) is equivalent to minimizing \( J'(\hat{A}(z)) \). Letting \( L = 2 \) in \( J'(\hat{A}(z)) \) yields

\[
J'(\hat{A}(z)) = \int_0^1 \int_0^1 \left| S_{M,0}(f) \right|^2 df
\]

\[
= \sum_{k=1}^{\infty} \ldots \sum_{k=M-1}^{\infty} \left| C_{M,0}(k) \right|^2 = J(\hat{A}(z)).
\]

(B5)

Hence, minimizing \( \xi_e \) is equivalent to minimizing \( J(\hat{A}(z)) \) which implies R(3). The assumption of a minimum-phase \( \hat{A}(z) \) is not restrictive due to (R1) [4].

Appendix C. Proof of (R4)

Let us solve for the optimum \( \hat{A}(z) \) by letting \( \xi_e = 1 \) (maximum value of \( \xi_e \)) which requires

\[
\left| S_{M,0}(f) \right| = \left| S_{M,x}(f) \cdot \hat{A}(f_1) \cdot \hat{A}(f_2) \ldots \hat{A}(f_{M-1}) \right|
\]

\[
\cdot \hat{A}(f_{M-1}) \cdot \left| \hat{A}(f_1) \cdot \hat{A}(f_2) \ldots \hat{A}(f_{M-1}) \right|
\]
\[
\begin{align*}
\gamma_0 \cdot \begin{vmatrix}
\hat{A}(f_1) & \hat{A}(f_2) & \cdots & \hat{A}(f_{M-1}) \\
\hat{A}_1(f_1) & \hat{A}(f_2) & \cdots & \hat{A}_1(f_{M-1}) \\
\hat{A}(f_1 + f_2 + \cdots + f_{M-1}) & \hat{A}_1(f_1 + f_2 + \cdots + f_{M-1}) & \cdots & \hat{A}_1(f_1 + f_2 + \cdots + f_{M-1}) \\
\hat{A}_1(f_1 + f_2 + \cdots + f_{M-1}) & \hat{A}(f_1 + f_2 + \cdots + f_{M-1}) & \cdots & \hat{A}_1(f_1 + f_2 + \cdots + f_{M-1}) \\
\end{vmatrix}
\end{align*}
\]

where \( K \) is a positive constant. Note that \( A(z) \) is minimum–phase by assumption and that \( \hat{A}(z) \) is also minimum–phase due to (R1) and (R3). Therefore, the pth–order rational function \( \hat{A}(z) / A(z) \), which is also minimum–phase, never forms a pth–order allpass filter which is maximum–phase. Thus, we conclude, from (C1), that

\[
\hat{A}(z)/A(z) = K^{'}
\]

where \( K^{'} \) is also a constant. Since \( a(0) = a(0) = 1 \), the constant \( K^{'} \) can only take the value of unity. In other words, \( \hat{A}(z) = A(z) \).

**Appendix D. Proof of (R5) and (R6)**

Proof: Let \( \mathfrak{L} \) be

\[
\mathfrak{L} = \Gamma(\hat{S}_{M,x}(f)) - \sum_{k \in R(M,p)} \lambda(k) \left\{ \hat{C}_{M,x}(k) - C_{M,x}(k) \right\}
\]

where \( \lambda(k) \) is the Lagrange multiplier. Note that \( \lambda(k) \) is real since \( \hat{C}_{M,x}(k) \) is real. Taking partial derivative of \( \mathfrak{L} \) with respect to \( \hat{C}_{M,x}(k) \) and setting the result to zero yield

\[
\frac{1}{2} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \left\{ \frac{e^{-j2\pi f_1 k}}{S_{M,x}(f)} + \frac{e^{-j2\pi f_2 k}}{S_{M,x}(f)} \right\} df dk = 0
\]

for all \( k \in R(M,p) \). (D2)

Let us define \( \Lambda(k) \) to be

\[
\Lambda(k) = \begin{cases} 
\lambda(-k), & k \in R(M,p) \\
0, & \text{otherwise}.
\end{cases}
\]

(D3)

It can be easily shown, from (D2) and (D3), that

\[
\frac{1}{\hat{S}_{M,x}(f)} = \sum_{k \in R(M,p)} \Lambda(k) \exp(-j2\pi f_1 k)
\]

(D4)

where \( \Lambda(k) \) also satisfies all the inherent symmetry properties of the Mth–order cumulant function since \( \hat{S}_{M,x}(f) \) is a polynomials of a non–Gaussian linear process by assumption. Moreover, one can see, from (D4), that \( 1/\hat{S}_{M,x}(f) \) can be viewed as the Mth–order polynomials of a non–Gaussian MA process of order \( p \) and therefore can be factored as the product

\[
\frac{1}{\gamma_M} \hat{A}(f_1) \cdots \hat{A}(f_{M-1})
\]

(D5)

where \( \hat{A}(f) = A(z = \exp(2j\pi f)) \) and \( \hat{A}(z) \) is a pth–order polynomial of \( z^{-1} \), or

\[
\frac{1}{\hat{S}_{M,x}(f)} = \frac{1}{\gamma_M} \hat{A}(f_1) \cdots \hat{A}(f_{M-1})
\]

(D6)

Substituting (D6) back into (8) gives rise to

\[
\Gamma(\hat{S}_{M,x}(f)) = \int \left| \hat{A}(f_1) \right|^2 \cdots \int \left| \hat{A}(f_{M-1}) \right|^2 df
\]

(D7)

Next, let us show that \( \hat{A}(z) \) associated with the maximum of \( \Gamma(\hat{S}_{M,x}(f)) \) given by (D7) is minimum–phase.

Assume that \( \hat{A}(z) \) is expressed as (A1) where \( |a_0| > 1 \). We can easily infer, from (A4) and (D7), that the value of \( \Gamma(\hat{S}_{M,x}(f)) \) is larger if \( \hat{A}(z) \) is replaced by \( B(z) \) given by (A3). Therefore, the optimum \( \hat{A}(z) \) has to be minimum–phase, and the MME polynamium \( \hat{S}_{M,x}(f) \) given by (D6) can be thought of as the Mth–order polynamium of a stationary non–Gaussian AR process of order \( p \).

Thus, we have shown (R5).

By (R5), solving for the MME polynamium is equivalent to solving for the minimum–phase \( \hat{A}(z) \) from \( \hat{C}_{M,x}(k) = C_{M,x}(k) \) for all \( k \in R(M,p) \). Obviously, \( \hat{A}(z) = A(z) \) is a solution. Hence, the associated AR polynamium \( S_{M,x}(f) \) given by (3) is an MME polynamium. Thus we have proven (R6).

**References**


AN ADAPTIVE ARMA LATTICE FILTER FOR SPECTRAL ESTIMATION WITH FREQUENCY WEIGHTING

Miki HASEYAMA, Nobuo NAGAI, and Nobuhiro MIKI
Research Institute for Electronic Science, Hokkaido University, Sapporo 060, JAPAN

This paper proposes a method of spectral estimation with frequency weighting (SEFW) using an adaptive ARMA lattice filter. By this SEFW lattice filter algorithm, we can obtain accurate spectral estimation in the frequency band focused. The SEFW lattice filter algorithm is constructed with two algorithms:

Algorithm(1): Realization algorithm of an ARMA lattice filter with frequency-weighting.
Algorithm(2): An algorithm which computes ARMA parameters from the coefficients of the lattice filter realized by Algorithm(1).

In this paper, we present the above two algorithms.

1. INTRODUCTION

Lattice filters for digital signal processing became popular in the mid-1970's [2] [5]. In this paper, we propose a method of spectral estimation with frequency weighting (SEFW) using an adaptive ARMA lattice filter. It is well-known that spectral estimation is significant for digital signal processing. Using this SEFW lattice filter algorithm, we can obtain accurate spectral estimation in the frequency band focused. In the SEFW, the band in which the spectrum is estimated more accurately is set by 'frequency-weighting function' which can be designed with an ARMA model arbitrarily [6] [4]. Actually, in spectral estimation of speech, since important information is not equally presented throughout all region of the half of the sampling frequency, the spectrum has often been estimated with frequency weighting, such as the mel spectral weighting method. It has not been derived such an SEFW lattice-filter algorithm by using other ARMA lattice filters.

The SEFW lattice filter algorithm proposed in this paper is constructed with the following two algorithms:

Algorithm(1): Realization algorithm of an ARMA lattice filter with frequency-weighting.
Algorithm(2): Algorithm which computes ARMA parameters from the coefficients of the lattice filter realized by Algorithm(1).

In this paper, in Sec.2 we present structure of an ARMA lattice filter which is used in the SEFW lattice algorithm. In Sec.3, we propose Algorithm(1) and Algorithm(2).

2. STRUCTURE OF AN ARMA LATTICE FILTER

The ARMA lattice filter used in the SEFW lattice algorithm is constructed with the following prediction equations:

\[ \hat{e}^f_{a}(i|k) \triangleq x(i) - h^T_{a+1}(i|k)R_{a+1}^{-1}(k-1) \]
\[ - H^T_{a+1}(k-1|k-1) \Delta \varpi(k) \]  

\[ \hat{e}^r_{a}(i|k) \triangleq y(i) - h^T_{a+1}(i|k)R_{a+1}^{-1}(k-1) \]
\[ - H^T_{a+1}(k-1|k-1) \Delta \varpi(k) \]  

where,

\[ \varpi(k-l) \triangleq [\varpi(k-l) \ldots \varpi(k-l+w+1)]^T \]
\[ y(k-m) \triangleq [y(k-m) \ldots y(k-m+w+1)]^T \]
\[ h_{a+1}(j) \triangleq [y(i) \ldots y(i-t+1) \varpi(j) \ldots \varpi(k-s+1)]^T \]
\[ H^T_{a+1}(k|k) \triangleq [y(k) \ldots y(k-t+1) \varpi(k) \ldots \varpi(k-s+1)] \]
\[ \Lambda \triangleq \text{diag} \left[ \prod_{j=1}^{m-1} \lambda(j) \prod_{j=1}^{s-1} \lambda(j) \right] \]
Further, by using the prediction errors defined in Eqs. (1) ~ (8), we can derive four types of order-update recursive formulas, AR0, MA0, ARMA-1 and ARMA-2 types. The AR0 order-update recursive formulas increase the AR order of the prediction errors by one when their MA orders are zero. And the MA0 order-update recursive formulas increase the MA order by one when the AR orders are zero. Using the ARMA-1 and ARMA-2 order-update recursive formulas, we can increase both the AR order and the MA order by one when both AR and MA orders are zero. Based on the order-update recursive formulas, the ARMA lattice filter elementary sections are realized as in Fig. 1 (a) ~ (d). And the structure of 'UNIT' used in Fig. 1 (e). where

\[ V_{x,i}(k) = \sum_{i=k}^{k+w} \lambda^{k-i} p_{x,i}(i) p_{x,i}(i-1) \]

\[ V_{x,i}(k) = \sum_{k-w}^{k} \lambda^{k-i} \{ p_{x,i}(i) \}^2 \]

\[ \rho_{x,i}(k) = \frac{V_{x,i}(k)}{V_{x,i}(k)} \]

In the above equations, \( w \) is a sliding window length [3] and \( \lambda \) is a forgetting factor [1], therefore, our algorithm can perform the SBEW with the exponentially weighted sliding window [5].

In order to realize the ARMA four-line lattice filter as adaptive form, we need to time-update the filter coefficients shown in Fig. 1. Since each elementary sections are constructed with some units shown in Fig. 1 (e), the filter coefficients can be time-updated by time-updating the coefficients of the unit.

For example, let us consider the time-update recursive formula for the coefficients of the first unit (UNIT 1) of the AR0 type elementary section. The coefficients of the unit are calculated with the correlations \( V_{x,i}(k) \), \( V_{x,i}(k) \), and \( V_{x,i}(k) \). The values \( V_{x,i}(k) \) and \( V_{x,i}(k) \) can be obtained from the lattice coefficients and correlations of the previous sections. The time-update recursive formulas of \( V_{x,i}(k) \) is shown heredue:

\[ V_{x,i}(k) = \lambda \{ V_{x,i}(k) - 1 \} + A_{x} p_{x,i}(k) + A_{x} p_{x,i}(k) \}

\[ + A_{x} p_{x,i}(k) - A_{x} p_{x,i}(k) \}

where

\[ A_{x} = (\lambda + \alpha_{x}(k-1))^{-1} \]

\[ A_{x} = \lambda^{-1} \{ 1 - \lambda^{w} \alpha_{x}(k) \} \]

\[ A_{x} = \alpha_{x}(k-w) \cdot A_{x} + \alpha_{x}(k) \]

(11)

Furthermore, in our algorithm, order-update recursive formulas of \( \alpha_{x}(k-i) \) defined in Eq.(12) are needed. These formulas are described as follows:

1) \( \alpha_{x}(k-i) \) which is used for time-updating co-

**Figure 1:** ARMA lattice filter elementary sections

(a) AR0 section (b) MA0 section (c) ARMA-1 section (d) ARMA-2 section (e) The structure of UNIT
coefficients of AR 0th section’s \( \text{UNIT} \), ARMA-1 section’s \( \text{UNIT} \), and ARMA-2 section’s \( \text{UNIT} \)

\[
\alpha_{s_{i+1}}(k - i | k - j) = \alpha_{s_{i}}(k - i | k - j)
\]

\[
+ \frac{\rho_{s_{i}}^{2}(k - i | k - 1) \alpha_{s_{i}}^{2}(k - j | k - 1)}{V_{s_{i}}^{2}(k - 1)}
\]

\( \alpha_{s} \) is used for time-updating coefficients of MA0 section’s \( \text{UNIT} \), ARMA-1 section’s \( \text{UNIT} \), and ARMA-2 section’s \( \text{UNIT} \).

\[
\alpha_{s_{i+1}}'(k - i | k - j) = \alpha_{s_{i}}'(k - i | k - j)
\]

\[
+ \frac{\rho_{s_{i}}^{2}(k - i + 1 | k) \alpha_{s_{i}}^{2}(k - j + 1 | k)}{V_{s_{i}}^{2}(k)}
\]

\( \alpha_{s} \) is used for time-updating coefficients of MA0 section’s \( \text{UNIT} \), ARMA-2 section’s \( \text{UNIT} \), and ARMA-2 section’s \( \text{UNIT} \).

\[
\alpha_{s_{i+1}}'(k - i | k - j) = \alpha_{s_{i}}'(k - i | k - j)
\]

\[
+ \frac{\rho_{s_{i}}^{2}(k - i + 1 | k) \alpha_{s_{i}}^{2}(k - j + 1 | k)}{V_{s_{i}}^{2}(k)}
\]

where

\( \alpha_{s_{i}}(k - i | k - j) = 0 \).

The Algorithm (1) shown in Sec. 3 has two parts; one is formed based on order-update recursive formulas and the other is formed based on time-update recursive formulas of the elementary-section coefficients.

3. Algorithm(1) and Algorithm(2)

The proposed lattice-filter structure is similar to that shown in Ref. [6]. Comparing Ref. [6] with Fig. 1, we can see that there is no significant differences between the number of their lattice-filter coefficients. However, the proposed algorithm is different from Ref. [6] not only on the ability of the SEFW, but also on the following points.

(1) In the case of AR order ≥ MA order, an input estimation procedure can be embedded in Algorithm(1) without increasing calculation costs. (The input estimation problem often occurs in the case of ARMA spectral estimation.)

(2) Using Algorithm(2), we can obtain the ARMA parameters. (The spectrum can be easily obtained from the ARMA parameters.)

3.1. Frequency-Weighting Method

The block diagram shown in Fig. 2 (a) expresses an error for the least-square (LS) estimation problem without frequency-weighting, where, in Fig. 2, \( u(k) \) is a white Gaussian noise. Minimizing the value

\[
\sum_{k=0}^{\infty} \lambda^{k} \rho_{s_{i}}^{2}(k)
\]

We can obtain the optimal estimated parameters of \( \hat{A}(q^{-1}) \) and \( \hat{B}(q^{-1}) \); the parameters can be estimated by the lattice filter denoted in Sec. 2.

Further, using a transfer function \( W(q^{-1}) \), where

\[
1 + \frac{1}{2\pi} \int_{|z|=1} W(z^{-1}) \frac{dz}{z} = 1,
\]

(19)

we try to modify Fig. 2 (a) frequency-weighting LS estimation problem. Comparing Fig. 2 (a) and (b), we can find frequency-weighting model identification is achieved by the lattice filter derived in Sec. 2.

Figure 2: Diagrams for LS and frequency-weighting LS

3.2. Algorithm(1)

Algorithm(1) achieves the realization of ARMA lattice filter with the frequency weighting and the input estimation. In the case of AR order ≥ MA order, the procedures of Algorithm(1) are shown in Table I. Algorithm(1) of case of AR order < MA order can be easily obtained.

Table I: Algorithm(1)

1. begin algorithm
2. initialize \( V_s, \beta_s, \beta_t, \rho_s, \rho_t, \) and \( \alpha \).
3. n=AR order; m=MA order; \( w = \) the rectangular window length; \( \lambda = \) the forgetting factor
4. \( \beta = \) 1
5. while \( y(k) \) exists
6. read \( y(k) \) \{\( y(k) \) is the observed signal\}
7. \( y'(k) = W(q^{-1}) y(k) \) \{If frequency-weighting is not needed, \( W(q^{-1}) = 1 \} \}
8. \( \hat{\beta}_{0, s}(k | k - 1) = y'(k) \)
9. \( s = m \)
10. for \( j = 1 \) to \( s \) step by 1
11. compute \( \hat{\beta}_{j, s}(k | k - 1) \)
12. end of for

\{by order-update recursive formula\}
13. for \( i = 0 \) to \( m - 1 \) step by 1
14. compute \( \hat{y}_a^{i+1,j,i+1}(k|i-1) \)
15. end of for
\( i \)
16. \( \hat{a}(k) = \frac{M-1}{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} \hat{y}_a^{i+1,j,i+1}(k|i-1)} \)
17. \( \hat{y}'(k) = W(q^{-1})\hat{a}(k) \)
18. \( \{ \text{the estimated input and output signals are fed into the lattice filter.} \}
19. \}
20. \( \hat{e}_a^{i,i}(k|i-1) = \hat{y}'(k) - \hat{y}_a^{i,i}(k|i-1) \)
21. \( \hat{e}_a^{i,i}(k|i-1) = \hat{y}_a^{i,i}(k|i-1) \)
22. \( \{ \text{the order-updating for the other prediction errors and } \alpha \} \}
23. \}
24. for \( j = 1 \) to \( s \) step by 1
25. update \( \alpha_4(k) \) by Eq.(13)
26. end of for
27. time-updating all of the lattice parameters by Eq.(11)
28. \( \tau_\chi = \lambda + \gamma^2(k) \)
29. \( \tau_\eta = \lambda + \gamma^2(k) \)
30. end of while
31. end of algorithm

3.3. Algorithm(2)

Algorithm(2) computes the estimated parameters of ARMA transfer function from the parameters of the ARMA lattice filter realized by Algorithm(1). The procedures of Algorithm(2) are shown in Table II. In Table II, the procedures on the case of AR order \( \geq MA \) order are shown. Similarly, the procedure of the case of AR order \(< MA \) order can be obtained.

<table>
<thead>
<tr>
<th>Table II: Algorithm(2)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. begin algorithm</td>
</tr>
<tr>
<td>2. define ( A = (3,2) ) or ((4,2)) matrix</td>
</tr>
<tr>
<td>define ( B = (3,3) ) or ((4,4)) matrix</td>
</tr>
</tbody>
</table>
| \( \{ \) each element of the matrices \( A \) and \( B \) is a polynomial of \( z^{-1} \); \( A = [a_k(z^{-1})] \) \( \} \)
| 3. \( k \)-the time index; \( B := 0; A := I \)
| 4. \( m \)-AR order; \( m := MA \) order; \( m := n \)
| 5. \( A := \left[ \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -z^{-1} \\ 0 & 0 & 0 & 1 \end{array} \right] \)
| 6. for \( j := 0 \) to \( t - 1 \) step by 1 |
| 7. \( B := \left[ \begin{array}{cccc} 1 & -\rho^{(t)}_{(k)} & 0 & 0 \\ -\rho^{(t)}_{(k)} & z^{-1} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] \)
| 8. \( A := B + A \)
| 9. \( B := \left[ \begin{array}{cccc} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right] \)
| 10. \( A := B + A \)
| 11. end of for

4. CONCLUSIONS

We present the SEFW lattice filter algorithm which is constructed with two algorithms: Algorithm(1) and Algorithm(2). Using this SEFW lattice filter algorithm, we can obtain the accurate spectral estimation in the frequency band focused. Therefore, this method is suitable for the realization of a reduced order model.

REFERENCES

Application of SVD to AR Spectral Estimation

Poul M. Rands Jensen* Kjeld Hermansen

Department of Communication Technology, Institute of Electronic Systems, Aalborg University, DK-9220 Aalborg Ø, Denmark

Abstract

During recent years much interest has been given to the application of Singular Value Decomposition in association with extended-order and overdetermined evaluation in the finite parametric spectral estimation domain. Such approaches have been shown to perform superior to other methods for the harmonic retrieval problem.

We apply a similar SVD approach to wide-banded AR processes embedded in white noise which has not been issued previously. This situation typically arises from feature extraction of speech signals as the pre-processing part of a speech recognition.

It is found that the extraneous poles of the lower rank solution influence the spectral estimate. A novel approach, the Direction Weighted Total Least Squares solution, which enforces the extraneous poles to be located at the origin while maintaining the good properties of the aforementioned approach, is therefore introduced.

Computer simulation experiments performed so far indicate this approach to be superior to existing overdetermined and extended- or parsimonious-order methods on noisy data.

1 Introduction

Since the introduction of parametric modeling techniques in spectral estimation like sinusoidal and rational models, these techniques have become widely used due to their higher resolution and spectral fidelity compared to the periodogram.

It has been shown by Cadzow [2] that upon using more than the minimal number of equations, a reduction in data-induced model parameter hyper sensitivity is obtained and an improvement in estimation performance is obtained. This is especially true for noisy environments. Further increased performance can often be obtained by using the Total Least Squares formula, cf. [3, 11]. Moreover, improvements can be obtained using an extended-order model and apply SVD to reduce the rank of the obtained matrix, cf. [2, 9]. In the spectral estimation domain most interest has been given to the harmonic retrieval problem, cf. [1, 2, 5, 10], in which the detection of the number of sinusoids in presence of white noise is issued.

We consider the application of SVD to spectral estimation of wide-banded AR processes embedded in white noise, like voiced speech in noisy environments. The aim of this approach is to remove noise influence in the LPC coefficients to achieve improved spectral estimates. This continuous frequency application presents a number of new problems compared to the discontinuous frequency harmonic retrieval problem associated with the influence of the spectral estimate by the extraneous poles, caused by the extended-order.

We therefore propose a new approach called the Direction Weighted Total Least Squares solution, which utilizes overdetermination, extended-order, TLS and rank reduction via SVD, but by weighting the direction of the solution, enforces the extraneous poles to be located at the origin while maintaining the good properties of the aforementioned approaches.

2 AR Spectral Estimation

For estimating voiced speech and alike signals, the most often applied parametric model is the autoregressive model:

\[
y(k) = b_0 \cdot u(k) - \sum_{i=1}^{p} a_i \cdot y(k-i)
\]  

(2.1)
where \( s(k) = y(k) + \nu(k) \) is the observable signal, e.g., speech, \( u(k) \) is an unknown white noise input and \( \nu(k) \) is white noise perturbation. The estimate, \( \hat{s}(k) \), of \( s(k) \) is then given as:

\[
\hat{s}(k) = -\sum_{i=1}^{p} a_i \cdot s(k-i), \tag{2.2}
\]

where \( \{a_i\}_i^p \) are the parameters to be determined. Expanding (2.2) we can write the estimation problem as a LS problem, \( \mathbf{A} \hat{\mathbf{a}} = \mathbf{b} + \mathbf{r} \), using the deterministic autocovariance approach [4]:

\[
\begin{bmatrix}
    s(k+N-1) & \cdots & s(k) \\
    s(k+N) & \cdots & s(k+1) \\
    \vdots & & \vdots \\
    s(k+M+N-1) & \cdots & s(k+M) \\
\end{bmatrix}
\begin{bmatrix}
    a_1 \\
    a_2 \\
    \vdots \\
    a_N \\
\end{bmatrix}
= 
\begin{bmatrix}
    e(k+N) \\
    e(k+N+1) \\
    \vdots \\
    e(k+M+N) \\
\end{bmatrix}
\tag{2.3}
\]

where \( \{e(k)\} \) is an unobservable random variable of zero-mean, introduced into the model to account for its inaccuracy and which obviously is to be minimized. Alternatively we could have used the autocorrelation method or even have used the autocorrelation matrix, \( \mathbf{R} \), since the things issued in below do apply in these cases too.

(2.3) is readily solved as \( \mathbf{a} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \mathbf{b} \). To improve performance, however, we may use an extended-order model, i.e., \( N > p \), and apply SVD to obtain a Truncated SVD solution of rank \( p \), \( \mathbf{e}_p \). This is given as:

\[
\mathbf{e}_p = \sum_{i=1}^{p} \frac{\mathbf{u}_i^T}{\sigma_i} \mathbf{b}, \tag{2.4}
\]

where \( \{\mathbf{u}_i\}_i^M \), \( \{v_i\}_i^N \), \( \{\sigma_i\}_i^N \) are the left and right singular vectors and singular values of \( \mathbf{A} \), respectively. SVD allows for removal of noise influence in \( a_p \) since (2.4) expresses a partitioning of the domain of \( \mathbf{A} \) into \( p \) dimensional signal- and a \( (N-p) \) dimensional noise space, and hence, an improved spectral estimate can be gained.

When \( \{\nu(k)\} \) is non-zero, \( \mathbf{A} \) as well as \( \mathbf{b} \) are perturbed which makes a TLS solution more feasible [5, 11]. Forming \( \mathbf{B} = [\mathbf{b} \ \mathbf{A}] \) and computing its SVD,

\[
\mathbf{B} = \mathbf{U} \cdot \Sigma \cdot \begin{bmatrix}
    \mathbf{V}_{11} & \mathbf{V}_{12} \\
    \mathbf{V}_{21} & \mathbf{V}_{22}
\end{bmatrix}, \tag{2.5}
\]

the Truncated TLS solution, \( \mathbf{a}_{TLS,p} \), is given as:

\[
\mathbf{a}_{TLS,p} = \mathbf{V}_{22}^T \frac{\mathbf{V}_{12}}{\|\mathbf{V}_{12}\|^2} \tag{2.6}
\]

with \( \mathbf{V}_{11} \in \mathbb{R}^{1 \times p}, \mathbf{V}_{12} \in \mathbb{R}^{1 \times N+1-p}, \mathbf{V}_{21} \in \mathbb{R}^{N \times p} \) and \( \mathbf{V}_{22} \in \mathbb{R}^{N \times N+1-p} \).

3 The Direction Weighted Total Least Squares solution

Revealing a proper value for \( p \) is non-trivial since the problem is severely ill-conditioned, cf. [1, 2, 7]. This problem has been issued in e.g. [1, 8] and is not within the scope of this paper. Hence, we consider the computation of the parameters, \( \mathbf{a} \).

This may readily be carried out using (2.4) for a TSVD solution or (2.6) for a TLS solution. The extraneous poles that obviously are contained in these solutions will ideally be uniformly distributed inside the unit circle [6]. However, for a perturbed system the extraneous poles do fall closer to, on or even outside the unit circle and are not likely to be uniformly distributed.

Unless the signals are very narrow-banded like sinusoids, the extraneous poles will influence the spectrum. Hence, for speech signals considered here and other such AR processes, this will lead to wrong spectral estimates. We therefore propose a novel solution, the DW TLS solution, characterized by only computing a \( p \)-elements vector for a rank \( p \) solution.

Reconsider the TLS problem resulting from \( \|\mathbf{b}\|_2 \approx 0 \), and assume \( \text{rank}(\mathbf{[b \ A]}) = p \) where \( p < N \). This implies that we only need \( p \) columns of \( \mathbf{A} \), and hence a \( \{a_i\}_i^p \) to form a linear combination equal to \( \mathbf{b} \).

The DW TLS solution enforces the extraneous poles to be located at the origin. That is, only \( \{a_i\}_i^p \) are allowed to differ from zero, while \( \{a_i\}_i^{N-p} \) are enforced to equal zero.

First, consider the matrix \( \mathbf{B} \in \mathbb{R}^{M \times (N+1)} \) formed as above, i.e., \( \mathbf{B} = [\mathbf{b} \ \mathbf{A}] \). Next, assume \( \mathbf{B} \) has a SVD, \( \mathbf{B} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^T \), as in (2.5). We then know that the (minimal norm) TLS solution, \( \mathbf{a}_{TLS,p} \), is given by (2.6) which, of course, is unique. However, any \( \mathbf{z} \) given as a linear combination of \( \{v_i\}_i^{N+1} \) and with its first element equal to 1 is a valid solution.

Using the a priori knowledge that we only need \( \{a_i\}_i^p \), we search for a solution of the form \( \mathbf{z} = [1 \ a_1 \cdots a_p \ 0 \cdots 0]^T \):

\[
\begin{bmatrix}
    v_{1,p+1} & \cdots & v_{1,N+1} \\
    v_{2,p+1} & \cdots & v_{2,N+1} \\
    \vdots & & \vdots \\
    v_{p,N+1} & \cdots & v_{p,N+1} \\
\end{bmatrix}
\begin{bmatrix}
    \xi_{p+1} \\
    \xi_{p+2} \\
    \vdots \\
    \xi_{N+1} \\
\end{bmatrix}
= \begin{bmatrix}
    a_1 \\
    a_2 \\
    \vdots \\
    a_p \\
\end{bmatrix}
\]

\[
\begin{bmatrix}
    u_{1,p+1} & \cdots & u_{1,N+1} \\
    u_{2,p+1} & \cdots & u_{2,N+1} \\
    \vdots & & \vdots \\
    u_{N,N+1} & \cdots & u_{N,N+1} \\
\end{bmatrix}
\begin{bmatrix}
    \xi_{p+1} \\
    \xi_{p+2} \\
    \vdots \\
    \xi_{N+1} \\
\end{bmatrix}
= \begin{bmatrix}
    1 \\
    a_1 \\
    a_2 \\
    \vdots \\
    a_p \\
\end{bmatrix}
\]
This leads to the requirement:

\[
\begin{bmatrix}
\xi_{p+1} \\
\xi_{p+2} \\
\quad \\n\xi_{N+1}
\end{bmatrix}
= \begin{bmatrix}
1 \\
0 \\
\quad \\
0
\end{bmatrix}
\quad (3.1)
\]

The DW TLS solution, \( a_{DW, P} \), with elements \( a_i \), is finally given as

\[
a_i = \sum_{j=p+1}^{N+1} \xi_j \cdot v_{i,j}, \quad i \in [1, p]
\quad (3.2)
\]

Also refer to [7].

4 Results

We have performed a number of computer simulations on known AR processes comparing extended- and parsimonious-order standard solutions with extended-order LS and TLS solutions using SVD and the DW TLS solution.

E.g. we have made a comparison between DW TLS and parsimonious-order standard solution, in terms of QRD for a known 2nd order process embedded in white noise. The process is characterized by centerfrequency, \( \omega = 2.20 \cdot 10^{-2} \), and filter \( Q = 5.63 \). We have used a model order of 12 and \( p = 2 \) for DW TLS in all cases while the parsimonious-order for the QRD solution has been slightly increased for decreased SNR. For each SNR 1000 Monte Carlo trials each of 92 samples have been carried out.

In Table 1 are summarized the results of this comparison showing the superior performance of DW TLS,

where the evaluation quantities are given by (4.1).

\[
\omega_{e} = \left\{ \omega \bigg| \max \left( \hat{H}(\omega) \right) \right\}
\quad Q = \frac{\omega_e}{\omega_{bs}}
\quad (4.1)
\]

\[
\Delta H(\omega) = E \left[ \left| H(\omega) - \hat{H}(\omega) \right| \right]
\quad \Delta H(\omega)_{\text{max}} = \max_{\omega} \left| H(\omega) - \hat{H}(\omega) \right|
\]

5 Conclusion

In this contribution we have considered the application of the SVD to finite parametric AR spectral estimation. It has been found that the extraneous poles of the TLS solutions spoil the spectral estimate. The presented DW TLS solution ensures these extraneous poles to be located at the origin while maintaining the good properties of the SVD approaches.

Computer simulation experiments carried out indicate that this approach in the case considered is superior performance of the novel approach compared to existing parsimonious-order methods on noisy data provided knowledge of the parsimonious-order. More work is needed, especially on real data such as voiced speech, which is an issue of our current research [8].

References


A NEW APPROACH TO ESTIMATION OF THE FREQUENCY OF A NOISY CISOID *

A. Ferrari, G. Alengrin and J. Ménez

LASSY Université de Nice-Sophia Antipolis, Département de l'FIS
41 Bd Napoléon III, 06041 CEDEX FRANCE
tel.: +33-93217935, e-mail: ferrari@mimosa.unice.fr

We present in this paper a method to estimate the frequency of a cisoid from a small number of noisy samples based on the Constrained Total Least Squares. The estimated frequency is the global minimizer of a function consisting in the sum of the spectral density of the signal weighted by a bank of windows. The principle of this estimation scheme is studied. This leads to a new frame of estimators based on the presented scheme and using window banks associated to different criteria. The windows associated to the maximum likelihood estimator are given.

1. INTRODUCTION

The estimation of the frequency of a cisoid from a small number of noisy samples is a problem that arises in many fields (i.e. Doppler estimation in RADAR signals). Many algorithms relying on the well known property that the maximum likelihood estimator (MLE) is given by the global maximizer of the periodogram are present in the literature. In this paper, we address the problem using the Constrained Total Least Squares of Abatzoglou and Mendel, [1].

2. CTLs estimation

The Total Least Squares (TLS) is used to solve an overdetermined system of linear equations where both right and left hand sides are noisy. The CTLs assumes that the noise on each column of these matrices is a linear transformation of a unique white random vector. This formulation can be mathematically stated as a minimisation under constraints. In [1], the authors have shown that the above problem is equivalent to the unconstrained minimisation of a highly nonlinear function \( \Lambda_{\text{CTLS}} \). We refer to [2] for a complete summary on this topic.

Let assume now that the observed signal vector \( \mathbf{x} \) is constituted by \( N \) samples of a complex noisy sine wave with frequency \( \omega_0 \) corrupted by a white noise sequence. The CTLs can then be applied to the \( N - 1 \times 2 \) forward linear prediction matrix associated to \( \mathbf{x} \). We have demonstrated in [4] that in this particular case, the function \( \Lambda_{\text{CTLS}} \) can be rewritten as:

\[
\Lambda_{\text{CTLS}}(\omega) = s_\omega^H \cdot X^H \cdot P_{\text{CTLS}}^{-1} \cdot X \cdot s_\omega
\]  

(1)

where the \( N - 1 \times 1 \) vector \( s_\omega \) is:

\[
s_\omega = (1 \exp j\omega \ldots \exp j(N - 1)\omega)^H
\]  

(2)

the \( N - 1 \times N \) rectangular matrix \( X \) is:

* Work partly supported by CESDA/DCN Toulon.
\[
X = \begin{pmatrix}
  x_1 & -x_2 & 0 & \ldots & 0 \\
  0 & x_2 & -x_3 & \ddots & \vdots \\
  \vdots & \ddots & \ddots & \ddots & 0 \\
  0 & \ldots & 0 & x_{N-1} & -x_N
\end{pmatrix}
\] (3)

and the \( N-1 \times N-1 \) matrix \( P_{\text{CTS}} \) is:
\[
P_{\text{CTS}} = \begin{pmatrix}
  3 & -1 & 0 & \ldots & 0 \\
  -1 & 2 & \ddots & \ddots & \vdots \\
  0 & \ddots & \ddots & \ddots & 0 \\
  \vdots & \ddots & \ddots & 2 & -1 \\
  0 & \ldots & 0 & -1 & 3
\end{pmatrix}
\] (4)

\((H\) denotes the conjugate transpose operator.\)

This expression can be simplified using the eigendecomposition of the matrix \( P_{\text{CTS}} \). If we denote \((\sigma_i^{\text{CTS}}, v_i^{\text{CTS}}), i = 1, \ldots, N-1\) the eigenvalue-eigenvector pairs of \( P_{\text{CTS}} \) and if we define the normalized eigenvectors \( v_i^{\text{CTS}} \) as:
\[
v_i^{\text{CTS}} = v_i^{\text{CTS}} / \sqrt{\sigma_i^{\text{CTS}}}
\] (5)

the previous expression of \( \Lambda_{\text{CTS}}(\omega) \) can be rewritten:
\[
\Lambda_{\text{CTS}}(\omega) = \sum_{i=1}^{N-1} |(v_i^{\text{CTS}})^H X s_0|^2
\] (6)

If we now define the \( N-1 \) weighting vectors of order \( N \) \( w_i^{\text{CTS}} ; i = 1, \ldots, N-1 \) as:
\[
w_i^{\text{CTS}} = \begin{pmatrix}
  v_i^{\text{CTS}} \\
  0
\end{pmatrix}
\]
we finally obtain the expression:
\[
\Lambda_{\text{CTS}}(\omega) = \sum_{i=1}^{N-1} |F(w_i^{\text{CTS}} \times x)|^2
\] (8)

\((F(z))\) denotes the discrete Fourier transform of \( z \) at the frequency \( \omega \) and \( w \times z \) represents the signal \( w \) windowed by \( z \).\)

According to this expression, we can make the following remarks:

- The above expression of \( \Lambda_{\text{CTS}}(\omega) \) has a highly parallel implementation. It consists in \( N-1 \) parallel windowed Fourier transform of the signal \( x \).
- In the preceding expression of \( \Lambda_{\text{CTS}}(\omega) \), the criterion of estimation is relegated to the choice of the \( N-1 \) time windows \( w_i^{\text{CTS}}, i = 1, \ldots, N-1 \). We can easily imagine to generalize this scheme of estimation using windows based on other criteria. This point will be investigated in the next section.
- It has been demonstrated in [4] that in the noiseless case, the function \( \Lambda_{\text{CTS}}(\omega) \) does not present as much oscillations as the periodogram. This property is an important advantage of the method in fact, due to the well known oscillations produced by the Dirichlet kernel, [5], the computation of the global maximum of the periodogram with a classical iterative Newton algorithm is strongly dependent of the initial point, [3].

3. Principle of the estimation scheme

We will derive in this section the principle of the estimation scheme and show how it can be generalized to other criteria.

Let first consider the \( i^{\text{th}} \) term of the sum (8). This term can be written:
\[
F(w_i^{\text{CTS}} \times x) = F(w_i^{\text{CTS}}) \otimes F(x)
\] (9)
where \( \otimes \) denotes the convolution operator. The Fourier transform of the window \( w_i^{\text{CTS}} \) is given by:
\[
F(w_i^{\text{CTS}}) = (1 - \exp j\omega) F(v_i^{\text{CTS}})
\] (10)

We see that \( F(v_i^{\text{CTS}}) \) is the key for the interpretation of the estimation scheme. According to the appendix, the \( N-1 \) components of the eigenvector \( v_i^{\text{CTS}} \) are the samples of a sinewave at
the frequency \(i\pi/(N-1)\). The Fourier transform of this vector is then the sum of two Dirichlet kernels \(K_D(\omega)\) centered at frequencies \(i\pi/(N-1)\) and \(-i\pi/(N-1)\):

\[
\mathcal{F}(V_i^{\text{CTLS}}) = K_D(\omega - \frac{i\pi}{N-1}) + K_D(\omega + \frac{i\pi}{N-1})
\]

More insight in the method can then be obtained considering the ideal case were \(\mathcal{F}(x)\) is a pure spectral line. With no loss of generality, we can assume that this line is located at frequency zero:

\[
\mathcal{F}(x) = A_0 \delta_0
\]

The term corresponding to equation (9) is then given by:

\[
\mathcal{F}(w_i^{\text{CTLS}} \times x) = A_0 (1 - \exp j\omega) \\
\cdot \left(K_D(\omega - \frac{i\pi}{N-1}) + K_D(\omega + \frac{i\pi}{N-1})\right)
\]

and finally, the function \(A_{\text{CTLS}}(\omega)\) will represent the sum of the contribution of all these kernels centered at frequencies regularly separated by \(\pi/(N-1)\) with the exception of the central-frequency zero. This result is illustrated by figure (1).

Moreover, we can make the following remarks:

- The residual contributions of all the kernels at the frequency zero will be cancelled by the effect of \(1 - \exp j\omega\). We can note that if the windows are constructed according to relation (7), this property will always hold.
- The eigenvectors being orthogonal, their Fourier transform will also be orthogonal according to Parseval theorem. The consequence of this property is a “regularization” of the effect of each window at the non zero frequencies.
- The effect of the normalization of each eigenvector by the square root of its associated eigenvalue is fundamental to obtain a function \(A_{\text{CTLS}}(\omega)\) as flat as possible for the non zero frequencies.
- The squared modulus in (8) is fundamental for the same reason. However, it introduces a non linearity in the method that leads to the impossibility of the application of this scheme in the case where more than one cosoid is present in the signal.
- The overall process can be viewed as the construction of a filter rejecting the frequency present in the signal.
- If the effect of truncation on the observation of the cosoid is taken into account in the expression of its Fourier transform \(\mathcal{F}(x)\), and if the noise spectrum is assumed flat, equations leading to a similar interpretation of the method are obtained.
- In the real case, the main difference is the non flatness of the noise spectrum. The windows \(w_i^{\text{CTLS}}\) are designed to minimize the influence of this effect according to a particular criterion.

It is clear that windows associated to other criteria can be used. From the above remarks, the principal conditions they must verify is to be orthogonal and to have a cumulative spectrum as constant as possible. We can mention for example a bank of exponential kernels with \(\Delta \omega/\omega\) constant.

4. Maximum likelihood estimation windows

Using [1] and developing the same calculus as in [4], it can be shown that the MLE estimator obeys the same scheme as the one above replacing the matrix \(P_{\text{CTLS}}\) by the matrix \(P_{\text{MLE}}\) defined by:
Appendix A

The eigendecomposition of the $N - 1 \times N - 1$ matrices $P_{C T L S}$ and $P_{M L E}$ can be obtained analytically. The eigenvalues of $P_{C T L S}$ are given by:

$$\sigma_i^{C T L S} = 4 \sin^2 \frac{i \pi}{2(N - 1)}$$  (A.1)

and the eigenvalues of $P_{M L E}$ by:

$$\sigma_i^{M L E} = 4 \sin^2 \frac{i \pi}{2N}$$  (A.2)

The coordinate $k$ of the eigenvectors of $P_{C T L S}$ are given by:

$$v_i^{C T L S}(k) = \sqrt{\frac{c_i}{N - 1}} \sin \frac{(2k - 1)i \pi}{N - 1}$$  (A.3)

where $c_i = 2$ for $i < N - 1$ and $c_{N-1} = 1$, and the coordinate $k$ of the eigenvectors of $P_{M L E}$ by:

$$v_i^{M L E}(k) = \sqrt{\frac{2}{N}} \sin \frac{k i \pi}{N}$$  (A.4)

References


SEPARATION OF A MIXTURE OF INDEPENDENT SOURCES THROUGH
A MAXIMUM LIKELIHOOD APPROACH

Dinh Tuan PHAM *, Philippe GARAT * and Christian JUTTEN **

*Laboratory LMC/IMAG, C.N.R.S., BP 53X, 38041 Grenoble Cedex, France
** Laboratory TIRF, INPG 44 Avenue Félix-Viallet 38031 Grenoble Cedex, France.

The problem of separation of a mixture of independent (non Gaussian) sources without a priori knowledge is
approached from the maximum likelihood principle, leading to the use of nonlinear functions of observations.
These functions are unknown as they depend on the densities of sources and the problem is how to choose them.
To this end we investigate the performance of the method in term of the chosen functions. We also derive a simple
way to determine the optimal choice from the data when the above functions are restricted to given vector space.
Finally an algorithm for constructing the estimator of the mixture weights is proposed and tested by simulation.

1. INTRODUCTION

This paper deals with the problem of separation of a mixture of independent sources without any precise
assumption on the structure of them. This problem has important applications in signal processing (e.g. speech
analysis, radar and sonar processing, ...). Many methods have been proposed in the literature ([1], [2], [5]...), most
of them are based on the use of higher order moments. Our approach is based on the maximum likelihood principle,
but we do not adopt a model of sources. Instead of higher order moments, we are led to consider other statistics
which are nonlinear functions of the observations (such functions arise naturally as we work with non Gaussian
signals). Since no model of source is assumed, such separating function have to be chosen arbitrarily.
However, from the performance study of the estimator, we can derive a method for choosing them optimally in a
given vector space of functions. Further, we propose an adaptive method in order to follow changes of the mixture weights.
A similar algorithm, derived in an ad-hoc manner, has appeared in [4]. The maximum likelihood principle is also
used in [3], but the source density is approximated by a Gram-Charlier expansion based on cumulants up to fourth
order, leading to a different solution.

2. FORMULATION OF THE PROBLEM AND THE PROPOSED PROCEDURE

The problem can be formalized as follows: one observes \( T \) random vectors \( X(t), t = 1, \ldots, T \), of
dimension \( K' \), each component \( X_{j}(t) \) of \( X(t) \) being a linear combination of \( K \) independent sources, i.e. \( X(t) = A S(t) \) where \( A \) is some unknown matrix and \( S(t), t = 1, \ldots, T \), are random \( K \)-vectors with independent compo-
nents \( S_{1}(t), \ldots, S_{K}(t) \). The aim is to reconstruct the
"sources" \( S(t) \) (up to a permutation and a scale factor) from the observations \( X(1), \ldots, X(T) \). To simplify, we
suppose here that the number of sources \( K \) is the same as
the number of sensors \( K' \). Suppose further that the \( S(t) \)
at different times \( t \) are independent identically distributed
(this independence assumption can be relaxed). Then, in the
case where the probability density distribution of the \( i \)-th
source \( S_{i}(t) \) is known up to a scale factor, i.e. of the
form \( f_{i}(s_{i}/\sigma_{i})/\sigma_{i} \), with known \( f_{i} \), the log likelihood
of the data can be written as:

\[
L_{T} = T \sum_{i=1}^{K} \hat{E} \ln [f_{i}(e_{i}^T A^{-1}X/\sigma_{i})/\sigma_{i}] - \ln \det A \] 

where \( \hat{E} \) denotes the time averaging operator: \( \hat{E} g(X) = \{ g(X(1)) + \ldots + g(X(T)) \}/T, \)
denotes the \( i \)-th column of the identity matrix (of order \( K \)) and \( A \) denotes the transpose.
The above function is to be maximized with respect to \( A \) and \( \sigma_{1}, \ldots, \sigma_{K} \). To this end, one may equate its
partial derivatives to zero. But is more convenient to work with its differential instead. From \( dA A^{-1} =
- A^{-1} dA A^{-1}, d \ln \det A = tr(A^{-1} dA), (tr denoting the
trace), the differential of \( T^{-1} L_{T} \) can be seen to be

\[
\sum_{i=1}^{K} \hat{E} [\psi_{i}(e_{i}^T A^{-1}X/\sigma_{i}) e_{i}^T A^{-1} dA A^{-1} X/\sigma_{i}] - tr(A^{-1} dA)
\]

where \( \psi_{i} = -(\ln f_{i})' \), \( \cdot \) denoting derivative. Note that the
first two terms of the above differential can be written as
\[ \sum_{i=1}^{K} \sum_{i=1}^{K} \mathbb{E}[\psi_i(\alpha_i^T A_{-1} X) / \alpha_i] \mathbb{E}[\psi_i(\alpha_i^T A_{-1} X) / \alpha_i] \partial_{ij} = \sum_{i=1}^{K} \partial_{ii}, \]

where \( \partial_{ij} \) denotes the general element of \( A_{-1} dA \). At the maximum likelihood estimators \( \bar{A}, \bar{\sigma}_i \) of \( A, \sigma_i \), the above differential must be zero for all infinitesimal increments \( d\alpha_i \) and \( d\sigma_i \), yielding the estimating equations:

\[ \bar{\sigma}_i = \mathbb{E}[\psi_i(\alpha_i^T A_{-1} X) / \alpha_i] \mathbb{E}[\psi_i(\alpha_i^T A_{-1} X) / \alpha_i], \quad i = 1, \ldots, K, \]

\[ \mathbb{E}[\psi_i(\alpha_i^T A_{-1} X) / \alpha_i] \mathbb{E}[\psi_i(\alpha_i^T A_{-1} X) / \alpha_i] = 0, \quad i \neq j, \quad i, j = 1, \ldots, K. \]

In practice, one does not know \( f_i \), but it can be seen that the above second system of equations still define a consistent estimator of \( A \), when one uses arbitrary functions \( \psi_i \), provided that the sources are independent and of zero mean. Note that the true sources satisfy \( \mathbb{E}[\psi_i(S_i / \sigma_i) S_j] \to 0 \) as \( T \to \infty \) by ergodicity. Of course the choice of \( \psi_i \) can greatly affect the performance of the method and this question will be studied later. By absorbing the scale factor \( \sigma_i \) in \( \psi_i \), our separation procedure simply consists in maximizing (with respect to \( \bar{A} \))

\[ (2.1) \quad \mathbb{E}[\psi_i(\alpha_i^T A_{-1} X) / \alpha_i] \mathbb{E}[\psi_i(\alpha_i^T A_{-1} X) / \alpha_i] = 0, \quad i \neq j, \quad i, j = 1, \ldots, K. \]

3. ASYMPTOTIC PROPERTIES OF THE ESTIMATOR

To derive the asymptotic properties of the estimator, we shall assume that \( \bar{A} \) is close to the true mixture matrix, now denoted by \( A \), so that one can make a Taylor expansion of (2.1) around \( A \) (it is assumed here that \( A \) is scaled in the same way as \( \bar{A} \) and its column may be permuted). Thus, one can write \( \bar{A} = A(I + \delta) \) where \( \delta \) is a small matrix. The right hand side of (2.1) can then be written approximatively (for large \( T \)) as

\[ \mathbb{E}[\psi_i(S_i) S_j] - \sum_{k=1}^{K} \left( \mathbb{E}[\psi_i(S_k) S_k] \delta_{ik} + \mathbb{E}[\psi_i(S_k) S_k] \delta_{jk} \right) \]

where \( S_i = e_i^T A_{-1} X \) and \( \delta_{ij} \) denotes the general element of \( \delta \). We now approximate \( \mathbb{E}[\psi_i(S_i) S_i] \delta_{ik} \) and \( \mathbb{E}[\psi_i(S_i) S_k] \delta_{ij} \) by the expectation of \( \psi_i(S_i) S_i S_k \) and \( \psi_i(S_i) S_k S_k \), which vanish unless \( k = j \) or \( k = i \), respectively. Thus, for \( i \neq j \) and the block diagonal matrix \( \mathbf{H}^{-1} \mathbb{E}(\Psi) \), \( \mathbf{H} \) being the block diagonal matrix with blocks

\[ \mathbf{H}_{00} = \begin{bmatrix} \mathbb{E}[\psi_i(S_i) \mathbb{E}(S_j)] & \mathbb{E}[\psi_i(S_i) S_j] \\ \mathbb{E}[\psi_i(S_i) S_j] & \mathbb{E}[\psi_i(S_j) S_j] \end{bmatrix} \]

indexed by the pair \((i, j), i \neq j, 1, \ldots, K\). By the central limit theorem, \( \mathbb{E}(\Psi) \) is asymptotically Gaussian with mean zero and covariance matrix \( G/T \), where \( G = \mathbb{E}(\Psi \Psi^T) \), if the \( S_i(t) \) are temporally independent (this assumption is made to simplify the computation). Thus, \( \Delta \) is asymptotically Gaussian with covariance matrix \( \mathbf{H}^{-1} G \mathbf{H}^{-1} / T \). Note that \( G \) is also block diagonal provided that \( \mathbb{E}[\psi_i(S_i)] = 0 \) (which is satisfied e.g. if \( S_i \) has a symmetric distribution and \( \psi_i \) is an odd function). In this case, the random vectors \( (\delta_{ii} \delta_{ij}) \), for different pairs \((i, j)\), are asymptotically independent with covariance matrix \( \mathbf{H}_{10} \mathbf{G}_{00} \mathbf{H}_{01} / T \) where

\[ \mathbf{G}_{00} = \begin{bmatrix} \mathbb{E}[\psi_i(S_i) \mathbb{E}(S_j)] & \mathbb{E}[\psi_i(S_i) S_j] \\ \mathbb{E}[\psi_i(S_i) S_j] & \mathbb{E}[\psi_i(S_j) S_j] \end{bmatrix} \]

We now show that the above covariance matrix is minimum when \( \psi_i = \psi_i^* = - \log f_i^*, f_i^* \) being the true density of the \( i \)-th source. We first note that by integration by parts, one has, for any differentiable function \( g \):

\[ (3.1) \quad \mathbb{E}[g(S_i) \psi_i^*(S_i)] = \mathbb{E}[g(S_i)], \]

in particular \( \mathbb{E}[\psi_i^*(S_i) S_j] = 1 \) and \( \mathbb{E}[\psi_i^*(S_i)] = 0 \). Thus \( \mathbf{H} = \mathbb{E}(\Psi \Psi^T) \), where \( \Psi^* \) is defined similarly to \( \Psi \) with \( \psi_i^* \) in place of \( \psi_i \). Now, the partitioned matrix

\[ \mathbb{E} \left[ \begin{bmatrix} \Psi^* \Psi^T \end{bmatrix} \right] = \begin{bmatrix} G & \mathbf{H} \\ \mathbf{H}^T & J \end{bmatrix} \]

where \( J = \mathbb{E}(\Psi^* \Psi^*^T) \), is non negative, yielding \( G - H \bar{J} H^T \geq 0 \). Thus \( H^{-1} G (H^{-1})^T \bar{J} / T \geq J^{-1} / T \), and it can be easily seen that equality is attained when \( \psi_i \) is proportional to \( \psi_i^* \). Since \( \mathbb{E}[\psi_i^*(S_i)] = 0 \) by (3.1), the matrix \( J \) is also block diagonal as \( H \), meaning that there is a decoupling of the separation into pairs of channels.

4. CONSTRUCTION OF THE ESTIMATOR

An iterative algorithm for finding the solution of (2.1), based on the Newton method is now given. By expanding the equation (2.1) around an initial estimate \( \bar{A} \) of \( A \), using the same computation as in section 3, the next step estimate, denoted by \( \bar{A}(I + \delta) \), can be constructed as the solution of
\[
\sum_{k=1}^{K} \{ E\{\psi_j(S_0)S_0S_0^T\delta_k \} + E\{\psi_j(S_0)S_0\delta_k \} = E\{\psi_j(S_0)S_0 \}
\]

where \( \tilde{S}_i = \epsilon_i^{\frac{1}{T}}x^{\frac{1}{T}}x \) and \( \tilde{\delta}_j \) denote the general element of \( \tilde{S} \). As before, \( E\{\psi_j(S_0)S_0S_0^T\} = E\{\psi_j(S_0)S_0S_0\} = 0 \) if \( k \neq j \) and \( E\{\psi_j(S_0)S_0\} = E\{\psi_j(S_0)S_0 \} = 0 \) if \( k = i \). Thus the above equation may be replaced by

\[
E\{\psi_i(S_0)S_0^T\delta_i \} + E\{\psi_j(S_0)S_0\delta_i \} = E\{\psi_i(S_0)S_0 \}
\]

\( i \neq j = 1, \ldots, K \). These equations allow to calculate \( \tilde{S} \) and one can iterate the procedure until convergence. Since the sources are defined up to a scale factor, it is recommended to rescale each column of the estimate at each step, according to a chosen rule.

The above algorithm can be easily modified for use in an adaptive context, in which the estimate is updated each time a new observation is available. Let \( \tilde{A}(t-1) \) be an estimate of \( A \) at time \( t-1 \). The application of the above iteration leads to a new estimate at time \( t \): \( \tilde{A}(t) = \tilde{A}(t-1) \{ I + \tilde{S}(t) \} \) where \( \tilde{S}(t) \) is the matrix with general element \( \tilde{S}_{ij}(t) \), solution of

\[
\sum_{j=1}^{K} \psi_j(S(x))\tilde{S}_{ij}(t) + \sum_{j=1}^{K} \psi_j(S(x))\tilde{S}_{ij}(t) \tilde{S}_{ij}(t) = \sum_{j=1}^{K} \psi_j(S(x))\tilde{S}_{ij}(t), \quad i \neq j = 1, \ldots, K.
\]

Note that \( \tilde{S}_i \) is now computed from \( \tilde{A}(t-1) \) and hence \( \sum_{j=1}^{K} \psi_j(S(x))\tilde{S}_{ij}(t) \) is close to 0. Thus we simply replace above the right hand side by \( \psi_j(S(x))\tilde{S}_{ij}(t) \), yielding the algorithm \( (i = 1, \ldots, K; j \neq i = 1, \ldots, K) \)

\[
\tilde{S}_{ij}(t) = \epsilon_i^{\frac{1}{T}}A(t-1) - X(t),
\]

\[
\Omega_i(t) = \lambda\Omega_i(t-1) + \psi_i(S(x))\tilde{S}_{ij}(t),
\]

\[
\Omega_j(t) = \lambda\Omega_j(t-1) + \psi_j(S(x))\tilde{S}_{ij}(t),
\]

\[
\Omega_{ij}(t) = \Omega_i(t) + \Omega_j(t) + \psi_i(S(x))\tilde{S}_{ij}(t),
\]

where \( \lambda = 0, 1 \) is a forgetting factor introduced to track the possible evolution of \( A \).

5. CHOICE OF SEPARATING FUNCTIONS

Since the optimal choice \( \psi_j^* \) depends on the unknown source distribution and is not easily estimated, we shall restrict the choice of \( \psi_j \) to a vector space of functions spanned by a basis \( \phi_1, \ldots, \phi_N \), say, with \( \phi_1(x) = x \). Thus \( \psi_j = \sum_{n=1}^{N} c_n \phi_n \) for some constants \( c_n \). The goal is to minimize the covariance matrix of the estimator, or equivalently to maximize \( H_{ij}G_{ij}H_{ij} \)

where \( H_{ij} \) and \( G_{ij} \) are defined in section 3. Note that \( H_{ij} = E\{\psi_j(S_0)\phi_m(S_0)\} \), \( G_{ij} = E\{\psi_j(S_0)\phi_m^2(S_0)\} \) where \( \psi_j(S_0) = [\psi_j(S_0)S_0^T \psi_j(S_0)] \) and \( \psi_j^* \) is defined similarly, with \( \psi_j^* \) in place of \( \psi_j \). We show here that the optimal choice of \( \psi_j \) is given by the projection of \( \psi_j \) onto the vector space spanned by the \( \phi_n \), i.e., by taking \( c_n \) equal to \( c_{in} \), the solution of

\[
5.1 \quad \sum_{n=1}^{N} c_{in} E\{\phi_n(S_0)\phi_m(S_0)\} = E\{\psi_i^*(S_0)\phi_m(S_0)\},
\]

\( m = 1, \ldots, N \).

We note that \( \psi_j \) may be written as \( C_{ij}(\psi_j) \) where

\[
C_{ij} = \begin{bmatrix} c_{i1} & c_{i2} & \cdots & c_{iN} \\ 0 & c_{j1} & \cdots & 0 \\ 0 & 0 & \cdots & c_{jN} \end{bmatrix}
\]

and \( \Phi_j \) is the vector with components \( \phi_1(S_0), \ldots, \phi_N(S_0) \). We denote \( \Phi_j \) by the projection of \( \psi_j \) on the \( \Phi_j \). Then

\[
E\{\psi_j(S_0)\Phi_j^T(S_0)\} = E\{\psi_j^*(S_0)\Phi_j^T(S_0)\}
\]

and from the non negativity of this matrix, \( H_{ij}G_{ij}H_{ij} \leq E\{\psi_j^*(S_0)\Phi_j^T(S_0)\} \) with equality if \( \psi_j = \psi_j^* \). Thus, it suffices to show that for \( C_{ij} \) of the same form as \( C_{ij} \) with \( c_{in} \) in place of \( c_{in} \), \( C_{ij}E\{\Phi_j\Phi_j^T(S_0)\} = E\{\psi_j^*(S_0)\Phi_j^T(S_0)\} \), since this implies \( \psi_j^* = C_{ij}E\{\Phi_j\Phi_j^T(S_0)\} \). Hence \( C_{ij} \) is the optimal choice of \( \psi_j \) and hence by choosing \( C_{ij} = C_{ij} \), optimality is achieved. Now, the first row of \( C_{ij}E\{\Phi_j\Phi_j^T(S_0)\} \) contains the elements

\[
\sum_{n=1}^{N} c_{in} E\{\phi_n(S_0)\phi_m(S_0)\} \text{ and } \sum_{n=1}^{N} c_{in} E\{\phi_n(S_0)\phi_m(S_0)\phi_n(S_0)\},
\]

\( m = 1, \ldots, N \) (they are repeated if \( m = 1 \)), which by (5.1) are equal to \( E\{\psi_j^*(S_0)\phi_m(S_0)\} \) and \( E\{\psi_j^*(S_0)\phi_m(S_0)\phi_n(S_0)\} \). Hence the first row of \( C_{ij}E\{\Phi_j\Phi_j^T(S_0)\} \) equals that of \( E\{\psi_j^*(S_0)\Phi_j^T(S_0)\} \). The equality between the second rows of these matrices can be proved in the same way.

The important point is that one can easily estimate the \( c_{in} \). Indeed, by (5.1), the right hand sides of (5.1) can be replaced by \( E\{\phi_m^2(S_0)\} \). Then, replacing the expectation \( E \) operator by the time averaging operator \( \tilde{E} \) one gets a system of equations for estimating the coefficients \( c_{in} \).

6. SOME EXPERIMENTAL STUDIES

Figure 1 illustrates the results for one typical simulation experiment testing the iterative (non adaptive) method of section 4. Two sources from the same uniform
distribution are mixed and our algorithm, with \( \psi(s) = s^3 \), is applied on 256 sample points. We normalise the rows of \( A^{-1} \) to have unit Euclidian norm, and represent its off diagonal elements as coordinates of a point in the graph. The two solid curves correspond to the two equations (2.1). Their intersections yield \( \tilde{A}^{-1} \), which unfortunately, is not unique. Actually, ignoring permutation, there is a value of \( \tilde{A}^{-1} \), lying quite close to \( A^{-1} \) (+ in the graph) and another spurious one. The iteration may diverge to the "right" (1st and 2nd iterations) or to the "wrong" solution (3rd iteration), depending on the initial value (it converges well) and fast unless one starts at a point on the circle for which the corresponding matrix is singular). To resolve the above ambiguity, one may compute some index of dependence (e.g. sum of squares of the correlations) between the reconstructed sources and retain the solution for which they are least dependent. Also, by starting at some good initial estimate, one is more likely to attain the "right" solution.

![Figure 1: Examples of iterative computation of the estimator.](image)

To study the effect of the functions \( \psi_i \), we consider the case where the sources have a same distribution from the family of densities \( f_p(s) = K_p \exp(-|s|^{p}/p) \), \( p \) being a shape parameter and \( K_p \) being the normalization constant. We take \( \psi_i \) to be a same function \( \psi \). The asymptotic variance is smallest when \( \psi = \phi_p \) where \( \phi_p(s) = \text{sign}(s)|s|^{p-1} \). The ratios of this variance over the actual asymptotic variances of the estimator are called efficiencies and plotted in figure 2. As choices of \( \psi \), the functions \( \phi_4, \phi_{1.5}, \phi_4 + c \phi_{1.5} \) (with \( c \) chosen by the method of projection of section 5 and optimally by trial) and \( \phi_4 + c_1 \phi_2 + c_2 \phi_{1.5} \) (obtained by the method of projection, which is optimal) are considered. It is seen that the efficiency is quite good for \( \phi_4 \) for \( p > 2 \) (unless \( p \) is very large) but poor for \( p < 2 \). The efficiency for \( \phi_{1.5} \), on the other hand, is good for \( p < 2 \) but poor for \( p > 2 \). By using the linear combination of \( \phi_4 \) and \( \phi_{1.5} \) with optimal coefficient, good efficiency can be achieved overall, whereas the linear combination from the method of projection (not optimal since the function \( s \) is missing) can lead to catastrophic result for \( p \) near 2 (see figure 2). Note that the efficiency curves (except for \( \phi_4 + c_1 \phi_{1.5} \)) seem to be smooth at \( p = 2 \) even though the asymptotic variance diverges to infinity as \( p \to 2 \). This divergence is a consequence of the results of section 3 and means that Gaussian sources cannot be separated.

![Figure 2: Asymptotic efficiencies of the estimator.](image)

7. CONCLUSION

We have shown that the proposed method work well and yield highly efficient estimators, although some care needs to be taken with regard to the ambiguity problem.

REFERENCES


A FRAMEWORK FOR AUTOREGRESSIVE THEORY

P.M.T. Broersen
H.E. Wensink

Faculty of Applied Physics, Delft University of Technology
P.O. Box 5046, 2600 GA Delft, The Netherlands
phones: +31 (0)15-786419 fax: +31 (0)15-786081

Summary: Three levels of approximation can be distinguished in Autoregressive theory: Probability Limits, Asymptotic Theory and Finite Sample Theory. The Finite Sample Theory for autoregression takes the real data as a starting point, rather than relying on strict mathematical derivations. It provides formulae, which describe the behaviour of quantities like residual variance and prediction error in the practical situation of AR model estimation. Not only does the Finite Sample Theory lead to a more accurate order selection criterion, FSC, but it also evaluates the existing criteria, indicating ways of improvement. Some formulae are derived for the variance of reflection coefficients and parameters. They provide the necessary addition for an accurate and precise interpretation of the finite sample estimation results in AR model estimation.

1. INTRODUCTION

Suppose that an autoregressive series is generated by an AR(K) process:

\[ x(n) = \sum_{k=1}^{K} \phi_k x(n-k) + \epsilon(n), \]

where K is the process order. The driving process \( \epsilon(n) \) is gaussian white noise, with zero mean and variance \( \sigma^2 \).

An AR model of order p can be fitted to a realization of \( x(n), \) \( n=1,...,N \). The entries \( \phi(p) \) of the parameter vector \( \phi \) are estimated and the order \( p \) has to be selected. Typical quantities in the estimation process are the residual variance \( S^2(p) \) and the Prediction Error \( PE(p) \). The residual variance, which is defined as the fit of the estimated model to the series \( x(n) \) from which the parameters have been estimated, is as follows:

\[ S^2(p) = \frac{1}{N-p} \sum_{n=p+1}^{N} [x(n) - \sum_{i=1}^{p} \phi(i)x(n-i)]^2. \]

The prediction error is defined as the average fit of the estimated model to an infinitely long realization \( y \) of the same stochastic process:

\[ PE(p) = \frac{1}{N-p} \sum_{n=p+1}^{N} [y(n) - \sum_{i=1}^{p} \phi(i)y(n-i)]^2. \]

This result is equal to the expectation of the one step ahead prediction error. The latter being defined as the error in predicting \( x(n+1) \) from a series, which is known up to \( x(n) \).

In general the search will be for the model with lowest \( PE(p) \). The prediction error is not solely a measure for the forecasting capacities of the model. Small \( PE(p) \) indicates the model will provide the best parametric description of the data. Furthermore it can be shown via \( NATFE(p) \) that such a model has optimal spectral properties [1,2]. The definition in \( (3) \) refers to a theoretical quantity, which can only be computed in simulation experiments. In practice estimation from the data \( x(n) \) is unavoidable.

2. AR THEORY

Lately a theoretical framework for autoregressive estimation has been suggested by the authors [3]. Three levels of approximation to autoregressive theory are distinguished: probability limits, asymptotic theory and finite sample theory.

The following remarks concerning the table can be made:

- The probability limits were used by Mann and Wald [4] to apply results of classical linear regression to AR estimation. As such it constitutes the theoretical background to Autoregression. In the probability limits estimates assume their statistical expectations. The probability density functions of residuals and predictions become one and the same; \( S^2(p) \) and \( PE(p) \) converge to \( \sigma^2 \) for \( N \rightarrow \infty \). This accounts for the fact, that everything is known at this level. So process and model order are equal, all parameters being zero for \( p>1 \).

- At the asymptotic level the number of observations \( N \) is a given, finite, number. As to modeling no benefit can be taken from considerations with plim. To get an accurate description of the data, order selection becomes necessary. At this level an accuracy of order \( O(1/N) \) is needed, because essential differences between competing model orders are of this magnitude order [5,6,7]. This level is a source of confusion in literature, because several heuristic reasonings will fortuitously give some partially correct results [8]. Problems arise from the fact that, although stochastic, the autocovariance function of the data is taken to be exact. The influence of the fact that the autocovariance is actually stochastic is easily evaluated by means of a second order Taylor expansion [8]. More generally all bias terms of \( O(1/N) \) can be computed with this expansion.

For order selection purposes the Final Prediction Error, \( FPE(p)[5] \), has been developed. Selecting the model with lowest \( FPE(p) \), it provides an asymptotically based estimate for the \( PE(p) \) of the estimated model. Later alternatives to \( FPE(p) \) were investigated, using concepts from information theory. Resulting criteria can together be described as a
Table: Three levels of approximation to AR theory

<table>
<thead>
<tr>
<th>probability limits</th>
<th>asymptotic theory</th>
<th>finite sample theory</th>
</tr>
</thead>
<tbody>
<tr>
<td>ML theory</td>
<td>Taylor approximation</td>
<td>real data</td>
</tr>
<tr>
<td>( N \to \infty )</td>
<td>( \mathbb{E}[S^2(y)] = \frac{\sigma^2}{\epsilon} (1 - p/N) ) ( p \approx K )</td>
<td>( \mathbb{E}[S^2(y)] = \epsilon^2 \sum_{i=0}^{p} \left[ 1 + v(\epsilon_i) \right] ) ( p \approx K )</td>
</tr>
<tr>
<td>( p \lim_{N \to \infty} \mathbb{E}[P(E(\phi))] = \frac{\sigma^2}{\epsilon} (1 + p/N) ) ( \approx K )</td>
<td>( \mathbb{E}[P(E(\phi))] = \epsilon^2 \sum_{i=0}^{p} \left[ 1 + v(\epsilon_i) \right] ) ( p \approx K )</td>
<td>( \mathbb{E}[P(E(\phi))] = \epsilon^2 \sum_{i=0}^{p} \left[ 1 + v(\epsilon_i) \right] ) ( p \approx K )</td>
</tr>
<tr>
<td>no order selection</td>
<td>( FPE(\phi) = \frac{\sigma^2}{\epsilon} \left( \frac{N + p}{N - p} \right) )</td>
<td>( FSC(\phi) = \frac{\sigma^2}{\epsilon} \left( \sum_{i=0}^{p} \left[ 1 + v(\epsilon_i) \right] \right) )</td>
</tr>
<tr>
<td></td>
<td>( GIC(\phi, \alpha) = \ln[\sigma^2(y)] + v \ln(\epsilon) )</td>
<td>( Bressen and Wensink (1985, 1990) )</td>
</tr>
<tr>
<td></td>
<td>Mann and Wald (1943)</td>
<td>Akaike (1970, 1974)</td>
</tr>
</tbody>
</table>

Generalized Information Criterion \( GIC(\phi, \alpha) \). By substituting for \( \epsilon \) quantities like 2 [6], \( \ln(N) \) [9,10] and \( 2\ln(N) \) [11], the \( AIC(\phi) \) and its consistent variants can be described.

At finite sample level, the characteristic behaviour of the data forms the starting point. The influence of the estimation method becomes important. It might be called a physical theory as it gives an accurate description of numerous simulation experiments. It is shown, that the variance of the last estimated parameter depends on the model order \( p \) currently under estimation. Moreover, each estimation method has its particular influence on the estimation results. This is the basis for the finite sample variance coefficients \( v(\epsilon_i) \), which are defined as the variance of parameters above the true model order. They are plotted in Figure 1, for an AR(3) process with reflection coefficients \( \epsilon_1 = 0.8, \epsilon_2 = 0.3 \) and \( \epsilon_3 = 0.5 \). For their determination four estimation methods have been used: the Yule-Walker method (YW), the method of Burg (Burg), the least squares method minimizing both forward and backward residuals (LSFB) and the least squares method with one-sided residuals only (LSF). To the experimental results empirical approximations have been made, leading to the following expressions [1]:

\[
\begin{align*}
\nu(\epsilon, \text{YW}) &= (N-1)/N(N+2) \\
\nu(\epsilon, \text{Burg}) &= 1/(N+1-l) \\
\nu(\epsilon, \text{LSFB}) &= 1/(N+1.5-1.5l) \\
\nu(\epsilon, \text{LSF}) &= 1/(N+2-2l)
\end{align*}
\]

For all methods \( \nu(\epsilon, \cdot) = 0 \). The \( \nu(\epsilon, \cdot) \) are motivated rather by some simple theoretical background, than by the most optimal fit to the simulation results. With \( \nu(\epsilon, \text{LSF}) \), the asymptotical fact is reflected, that parameters are estimated for \( N-l \) residuals, leaving \( N-2l \) degrees of freedom. The constant 2 in the denominator is used to obtain \( 1/N \) for first order, which is the best approximation. The same holds for Burg and LSFB. The coefficients for Burg reflect the N-4 contributions to the estimated reflection coefficient \( \hat{\phi}(l) \) in the lattice structure.

Thus the finite sample theory not only shows the deviation of practical results from the asymptotic variance 1/N, but it also explains the difference between the Yule-Walker method of estimation and all other methods; only in the Yule-Walker method \( \nu(\epsilon, \cdot) \) are smaller than 1/N.

For order selection is done making use of the Finite Sample Criterion for order selection, \( FSC(\phi) \). The criteria based on Information theory can be improved by means of finite sample concepts. This gives rise to the Finite sample InformationCriterion \( FIC(\phi, \alpha) \), which is the finite sample improvement for \( GIC(\phi, \alpha) \), using the same \( \epsilon \).

3. ORDER SELECTION

The Framework can now be used to tackle the problem of selecting a model order with low prediction error. Therefore first of all the \( PE(\phi) \) will be evaluated. This has been done in 500,000 Monte Carlo simulation runs with 30 observations each, generated by an AR(3) process. The results are plotted in Figure 2. The \( PE(\phi) \) is shown to have a minimum at the optimal model order. Moreover, the behaviour clearly depends on the model order and the estimation method used for parameter estimation. It can be described for \( \alpha K \), because the Finite Sample Theory provides the expression for its statistical expectation:

\[
\mathbb{E}[PE(\phi)] = \epsilon^2 \sum_{i=0}^{p} \left[ 1 + v(\epsilon_i) \right].
\]
In practical situations the PE(p) is not known. In the quest for the optimal model the PE(p) must be estimated from the same data that have been used to estimate the parameters. As a counterpart of the asymptotical FPE(p), the Finite Sample Criterion FSC(p) has been developed for the finite sample case. Not only is it expected to select models with low PE(p), but also does the criterion value provide an estimate of the actual prediction error of the finally estimated model.

Like Akaike's FPE(p), the FSC(p) is based on the insight that each extra parameter above the best model order will increase PE(p) and decrease S^T(p). The accuracy of FSC(p) as an estimator of the prediction error can be determined in simulation experiments, where an independent realization of the same process can be used to compute PE(p). Figure 3 shows the average observed PE(p), the theoretical expectation E(PE(p)) and estimates provided by the asymptotical FPE(p) and the finite sample FSC(p). The results of Burg estimates are presented, but one-sided residuals LSF and least squares method with forward and backward residuals LSFB give similar results. The figure shows FSC(p) and FPE(p) to be quite alike for low model orders; the v(l_i) are still close to the asymptotical value of 1/N. The values of the two criteria separate for higher orders and FPE(p) will eventually have a wrong high order minimum.

4. VARIANCE OF PARAMETERS

The computation of the theoretical variance of finite sample parameters will now be treated. The covariance matrix C(θ) will be treated for the three levels of approximation to AR theory. For an accurate computation at finite sample level the bias of the parameters has to be taken into account. Therefore the Levinson recursion is used to compute the parameter variance from the finite sample covariance matrix of the reflection coefficients.

In the probability limit the variance of N times the estimated parameters \( C_{\text{lim}}(\theta) \) is given by [4]:

\[
C_{\text{lim}}(\theta) = \sigma^2_\theta \cdot R^{-1}.
\]  

(5)

The matrix R is the theoretical symmetric Toeplitz KxK auto-covariance matrix of the series \( x(n) \), which has expectation \( E(x(n)x(n+i)) \), for \( i = 0, \ldots, K-1 \).

At asymptotic level the variance of the parameters is given with accuracy of \( O(1/N) \):

\[
C_{\text{asy}}(\theta) = \sigma^2_\theta / N \cdot R^{-1}.
\]  

(6)

The asymptotic expression of the variance of estimated reflection coefficients \( k \) is given by [12]:

\[
C_{\text{asy}}(k) = \sigma^2_\theta / N \cdot D \cdot R^{-1} \cdot D^T,
\]  

(7)

where D is the matrix with entries \( d(i,j) = \frac{\partial}{\partial \theta_k} \theta_j ; \), \( \theta_k \) denotes the theoretical reflection coefficient indexed i. This is the usual first-order Taylor approximation for non-linear functions of a stochastic variable.

At finite sample level the covariance matrix is defined as:

\[
C_{\text{fin}}(\theta) = \sigma^2_\theta \cdot V \cdot R^{-1} \cdot D^T \cdot V,
\]  

(8)

where the matrix V is the diagonal matrix with elements \( \lambda(v_i^2) \). For white noise \( C_{\text{fin}}(\theta) \) becomes the diagonal matrix with elements \( v_i^2 \), because \( D = I \) and \( R = \sigma^2_\theta \cdot I \), where I is the diagonal unity matrix. For non-white processes, \( K > 0 \), the matrix \( C_{\text{fin}}(\theta) \) contains a KxK sub-matrix in the upper left corner with the covariance structure of the reflection coefficients of the AR(K) process. Its diagonal elements are 0.5209v(1,1), 0.5460v(2,2) and 0.7500v(3,3) in the example of Figure 1 (see Fig. 1).

The parameter variance is now build up applying essentially the Levinson recursion. The variable \( \gamma_p \) is introduced as \( \gamma_p = k_p + B(\lambda_p) \); the theoretical reflection coefficient corrected by its bias \( B \). The finite sample bias is taken to be equal to the asymptotic bias, which follows from a second order Taylor expansion of \( \lambda \) written as a function of \( \bar{R} \). This is accounted for the bias and the recursion becomes as follows:

...
Simulation experiments have been carried out to evaluate the theoretical description provided by the formula. It turns out that expression (10) together with (11) provides an accurate description. For LSFB and LSF it rather sketches the general behaviour, still being much more accurate than the asymptotic results.

5. CONCLUSIONS

Three different levels of approximation to AR theory are described by the Framework for Autoregressive Theory. The levels differ from one another in theoretical description and practical applicability. The Finite Sample Theory converges to asymptotic theory and further to plim results when $n \to \infty$. Its basis, however, does not result from theoretical derivations, but is founded on real data results. Therefore it may be called a physical theory. Its formula provides descriptions of the results, which actually occur in estimating AR models. Concluding it is stated, that the Finite Sample Theory gives immediate practical applicable results in AR model estimation from finite data sets.

REFERENCES

A NEW IDENTIFICATION ALGORITHM FOR ALLPASS SYSTEMS BY HIGHER-ORDER STATISTICS

Chong-Yung Chi and Jung-Yuan Kung
Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, ROC.

Abstract: In this paper, based on higher-order statistics (HOS), we propose a new allpass system identification algorithm with only non-Gaussian output measurements. Besides, not only the order determination but also the determination of whether the system of interest is an allpass system are also included in the proposed identification algorithm. It is applicable in channel equalization for the case of phase distortion channel [2]. It can be used to remove the remaining phase distortion of nonminimum-phase source wavelet in the predictive deconvolved data. On the other hand, the identification of an ARMA system \( H(z) \) can be converted into the identification of an allpass system [3,4] when minimum-phase (MP) - allpass (AP) decomposition based methods are used, and then it can be applied even when \( H(z) \) includes allpass factors whereas Chi-Kung’s allpass system classification algorithm [5] is not applicable in this case. Some simulation results are provided to support that the proposed algorithm works well.

1. Introduction
Cumulant (higher-order statistics) based identification of nonminimum-phase linear time-invariant (LTI) systems with only non-Gaussian output measurements has drawn extensive attention in various signal processing applications. A general parametric model for the LTI system is known as autoregressive moving average (ARMA) model, denoted \( H(z) \). One can preprocess the output of \( H(z) \) by the inverse filter \( 1/H_{MP}(z) \) where \( H_{MP}(z) \) is the spectrally equivalent minimum-phase system obtained by correlation based spectral estimators such that the identification of \( H(z) \) is equivalent to the identification of the allpass system \( H_{AP}(z) = H(z)/H_{MP}(z) \) [3,4]. When \( H(z) \) does not include any allpass factors, the poles of each possible candidate \( H_{AP}(z) \) are also zeros of \( H_{MP}(z) \). Chi and Kung [5] proposed an allpass system classification algorithm by a single cumulant sample for determining the \( H_{AP}(z) \) for the case that \( H_{MP}(z) \) is given in advance. On the other hand, in some applications, the LTI system \( H(z) \) is known to be an allpass system without any prior information about pole locations such as the phase distortion channel (allpass system) in channel equalization [2] where the channel input is often a sequence of non-Gaussian M-ary symbols. Another interesting instance is that the source wavelet in the well-known predictive deconvolution [6] is assumed to be minimum-phase but it might be nonminimum-phase in practice. Therefore, the deconvolved results can be viewed as the output of a phase distortion channel whose input is a non-Gaussian sparse reflectivity sequence, and removing the remaining phase distortion of nonminimum-phase source wavelet is needed in order to improve the quality of the deconvolved results.

2. The Allpass System Identification Algorithm
Assume that data \( x(k), k=0, 1, ..., N-1 \), are noisy output measurements of the allpass system \( h(k) \) to be identified as follows:

\[
x(k) = u(k) * h(k) + w(k) \tag{1}
\]

under the following assumptions about \( h(k), u(k) \) and \( w(k) \).

(A1) The allpass system \( h(k) \) is a real causal and exponentially stable pth-order allpass system with the transfer function \( H(z) \)

\[
H(z) = \frac{B(z)}{A(z)} \tag{2}
\]

where the minimum-phase \( A(z) \) is given by

\[
A(z) = A_p(z) = 1 + a_1 z^{-1} + \cdots + a_p z^{-p} \tag{3}
\]

and \( B(z) \) (maximum-phase) is given by

\[
B(z) = B_p(z) = A_p(z^{-1})z^{-p}. \tag{4}
\]

(A2) The input \( u(k) \) is real, zero-mean, independent identically distributed (i.i.d.), non-Gaussian with \( n \)-th-order cumulant \( \gamma_n \). Moreover, \( |\gamma_n| < \alpha \) for \( 2 \leq n \leq 2M \).

(A3) The measurement noise \( w(k) \) is real Gaussian which can be white or colored with unknown statistics, and is statistically independent of \( u(k) \).

This work was supported by the National Science Council under Grant NSC81-0404-E-007-001 and the Telecommunication Laboratories under Grant TL-NSC-81-5201.
For ease of later use, let \( k = (k_1, \ldots, k_{M-1}) \) and \( S_{AP}(L) \) denote the set of all Lth-order anticausal stable allpass filters as follows:

\[
\hat{H}_{AP}(z) = \frac{A_L(z)}{B_L(z)} \quad (5)
\]

where \( A_L(z) \) and \( B_L(z) \) have been defined in (3) and (4), respectively. Note that \(|\hat{H}_{AP}(f)| = |\hat{H}_{AP}(z = e^{i2\pi f})| = 1 \) for all \( f \), and that \(|H(0)| = 1 \) for all \( f \) and \( 1/H(z) \in S_{AP}(p) \). The new identification algorithm is based on Theorem 1 and Fact 1 below.

**Theorem 1.** Assume that \( x(k) \) was generated from (1) under the assumptions (A1), (A2) and (A3). Let \( y(k) \) be the output of a \( p \)-th-order allpass filter \( \hat{H}_{AP}(z) \in S_{AP}(p) \) with the input \( x(k) \). Then the absolute \( M \)-th-order \( (M \geq 2) \) cumulant \( |C_{M,y}(k=0)| \) of \( y(k) \) is maximum if and only if \( \hat{H}_{AP}(z) = 1/H(z) \). Furthermore, \( \max\{ |C_{M,y}(k=0)| \} = |\gamma_M| \).

**Fact 1.** Assume that \( x(k) \) was generated from (1) under the assumptions (A1), (A2) and (A3). Let \( y(k) \) be the input of an arbitrary allpass system with amplitude response equal to unity and \( y(k) \) be the corresponding output. Then

\[
\sum_{k_1=\infty}^{\infty} \cdots \sum_{k_{M-1}=\infty}^{\infty} C_{M,y}^2(k) = \sum_{k_1=\infty}^{\infty} \cdots \sum_{k_{M-1}=\infty}^{\infty} C_{M,x}^2(k) = \gamma_M^2 \quad (6)
\]

The new identification algorithm is basically to maximize the following objective function

\[
J(L) = \hat{C}_{M,y}(k=0) \quad (7)
\]

where \( \hat{C}_{M,y}(k=0) \) is the sample cumulant of the output \( y(k) \) of an \( L \)-th-order allpass system \( \hat{H}_{AP}(z) \in S_{AP}(L) \) with input \( x(k) \). Note that \( y(k) \) must be computed from \( x(k) \) recursively backwards since \( \hat{H}_{AP}(z) \) is an anticausal stable IIR filter and that \( J(L) \) is a nonlinear function of the coefficients of \( \hat{H}_{AP}(z) \). We next, illustrate the identification procedure.

**Parameter Estimation:** (The order is known a priori.)

\( \text{(s0)} \) Set \( L = p \).

\( \text{(s1)} \) Search for the \( J_{max}(L) \) (maximum of \( J(L) \)) and associated \( \hat{H}_{AP}(z) \in S_{AP}(L) \) by a numerical optimization algorithm such as Newton–Raphson type algorithm.

Thus the optimum \( \hat{H}(z) = 1/\hat{H}_{AP}(z) \) and the optimum estimate \( \hat{\gamma}_M = \hat{C}_{M,y}(k=0) \).

**Order Determination and Parameter Estimation:**

\( \text{(s2)} \) Set \( L = 0 \) and compute \( J_{max}(0) = \hat{C}_{M,x}(k=0) \).

\( \text{(s3)} \) Set \( L = L + 1 \).

\( \text{(s4)} \) Execute (s1).

\( \text{(s5)} \) If \( [J_{max}(L) - J_{max}(L-1)]/J_{max}(L-1) < \xi \), then compute the ratio

\[
R(L) = J_{max}(L) / D(L) \quad (8)
\]

where

\[
D(L) = \sum_{k_1=\infty}^{\infty} \cdots \sum_{k_{M-1}=\infty}^{\infty} C_{M,y}^2(k), \quad (9)
\]

otherwise go to (s3).

If \( R(L) < 1 \), \( H(z) = 1/\hat{H}_{AP}(z) \) and the optimum estimate \( \hat{\gamma}_M = \hat{C}_{M,y}(k=0) \); if \( R(L) < 1 \), \( \hat{H}(z) \) is not an allpass system. Remark that we calculate \( D(L) \) in (s5) over \( F(q) \) for a preassigned value of \( q \) where

\[
F(q) = \text{Domain of support of the } M \text{-th order cumulant function of non-Gaussian MA(q) process} \quad (10)
\]

The proposed algorithm works well due to the following characteristics:

\( \text{(C1)} \) The proposed algorithm is guaranteed for convergence no matter whether \( H(z) \) is an allpass system or not.

\( \text{(C2)} \) If \( H(z) \) is indeed an allpass system, \( R(L) = 1 \) as well as \( \hat{H}_{AP}(z) = 1/H(z) \in S_{AP}(p) \) (by both Theorem 1 and Fact 1) for all \( L \geq p \) as \( N \) is large. On the other hand, if \( R(L) < 1 \), one can infer that \( H(z) \) is not an allpass system.

\( \text{(C3)} \) With no doubt, various cumulant based least-squares (LS) methods such as [1] can be applied to estimating the AR parameters of \( H(z) \) by treating \( H(z) \) as a general ARMA model, while the MA parameters of \( H(z) \) are automatically determined (see (4)). The proposed algorithm outperforms cumulant based LS estimators simply due to more accurate model (allpass model) used by the former.

\( \text{(C4)} \) The proposed allpass system identification algorithm is a consistent estimator.

\( \text{(C5)} \) The proposed identification algorithm can be used for any \( M \geq 3 \) as long as \( \gamma_M \neq 0 \).
3. Simulation Results

Two simulation examples are to be presented to justify the good performance of the proposed allpass system identification algorithm. We calculated the quantity $D(L)$ over $F(q=15)$ (see (9) and (10)) in our simulation.

Example 1.

The driving input $u(k)$ was exponentially distributed with zero mean and skewness $\gamma_3=2$. Hence the cumulant order $M=3$ was used. A second-order allpass model (taken from [1]) was used for the case of $w(k)$ being colored Gaussian noise which was generated from a highpass second-order FIR filter $B(z) = 1 - 1.2 z^{-1} + 0.32 z^{-2}$ driven by a zero-mean white Gaussian noise sequence. The total data used in each run was $N=1024$.

First of all, let us consider the case that the order of $H(z)$ is known a priori. The simulation results are shown in Table 1, from which one can observe that AR parameter estimates show smaller bias and smaller standard deviation as SNR is larger, and that the values of bias and standard deviation for SNR=1 (a quite low SNR) are also small. These simulation results support the good performance of the proposed allpass system identification algorithm.

Next, let us consider the case that the order of $H(z)$ is unknown. We only show the simulation results for SNR=10 since those for other SNR’s are similar in every respect. Thirty $J_{\text{max}}(L)$’s and associated $R(L)$’s are shown in Fig. 1. In the top part of Fig. 1, different kinds of lines are used to make each single $J_{\text{max}}(L)$ discernible from other $J_{\text{max}}(L)$’s. The bottom part of Fig. 1 depicts thirty $R(L)$’s in the same manner. Note, from Fig. 1, that each $J_{\text{max}}(L)$ converges monotonically with $L$ and that each $J_{\text{max}}(L)$ converges for all $L>p=2$ (true order), and that all $R(L)$’s are larger than 0.85 close to unity. These simulation results also justify (C1) and support (C2).

Example 2.

Assume that $h(k)$ is not an allpass system but a third-order nonminimum-phase source wavelet as follows:

$$H(z) = \frac{1 + 0.1 z^{-1} - 3.275 z^{-2} + 1.41125 z^{-3}}{1 - 1.9 z^{-1} + 1.1525 z^{-2} - 0.1625 z^{-3}}$$

whose zeros are located at $-2.0415, 1.4719$ and $0.4686$. The input $u(k)$ was a sparse spike sequence with random amplitudes generated by use of the well-known Bernoulli–Gaussian (B–G) model [7] which has been used in seismic deconvolution and ultrasonic biomedical imaging. The cumulant order $M=4$ was used since $\gamma_3 \neq 0$ but $\gamma_2 = 0$ for this case.

The synthetic data $x(k)$ for SNR=100 shown in Fig. 2(a) were generated for the case of $w(k)$ being white Gaussian. We preprocessed $x(k)$ by the minimum-phase prediction error filter (PSE) of order equal to 40, which was obtained from $x(k)$ by Burg’s algorithm [8], to get the deconvolved data $e(k)$ which is shown in the top part of Fig. 2(b), from which, one can see that each spike in $u(k)$ (solid line) is associated with a wavelet in $e(k)$ (dotted line) which begins with two opposite peaks and gradually decays due to the remaining phase distortion of source wavelet. Then we processed $e(k)$ with the proposed algorithm. The output $y(k)$ (dotted line) of the optimum $H_{\text{AP}}(z)$ is shown in the bottom part of Fig. 2(b), from which, one can see that $y(k)$ approximates $u(k)$ very well except for a scale factor. In other words, the phase distortion in $e(k)$ has been considerably removed by the proposed algorithm. Hence, the previous simulation results justify that our allpass system identification algorithm can remove the remaining phase distortion of nonminimum-phase source wavelet and improve the quality of the predictive deconvolved results.

![Fig. 1. Simulation results for order determination associated with Example 1](image)

![Table 1](image)

<table>
<thead>
<tr>
<th>SNR</th>
<th>$\hat{\gamma}_1$</th>
<th>$\hat{\gamma}_2$</th>
<th>$\hat{\gamma}_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\infty$</td>
<td>$-0.3021 \pm 0.0147$</td>
<td>$-0.4001 \pm 0.0162$</td>
<td>$1.9892 \pm 0.0638$</td>
</tr>
<tr>
<td>100</td>
<td>$-0.3032 \pm 0.0182$</td>
<td>$-0.3974 \pm 0.0187$</td>
<td>$1.9765 \pm 0.0823$</td>
</tr>
<tr>
<td>10</td>
<td>$-0.3048 \pm 0.0246$</td>
<td>$-0.3952 \pm 0.0393$</td>
<td>$2.0287 \pm 0.1303$</td>
</tr>
<tr>
<td>1</td>
<td>$-0.3276 \pm 0.0715$</td>
<td>$-0.3723 \pm 0.0658$</td>
<td>$2.0474 \pm 0.2918$</td>
</tr>
</tbody>
</table>
4. Conclusions

In this paper, we have presented a new cumulant based allpass system identification algorithm with only non-Gaussian output measurements. Besides, not only the order determination but also the determination of whether the system of interest is an allpass system are also included. The proposed algorithm possesses five nice characteristics described in (C1) through (C5) in Section 2. It is applicable for any LTI phase distortion systems such as phase-distortion communication channels and phase distortion of source wavelet in predictive deconvolved seismic signals. We also showed two simulation examples to support the proposed algorithm. As mentioned in Section 1, the identification of an ARMA(p,q) system \( H(z) \) can be converted into the identification of an allpass system when MP–AP decomposition based methods are used, and our allpass system identification algorithm can be applied no matter whether \( H(z) \) includes allpass factors, whereas Chi–Kung’s [5] algorithm is not applicable when \( H(z) \) includes allpass factors.

References


Fig. 2. Simulation results associated with Example 2. (a) Synthetic noisy data for SNR = 100. (b) The top part includes true input signal \( u(k) \) (solid line) and the predictive deconvolved data \( e(k) \) (dotted line); the bottom includes true input signal \( u(k) \) (solid line) and the output \( y(k) \) (dotted line) of the optimum \( \hat{H}_{AP}(z) \) of order equal to 2.
OPTIMAL BAYESIAN DETECTION OF SPECTRAL CHANGES IN RANDOM SIGNALS

E. LE CARPENTIER - C. DONCARLI - B. VOZEL

Laboratoire d'Automatique de Nantes, E.C.N. / C.N.R.S.
1 rue de la Noé, 44072 NANTES CEDEX, FRANCE

The purpose of this paper is the presentation of a new sequential parametric procedure to detect and estimate simultaneously abrupt spectral changes in random signals. After a recall of the problem position, a classification of practically used methods is then considered. The application of the attractive optimum bayesian technique to the detection and estimation problem, inspired by a previous work of the authors, dealing with supervised classification of finite length random signals, is afterwards developed and the corresponding decision criteria are then derived. Finally, a simulation study is presented to light the improvement of the results to this new approach before concluding.

1 INTRODUCTION

The early detection and diagnosis of abnormal behaviors in the dynamic properties of random signals are realistic problems which are encountered in many fields of application. It is a natural way to improve performances of very numerous and applied processes (mechanical devices, medical monitoring,...), for which we might be interested in taking action after detecting the change instant, with minimum time delay.

Moreover, tracking of time-varying phenomena leads also to improve performances of adaptive filters, which from a theoretical viewpoint can only deal with slow changes. In this paper, we mainly focus our attention on the on-line parametric approach to detect spectral abrupt changes.

After the definition of the studied signals and changes (§2), the state of the art is briefly described (§3). Then, a new sequential procedure, based on a specific distance, is introduced (§4) and a comparison study with classical methods is performed on a very large set of signals (§5). Finally, some concluding comments are given.

2 PROBLEM STATEMENT

The signals are assumed to be described by a linear parametric model (AR or ARMA), which can be written in the following form:

\[ y_k = \Theta^T \phi_k + e_k \]

with \((e_k)\) zero mean i.i.d. gaussian sequence, \(\text{var}(e_k) = \sigma^2_e\)

\[ \Theta^T = (a_1, a_2, \ldots, a_p) \]

\[ \phi_k = (-y_{k-1}, -y_{k-2}, \ldots, -y_{k-p}) \text{ for an AR (p) model} \]

\[ \Theta^T = (a_1, \ldots, a_p, c_1, \ldots, c_q) \]

\[ \phi_k = (-y_{k-1}, \ldots, -y_{k-p}, \phi_{k}, \ldots, \phi_{k-q}) \text{ for ARMA (p,q) model} \]

Then, spectral changes are either change in the variance or in the dynamic parameter vector of the signal model.

The detection of a change can be stated as follows: given the samples \(Y_{1:}\) of a stochastic process \((Y_t)\) with conditional density \(p(y_t|Y_{t-1})\) where \((Y_{t-1})^T = (y_1, \ldots, y_{t-1})\),

- a nominal time invariant parameter vector \(\Theta^0 = (\Theta^T, \sigma^2)^T\) possibly identified, we test on-line if the observations are generated by the nominal parameter vector or by a different set of parameters. We assume that no change occurs in the model order or in the statistical distribution either.

Formally, observing the sample \(y_0, y_1, \ldots, y_t\) up to time \(t\), we have to decide between the two hypotheses:

\( H_0: \Theta = \Theta_0 \)

\( H_1: \) there exists an instant \(r(1 \leq r \leq t)\) such that:

\( \Theta = \Theta_0 \)

\( 1 \leq k \leq r-1 \)

\( \Theta = \Theta_1 \)

\( r \leq k \leq t \)

If \((H_1)\) is decided, then the estimation of the change time \(r\) and possibly \(\Theta_1\) are required.

According to the assumption that \(r\) is a deterministic unknown value, the standard optimality criteria [1] of interest in this paper (on-line detection theory) consists in minimizing the mean detection delay for a fixed false alarm rate, namely the mean time between false alarms.

Let \(t_o\) be the alarm time at which a detection occurs, the delay to be minimized is:

\[ t = \sup_{r \geq 1} \sup_{\Theta_1 \neq \Theta_0} \sup_{\Theta_0} \{ y_k = 1 / Y_{1:r}^1, t_o \geq r \} \]

while the mean time between false alarms satisfies:

\[ T = E_{\Theta_0}(t_o) \geq T \]

But the factors which allow us to estimate the advantages and drawbacks of the different procedures (rate, false alarms rate) are obviously non linear and complicated functions of the random variables. Moreover, the difficulty to evaluate the efficiency of the various methods is strongly related to the difference between \(\Theta_0\) and \(\Theta_1\) and the value of the change time \(r\). As a result, a precise analytical assessment of their efficiency becomes quickly difficult, and a comparison based on extensive simulations is often considered.[2],[3],[4]

3 STATE OF THE ART

Roughly speaking, two classes of sequential procedures for detecting and estimating a change in the signal parametric model can be distinguished and briefly recalled:

3.1 First Class

In the first group, we find algorithms which test the statistical properties of a prediction error computed via a suitable filter: LPE (Linear Prediction Error)[5], EKF (Extended Kalman Filter)[6], RML (Recursive Maximum
Likelihood\cite{7}.

According to the method used to design the filters, the prediction error is studied under both hypotheses $H_0$ and $H_1$. For any of these methods, we can consider the prediction error $e_k$ as the output of a linear system having $e_k$ as input. As a result, the sequence $e_k$ is centered under both the hypotheses but two types of behavior may be expected in case of change: a change in $\theta$ makes the prediction error $e_k$ becoming correlated under $H_1$ and probably more energetic while independent under $H_0$. A change in $\sigma^2$ makes only $\sigma^2_k$ suddenly increasing after the change time.

Thus, each method implements a specific test to detect the expected behavior of the prediction error when a change occurs.

3.1.1 LPE

The linear prediction error is defined by:

$$e_k = y_k + \sum_{i=1}^{p} a_i y_{k-i}$$

where $(a_i)\ 1 \leq i \leq p$ denotes an identified AR model of the signal (before change).

Then, for each type of changes (in $\theta$ or in $\sigma^2$), the test consists in finding the characteristic number of times $L\cdot e_k$ overreaches a threshold $d$ \cite{5}. The threshold can be computed a priori by fixing an upper false alarm rate bound $T$ and approximating the false alarm rate in the following form:

$$P_{FA} \approx \text{Prob}\{|1_{e_k}| > d\} \leq T$$

where, under the gaussian assumption

$$\text{Prob}\{|1_{e_k}| > d\} = \frac{1}{\sqrt{2\pi}\sigma_f} \int_{1_{e_k} > d} e^{-\frac{1_{e_k}^2}{2\sigma_f^2}}$$

If $e_k > d$ at least $L$ consecutives samples, $H_1$ has to be decided, $H_0$ otherwise.

3.1.2 EKF

The prediction error (pseudo-innovation process) is now defined by:

$$e_k = y_k + \sum_{i=1}^{p} a_i y_{k-i} - \sum_{i=1}^{q} c_i e_{k-i}$$

where $(a_i)\ 1 \leq i \leq p$; $(c_i)\ 1 \leq i \leq q$, denote the coefficients of an identified ARMA model of the signal by EKF.

The test consists in computing an ergodic estimation of the correlation between $e_k$ and $\epsilon_{k-1}$:

$$\rho_k = \alpha \rho_{k-1} + (1-\alpha) e_{k-1}$$

A change is detected when $|\rho_k|$ suddenly increases.

3.2 Second class

The second class musters algorithms which are based on Maximum Likelihood Estimation of the time and magnitude of the jumps, in order to get statistical performance near to optimum.

The basis statistical approach is the GLR method \cite{8}.

3.2.1 GLR

Considering an AR or ARMA modelisation of the signal, and assuming the following model for the joint density of the observations $Y_k$:

$$p(Y_{1:k}, k \geq r) = p_{\theta_0}(y_1) \prod_{i=2}^{r-1} p_{\theta_0}(y_i) p_{\theta_1}(y_r) \prod_{i=r+1}^{k} P_{\theta_1}(y_i)$$

The log-likelihood ratio between the two hypotheses for an n-size sample is:

$$S^n_r (H_0'/H_1) = \sum_{k=1}^{n} \log \frac{P_{\Theta_1}(y_k / Y_{1:k-1})}{P_{\Theta_0}(y_k / Y_{1:k-1})}$$

The change instant $r$ and the parameter vector $\hat{\Theta}$ which are unknown are then replaced by their maximum likelihood estimates:

$$\hat{r} = \arg\max_r S^n_r (H_0'/H_1)$$

$$\hat{\Theta} = \arg\max_{\Theta} S^n_r (H_0'/H_1)$$

Then, the change detector, conditioned by a threshold $\lambda$ is given by:

$$s_n = \max_{1 \leq r \leq n} S^n_r (H_0'/H_1) \leq \lambda$$

When $s_n$ exceeds $\lambda$, $H_1$ has to be decided, $H_0$ otherwise.

But the double maximisation in the previous sum is complex and time consuming and so complicates the recursive structure of the algorithm. Thus, the practically used methods (Brandt's GLR \cite{9}, the divergence test \cite{4}) consist in an approximated on-line implementation of the GLR. Roughly speaking, they can be described as follows: At time $t$, two or eventually three observation windows of different lengths are sequentially considered (according to the selected method): a reference large one which takes into account all or almost all the past of the signal, and a test short one which only contains the L past observations. Within these observation windows, an AR or ARMA parametric model is identified. When a change occurs, the large windowed data model can be supposed to remain constant; on the other hand, the short windowed data one, obtained from a finite length data set quickly changes. Then, the detection process consists in comparing the two previous models and to decide if they are different by the mean of a suitable distance.

3.2.2 Brandt's GLR

Here, the GLR test consists first in detecting a change in the signal model variance $\sigma^2_k$ by computing an estimation of a signal AR model within three observation windows defined as follows:

$$1 \quad W_1 \quad k-1 \quad k \quad W_2 \quad t$$

where $W_1$ is a growing reference window starting from the last boundary detected; $W_2$ a sliding test window of constant length ($L$), and $W_0$ the pooled window. Then, the estimation of the change instant is performed. Assuming the change is detected at $t = l_0$ (end of $W_2$), a new GLR test is applied between the hypotheses:

$$H_i$: there exists an instant $r$ ($1 \leq r \leq l_0 - L$) such that:

$$\Theta = \Theta_0 \quad 1 \leq k \leq r$$

$$\Theta = \Theta_1 \quad r < k \leq l_0$$

$$H_r: \quad \Theta = \Theta_1 \quad 1 \leq k \leq l_0 - L$$

This test is applied from $n = l_0$ to $l_0 + L$ and the change instant is given by the last value taken by $r$.

3.2.3 The Divergence Test

A cumulative distance measure $W_r = \sum_{k=1}^{r} w_k$ between two AR models $\Theta_0$ and $\Theta_1$ as shown in the next figure (cross entropy between the conditional distributions of the two models) is computed.
This cumulative sum can be shown to have a zero conditional drift under $H_0$ and a negative one under $H_1$. A procedure of Page-Hinkley [10] is then applied to detect such a behavior of the statistics $W_t$, $1 \leq t \leq n$.

\[
\begin{array}{c|c|c}
    t-L+1 & t & t+1 \\
\hline
    \Theta_0 & \Theta_1 & \Theta_2
\end{array}
\]

Many other distance measurements, which depend on the choice of the representation space (model or cepstral coefficients, spectral distribution, ..) can be used in a same way.

However, both the previous algorithms, used with any distance assume the models to be compared are exactly known. Unfortunately, even if the large windowed data model can be reasonably supposed very close to the true model (before change), the short windowed data one must be considered as an estimation obtained from just a finite length data set and thus, must be processed as a random variable.

The original part of this paper consists then in introducing a sequential procedure for detecting and estimating parameter changes via the use of a specific distance, where the random character of the short windowed data model is taken into account. This method is derived from a previous one dealing with supervised classification of finite length random signals [12], where the random character of the estimated models is introduced into the decision laws.

4 A NEW SEQUENTIAL PROCEDURE

4.1 Introduction

In a previous work [12], the authors pointed out that, to classify ARMA models, one must take into account that considered models are just estimated from finite length data sequences. Then, the uncertainty of knowledge must be included in the decision process.

Let us consider a $N$-points data sequence $\{y_k\}_{1 \leq k \leq N}$ of a random signal driven by an unknown ARMA(p,q) model $\theta = [a_1, a_2, \ldots, a_p, c_1, \ldots, c_q]$: $y_k = \Phi_k(\theta) + e_k$

with $\Phi_k = [-y_{k-1}, \ldots, -y_{k-p}, c_1, \ldots, c_q]$

The Maximum Likelihood (ML) estimator $\hat{\theta}$ of $\theta$ is:

\[x = \arg \min_{\theta} \sum_{k=1}^{N} e_k^2(\theta)\]

where $e_k$ is recursively defined by:

\[e_k = y_k - \Phi_k(\hat{\theta})\]

\[\Phi_k(\theta) = [-y_{k-1}, \ldots, -y_{k-p}, e_{k-1} + \ldots, e_{k-q} + \theta] \]

Then $\sqrt{N(x-\theta)}$ is approximately gaussian with zero mean and covariance matrix $V$ defined by:

\[V = \sum_{k=1}^{N} e_k^2(x) \sum_{k=1}^{N} \Psi_k(x) \Psi_k^T(x)\]

where $\Psi_k(\theta) = -\frac{d}{d\theta} e_k^2(\theta)$

Then, a bayesian viewpoint led the authors to propose in [12] the following distance measure between two ARMA models $x_1$ and $x_2$ identified by ML method with data sequences of length $N_1$ and $N_2$, which corresponds to covariance matrices $P_1 = V_1/N_1$ and $P_2 = V_2/N_2$:

\[d = \log \det (P_1 + P_2)^{-1} + (x_2 - x_1)^T (P_1 + P_2)^{-1} (x_2 - x_1)\]

4.2 Application to segmentation

We propose to use the two models approach, as in the divergence test, where the short-time model and the long-time one are identified by the Recursive Maximum Likelihood (RML) method.

According to [7], we use the following recursive algorithm, with initializations $x_0, P_0, \theta_1, \psi_1$:

\[e_k = y_k - \xi_k^T x_k\]

\[\mu_k = p_k - 1 \mu_{k-1} + \gamma_k (\xi_k^T \xi_k - \mu_{k-1})\]

\[K_k = P_{k-1} - K_k \xi_k \xi_k^T P_{k-1} \]

\[x_k = x_{k-1} + K_k e_k\]

\[P_k = [P_{k-1} - K_k \xi_k \xi_k^T P_{k-1}] K_k\]

see [7] to compute recursively $\psi_k$ and $\psi_k$.

$\gamma_k$ and $\lambda_k$ vary according to:

\[\gamma_0 = 1; \gamma_k = \frac{1}{1 + \lambda_k \gamma_{k-1}}\]

\[\lambda_k = \alpha \lambda_{k-1} + (1-\alpha) \lambda_0\]

with $0 < \alpha < 1, 0 < \lambda_0 < 1, 0 \leq \gamma < 1$

if $\lambda_k = 1$ then $\gamma_k = 1$

if $\lambda_k < 1$ then $\gamma_k = (1-\lambda_k) (1-\gamma_{k-1})$, $\gamma_k = 1 - \lambda_k$ the filter is adaptive and $\lambda_k$ sets the convergence speed.

Using $\lambda_k = \alpha \lambda_{k-1} + (1-\alpha) \lambda_0$, with $\lambda_0 \leq 1$, then the filter is adaptive in transient case (to quickly forget the initializations), and $\lambda_0$ sets the convergence speed in permanent case.

Then, $\gamma_k^T P_k$ is nothing but the covariance matrix $V$ defined above.

Here, we use the following values for the long-time model and the short-time one:

\[\lambda_{long} = 0.95; \lambda_{short} = 0.99; \alpha_{long} = 0.95; \alpha_{short} = 0.95\]

Then, in transient case, the 2 filters have the same behaviour, and (omitting the time-index $k$):

\[P_{short} + P_{long} = (\gamma_{short} + \gamma_{long}) V = \gamma_{short} + \gamma_{long} P_{short}\]

\[P_{short} + P_{long} = (\gamma_{short} + \gamma_{long}) P_{short}\]

In permanent case, $\gamma_{long} \equiv 0, P_{long} \equiv 0$, and

\[P_{short} + P_{long} = (\gamma_{short} + \gamma_{long}) P_{short}\]

Then, according to 4.1, the distance measure between the 2 models is defined by:

\[d = \log \det (\gamma_{short}^T P_{short}\gamma_{long} + (x_{short} - x_{long})^T (\gamma_{short}^T P_{short})^{-1} (x_{short} - x_{long})\]

Since the first term in $d$ just depends on the short-time model, we can pick it out. Then the distance used for the segmentation is:

\[\Delta = (x_{short} - x_{long})^T (\gamma_{short}^T P_{short}^{-1}) (x_{short} - x_{long})\]

5 SIMULATIONS

5.1 Method

We simulated an AR or ARMA process, with a jump at time 1000. For three different distances (namely cepstral distance, Kullback divergence between joint laws,
proposed distance), and for each values of a varying threshold, we measured the false alarm percentage between times 1 and 1000, and the detection delay. Then, representing the false alarm rate versus the detection delay, we obtain a curve from point (0,100) (nil threshold) to $(+\infty, 0)$ (infinite threshold). Then, we can conclude that the best distance is the one which gives the curve which is the closest from the point (0,0).

5.2 Results

We made two trials

- AR (3) models (fig. 1):
  - before jump: [0.65, -0.33, -0.05]
  - poles: -0.944, 0.42, -0.125
  - after jump: [0.5, -0.55, -0.1]
  - poles: -0.963, 0.628, 0.165

- ARMA (2,2) models (fig. 2):
  - before jump: [-1.79, 0.836, 0.146, 0.141]
  - poles: 0.895 ± 0.187 (norm 0.914)
  - zeros: -0.073 ± 0.368 (norm 0.375)
  - after jump: [-1.78, 0.829, 0.391, 0.244]
  - poles: 0.89 ± 0.192 (norm 0.91)
  - zeros: -0.196 ± 0.454 (norm 0.494)

We obtain the following curves.

6 CONCLUSION

The two previous trials show a significant improvement with the proposed method, which can be explained because we explicitly take into account the random nature of the identified models.

Therefore, we restricted ourselves to compare the jump in the distance between the two models of the two-models approach. We must point out the problems of methodology to compare objectively two different segmentation methods, because of the numerous parameters which are involved (choice of model, distance, window lengths, identification method...).

REFERENCES

LOWERING THE SNR THRESHOLD IN SVD-BASED FREQUENCY ESTIMATION ALGORITHMS

J. I. Portillo-García, J. R. Casar-Corredera

Dpto. Señales, Sistemas y Radiocomunicaciones
E. T. S. Ing. de Telecomunicacion - Universidad Politécnica de Madrid
Ciudad Universitaria s/n
28040 MADRID (SPAIN)
Tfn: 34-1-5495700 x 206

Some high resolution methods for frequency estimation obtain a vector in the noise subspace of the estimated covariance matrix. The frequencies are estimated from the phases of the complex roots of the polynomial associated to that vector. Those methods need a number of components of the noise subspace estimated vector reasonably greater than the number of frequencies to be estimated. However, this makes difficult to distinguish between the signal roots and the spurious roots originated by the higher degree of the polynomial. Minimum-norm method overcomes this difficulty moving the spurious roots inside the unit circle, which allows to pick the roots nearest to the unit circumference. We propose in this communication a new method to perform the correct signal-root selection. The procedure improves the performance of the minimum-norm method and allows to employ other vectors in the noise subspace. As a side result, a sub-optimum root selection method improves the performance of FBLP, and avoids the necessity to compute the SVD of the data covariance matrix.

1. INTRODUCTION

Frequency estimation of complex exponentials in noise is a very important subject in applications such as: radar, sonar, radioastronomy, speech and music processing, etc. There have been proposed a lot of solutions to this problem: some of them are based on classical spectral analysis, some other are based on maximum-likelihood approaches, or on linear prediction methods (such as FBLP) [1] [2]. FBLP method [3] [4] have been widely used, due to its good behaviour in moderate-high SNR. However, when SNR lowers, the performance gets worse very quickly.

Lately, a family of high resolution methods have been applied to this problem, with good performance at low SNR. These methods are based on computing the SVD of the data covariance matrix [5]. Some of them, such as minimum-norm [6], obtain a vector in the noise subspace of the data covariance matrix. Using that vector, an associated polynomial is formed:

$$G(z) = \sum_{i=0}^{L} g_i z^{-i}$$  \hspace{1cm} (1)

where $g_i$ is the $i$-th component of the minimum-norm vector. Supposing we know we have $p$ complex exponentials, the $p$ roots which are nearest to the unit circumference are selected. The $p$ frequencies are then estimated as:

$$f_i = \tan^{-1} \frac{\Im(z_i)}{\Re(z_i)}$$  \hspace{1cm} (2)

where $z_i$ is one of the $p$ roots selected. Minimum-norm property in the noise subspace vector selected is very desirable in the sense that it tends to locate $p$ signal roots on the unit circumference, with the adequate phases, and to
move the spurious \((L-p)\) roots inside the unit circle, far away from the \(p\) signal roots. It is then easy to select the correct signal roots.

However, when SNR is too low, the spurious roots show an erratic behaviour, and some of them move towards the unit circumference. Besides, some signal roots may move away from the unit circumference. If one uses the root selection procedure previously commented, it will result in a wrong root selection, which in turn causes an increment in the mean squared value of the frequency error. The final effect is a threshold effect: at some value of SNR, the frequency errors grow very fast, due to the effect of the noise spurious roots.

2. PROPOSED METHOD

We propose in this contribution a new method to select the correct roots of the polynomial defined in (1). Let us define the steering vector at frequency \(f\) as:

\[
d(f) = \begin{bmatrix} 1 & e^{j2\pi f} & \cdots & e^{j2\pi Lf} \end{bmatrix}^T
\]  

(3)

Theoretically, the steering vectors corresponding to the signal frequencies will be in the signal subspace, and the steering vectors corresponding to the spurious frequencies will not be in that subspace. Therefore, we may expect the norm of the projection of steering vectors corresponding to signal frequencies onto the signal subspace to be larger than the projection of steering vectors corresponding to spurious frequencies. We then propose to select the \(p\) frequencies which maximize the norm of the associated steering vector onto the signal subspace:

\[
P(f) = d^H(f)P_s d(f)
\]  

(4)

where \(P_s\) is the signal subspace projection matrix, defined as:

\[
P_s = \sum_{i=0}^{p-1} u_i u_i^H
\]  

(5)

and \(u_i\) are the eigenvectors of the data correlation matrix which span the signal subspace.

The employment of this root selection procedure has two main advantages:

1) It improves the performance of the minimum-norm method. Since the method is not directly based on computing distances to the unit circumference, it is more robust against root movements in complex plane due to the low SNR. The threshold lowers, as can be seen in an example below.

2) It allows the employment of some other vectors in the noise subspace. Minimum-norm is a well-behaved method if one uses a root selection criterion based on the position of the roots with respect to the unit circumference. If the root selection criterion changes, some other noise subspace vectors may be used, even if they locate the spurious roots near of the unit circumference.

As a side result, we will examine an improvement of the FBLP method. We use \(p\) columns of the data covariance matrix to estimate a sub-optimum signal subspace projection matrix:

\[
P_s' = \begin{bmatrix} r_{i_1} & r_{i_2} & \cdots & r_{i_p} \end{bmatrix} \begin{bmatrix} r_{i_1} & r_{i_2} & \cdots & r_{i_p} \end{bmatrix}^H
\]  

(6)

where \(r_{i_k}\) is the \(i_k\)-th column of the data covariance matrix. We select \(p\) roots, using the function defined in (4), as explained before, using the sub-optimum projection matrix \(P_s'\). Note that this procedure avoids SVD computing. Therefore, its computational cost will be lower than SVD-based methods, with good results, as it is shown below.

3. SOME RESULTS

In this section, we show some results that illustrate the improvement of the proposed root selection method. 25 points of the following signal have been generated:

\[
x[n] = e^{j(2\pi 0.1 n)} + e^{j(2\pi 0.13 + \eta n)} + w[n]
\]  

(7)

where \(w[n]\) is a Gaussian, white noise. SNR is varied from -2 dB to 25 dB in steps of 1 dB. 500 data registers have been generated at each SNR.
Figure 1 shows the improvement in the MSE of the normalized frequency 0.1, when we select the roots in the minimum-norm method using the proposed procedure. Note the lower threshold obtained.

![Figure 1](image)

The figure shows the results for the proposed root selection method. Minimum-norm vector, $f = 0.1$.

- Classical root selection.
- Proposed root selection.

As we stated before, the proposed procedure allows the employment of some other noise subspace vectors. Figure 2 shows the improvement in the MSE of the normalized frequency 0.1, when we use a noise subspace proposed vector, $\mathbf{g}_{p0}$: the projection onto the noise subspace of the FBLP prediction error filter $\mathbf{g}_f$:

$$\mathbf{g}_{p0} = \mathbf{P}_n \mathbf{g}_f$$  \hspace{1cm} (8)

where $\mathbf{P}_n$ is the projection matrix onto the noise subspace. Note the improvement obtained using the proposed root selection procedure. Without it, this vector would not be useful.

Figure 3 is a comparison between minimum-norm and the proposed vector, both cases using the proposed root selection method. We can appreciate the improvement in the threshold using the proposed noise subspace vector. Furthermore, this vector, as we will show in a future contribution, is more robust when there are errors in the estimation of the model order ($p$).

![Figure 2](image)

Figure 2. Results for the proposed noise subspace vector, $f = 0.1$.

- Classical root selection.
- Proposed root selection.

![Figure 3](image)

Figure 3. Comparison of minimum-norm vector results and the proposed noise subspace vector results, $f = 0.1$.

- Minimum-norm.
- Proposed vector.

Finally, figure 4 shows the performance of FBLP method using the suboptimal root selection procedure, with the matrix $\mathbf{P}_s$, defined in (6). Note the improvement obtained, compared with the use of FBLP and selection of the $p$ roots closest to the unit circumference.

4. CONCLUSIONS

In this paper, we propose a different method for the selection of the signal roots of polynomials associated to vectors in the noise subspace of the
Figure 4. Improvement in FBLP results using the suboptimum root selection procedure.

\[ f = 0.1. \]

- Classical root selection.
- --- Proposed root selection.

Data covariance matrix. The method is based on constructing a steering vector for each root, and selecting the \( p \) roots for which the projection of that steering vector onto the signal subspace is maximum. Therefore, the method is not based on computing any distance to the unit circumference, and offers better results when it is applied to minimum-norm vector, as it can be seen in simulation examples. Besides, this root selection procedure is general, and may applied to some other vectors in the noise subspace. We propose the use of a noise subspace vector obtained as the projection of the FBLP error prediction filter vector onto the noise subspace. Again, simulations present examples of the better performance of this vector. However, the roots of the polynomial associated to this vector are very bad behaved, and, without the use of the root selection procedure proposed, this vector could not be employed. As a side result, we apply the root selection procedure to FBLP method, using a suboptimal signal subspace projection matrix, with improved results.

5. REFERENCES

DEGENERACY IN NON-LINEAR PREDICTORS

B. Mulgrew, K. Nisbet & S. McLaughlin
The University of Edinburgh

Abstract This paper describes an initial attempt to reduce the complexity of Volterra series and radial basis function nonlinear predictors. The algorithm exploits signal subspace concepts to reduce the number of nonlinear terms, thereby maintaining the same MSE performance.

1. Introduction

At present, two of the major choices in nonlinear signal prediction are the radial basis function (RBF) network and the Volterra series combiner (VSC). They share a common architecture—a non-linear expansion followed by a linear combiner. The resultant linear least squares (LS) optimisation problem is usually overdetermined and resort is often made to a minimum norm solution [1].

The recent work reported in [2] and [3] have proposed exploiting the overdetermined nature of the problem in a positive way to select centres [2] or to remove polynomial terms [3]. If a system of linear equations is overdetermined, then the amount by which it is overdetermined tells us how many parameters or elements in the weight vector can arbitrarily be set to zero without degrading the performance of the predictor. Setting a weight to zero effectively removes an RBF centre or a polynomial term from a VSC. These methods, which we might call state reduction techniques, tell us how many elements we can remove but do not tell us which ones would be the best choice.

In this paper we present a geometrical interpretation of the problem in terms of the signal subspace. The geometrical interpretation indicates that some choices of reduced state subspace may be better than others with respect to numerical conditioning. Further it is suggested that the reduced state subspace which is "close" to the original subspace is the best choice. Initial results indicate that the proximity of the reduced state subspace to the original signal subspace leads to solutions which tend to preserve the good numerical properties of the minimum norm solution.

2. Background

If, for example, we have an \( L \)-element signal vector \( \mathbf{x}(k) \) which contains the last \( L \) signal samples. This vector is used to predict the next signal sample \( \mathbf{x}(k+1) \). A fixed non-linearity such as a Volterra series expansion or a radial basis function expansion is applied to each vector \( \mathbf{x}(k) \) to produce an \( N \)-vector \( \mathbf{z}(k) \) [4]. Usually \( N \) is greater than \( L \). The sequence of \( \mathbf{z}(k) \) vectors has associated with it an \( (N \times N) \) autocorrelation matrix \( \phi_{\mathbf{z}} \). The matrix has rank \( M \) which is usually less than \( N \). In order to facilitate the development of a geometrical interpretation consider the simple case where \( N = 3 \) and \( M = 2 \). Since the autocorrelation matrix has rank 2, the sequence of 3-vectors \( \mathbf{z}(k) \) spans a subspace of order 2. This plane is illustrated in Figure 1 and is labelled the "signal subspace". The axes set \( \mathbf{z}_1, \mathbf{z}_2 \) and \( \mathbf{z}_3 \) define orthonormal directions for outputs from individual non-linear functions in the Volterra or RBF expansions. For example:

\[
\mathbf{z}(k) = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} = 3 \mathbf{z}_1 + 2 \mathbf{z}_2 + \mathbf{z}_3
\]

The signal subspace is spanned by the eigenvectors \( \mathbf{z}_2 \) and \( \mathbf{z}_3 \) which are associated with the non-zero eigenvalues \( \lambda_2 \) and \( \lambda_3 \). The remaining eigenvector \( \mathbf{z}_1 \) is associated with the zero eigenvalue \( \lambda_1 \) and is by definition orthogonal to the signal subspace.

3. The Nearest Subspace

In reducing the number of terms in a Volterra expansion or selecting centres for an RBF, we essentially wish to throw away some of the axes vectors while maintaining the quality of the prediction. One possible way of doing this is to choose a subset of axes vectors which are close to the signal subspace. Thus we have to define a measure of distance between a unit axis vector (e.g., \( \mathbf{z}_1 \)) and the signal subspace. Such a measure could be obtained by calculating the length of the projection of a unit axis vector onto the signal subspace. The greater the length the "closer" the axes vector to the subspace. The projection can be calculated using the eigenvectors which define the signal subspace. The algorithm can be summarised as follows: (i) calculate eigenspectrum and hence rank \( M \); (ii) measure the distance between all axis vectors and the signal subspace; (iii) choose the \( M \) axis vectors which are closest to the subspace; (iv) recalculate prediction coefficients and compare prediction quality.

4. Other Possible Solutions

The minimum norm solution \( \mathbf{h}_{\text{min}} \) lies in the signal subspace. Other equally good solutions (in a mean square
error (MSE sense) are defined by the minimum norm solution plus a weighted sum of the eigenvectors which are orthogonal to the signal subspace. In the simple example illustrated in Figure 1 this would be:

\[ h = h_{\text{min}} + \alpha \varepsilon_1 \]

where \( \alpha \) is a constant. A more general way of looking at this effect is facilitated if a rank 1 autocorrelation matrix is considered. This situation is illustrated in Figure 2. The signal subspace is now a line defined by the eigenvector \( \varepsilon_2 \). The minimum norm solution lies in this subspace at the point \( h_{\text{min}} \). All other solutions lie in a subspace which is orthogonal to \( \varepsilon_2 \) and intersects it at \( h_{\text{min}} \). Thus the "plane of other possible solutions" is defined:

\[ h = h_{\text{min}} + \alpha_1 \varepsilon_1 + \alpha_2 \varepsilon_2 \]

In [3] this property was utilised to arbitrarily set elements of \( h \) to zero. In the example of Figure 2 we could set the components along the \( \varepsilon_1 \) and \( \varepsilon_2 \) axes to zero in which case the solution for \( h \) would be found at the point where the "plane of other possible solutions" intersects the axis \( \varepsilon_2 \). However it would be equally valid to set the components along \( \varepsilon_1 \) and \( \varepsilon_2 \) to zero in which case the solution for \( h \) would be found at the point where the "plane of other possible solutions" intersects the axis \( \varepsilon_2 \). The most worrying possibility would be to choose a solution along \( \varepsilon_1 \) since the "plane" is almost parallel to that axis and hence they will intersect at a very large value of \( h \). This is why the minimum norm solution is often chosen as the solution to a set of overdetermined linear equations - because it is numerically well-conditioned.

Step (iv) of the nearest subspace method appears at first to be an approximation. It is indeed an approximation to the minimum norm solution. A more important question to ask is whether it can find one of the other possible solutions which is close to the minimum norm solution? Consider Figure 2 once more. The axis \( \varepsilon_2 \) is clearly the nearest axis to the signal subspace defined by \( \varepsilon_2 \). We would suggest that the solution which lies on \( \varepsilon_2 \) defines a "best solution" to the problem of removing degeneracy from a nonlinear predictor in that it is both close to the minimum norm solution and sets 2 components to zero?

5. Algorithm

We can summarise the steps towards reducing the number of centres or non-linear elements in a systematic way as follows:

(i) calculate eigenvectors and eigenvalues of the matrix 
\[ \sum_k \varepsilon(k) \varepsilon^T(k) \] which defines the LS problem

\[ \sum_k \varepsilon(k) \varepsilon^T(k) h = \sum_k \varepsilon(k) x(k+1) \] (1)

and hence deduce the rank \( M \); the eigenvectors are collected into a matrix \( \mathbf{V} = [\varepsilon_1 \varepsilon_2 \ldots \varepsilon_M] \).

(ii) project each of the axis vectors \( \varepsilon_1, \varepsilon_2, \ldots, \varepsilon_M \) onto the signal subspace; in general the length of this projection has the form \( d_j = \varepsilon_j^T \mathbf{Y}_s \mathbf{Y}^T_s \varepsilon_j \) where \( \mathbf{Y}_s \) is the matrix of eigenvectors which span the signal subspace. The indices or addresses of the \( M \) largest elements of \( d \) are the indices or addresses of the centres or nonlinearities which are selected. The remaining indices indicate the weights which will effectively be set to zero.

(iii) Other possible solutions with the same MSE are given by: \( h = h_{\text{min}} + \varepsilon_i h \) where \( \varepsilon_i \) is the matrix of eigenvectors orthogonal to the signal subspace and \( h \) is a column vector with \( N = M \) elements. By suitable manipulation the appropriate elements of \( h \), identified in (ii), can be set to zero and hence the remaining non-zero elements of \( h \) can be calculated. Alternatively appropriate rows and columns of (1) can be forced to zero and the reduced state LS problem can be solved directly.

6. Results

To demonstrate the use of the technique we consider Duffing's equation [5] in the chaotic region as the signal generation mechanism i.e.

\[ \frac{d^2 x}{dt^2} + 0.25 \frac{dx}{dt} - x + x^3 = 0.3 \cos(t) \]

This is solved using fourth order Runge-Kutta with a step size \( \Delta t = 0.1 \). An appropriate choice for the embedding dimension is made - in this case \( L = 6 \). The predictions are initially constructed using a 3-term (cubic) Volterra series with a full compliment of \( N = 84 \) polynomial terms. The performance of this "full compliment" predictor sets the lower bound on performance in this example. Reduced state predictors are also formed from the 84 possible polynomial terms using the method described in section 5 with \( M = 69 \) and \( M = 42 \).

Figure 3 illustrates the normalised MSE performance of these predictors against iterate or sample number \( n \) for a sequence of 10000 data samples. The normalised MSE is defined as:

\[ \text{MSE}(n) = 10 \log_{10} \left( \frac{1}{n} \sum_{k=1}^{n} e^2(k) \right) \]

where \( e(k) \) is the one-step prediction error and \( x(k) \) is the signal at sample \( k \). The performance of the reduced state predictor with \( M = 69 \) polynomial terms (Figure 3(b)) is within 2 dB of the performance of the full predictor (Figure...
FIGURE 1 Rank-2 Signal Subspace and Associated Eigenvectors.

FIGURE 2 Other Possible Solutions (rank-1 signal subspace).
3(a)) with $M=84$ terms. Reducing the number of term to $M=42$ degrades the performance of the predictor by a further $15$ dB to approximately $-43$ dB (Figure 3(c)).

7. Discussion

In this paper some initial results are presented which demonstrate how the concept of a "nearest subspace" can be used to systematically reduce the number of polynomial terms in a Volterra series predictor used on a chaotic process. In principal the technique can also be used in conjunction with RBF predictors to remove centres in a similar manner to the technique described in [2]. In the particular example illustrated in Figure 3 it should be noted that 15 of the 84 polynomial terms can be removed without significantly degrading the quality of the predictions.

8. Acknowledgement

The authors wish to acknowledge the support of the UK Science and Engineering Research Council.

References


\[\text{FIGURE 3 Normalised MSE performance of cubic Volterra one-step predictor on chaotic Duffing's equation: (a) full set of 84 polynomial terms; (b) reduced state with 69 polynomial terms; (c) reduced state with 42 polynomial terms.}\]
POLYNOMIAL PHASE SIGNAL ESTIMATION

Z. FARAJ and F. CASTANIE (Member EURASIP)
GAPSE-ENSEEIH, 2 rue Camichel, 31071 TOULOUSE, FRANCE

ABSTRACT: To overcome the problem of estimating the parameters of complex polynomial FM signals, a new algorithm, based on a combination of two new transforms, is proposed. The polynomial autocorrelation function and the modified Fourier transform, which can be considered as a generalization of the standard autocorrelation function and the standard Fourier transform, are adapted for polynomial phase signals. Numerical examples are given in order to illustrate the performances of this algorithm and to show the good agreement with the theory.

I. INTRODUCTION

Signals having constant amplitude and polynomial phase are common in radar and communication applications [1]. The estimation of the coefficients of the phase polynomial is important in various applications such as are found in deep space telemetry systems [2].

In a recent correspondence [1], a new transform was introduced for estimating the phase of complex signals that have constant amplitude and polynomial phase. Let \( g(t) \) be a complex valued function of a real variable \( t \), \( Q \) an integer, \( \tau \) a positive real number and \( M \) a positive integer. The Polynomial Transform \( PT_M[g, \omega, \tau] \) is defined as the Fourier transform of \( P_M[g(t), \tau] \) over the interval \([0, T]\), i.e.,

\[
PT_M[g, \omega, \tau] = \int_0^T P_M[g(t), \tau] \exp(-j\omega t) \, dt
\]

where:

\[
P_M[g(t), \tau] = \sum_{q=0}^{M-1} [g^{*q}(t-q\tau)]^{(M-1)q}
\]

\( g^{*q}(t) = g(t) \) if \( q \) is even.

\( g^{*q}(t) = g^*(-t) \) if \( q \) is odd.

\( .^* \) denotes complex conjugation.

The following special cases illustrate this definition:

\[
PT_1[g, \omega, \tau] = \int_0^T g(t) \exp(-j\omega t) \, dt
\]

\( PT_1 \) is the standard Fourier transform. \( PT_2 \) is the ambiguity function. \( PT_3 \) is new and so are the higher order transforms. One of the basic properties of \( PT_M \) is that \( |PT_M[g, \omega, \tau]| \) has a global maximum at \( \omega = M! \tau^{M-1} a_M \) if \( g(t) \) is a complex function with constant amplitude and polynomial phase on the interval \([0, T]\), i.e.,

\[
g(t) = b_0 \exp\left\{ \sum_{n=0}^M a_n t^n \right\} \quad 0 \leq t \leq T
\]

The choice of \( \tau \) is arbitrary, but in practice it will affect the accuracy of the estimated coefficients.

In this paper, two new transforms are proposed for estimating the phase of complex signals of this type:

- The Polynomial Autocorrelation Function.
- The Modified Fourier Transform.

II. POLYNOMIAL AUTOCORRELATION FUNCTION

The Polynomial Autocorrelation Function (P.A.F) \( C_M[g(t), \tau] \) is defined by:

\[
C_M[g(t), \tau] = E \left\{ P_M[g(t), \tau] \right\} = E \left\{ \sum_{n=0}^M [g^{*q}(t-q\tau)]^{(M-1)q(n)} \right\}
\]

\( E \{ . \} \) denotes the expectation operator.

The following special cases illustrate this definition:
Theorem 1: Let $g(t)$ be a complex function with constant amplitude and polynomial phase, i.e.,

$$g(t) = b_0 \exp \left( \sum_{m=0}^{M} a_m t^m \right)$$

Then:

$$C_M[g(t), \tau] = C_M[\tau] - b_0^{(2^M)} \exp \left( J M! a_M \tau^M \right)$$

Proof:

$$P_{M+1}[g(t), \tau] = \frac{1}{1 + \left( g^{*} (1-q \tau) \right)^{(M)}}$$

$$P_{M+1}[g(t), \tau] = \frac{1}{1 + \left( g^{*} (1-q \tau) \right)^{(M)}}$$

$$P_{M+1}[g(t), \tau] = \frac{1}{1 + \left( g^{*} (1-q \tau) \right)^{(M)}}$$

$$P_{M+1}[g(t), \tau] = \frac{1}{1 + \left( g^{*} (1-q \tau) \right)^{(M)}}$$

By Lemmas A.1 and A.2 [1], we have:

$$P_{M+1}[g(t), \tau] = b_0^{(2^M)} \exp \left( J M! a_M \tau^M \right)$$

Therefore,

$$C_M[g(t), \tau] = E \left( P_{M+1}[g(t), \tau] \right) = b_0^{(2^M)} \exp \left( J M! a_M \tau^M \right)$$

Lemma A.1: \[ \sum_{n=0}^{M} (-1)^n (M)_n \tau^M = 0 \quad 0 \leq n \leq M + 1 \]

Lemma A.2: \[ \sum_{n=0}^{M} (-1)^n (M)_n \tau^M = 0 \quad 0 \leq n \leq M + 1 \]

Theorem 2: Let $g(t)$ be a complex function having constant amplitude and polynomial phase with an additive noise component, i.e.,

$$g(t) = b_0 \exp \left( \sum_{m=0}^{M} a_m t^m \right) + B(t)$$

where $B(t)$ is a complex white Gaussian noise with zero mean and variance $\sigma_B^2$. Then:

$$C_M[g(t), \tau] = C_M[\tau] - b_0^{(2^M)} \exp \left( J M! a_M \tau^M \right) + P_{M+1}[g(t), \tau]$$

$$P_{M+1}[g(t), \tau] = \frac{1}{1 + \left( g^{*} (1-q \tau) \right)^{(M)}}$$

The proof of this theorem do not appear in this paper for lack of space. It can be found in the forthcoming thesis [4].

Suppose the signal is sampled at interval $T_s$, and let $k$ be a positive integer. Then the unbiased Polynomial Autocorrelation Function estimator is defined by:

$$C_M[g(n), k] = \frac{1}{N - Mk} \sum_{n=0}^{N} \left( g^{*} (n-k) \right)^{(M)}$$

where:

$$g(n) = b_0 \exp \left( \sum_{m=0}^{M} a_m n^m \right) + B(n) \quad 1 \leq n \leq N$$

III. THE MODIFIED FOURIER TRANSFORM

The Modified Fourier Transform (M.F.T.) is introduced for estimating the highest order coefficient $a_M$. Let $x(\tau)$ be a complex valued function of a real variable $\tau$ over the interval $[0, T]$, the Modified Fourier Transform is defined by:

$$FT_M[x(\tau), \omega] = \int_{0}^{T} x(\tau) \exp (-j \omega \tau^M) d\tau$$

The following special cases illustrate this definition:

$$FT_1[x(\tau), \omega] = \int_{0}^{T} x(\tau) \exp (-j \omega \tau) d\tau$$

$$FT_2[x(\tau), \omega] = \int_{0}^{T} \tau x(\tau) \exp (-j \omega \tau^2) d\tau$$

$FT_1$ is the standard Fourier transform. $FT_2$ is new, and so are the higher order transforms.
One of the basic properties of the Modified Fourier Transform is given in the following theorem:

**Theorem 3**: Let \( x(t) \) be a complex function with constant amplitude and polynomial phase where only the highest order coefficient is nonzero on the interval \([0, T]\), i.e.,

\[
x(t) = A \exp\left(j \omega_M T^M\right) \quad 0 \leq t \leq T
\]

Then \( FT_M[x(t), \omega] \) has a global maximum at \( \omega = \omega_M \), i.e.,

\[
\arg \max |FT_M[x(t), \omega]| = \omega_M
\]

**Proof**:

\[
x(t) = A \exp\left(j \omega_M T^M\right) \quad 0 \leq t \leq T
\]

\[
FT_M[x(t), \omega] = \int_0^T T^{M-1} x(t) \exp(-j \omega T^M) dt
\]

\[
FT_M[x(t), \omega] = \int_0^T A T^{M-1} \exp\left(j \omega_M - \omega \right) T^M dt
\]

Put \( t = \tau^M \), then \( dt = M \tau^{M-1} d\tau \). This gives,

\[
FT_M[x(t), \omega] = \int_0^T A M \exp\left(j \omega_M - \omega \right) \tau^M \sin\left(\left[\omega_M - \omega\right]^{M/2}\right)
\]

Hence,

\[
\arg \max |FT_M[x(t), \omega]| = \arg \max |\sin\left(\left[\omega_M - \omega\right]^{M/2}\right)| = \omega_M
\]

Suppose the signal is sampled at interval \( T_s \). Then the Discrete Modified Fourier Transform (D.M.F.T.) is defined by:

\[
DFT_M[x(k), \omega] = \sum_{k=1}^{N_M} k^{M-1} x(k) \exp\left(-j \omega T_s^M k^M\right)
\]

where:

\[
x(k) = A \exp\left(j \omega_M T_s^M k^M\right) \quad 1 \leq k \leq N_M
\]

**IV. POLYNOMIAL PHASE SIGNAL ESTIMATION**

Under the assumptions of theorems 1, 2, and 3, the highest order coefficient \( \alpha_M \) of a polynomial phase signal \( g(t) \) is given by:

\[
\alpha_M = \frac{1}{M!} \arg \max |FT_M[C_M(g(t), \tau_1, \omega)|
\]

We can also estimate the coefficients \( (a_n) \) sequentially, starting at the highest order coefficient \( \alpha_M \). At each step, the effect of the phase term of the higher order is removed:

1) Substitute \( m = M \) and \( g_m(n) = g(n) \quad 1 \leq n \leq N \)

2) Estimate the P.A.F. by:

\[
\hat{C}_m[g_m(n), k] = \frac{1}{N - mk} \sum_{n=mk+1}^{N} \left[ g_m^2(n - qk) \right]^{(m)/k}
\]

\[
1 \leq k \leq N_M \quad N_M = \frac{N - 1}{m}
\]

3) Compute the D.M.F.T. by:

\[
DFT_m[\hat{C}_m[g_m(n), k], \omega] = \sum_{k=1}^{N_M} k^{m-1} \hat{C}_m[g_m(n), k] \exp(-j \omega T_s^M k^M)
\]

4) Estimate \( a_m \) by:

\[
a_m = \frac{1}{m!} \arg \max |DFT_m[\hat{C}_m[g_m(n), k], \omega]|
\]

5) Let \( g_{m-1}(n) = g_m(n) \exp(-j a_n T_s^M) \quad 1 \leq n \leq N \)

and set \( m \leftarrow m - 1 \)

6) if \( m \geq 1 \), go back to Step 2. Else go to Step 7.

7) Estimate \( \alpha_0 \) from \( g_0(n) \) using the formula:

\[
\hat{a}_0 = \frac{\pi}{N} \left[ \log \left( \sum_{n=1}^{N} g_0(n) \right) \right]
\]

The phase polynomial in the following example has degree 2. The instantaneous frequency is given by: \( f(n) = f_0 + f_1 T_s n \). The signal parameters are:

\[
f_0 = 10.34 \text{ Hz} \quad f_1 = 5.24 \text{ Hz} / s \quad T_s = 0.01 \text{ s} \quad N = 201
\]

The Signal to Noise Ratio which is defined by \( SNR = b_n / \sigma_x^2 \) is equal to 10 dB

Figures (1a) and (1b) show, respectively, the real and the imaginary part of the P.A.F. : \( \tilde{C}_z(g_z(n), k) \). Figure (1c) shows the magnitude of the D.M.F.T. : \( DFT_z[\tilde{C}_z(g_z(n), k), \omega] \).

Figures (2a) and (2b) show, respectively, the real and the imaginary part of the P.A.F. : \( C_z(g_z(n), k) \) after removing estimated quadratic phase term. Figure (2c) shows the magnitude of the D.M.F.T. : \( DFT_z[C_z(g_z(n), k), \omega] \).
V. CONCLUSION

In this paper, two new transforms are presented for estimating the phase of complex signals that have constant amplitude and polynomial phase: the polynomial autocorrelation function and the modified Fourier transform.

To overcome the problem of estimating the parameters of complex polynomial FM signals, a new algorithm, based on a combination of the two new transforms, is proposed and numerical examples are given in order to illustrate the performances of this algorithm and to show the good agreement with the theory.

This work is a part of Z. Faraj's thesis and has been supported by the French Space Agency (CNES) under contract N° 874 CNES 89579200.

REFERENCES


THE MV SPECTRAL CONVERGENCE APPLIED TO RANDOM FIELDS

(1) D. E. LYON, (2) P. J. SHERMAN, (3) A. E. FRAZHO

(1) University of New Brunswick, Fredericton, N.B., Canada E3B 5A3
(2) Iowa State University, Ames, IA 50011, U.S.A.
(3) Purdue University, West Lafayette, IN 47907, U.S.A.

ABSTRACT

This work addresses the problem of identifying a multichannel harmonic signal field that has been corrupted by an unknown homogeneous noise field. Because recursive spectral-based algorithms such as the Levinson algorithm do not exist in the d-dimensional setting (for \( d > 1 \)), the computational expense associated with multichannel spectral analysis on a d-dimensional field can play a major role in the types of models that can be considered and in the resolution of the d-dimensional lattice on which it is computed. This is particularly true of a recently developed method of Frazho and Sherman [1] which extends the multichannel convergence-based method of [2] to the d-dimensional setting. As noted [2], in the random process setting (\( d = 1 \)) the MV(n) spectra can be computed in a recursive fashion. This work presents a method for utilizing this recursive nature for the \( d > 1 \) setting, as a simple approach would be to apply the MV(n) method to each dimension of the field. This has the disadvantage of not utilizing the joint information contained in other dimensions of the field. The approach presented here allows this simple approach as a special case. More generally however, it allows information from the other dimensions to be utilized in a compact way.

1. BACKGROUND

The problem of identifying the periodic components of a multichannel random field arises in a variety of areas, including radar and sonar processing, seismic analysis, active noise control, and acoustic holography, to name a few. The most common approach to solving this problem is to identify the peaks in the autospectra of the estimated spectral matrix. This matrix can be obtained by a number of methods. The most popular method is the discrete Fourier transform (DFT) approach. Improved algorithms for computation of the multidimensional DFT have been proposed by [3], [4], while a real time implementation using an FFT chip has recently been proposed by [5]. The autoregressive (AR) method has been popular in settings involving multichannel random processes, and single-channel 2-D random fields (see [6] for references), but has not been considered to any degree for the more general multichannel random field setting. One obvious reason for this is no doubt related to the complexities involved in extensions to the random field setting from the scalar setting (e.g. [7]). The minimum-variance (MV) method, otherwise known as Capon's maximum likelihood method, was originally proposed by Capon [8] in the single-channel 3-D random field setting, and was perhaps the first alternative to the DFT in this setting. Actually, Capon first computed the DFT of the time domain signal, and then applied the MV method to the 2-D spatial dimensions of this transformed data at each time frequency. It has been a popular alternative to the DFT approach in sonar processing, but only as a fixed order estimate. In the d-dimensional setting the MV(n) spectral estimate has the general form [1]:

\[ \text{MV}_n(z) = \left[ E(z) \bar{R}_n^{-1} E(z)^* \right]^{-1} \]

where \( E(z) = [I, z^{-1}, \ldots, z^{-(n-1)}] \). Here \( I \) is the identity on \( C^p \) (\( p \) equals the number of channels), \( z \in C^p \) has elements of modulus one and takes the form \( e^{i\omega} \) for \( d = 1 \), \( \{ \sigma_j \}_{j=1}^d \) is an ordering of the d-dimensional lattice, and \( z_j = \prod_{k=1}^d \sigma_j^{(k)} \). The \( p \times p \) block covariance matrix \( \bar{R}_n \) is in general not Toeplitz. Thus, for each order, \( n \), the computation of the \( \text{MV}(n) \) spectrum will entail \( O(p^n) \) operations for each value of \( n \) using conventional methods.

Even though the results in [1] provide the theoretical elements for utilizing the convergence properties of the \( \text{MV}(n) \) spectra in the d-dimensional setting, to implement these results in the fashion addressed in [9] would entail a notable effort. For example, the rate of convergence, so central to that work, will clearly depend on the order in which one proceeds along the d-dimensional lattice (i.e. the ordering \( \{ \sigma_j \}_{j=1}^d \)). For this reason, the method of [1] will not be addressed in this work. Instead, we will present an approach for using the \( \text{MV}(n) \) spectral approach in [2] to identify harmonic signals in the random field setting. There are instances wherein the random field spectral information can be obtained by considering only a subset of the random field data. For example, if the field is known to be isotropic [e.g. the correlation function \( R(x) = R(||x||) \)] and \( x \) is in \( C^d \), then all of the spectral information is contained in the correlation sequence \( R(x) \), corresponding to the random process associated with any \( j^{th} \) component of the random field. Theoretically then, one could
obtain the desired spectral information by utilizing
spectral methods for random processes. Similarly, if
the random field consists of harmonic plane waves
then theoretically one could recover their wave-
numbers from information contained in the com-
ponent correlation sequences. Even though this
approach would significantly reduce the computa-
tional load, it is suboptimal in the sense that it does not
utilize all of the information contained in the random
field. A compromise between these two extremes
would be to somehow include all of this information in
the component correlation sequences. The purpose of
this work is to present a method for doing just that, in
order to identify harmonic signals in arbitrary un
known homogeneous noise fields.

2. WAVENUMBER IDENTIFICATION

Consider a p-channel random field

\[ Y(x) = S(x) + N(x), \quad x \in \mathbb{C}^d \]

where the noise field \( N(x) \) is homogeneous and purely
deterministic, and the signal field is purely
deterministic and has the form

\[ S(x) = \sum_{n=1}^{Q} a_n e^{i(\langle \text{Kn}, x \rangle + \phi_{1n})}, \cdots, e^{i(\langle \text{Kn}, x \rangle + \phi_{pn})} \]

(2)

Here \( \langle x, y \rangle = x^* y \) denotes the usual inner product.
The purpose of this work is to present a new technique
which uses the multichannel random process signal
recovery method of [2] to identify the signal
wavenumbers \( \{\text{Kn}\} \) and amplitude matrices \( \{a_n\} \).
Under the assumption that the signal phases \( \{\phi_{jn}\}
\) \( n = 1, \ldots, 1 \) and \( j = 1, \ldots, p \) are mutually independent
and identically distributed uniformly over \([0, 2\pi]\), the

\[ R_Y(x) = R_S(x) + R_N(x) \]

\[ = \sum_{n=1}^{Q} a_n^* e^{i\langle \text{Kn}, x \rangle} + R_N(x), \quad x \in \mathbb{C}^d. \]

(3)

Note, when \( d = 1 \) then (1) is termed a p-channel
random process, and the signal information can be
identified using the method of [2].

Our proposed method for identification of the
wavenumbers \( \{\text{Kn}\} \) proceeds in two stages. Denote
\( \text{Kn} = [k_{n1}, k_{n2}, \ldots, k_{nd}]^t \). The first stage successively
identifies the collections \( \{k_{1n}\}_{n=1}^{1} \) for \( j = 1, 2, \ldots, d \); that
is, the collection of first components of each
wavenumber are estimated initially, then the second,
and so forth. The second stage determines the appro-
priate combination of \( d \) components, one from each
collection, to form the \( n \)th wavenumber \( \text{Kn} \) for each
\( n = 1, 2, \ldots, Q \).

We begin by identifying \( \{k_{1n}\} \) \( n = 1 \) in no particular
order. The same procedure can be applied repeatedly
to recover the remaining collections of
components. Denote

\[ \text{Kn} = [k_{n1}, k_{n2}, \ldots, k_{nd}]^t \in \mathbb{C}^{d} \]

and in (3) let \( x = [x, \bar{x}]^t \) where \( x \) member \( C \) and
\( \bar{x} \in \mathbb{C}^{d-1} \). The information associated with \( \bar{x} \) will be
condensed by selecting a collection of functions \( \{f_k\}_{k=1}^{1} \)
with values in \( \mathbb{C}^{d} \), from the space

\[ L^2(\mathbb{C}^{d}, \mathbb{C}^{d-1}) = \{f(x) \in C^d \mid \sum ||f(x)||^2 \, d\mu(x) < \infty \} \]

(4a)

where the inner product in this space is denoted

\[ \langle f, d x \rangle = \sum \langle f, d x \rangle \]

(4b)

where \( \mu(\bar{x}) \) is a specified measure. The following p-
channel random process can now be formed:

\[ y(x) = [y_1(x), y_2(x), \ldots, y_p(x)]^t \]

(5)

where

\[ y_j(x) = \langle Y(x), y_j(x) > \mu \]

From (1) and (2) it follows that

\[ y_j(x) = \sum_{n=1}^{Q} A_j n_{kn} e^{i(k_{n1} x + \phi_{jn})}, \cdots, e^{i(k_{np} x + \phi_{jn})} \]

\[ + n_j(x), \quad j = 1, 2, \ldots, p \]

(6a)

where \( A_j \) is a row-vector of the form

\[ A_j = \int \bar{f}_j(\bar{x}) d\mu(\bar{x}) a_n \]

(6b)

and the noise random process is

\[ n_j(x) = \langle N(x), y_j(x) > \mu \]

(6c)

In vector form (6) becomes

\[ y(x) = \sum_{n=1}^{Q} A_j n_{kn} e^{i(k_{n1} x + \phi_{jn})}, \cdots, e^{i(k_{np} x + \phi_{jn})} \]

\[ + n(x) = \bar{s}(x) + n(x) \]

(7)

where \( A_j = [A_1^t, A_2^t, \ldots, A_p^t]^t \). The matrix-valued
correlation function \( r_Y(x) \) is set equal to \( \nu[y(x+w)\bar{y}(w)] \)
corresponding to the p-channel random process (7) has
elements computed from (5) as

\[ [r_Y(x)]_{jk} = \langle Y_j(x), Y_j(x) > \mu \quad j, k = 1, \ldots, p \]

(8a)

where the operator

\[ \bar{r}_Y(x) = \int \bar{r}_Y(x, \bar{x} - \bar{w}) f_k(\bar{w}) d\mu(\bar{w}) \]

(8b)

is on \( L^2(\mathbb{C}^{d}, \mathbb{C}^{d-1}) \). Notice that (8) admits utilization
of all of the information in \( R_Y(x) \). The correlation
information associated with the first component of
\( x = [x, \bar{x}] \) is retained, while the remainder of the
information is condensed, and depends on the selected
functions \( \{f_k\}_{k=1}^{1} \), as well as the measure \( \mu \). The form
(7) reduced (8) to
\[ r(x) = \sum_{n=1}^{q} A_n A_n^* e^{i k_n x} + n(x) \triangleq r_0(x) + r_n(x). \]

where \( r_n(x) \) is identical to (8) upon replacing the observed field \( Y \) replaced by the noise field \( N \). The results of (2) for the \( d = 1 \) setting are now applicable. In particular, they will identify the collection \( \{k_{m2}\}_{n=1}^{q} \) as well as \( \{a_n a_n^*\} \).

Recovery of \( \{a_n a_n^*\}_{n=1}^{q} \) proceeds in the same fashion as above. In this case, one obtains another set of amplitudes which generally will differ from the above set \( a_n a_n^* \). Also, one might desire to choose a different set \( \{k_{m1}\} \) and a different measure \( \mu \). Repeated application will ultimately result in \( \{k_{m1}\}_{n=1}^{q} \) and \( \{a_n a_n^*\}_{n=1}^{q} \) for \( j = 1, 2, \ldots, d \). The signal spectral matrices \( \{a_n a_n^*\} \) can then be recovered from this information. Prior knowledge concerning the signal wavenumbers can be incorporated into the selection of the \( \{k_{m1}\}_{n=1}^{q} \) as well as \( \{a_n a_n^*\} \) for \( j = 1, \ldots, d \) to enhance the rate of convergence of the MV(\( q \)) spectra to the point spectrum matrix, as demonstrated in the next section.

To identify the wavenumber vectors from the collections \( \{k_{m1}\}_{n=1}^{q} \) for \( j = 1, \ldots, d \), without loss of generality, assume that \( \{k_{m1}\}_{n=1}^{q} \) has the most distinct components \( \leq q \). To recover \( k_{m2} \) corresponding to \( k_{m1} \) we use (3) to compute:

\[ \frac{T}{2} \int_{-T/2}^{T/2} R_Y \begin{pmatrix} x_1 \\ x_2 \\ 0 \end{pmatrix} e^{-i k_{m1} x_1} dx_1 \cong a_{n2} a_{n2}^* e^{-i k_{m2} x_2}. \]

The approximate equality in (10) will hold for sufficiently large \( T \). A subsequent Fourier transformation of any \( (i, k) \)th component of the matrix-valued function of (10) will yield a peak at location \( k_{n2} \) with a magnitude approximately equal to the value of \( (a_n a_n^*)_{k_i} \). The choice of the \( (i, k) \) element in \( R_Y(x) \) admits verification. Thus, identification of \( k_{n2} \) proceeds by noting the location where the maximum occurs in the Fourier transform of the left-hand side of (10). The element in \( \{k_{n2}\}_{n=1}^{q} \) closest to this peak is identified as \( k_{n2} \). Alternatively, one can apply the multichannel algorithm of (2) to (10). This would yield an estimate of \( k_{n2} \) as well as \( a_{n2} a_{n2}^* \). We remark that the approximate equality in (10) is not necessary. All that is required is that the DFT of (10) yield its largest peak nearer to \( k_{n2} \) than to any of the other components. A large value of \( T \) increases the chances of this, and admits an estimate of \( a_{n2} a_{n2}^* \).

If \( k_{n1} \) is repeated then such peaks will be present. In the case of a unique \( k_{n1} \) we can next recover \( k_{n3} \) by computing:

\[ \frac{T}{2} \int_{-T/2}^{T/2} R_Y \begin{pmatrix} x_1 \\ 0 \\ x_3 \end{pmatrix} e^{-i k_{n1} x_1} dx_1 \cong a_{n3} a_{n3}^* e^{-i k_{n3} x_3}. \]

and taking a subsequent Fourier transform. If \( k_{n3} \) has identified two second components \( k_{n2} \) we compute

\[ \frac{T}{2} \int_{-T/2}^{T/2} R_Y \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} e^{-i (k_{n1} x_1 + k_{n2} x_2 + k_{n3} x_3)} dx_1 dx_2 \cong a_{n1} a_{n1}^* e^{-i k_{n3} x_3}. \]

Example. Consider a 3-channel 2-dimensional random field setting. The noise field, \( N(x) \), has a correlation function given by

\[ R_N(x) = -0.5 |x_1| -0.5 |x_2| I. \]

This type of noise field is found in image processing. The signal field includes two components with amplitude matrices and wavenumbers in Table 1.

Table 1. Signal wavenumber and amplitude information.

<table>
<thead>
<tr>
<th>( K_{1/2} )</th>
<th>( a_1 )</th>
<th>( K_{2/2} )</th>
<th>( a_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.129</td>
<td>10</td>
<td>0</td>
<td>0.254</td>
</tr>
<tr>
<td>0.000</td>
<td>0</td>
<td>13</td>
<td>0</td>
</tr>
<tr>
<td>0.000</td>
<td>0</td>
<td>16</td>
<td>0</td>
</tr>
</tbody>
</table>

Our measure is chosen to be \( \mu(x) = 1 \) for \( |x| < 64 \), and our 3-channel functions are \( f_1(x) = [1, 0, 0]^T \), \( f_2(x) = [0, 1, 0]^T \), and \( f_3(x) = [0, 0, 1]^T \) for all \( x \in C \).

The collection \( (0.129, 0.254) \) of first components, normalized by \( 2q \), are easily identified by the locations of convergence on the minimum eigenvalue plot of Figure 1(a). That 0.125 is present on all three channels, while 0.254 is present on only two is suggested by Figures 1(b-c). Identification of the collection \( (0.0, 0.160) \) of normalized second components and their presence on the appropriate number of channels proceeds in exactly the same way using the information in Figure 2. To determine which second component corresponds to the first component, \( k_{n1} = 0.129 \), the Fourier transform of the (1, 1) entry of (10) was computed. The peak correctly identified \( k_{n2} = 0.0 \). The value of this peak also gave a good estimate of \( a_{n1} a_{n2}^* \). In particular, (6b) yielded

\[ A_n A_n^* = \frac{\sin(\theta_{n2}^*)}{k_{n2}^2} a_{n1} a_{n2}^*. \]

Using the estimate of \( A_n A_n^* \) obtained from the MV(160) spectrum resulted in the power estimates given in Table 2.

Table 2. Signal spectral power estimates obtained from (14).

<table>
<thead>
<tr>
<th>( a_{n1}^* )</th>
<th>( a_{n2}^* )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100(160)</td>
<td>0(0)</td>
</tr>
<tr>
<td>100(160)</td>
<td>0(0)</td>
</tr>
<tr>
<td>100(160)</td>
<td>256(256)</td>
</tr>
</tbody>
</table>
The estimate of $a_1 a_1^*$ is perfect, while that of $a_2 a_2^*$ is slightly high. Notice that because one can only recover $a_1$ up to a right unitary transformation, the information in $a_2 a_2^*$ does not readily yield the relative source strengths measured at a given receiver. Its diagonals do give the total power measured at that channel. The diagonal structure of $a_1 a_1^*$ in Table 2 does reflect the fact that each receiver is related to a distinct source.

REFERENCES


Figure 1. MV(k) spectra corresponding to the collection of first wavenumber components in Example 3. The eigenvalues $\lambda_1(k) \leq \lambda_2(k) \leq \lambda_3(k)$ corresponding to MV(k) are given in (a)-(c), respectively.

Figure 2. MV(k) spectra corresponding to the collection of second wavenumber components in Example 3. The eigenvalues $\lambda_1(k) \leq \lambda_2(k) \leq \lambda_3(k)$ corresponding to MV(k) are given in (a)-(c), respectively.
A recursive least square approach with exponential weighting factor to short time Fourier analysis

P. Gruber
C-R&D Landis & Gyr Betriebs AG
CH-6301 Zug, Switzerland

A recursive filter is derived which is able to track the amplitudes and phases of frequency components with known frequencies that are part of an arbitrary measured signal. The filter is based on the recursive least square method with exponential weighting factor. In the case of estimating the phase and amplitude of one frequency only, the complete parametrized filter is solved analytically. The resulting filter is closely related to the Kalman filter solution to that estimation problem based on a random walk model. Its structure is also similar to a discrete quadrature/correlator structure.

1. INTRODUCTION

In many application one is interested in estimating in real time the parameters $a_i(k)$, $i=0,...,N$ and $b_i(k)$, $i=1,...,N$ of a signal consisting of a sum of sinusoidal functions with time varying amplitudes and phases but with known and constant frequencies, which is part of an arbitrary measured signal $y(k)$. $y(k)$ can thus be represented as

$$y(k) = \sum_{i=1}^{N} \left[ a_i(k) \cos(\omega_i k T - \psi_i(k)) \right] + w(k)$$

$$= a_0(k) + \sum_{i=1}^{N} \left[ a_i(k) \cos\omega_i k T + b_i(k) \sin\omega_i k T \right] + w(k) \quad (1)$$

$a_i(k) = A_i(k) \cos\omega_i(k) \quad i=1,...,N$

$b_i(k) = A_i(k) \sin\omega_i(k)$

$T$: sampling period

$w(k)$: measurement noise, disturbances

The frequencies can be chosen arbitrarily except for the condition that the time variation of the signal parameters $a_i(k)$ and $b_i(k)$ are much slower than the frequencies $\omega_i$. If, for instance,

i) the frequencies $\omega_i$ are related by $\omega_i = i \omega_0 \quad i=1,...,N \quad \omega_0 = 2 \pi / \tau$

$\tau = NT \quad N = 2M$

ii) the measurement noise $w(k)$ is equal to zero for all $k$ and

iii) the parameters are constant, then the right hand side of (1) is the Fourier representation of the signal $y(k)$. For the estimation of the time varying parameters various methods can be used. The classical short time Fourier analysis using either block transforms or discrete running Fourier series methods is strongly noise dependent and does not allow an arbitrary selection of frequencies $\omega_i$. That's why in [1] a random walk model was chosen for the variation of the parameters and then the Kalman filter theory has been applied to solve the estimation problem, resulting in an optimal filter with periodic coefficients. Another possibility is to assume an underlying oscillator model leading to a different Kalman filter structure [2]. Comparisons of the two above methods and the Fourier transform method are given in [3]. Here a recursive least square method with a constant exponential weighting factor $\lambda$ is used for the estimation of the time varying parameters. Fig. 1 shows the signal processing scheme for the estimation of the amplitudes and phases. In a first stage the coefficients $a(k)$ and $b(k)$ are estimated by a linear estimator. In a second stage the estimates of the amplitudes $A(k)$ and the phases $\psi(k)$ are determined by a nonlinear transformation:

$$\hat{A}(k) = \{ \hat{a}_1(k)^2 + \hat{b}_1(k)^2 \}$$

$$\hat{\psi}(k) = \arctan(\hat{b}_1(k) / \hat{a}_1(k))$$

$$y(k) \rightarrow \hat{a}_1(k) \rightarrow \hat{b}_1(k) \rightarrow \hat{A}(k) \rightarrow \hat{\psi}(k)$$

Fig. 1: Estimation scheme

2. DERIVATION OF THE FILTER EQUATIONS

The performance index to be minimized is given by [4]

$$J(k, \lambda, x(k)) = \sum_{i=1}^{N} \left[ \frac{1}{2} \lambda^k - \lambda \right] a_i^2(k) x_i^2(k) + \lambda^k J_0$$

$$J_0 = \sum_{i=1}^{N} \left[ \frac{1}{2} \lambda^k - \lambda \right] c_i^2(k) x_i^2(k) + \lambda^k J_0$$

$c_i^2(k) = \{ 1, \cos(\omega_i k T), \sin(\omega_i k T), \ldots \}$

$x_i(k) = \{ a_i(k), b_i(k), h_i(k), \ldots \}$

To simplify the expressions we concentrate here on one frequency only and set $x_1(k) = a_1(k), x_2(k) = b_1(k)$
Then $J(k,\lambda,\xi(k))$ is given by
\[ J(k,\lambda,\xi(k)) = \sum_{i=1}^{k} \left[ \frac{\lambda^k}{2} \left( y(i) - x_1(i) \cos(\omega_i T) - x_2(i) \sin(\omega_i T) \right)^2 + \lambda^k J_0 \right] \]
(2)

The recursive solution for the estimates $\hat{\xi}(k)$ minimizing (2) can be written as
\[ \hat{\xi}(k) = \hat{\xi}(k-1) + K(k)(y(k) - \hat{c}'(k)\hat{\xi}(k-1)) \]
(3)
\[ K(k) = \frac{P(k-1)c'(k)[I + c'(k)P(k-1)c(k)]^{-1}}{P(k-1) - K(k)c'(k)P(k-1)} \]
(4)
\[ P(k) = \lambda^{-1}(P(k-1) - K(k)c'(k)P(k-1)) \]
(5)
\[ \hat{\xi}(0) = \hat{\xi}_0 \quad P(0) = P_0 \quad \text{positive definite} \quad 0 < \lambda < 1 \]

If the least square method is formulated in a stochastic way, $P(k)$ corresponds to the symmetric error covariance matrix $P(k) = \text{Var}[\hat{\xi}(k) - \hat{\xi}(k)]$.

3. DIFFERENCE EQUATIONS FOR $P(k)$

By introducing the new variables $Z(k) = p_1(k)c_1(k+1)^2 + p_2(k)c_2(k+1)^2$
\[ + 2p_2(k)c_1(k+1)c_2(k+1) \]
(6)
$D(k) = p_1(k)p_2(k) - p_3(k)^2$
(7)

one can rewrite equations (4) and (5) in the following way:
\[ p_1(k+1) = \lambda^{-1}[D(k)c_2(k+1)^2 + rp_1(k)]/Z(k) \]
(8)
\[ p_2(k+1) = \lambda^{-1}[D(k)c_1(k+1)^2 + rp_2(k)]/Z(k) \]
(9)
\[ p_3(k+1) = [rp_2(k) - D(k)c_2(k+1)c_1(k+1)]/\lambda S(k) \]
(10)

This represents a set of three nonlinear time varying difference equations. The solutions can be determined independent of the data $y(k)$. The new variable $D(k)$ is the determinant of the positive definite matrix $P(k)$ and is therefore $>0$, and $Z(k)$, the denominator of equation (5), is a quadratic function of $\hat{c}(k)$ and thus also $>0$ for $\lambda > 0$. Fig. 2 shows the behaviour of the elements $p_1(k), p_2(k)$ and $p_3(k)$ of $P(k)$ for the case $N = 20$, $\lambda = 0.9$ and $p_1(0) = p_2(0) = 0.1$ and $p_3(0) = 0$.

Due to the periodicity in $c(k)$, periodic solutions are obtained for $P(k)$ in the steady state. For $\lambda$ values close to 1 (e.g. $\lambda = 0.99$), the steady state is much slower than for smaller $\lambda$ values (e.g. $\lambda = 0.9$). In the limiting case $\lambda = 1$ all $p_1(k)$ values tend to zero for $k \rightarrow \infty$, which means that the gain $K(k)$ tends towards zero. In that case no adaptation would be possible. The periodic solutions of $P(k)$ are sinusoidal with an offset for $p_1(k)$ and $p_2(k)$ and frequency $2\omega_0$ (double frequency of $c(k)$ frequency content) and with a $180^\circ$ phase shift between $p_1(k)$ and $p_2(k)$ and a $+90^\circ$ phase shift between $p_3(k)$ (or $p_2(k)$) and $P(k)$.

Beside $\hat{\xi}(k)$ and $D(k)$ the following two functions of $P(k)$ are of interest as can be seen later
\[ T(k) = p_1(k) + p_2(k) = \text{trace of } P(k) \]
(11)
\[ S(k) = [c_2(k+1)^2 - c_2(k)^2]p_3(k) + c_1(k+1)c_2(k+1) \]
(12)

The steady state of the variables $Z(k), T(k)$ and $S(k)$ are constant and denoted by $Z^*$, $T^*$, $S^*$ and $T^0$. The values of Table 1 and 2 were found by simulations. They are a function of $N$ and $\lambda$. It will be shown in part 4 that these steady state values can be derived by solving a system of algebraic equations. The maximum of $p_1$ and $p_2$ is a function of $N$ and $\lambda$ too. It can easily be computed from the steady state values $D^*$ and $T^0$. From equations (7) and (11) follows
\[ p_1(k) = T^0/2 + \sqrt{(T^0/2)^2 - D^0 - p_3(k)^2} \]
(13)

Equation (13) clearly has a maximum if $p_1(k) = 0$, that means
\[ P_{\text{max}} = T^0/2 + \sqrt{(T^0/2)^2 - D^0} \]
(14)

Table 1 shows that for a given $N$ the denominator $Z^*$ decreases with increasing $\lambda$ with a limit $Z^* = 1$ for $\lambda = 1$, whereas the other variables $T^0$, $D^*$ and $S^*$ decrease towards zero. Table 2 shows the dependence of the steady states on the number of samples $N$ per period. It is interes-
\[
\lambda \quad Z^0 \quad T^0 \quad D^0 \quad S^0 \quad P_{\lambda\max}
\]
\[
0.5 \quad 4. \quad 9.236 \quad 9.236 \quad 3.078 \quad 8.095
0.8 \quad 1.562 \quad 1.631 \quad 0.283 \quad 0.192 \quad 0.758
0.9 \quad 1.235 \quad 1.057 \quad 0.051 \quad 0.038 \quad 0.267
0.95 \quad 1.108 \quad 0.212 \quad 0.011 \quad 0.009 \quad 0.112
0.99 \quad 1.022 \quad 0.043 \quad 0.0005 \quad 0.0003 \quad 0.026
1 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0
\]

Table 1: Steady state values of \(Z, T, D\) and \(S\) and \(P_{\lambda\max}\) for different \(\lambda\) values and fixed \(N (N=20)\) and \(r (r=1)\)

\[
N \quad Z^0 \quad T^0 \quad D^0 \quad S^0 \quad P_{\lambda\max}
\]
\[
4 \quad 1.235 \quad 0.446 \quad 0.050 \quad 0.005 \quad 0.236
8 \quad 1.235 \quad 0.447 \quad 0.050 \quad 0.012 \quad 0.239
16 \quad 1.235 \quad 0.453 \quad 0.050 \quad 0.030 \quad 0.258
32 \quad 1.235 \quad 0.477 \quad 0.053 \quad 0.062 \quad 0.301
64 \quad 1.235 \quad 0.573 \quad 0.064 \quad 0.125 \quad 0.422
128 \quad 1.235 \quad 0.957 \quad 0.106 \quad 0.251 \quad 0.829
256 \quad 1.235 \quad 2.494 \quad 0.277 \quad 0.503 \quad 2.378
512 \quad 1.235 \quad 8.643 \quad 0.960 \quad 1.06 \quad 5.350
1024 \quad 1.235 \quad 33.236 \quad 3.693 \quad 2.012 \quad 33.124
\]

Table 2: Steady state values of \(Z, T, D\) and \(S\) and \(P_{\lambda\max}\) for different \(N\) values and fixed \(\lambda (\lambda=0.9)\) and \(r (r=1)\)

When \(T^0, D^0\) and \(S^0\) are increasing with increasing \(N\).

4. DIFFERENCE EQUATIONS FOR \(D(k), T(k), Z(k)\) AND \(S(k)\)

Writing down the difference equations for the new variables \(T(k), D(k), S(k)\) and \(Z(k)\) yields after some tedious manipulations with \(\varphi=2\pi/N\)

\[
T(k+1)=\lambda^{-1}[D(k)+rT(k)]/Z(k) \quad (15)
\]

\[
D(k+1)=\lambda^{-2}rD(k)/Z(k) \quad (16)
\]

\[
Z(k+1)=\lambda^{-1}[D(k)+rT(k)][\sin^2\varphi+r(Z(k)-r)]^2
\]

\[
S(k+1)=\lambda^{-1}[(0.5[D(k)+rT(k)]-2r(Z(k)-r)]
\]

\[
\sin^2\varphi+rS(k)[\cos^2\varphi]/Z(k) \quad \text{(17)}
\]

It is interesting to note that the above system of nonlinear difference equations is time invariant. The steady state of this system correspond to the constant values \(Z^0, T^0, D^0\) and \(S^0\). These values can be found by solving the following algebraic equations:

\[
T^0=\lambda^{-1}[D^0+rT^0]/Z^0 \quad (19)
\]

\[
D^0=\lambda^{-2}rD^0/Z^0 \quad (20)
\]

\[
Z^0=\lambda^{-1}[(D^0+rT^0)\sin^2\varphi+r(Z^0-r)\cos^2\varphi]
\]

\[
+rS^0[\sin^2\varphi+rZ^0]/Z^0 \quad (21)
\]

\[
S^0=\lambda^{-1}[(0.5[D^0+rT^0]-2r(Z^0-r)]\sin^2\varphi
\]

Although the equations are nonlinear they can be solved analytically. If a Kalman filter is used for the estimation the corresponding system of nonlinear equations (19)-(22) cannot be solved analytically. From equation (20) follows

\[
Z^0=r/\lambda^2 \quad (23)
\]

which is indeed independent on \(N\) or \(\varphi\). Equation (19) yields with (23)

\[
D^0=(1-\lambda)T^0/\lambda \quad (24)
\]

If \(D^0\) and \(Z^0\) are plugged into equations (21) and (22) one obtains the following system of linear equations:

\[
A \begin{bmatrix} S^0 \\ T^0 \end{bmatrix} = b; \quad A = \begin{bmatrix} \sin^2\varphi & \sin^2\varphi \\ \lambda^2 \lambda \end{bmatrix}
\]

\[
\begin{bmatrix} r \left(1-\lambda^2\right) & 0 \end{bmatrix} \begin{bmatrix} \sin^2\varphi \\ \sin^2\varphi \end{bmatrix}
\]

\[
A \begin{bmatrix} \sin^2\varphi \\ \sin^2\varphi \end{bmatrix} = b = \begin{bmatrix} r \left(1-\lambda^2\right) \cos^2\varphi \\ r \left(1-\lambda^2\right) \cos^2\varphi \end{bmatrix}
\]

This can be solved leading to

\[
S^0=r\left(1-\lambda^2\right)^2 \frac{1}{\lambda^2} \quad (25)
\]

\[
T^0=r(1-\lambda)(1+\lambda^2-2\lambda^2-\cos^2\varphi)/\sin^2\varphi \quad (26)
\]

Equations (23)-(26) represent a complete set of solutions. All steady states are linearly dependent on \(r\).

5. THE RESULTING FILTER

The definition equations of \(S(k), T(k)\) and \(Z(k)\) can be put into the following form:

\[
\begin{bmatrix} 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} p_1(k) \\ p_2(k) \\ p_3(k) \end{bmatrix} = \begin{bmatrix} T(k) \\ S(k) \\ Z(k)-r \end{bmatrix}
\]

\[
\begin{bmatrix} -\alpha & \alpha & \beta-\alpha \\ \beta & \gamma & 2\alpha \\ \gamma & -2\alpha & \beta-\gamma \end{bmatrix} \begin{bmatrix} p_1(k) \\ p_2(k) \\ p_3(k) \end{bmatrix} = \begin{bmatrix} T(k) \\ S(k) \\ Z(k)-r \end{bmatrix}
\]

with \(\alpha=c_1(k)c_2(k)=0.5\sin 2k\varphi\)

\(\beta=c_1(k)^2=\cos^2 k\varphi; \gamma=c_2(k)^2=\sin^2 k\varphi\)

From this we get the element of \(P(k)\):

\[
\begin{bmatrix} p_1(k) \\ p_2(k) \\ p_3(k) \end{bmatrix} = \begin{bmatrix} \gamma & -2\alpha & \beta-\gamma \\ \beta & 2\alpha & \gamma-\beta \\ \gamma & -2\alpha & \beta-\gamma \end{bmatrix} \begin{bmatrix} T(k) \\ S(k) \\ Z(k)-r \end{bmatrix}
\]

The gain vector \(K(k)\) for the correction of the estimates \(x(k)\) are obtained from \(P(k)\) leading to time varying gains. The gains can be computed offline. They are given with equations (4) and (27) by

\[
K_1(k)=-\sin k\varphi(S(k)/Z(k))+\cos k\varphi(1-r/Z(k))
\]
\[ K_1(k) = \cos(\phi) \frac{S(k)}{Z(k)} + \sin(\phi)(1 - \frac{r}{Z(k)}) \]

Fig. 3: Gain vector \( K(k) \) \( [N=20, \lambda=1] \)

Fig. 3 shows the gains for the example mentioned above. In the steady state the gains reduce to the simple forms:

\[ K_1(k) = -\sin(\phi)(1-\lambda)^2 \cot\phi + \cos(\phi)(1-\lambda^2) \]
\[ K_2(k) = \cos(\phi)(1-\lambda)^2 \cot\phi + \sin(\phi)(1-\lambda^2) \]

It is interesting to note that the steady state filter is independent on \( r \). Fig 4 shows the resulting steady state filter structure for the fundamental harmonic component. The gain is split into a harmonic part (I) and a time-varying part (II) whose coefficients converge to the above constant values in the steady state.

![Filter structure diagram](image)

Fig. 4: Filter structure of the recursive least square estimator with exponential weighting factor

The resulting estimator structure is similar to the one that is obtained if a Kalman filter estimator based on a random walk model is used (see [1]). It is also interesting that the structure is related to the quadrature demodulation scheme of Fig. 5.

The difference of the quadrature demodulator are:

1) the missing time varying feedback

![Discrete quadrature demodulator](image)

Fig. 5: discrete quadrature demodulator front end

- loop generating the error signal \( y(k) \)
- the links in the forward signal \( y(k) \)
- summation over a finite path between the sin and cos path

3) sum over the determination of the estimates.

6. CONCLUSION

An estimator has been derived that allows to estimate on line the time varying amplitude and phase of a sinusoid of given frequency in the presence of other sinusoidal disturbances with known frequencies and noise. It is based on the recursive least square method with exponential weighting factor. The method can be extended to the determination of a higher order filter for the estimation of time-varying amplitudes and phases of several sinusoids of known frequencies. Analytical results however are not yet available for that case. The structure of the optimal filter has some interesting properties related to Kalman filters and quadrature demodulators.

7. ACKNOWLEDGEMENT

The author likes to thank J. Tödtli for his valuable discussions.

[1] F. Gruber, J. Tödtli: A recursive estimator for the determination of unknown constant or slowly varying Fourier coefficients, EUSIPCO 88, p.1433-1435, Grenoble
Doubly Normalized Schur RLS Lattice Filter and CORDIC Array Architecture
Peter Strobach
Siemens AG, Zentralabteilung Forschung und Entwicklung
ZFE ST SN 6f, Otto-Hahn-Ring 6, W-8000 Munchen 83, Germany

Abstract - This paper presents the following novel results in Schur RLS adaptive filtering: (1) It is shown that both boundary and internal cells of the square-root Schur RLS adaptive filter array can be realized with the same elementary CORDIC cell. (2) This CORDIC cell can be used additionally in a second order moment estimation section based on recursive Gray-Markel lattice filters. (3) The structure of the square-root Schur RLS adaptive filter is invariant with respect to an additional input power normalization. (4) All internal variables of the doubly (input and residual) normalized algorithm are scaled to a value of less or equal one in magnitude and fully utilize this interval in successive stages due to the autoscaling feature of hyperbolic rotations. These important properties make the presented CORDIC array of the doubly normalized Schur RLS adaptive filter an ideal candidate for a systolic VLSI implementation.

I. INTRODUCTION
Schur algorithms have been a subject of widespread interest in signal processing [1-8]. Their square-root forms have the key feature that they provide a Cholesky factorization of an underlying covariance matrix by local hyperbolic rotations and hence are the natural counterparts of QR-based algorithms which provide a Cholesky factorization by local circular rotations. Recent investigations [6-8] have shown that the square-root Schur approach extends to the recursive least squares (RLS) adaptive filtering case. The resulting square-root Schur RLS adaptive filter [7] has outstanding numerical and structural properties. It provides an efficient Cholesky factorization of the (non-Toeplitz) data covariance matrix and facilitates an easy incorporation of smooth finite or infinite forgetting functions in the RLS time updating process. Thus Schur RLS adaptive filters can outperform other types of RLS adaptive filters which are restricted on the application of simple exponential or rectangular forgetting functions, or simple modifications thereof.

It is the purpose of this paper to note some further properties of the square-root Schur RLS adaptive filter presented in [7]. First, we want to show that even in the true nonstationary case, the boundary cell of the square-root Schur RLS adaptive filter array of [7] can be computed with the same hyperbolic CORDIC cell as is used for internal cells. The resulting pure CORDIC structure suggests that the second order momentum inputs of the triangular array may also be computed with a CORDIC based structure. We show how this can be accomplished on the basis of recursive Gray-Markel lattice filters. Finally, we discovered the particularly useful property that the square-root Schur RLS adaptive filter is invariant with respect to input power normalization. In case of input power normalization, all internal variables of the algorithm remain bounded in the unit interval and fully utilize it in successive stages due to a nice autoscaling feature of the involved hyperbolic rotations. These properties have a significant impact on the VLSI realization of the algorithm. It is conjectured that the presented structures can become a useful part of future high-speed adaptive equalizer/demodulator circuits.

II. THE SQUARE-ROOT SCHUR RLS ADAPTIVE FILTER
The square-root Schur RLS adaptive filter [7] is based on the feed-forward square-root lattice form, a cascaded structure of hyperbolic lattice rotations,
\[
\begin{bmatrix}
\mathbf{v}_m(t) \\
\mathbf{w}_m(t)
\end{bmatrix} \leftarrow \Theta_m(t) \begin{bmatrix}
\mathbf{v}_{m-1}(t) \\
\mathbf{w}_{m-1}(t)
\end{bmatrix},
\]
where \(\Theta_m(t)\) is the square-root lattice rotor,
\[
\Theta_m(t) = \begin{bmatrix}
1 - \alpha_m(t) & 1 \\
\alpha_m(t) & 1
\end{bmatrix}^{-1/2} \begin{bmatrix}
1 & \rho_m(t) \\
\rho_m(t) & 1
\end{bmatrix},
\]
and
\[
\mathbf{v}_m(t) = \mathbf{e}_m(t) \mathbf{v}_{m-1}(t),
\]
\[
\mathbf{w}_m(t) = \mathbf{r}_m(t) \mathbf{w}_{m-1}(t),
\]
are normalized residual vectors with initial condition \(\mathbf{e}_0(t) = \mathbf{r}_0(t) = \mathbf{x}(0)\), where \(\mathbf{x}(t) = [x(t), x(t-1), x(t-2), \ldots ]\)
is a sliding or growing window observation of \(x(t)\). Note that \(\mathbf{R}_m(t)\) and \(\mathbf{R}_m(t)\) are forward/backward residual energies:
\[
\mathbf{E}_m(t) = \mathbf{e}_m(t)^T \mathbf{e}_m(t),
\]
\[
\mathbf{R}_m(t) = \mathbf{r}_m(t)^T \mathbf{r}_m(t).
\]

Thus, by definition, \(\mathbf{v}_m(t)^T \mathbf{v}_m(t) = 1\) and the same holds for \(\mathbf{w}_m(t)^T \mathbf{w}_m(t)\). The interesting point with \(\Theta_m(t)\) is that this rotor is J-unitary, i.e., \(\Theta_m(t)\) represents a standard hyperbolic rotation and thus can be implemented easily by using standard vector rotation hardware, such as Coordinate Rotation Digital Computer (CORDIC) processors [9].

The square-root lattice rotor is uniquely determined by the square-root lattice parameter \(\rho_m(t)\),
\[
\rho_{m-1}(t) = -\frac{\mathbf{v}_m(t)^T \mathbf{w}_m(t)}{\mathbf{w}_m(t)^T \mathbf{w}_m(t-1)},
\]
which has also the physical meaning of a reflections coefficient in scattering theory. This physical interpretation suggests that we must have \(|\rho_m(t)| \leq 1\). An order recursion of \(\rho_m(t)\) can be defined on the basis of square-root generalized covariances (inner products of shifted data and normalized residual vectors):
\[
\rho^f_{m-1}(t) = x^T(t-\beta) \mathbf{v}_m(t),
\]
\[
\rho^b_{m-1}(t) = x^T(t-\beta) \mathbf{w}_m(t).
\]
The key result in square-root Schur RLS adaptive filtering is that an order recursion of the square-root generalized covariances exists and this recursion has exactly the structure of the square-root lattice filter:
\[ \begin{bmatrix} D_{m,j}^f(t) \\ D_{m,j}^b(t) \\ \rho_{m,j}^f(t) \\ \rho_{m,j}^b(t) \end{bmatrix} = \begin{bmatrix} \Theta_m(t) \\ D_{m-1,j}^f(t) \\ D_{m-1,j}^b(t) \end{bmatrix}, \]  

(7)

Next note that from (2), (4), (6) and (6a) it follows that the reflection coefficient can be expressed as the product of the normalized generalized forward covariance of degree \( m \) and the inverse square-root backward residual energy:

\[ \rho_{m,j}^f(t) = \sqrt{T_m(t)} \omega_m(t-1) - D^f_{m-1,j}(t) H_{m-1,j}(t-1) \]  

(8)

In order to find a recursion for \( R_m(t) \) one exploits a result from linear prediction theory [5], namely,

\[ R_m(t) = x^T(t-m) \omega_m(t-1) - x^T(t-m) \omega_m(t) R_m^{1/2}(t) \]  

(9)

Thus it is easy to check [comparison with (8b)] that

\[ R_m^{1/2}(t) = x^T(t-m) \omega_m(t) = D_{m,m}^b(t) \]  

(10)

Strictly speaking, expression (10) reveals that the square-root of the backward residual energy is identical with the normalized generalized backward covariance of order \( m \) and degree \( m \). Thus we can ultimately use recursion (7) for a computation of \( R_m^{1/2}(t) \), or equivalently, \( D_{m,m}^b(t) \) as follows:

\[ \begin{bmatrix} D_{m,m}^f(t) \\ D_{m,m}^b(t) \end{bmatrix} = \begin{bmatrix} \Theta_m(t) \\ D_{m-1,m}^f(t) \end{bmatrix}. \]  

(11)

Even more notable is the role of the quantity \( D_{m,m}^f(t) \), which appears as a by-product in the computation of \( D_{m,m}^b(t) \) via (10). From (10) and (8) it follows that

\[ \rho_{m}(t) = - D^f_{m-1,m}(t) / D^b_{m-1,m}(t-1). \]  

(12)

Thus an evaluation of the upper equation in (10) gives

\[ D_{m,m}^f(t) = 0 \]  

(13)

for all practical values of \( \rho_{m}(t) \). Hence one may employ a singleCORDIC processor for annihilation of the top component \( D_{m,m}^f(t) \) by iterative hyperbolic rotation (vectoring mode) according to

\[ \begin{bmatrix} D_{m,m}^f(t) \\ D_{m,m}^b(t) \end{bmatrix} \rightarrow \begin{bmatrix} \Theta_m(t) \\ D_{m-1,m}^f(t) \end{bmatrix}. \]  

(14)

As soon as \( D_{m,m}^f(t) \) is sufficiently close to zero one obtains the desired value of \( D_{m,m}^b(t) \) at the second output of the CORDIC processor. Most notable is the fact that this operation additionally produces the reflection coefficient \( \rho_m(t) \) implicitly as a "by-product" in terms of the resulting "angular variable" \( \varphi_m(t) \) because

\[ \sin \varphi_m(t) = \rho_m(t) \left[ 1 - \rho^2_m(t) \right]^{-1/2}, \]  

(54a)

\[ \cos \varphi_m(t) = \left[ 1 - \rho^2_m(t) \right]^{1/2}. \]  

(54b)

The recursions of the square-root Schur RLS adaptive filter can be summarized as follows:

1. **Boundary Cell:** Compute \( \rho_m(t) \) and \( D_{m,m}^b(t) \) via (14) [CORDIC processor in hyperbolic vectoring mode. See Fig. 1].
2. **Internal Cell:** Compute order recursion of \( D_{m,i}^f(t) \) and \( D_{m,i}^b(t) \) via (7) [CORDIC processor in hyperbolic rotation mode. See Fig. 1].
3. **Filter Cell:** Compute adaptive square-root lattice filter via (14,b) [CORDIC processor in hyperbolic rotation mode. See Fig. 1].

III. **Input Normalization**

The recursions of the square-root Schur RLS adaptive filter are initialized from the square-root normalized second-order information of an observed process. This information has a compressed dynamic range, yet it is not bounded in the unit interval. In this section, we demonstrate that power normalization can be applied on the shifted data vectors \( x(t) \), \( x(t-j) \), \ldots, \( x(t-m) \). In a geometrical interpretation the power normalized second-order information is just the set of cosines of the angles between the shifted data vectors and is hence bounded in the unit interval. It is shown that the recursions of the square-root Schur RLS adaptive filter are invariant to input power normalization. Further it is shown that all internal variables of the square-root Schur RLS adaptive filter remain bounded in the unit interval when initialized from the power normalized second-order information.

Consider the power normalized data vectors

\[ \tilde{x}(t-j) = x(t-j) / c_{j,t}^{1/2} \]  

(65)

Clearly,

\[ \tilde{x}^T(t-j) \tilde{x}(t-j) = 1 \]  

(66)

A quick inspection of (7) shows that the power normalized second-order information [short term autocorrelation coefficients] \( \tilde{E}(t) \) are just the cosines of the angles between shifted data vectors. Moreover, one can show that with this type of normalization, on obtains normalized residual vectors of the type

\[ \tilde{\omega}_m(t) = - \omega_m(t) c_{t,m}^{1/2} \]  

(68a)

\[ \tilde{R}_m(t) = R_m(t) c_{t,m} \]  

(68b)

Thus:

\[ E_m(t) = E_m(t) c_{t,m}^{1/2} \]  

(69a)

\[ R_m(t) = R_m(t) c_{t,m} \]  

(69b)

With these normalized quantities one finds that:

\[ \tilde{E}_m(t) = \tilde{E}_m(t) \tilde{E}_{m-1}^{1/2} = \tilde{E}_m(t). \]  

(20a)

\[ \tilde{W}_m(t) = \tilde{W}_m(t) \tilde{W}_{m-1}^{1/2} = \tilde{W}_m(t). \]  

(20b)

Strictly speaking, the square-root normalized residual vectors remain unaffected by additional power normalization of the shifted data vectors. Moreover:

\[ \hat{\omega}_m(t) = - \tilde{E}_m(t) \tilde{W}_m(t) = - \tilde{E}_m(t)^2 \]  

(21)

which means that the same statement holds for the reflection coefficients.

Finally it remains to determine the dynamic range requirements of the square-root normalized covariances when initialized from the time varying cosines \( c(t) \). A normalization of the generalized covariance in consideration of (20a,b) gives:

\[ \tilde{E}_{m,j}^f(t) = \tilde{x}^T(t-j) \tilde{E}_m(t) \tilde{x}(t-j) \]  

(22a)

\[ \tilde{E}_{m,j}^b(t) = \tilde{x}^T(t-j) \tilde{W}_m(t) \tilde{x}(t-j) \]  

(22b)

Again inner products of unit energy vectors are always less or equal one in magnitude and hence:

\[ | \tilde{E}_{m,j}^f(t) | \leq 1, \]  

(23a)

\[ | \tilde{E}_{m,j}^b(t) | \leq 1. \]  

(23b)

Thus we have shown that all internal variables of the square-root Schur RLS adaptive filter remain bounded in the unit interval when only the square-root generalized covariances are initialized from the power normalized second-order information (time varying cosines) of the observed process.
\( \hat{D}_{Q,m}(t) = \hat{D}_{Q,m}^{(j)}(t) = \xi_j(t), \quad 1 \leq j \leq N. \)   

IV. ESTIMATION OF SECOND-ORDER INFORMATION

The second-order information \( \xi_j(t), 0 \leq j \leq N \) can be expressed as a weighted inner product of shifted data vectors:

\[ \xi_j(t) = x^T(t)Wx(t-j); \quad W = \text{diag}[w_0, w_1, w_2, \ldots]. \]  

Usually this weighted inner product is updated recursively. Such recursions typically exhibit the structure of recursive low-pass filters with the sample products \( x(t)x(t-j) \) as input and the "cumulative" second-order information as output. Particularly interesting cases arise with recursive filters with transfer function

\[ H(z) = \text{const.} \frac{z}{(z-\alpha)^n}; \quad 0 < \alpha < 1. \]

Such systems have \( n \)-fold poles at \( z = \alpha \) on the real axis in the \( z \)-plane. They can be realized by the following order \( n \) cascaded Gray-Markov feed-back lattice filter \([10]\), with instantaneous autocorrelation coefficients \( x(t)x(t-j) = P_j \) as input and the estimated second-order information \( \hat{\xi}_j(t) = P_j(t) - q_j(t) \) as output:

\[
\begin{bmatrix}
    \hat{q}_j^{(0)}(t) \\
    \hat{q}_j^{(1)}(t) \\
    \vdots \\
    \hat{q}_j^{(n-1)}(t)
\end{bmatrix} = 
\begin{bmatrix}
    1 - K_m & 0 & \cdots & 0 \\
    K_m & 1 - K_m & \cdots & 0 \\
    0 & K_m & \cdots & 1 - K_m \\
    \vdots & \vdots & \ddots & \vdots \\
    0 & 0 & \cdots & 1 - K_m \\
\end{bmatrix}
\begin{bmatrix}
    \hat{q}^{(0)}(t) \\
    \hat{q}^{(1)}(t) \\
    \vdots \\
    \hat{q}^{(n-1)}(t)
\end{bmatrix} \quad (27)
\]

where \( K_m, 1 \leq m \leq n \) are the fixed parameters of the Gray-Markov lattice filter of order \( n \) which are determined such that the overall filter has the desired transfer function \([26]\). Clearly, the classical exponential forgetting function drops out such that the special case \( n = 1 \) and the second-order (or modified Busswell) window \([6]\) is just the special case of an order \( n = 2 \) filter.

The most interesting point with this type of recursive forgetting is in fact the observation that the Gray-Markov lattice recursion \([27]\) is just a circular rotation. Thus this type of recursive lattice filter can be implemented by a cascade of CORDIC processors performing circular rotations as described in \([11]\). This CORDIC-based estimator of the second-order information fits very well into the overall CORDIC array structure of the square-root Schur RLS adaptive filter.

V. SYSTOLIC ARRAY

Fig. 1 illustrates the described hyperbolic rotation and hyperbolic vectoring modes of the CORDIC elementary cell which is used in the adaptation and in the filter sections of the square-root Schur RLS adaptive filter. Fig. 2 shows the CORDIC array architecture of the parameter adaptation and adaptive filter sections. The triangular part of the array (parameter adaptation section) computes the angles \( \phi_0(t) - \phi_1(t) \) from the second order information (time varying cosines) provided at the inputs. The angles \( \phi_0(t) - \phi_1(t) \) are subsequently transmitted to the adaptive filter section at the bottom of the triangular array. In the case of a time invariant second-order information, the proposed algorithm and structures reduce to the Toeplitz system solver of Jou, Hu and Peng \([22]\).

It becomes apparent that in an all-CORDIC realization of the square-root Schur RLS adaptive filter, one prefers to operate with the angular variable \( \phi_0(t) \) rather than with the explicit reflection coefficient \( \rho(t) \). Once the angular variable has been determined by a CORDIC processor in vectoring mode, it is fed through all rotational processors in column \( m \) of the array and the angular variable input of these processors is preset by \( \phi_0(t) \) to obtain a prescribed rotation by iterative annihilation of the angular variable (rotational mode) \([9]\).

VI. CONCLUSIONS

A doubly normalized Schur RLS adaptive filter and its fully systolic structure have been presented. The algorithm is very amenable to a VLSI realization because it consist of a mere locally connected CORDIC cells. Moreover, it has been shown that the algorithm has a particularly useful auto-scaling feature in that the hyperbolic rotators always fully utilize the given dynamic range. Thus it is conjectured that the proposed structure can become a useful technique in high-speed data equalization and adaptive receiver applications. But we may not put our hopes in the future ahead of a pure digital implementation of this array. Since all elementary cells perform passive or J-passive operations the computations could probably been mapped on even more exciting structures with light or other electromagnetic waves, depending on the medium chosen for implementation (e.g. fiber optics, surface acoustic wave (SAW) devices \([32] \), etc.).

Figure 1: CORDIC processor configurations for rotational and boundary cell operations of the doubly normalized Schur RLS adaptive filter.

References

Figure 2: Division-free systolic all-CORDIC array architecture of the doubly normalized Schur RLS adaptive filter (order N=4 example).
A RLS with an adaptive forgetting factor

Alison Carter, Pascal Ruiz

Techniphone Développement, BP 44, F-13610 PUY STE REPARADE
Phone Number : 42338733 ; Fax : 42618961

We present here a Fast RLS algorithm with an adaptive forgetting factor. The performances are compared with a classical Fast RLS algorithm, on both simulations and real situations. The achievement of the former algorithm is then showed, whereas the calculation cost is still "reasonable".

1. Introduction

Adaptive signal processing is highly conditioned by the performances of the algorithms which minimize an energy function [1,2]. The challenge has always been the compromise between the speed convergence of the algorithm, and if whether or not the solution obtained is the optimum one. Moreover, real time applications has involved the birth of fast algorithms, with other problems of convergence. Nowadays, the LMS and Fast RLS algorithms have been largely used, because of their properties of stability. Therefore, several parameters must be initialized, as the forgetting factor, or the adaptation step. For the LMS, the adaptation step \( \mu \) will influence the convergence speed and the bias of the solution. A weak value of \( \mu \) will be useful near the minimum of the performance surface, whereas a larger value at the beginning of the minimization will accelerate the descent towards the minimum. A solution for improving the algorithm would consist in the calculation of an adaptive \( \mu \). Such a procedure, from a calculation point of view, implies a greater but still reasonable cost since it remains in \( O(N) \) per sample, and not in \( O(N^2) \) (\( N \) is the order of the FIR filter to estimate). P. Bragard [3] has proposed an adaptive optimisation of \( \mu \) in the LMS algorithm. The accuracy of such an algorithm has been proved. Since the RLS algorithm has better performances than the LMS, we propose here an adaptive optimisation of the RLS algorithm. Part 2 will detail the algorithm, and Part 3 will show some results.

2. The adaptive RLS

We present here how we have defined the adaptive RLS algorithm with a gradient procedure, in minimizing the mean square error defined versus the filter coefficients, but also versus the forgetting factor \( w \).

2.1 The RLS algorithm [4,5,6]

Let \( x_n \) be the \( n \)th sample of the observed signal (primary input), and \( X_n \) the \( N \)-dimension observation vector. The purpose of the RLS algorithm is to minimize
\[ J_d(n) = \sum_{p=0}^{n} (e_p)^2, \text{ where } e_p = s_p - H_{p-1}X_{p-1}. \]

The RLS algorithm estimates \( H_p \) in a recursive way:
\[ H_p = H_{p-1} + R^{-1}(p)X_p(s_p - H_{p-1}X_p), \]
where \( R \) is the estimated covariance matrix.

In a non-stationary case, one can apply a forgetting factor \( \omega < 1 \), and minimize now:
\[ J_d(n) = \sum_{p=0}^{n} \omega^{n-p}(e_p)^2 \]

### 2.2 The Fast RLS (FRLS) algorithm

The FRLS consists in estimating \( R^{-1}(p) \) versus \( R^{-1}(p-1) \), among the determination of the backward and forward prediction coefficients \( B_n^t \) and \( F_n^t \) which minimize a function of \( E_f(n) \) and \( E_b(n) \), where:
\[ E_f(n) = \sum_{p=1}^{n} \omega^{n-p} \left| x_p - F_n^t X_{p-1} \right|^2 \]
\[ E_b(n) = \sum_{p=1}^{n} \omega^{n-p} \left| x_{p-N} - B_n^t X_p \right|^2 \]

Our problem is now reduced in estimating the backward and forward prediction coefficients \( B_n^t \) and \( F_n^t \) which minimize a function \( J(E_f, E_b) \) of \( E_f(n) \) and \( E_b(n) \). Noticing that \( J(E_f, E_b) \) is also a function of \( \omega \), our purpose is here to make \( \omega \) be dependent of \( n \), and to update it in a recursive way.

### 2.3 The adaptive FRLS

In that algorithm, the forgetting factor \( \omega \) will be dependent of \( n \) : the way of updating it would be:
\[ \omega_{n+1} = \omega_n - \alpha \frac{\partial J(E_f, E_b)}{\partial \omega_n}, \]
where \( J(E_f, E_b) \) is a function of the energies (we take here \( J(E_f, E_b) = E_f + E_b \)), and where \( \alpha \) is the second adaptive step. From the previous formula we obtain:
\[ \omega_{n+1} = \omega_n - \alpha \left( \sum_{p=1}^{n} (n-p) \omega^{n-p-1} \left( \left| x_p - F_n^t X_{p-1} \right|^2 + \left| x_{p-N} - B_n^t X_p \right|^2 \right) \right) \]

#### 2.3.1 The algorithm

The complete FRLS with an adaptive forgetting factor will be:
\[ e_{n+1}^f = x_{n+1} - F_n^t X_n; \text{ forward prediction error} \]
\[ e_{n+1}^b = x_{n+1} - B_n^t X_n; \text{ backward prediction error} \]
\[ E_{n+1}^f = E_n + \omega_{n+1} e_{n+1}^f; \text{ forward prediction coefficients} \]
\[ E_{n+1}^b = E_n + \omega_{n+1} e_{n+1}^b; \text{ forward prediction error (a posteriori)} \]
\[ w_{n+1} = \omega_n - \alpha \left( \sum_{p=1}^{n} (n-p) \omega^{n-p-1} \left( \left| x_p - F_n^t X_{p-1} \right|^2 + \left| x_{p-N} - B_n^t X_p \right|^2 \right) \right) \]
\[ E_{n+1} = E_n + \omega_{n+1} e_{n+1}; \text{ cumulated forward prediction error update} \]
\[ e_{n+1}^b = x_{n+1} - B_n^t X_n; \text{ backward prediction error} \]
\[ G_{n+1} = \frac{M_{n+1} - m_{n+1}}{m_{n+1}}; \text{ Kalman gain} \]
\[ H_{n+1} = H_n + \frac{b}{e_{n+1}}; \text{ backward prediction coefficients update} \]
\[ \hat{y}_{n+1} = y_{n+1} - H_n^t X_{n+1}; \text{ error signal} \]
Because the forgetting factor cannot be negative, a positivity constraint is applied:

\[ w_{n+1} = \frac{w_n + \lambda w_{n-1}}{2} \]

- The choice of \( \alpha \) would be such that \( w_n \) converges between two transient signal, with slow fluctuations. That is quite an empirical choice. We chose it here around 0.001.
- The calculation cost will be around 11 N multiplications/divisions and 10 N additions/subtractions, where N is the order of the filter.

3. Results

3.1. Results on simulated signals

We compare the performances between the FRLS and the FRLS algorithm with a variable forgetting factor. The reference input \( x(n) \) is a sine with a period of 32 points. The primary input is \( y(n) = h(n)*x(n) + b(n) \), where \( h(n) \) is a F.I.R. filter of order 10 which varies slowly, where \( * \) is the convolution operator, and where \( b(n) \) is an additive noise uncorrelated with \( x(n) \). The signal to noise ratio between \( x(n) \) and \( b(n) \) is approximatively 20 dB. In figure 1 as in figure 2 we represented the ratio

\[ \frac{E([y(n) - h(n)*x(n)]^2)}{E(y^2(n))} \]

for the LMS algorithms (\( \mu = 0.02 \)), the FRLS algorithm (\( w = 0.98 \)), and the FRLS with a variable forgetting factor (\( w_0 = 0.98 \) and \( a = 10^{-2} \)). The figures show the accuracy of the FRLS with a variable forgetting factor, compared with the FRLS.

3.2 Results on real situations

We compare here the performances between the FRLS, LMS, and our FRLS with a variable forgetting factor. The algorithms are applied on real signals: the reference input (figure 3) is a motor noise, the primary input is the motor noise (figure 4) inside a cabin, plus an uncorrelated desired response. In figure 5 we represent the input versus

\[ \sum_{i=1}^{n} y^2(i) \]

and the residual output \( \sum_{i=1}^{n} (y(i) - h(i)*x(i))^2 \) for the different algorithms. The performances of the adaptive FRLS are still better than the other algorithms.

References:

A FOURTH ORDER CUMULANT BASED ADAPTIVE ALGORITHM FOR
THE IDENTIFICATION OF NONLINEAR HAMMERSTEIN MODELS

Deergha Rao Korrai and Devender C Reddy

R & T Unit for Navigational Electronics, Osmania University
Hyderabad-500007, India.

An adaptive algorithm known as ORIV and based on HOS has been proposed [1] for
the identification of IIR systems. Although powerful, yet nevertheless the method has the
shortcoming that a proper choice of instrument variables be made, which at times is diffi-
cult. A new on-line technique that uses ORLS in conjunction with RLS is now proposed
for the identification of non-linear Hammerstein models driven by Gaussian input (this is
equivalent to the identification of linear ARMA models driven by non-Gaussian inputs)
which is relatively easy to apply. However, due to paucity of time simulation studies in
respect of an MA process only using ORLS was attempted. The results obtained were
found to be quite satisfactory.

1. INTRODUCTION

Both the theory and practice of adaptive signal processing are almost exclusively based on the second order
statistics of the underlying process. This is because of the implicit assumption that the process is Gauss-
ian. But non-Gaussian processes are quite common [1]. The tendency in contemporary Signal Processing by and
large is to ignore the non-Gaussian nature of underlying signal. However, substantial benefits may accrue by
making use of the non-Gaussian nature of these signals. This has been made possible by the use of higher or-
der order statistics (HOS). Since HOS provide an operational calculus for studying non-Gaussian processes and non-
linear systems, vanish for coloured Gaussian noise and convey additional (phase) information about the under-
lying process they have found direct application in nonlinear filter identification, parameter estimation, speech
and signal processing, phase estimation of linear processes, blind deconvolution, tests for linearity, measur-
ing association, detection and classification [2].

This being the case the search for HOS-based algorithms with improved accuracy continues.

1.1. HOS in the Identification of Non-linear System

As has been mentioned above, HOS seem to be natural candidates to use when one is confronted with non-
Gaussian models. A case in point is the nonlinear Hammerstein model which is composed of a zero memory
nonlinear part $N(\cdot)$, a linear ARMA$(p,q)$ part and driven by Gaussian white noise (see fig. 1). Observe that al-
though the input to the system is Gaussian nevertheless the input to the ARMA part of the system is non-
Gaussian.

One of the important problems in system identification is the determination of the system parameters of such a
model. A noniterative procedure, using HOS of the observed data, has been suggested [3] for the estimation of
parameters of the linear MA$(q)$ part of the Hammerstein model (fig. 1). However, literature seems to be lacking in
adaptive techniques that are based on HOS for the identification of nonlinear, ARMA Hammerstein models.

The object of this article, therefore, is to suggest an adaptive algorithm for the identification of nonlinear
Hammerstein models driven by Gaussian white noise, using fourth order cumulants. Fourth order cumulants are
to be used when the random process is symmetrically distributed [6] for this the third order cumulants
equal zero.

We shall therefore consider such a system in which the third order cumulants are zero but the fourth order
cumulants are non zero and develop an algorithm for parameter identification. Towards this end, as a first step
we shall consider a MA process whose parameters are to be estimated.

2. THE OVERDETERMINED RECUR-
SIVE LEAST SQUARES ALGORITHM FOR
MA PROCESSES

Let the output of the observed process be defined by
the eq: 
\[ y_t = u_t + \sum_{k=1}^{\infty} b_k u_{t-k} \]  

(1)

where 
\( \{u_t\} \) is a zero mean, non-Gaussian white noise sequence 
and \( b_k \)'s are constant coefficients.

For this process, it has been shown [4] that the 

\[ r(m) + \sum_{k=1}^{\infty} b_k r(m-k) = \varepsilon + \sum_{k=1}^{\infty} b_k c(m-k) \]

(2)

where

\[ r(m) = E[y_t y_{t+m}] \]
\[ c(m) = E[y_t y_{t+m}] - 3r(m) r(m) \]
and

\[ \varepsilon = \sigma^2 / \gamma_4 \sigma^2 = E[u_t^2] \]
\[ \gamma_4 = E[u_t^4] - 3 \sigma^4 \]

Eq. (2) may also be written in the more compact matrix

\[ \begin{bmatrix} c(q) & 0 & \cdots & 0 \\ c(q+1) & c(q) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ c(q+q) & c(q+q-1) & \cdots & c(q) \\ 0 & c(q+q) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & c(q) \end{bmatrix} \begin{bmatrix} \varepsilon \\ \varepsilon b_1 \\ \vdots \\ \vdots \\ \vdots \\ \varepsilon b_q \end{bmatrix} = \begin{bmatrix} r(q) \\ r(q-1) \\ \vdots \\ \vdots \\ \vdots \\ r(q) \end{bmatrix} \]

(3)

form:

\[ Y \theta = \varepsilon \]

(4)

where the definition of \( Y, \theta \) and \( \varepsilon \) are obvious from eq. (3).

Eq. (3) represents a set of overdetermined set of equations in the parameter vector \( \theta \).

The least squares solution of this overdetermined system of equations is given by the eq.:

\[ \theta_{LS} = (Y^T Y)^{-1} Y^T \varepsilon \]

(5)

Observe that eq. (5) is not an iterative solution. Friedlander and Porat [1] has proposed an adaptive algorithm, known as ORIV, for solving equation (3). However, this method requires the choice of instrument variables which at times may not be too obvious. We therefore propose a different on-line technique, known as ORLS (overdetermined recursive least squares) for this purpose.
\[ \dot{\theta}(t) = \left[ R^T(t) R(t) \right]^{-1} R^T(t) \sum_{k=1}^{1} z(k) \]  
\tag{10} 

An exponential forgetting factor \( 0 \leq \alpha \leq 1 \) which enables the algorithm to track time varying parameters can be introduced by premultiplying eq.(10) by \( \text{diag} \{ \alpha^t, \alpha^{t-1}, \ldots, \alpha, 1 \} \). In this derivation, the forgetting factor \( \alpha \) is assumed to be unity.

Next let
\[ P_t = \left[ R^T(t) R(t) \right]^{-1} \]  
\tag{11} 

Hence
\[ P_{t}^{-1} = R^T(t) R(t) \]  
\tag{12} 

Also it follows that
\[ P_{t}^{-1} = \left[ R^T(t-1) + Y_1(t) Y_1^T(t) \right] \left[ R(t-1) + Y_1(t) Y_1^T(t) \right]^{-1} \]  
\[ = R^T(t-1) R(t-1) + Y_1(t) Y_1^T(t) R(t-1) + Y_1^T(t) Y_1(t) R(t-1) \]  
\[ + Y_1^T(t) Y_1(t) Y_1^T(t) \]  
\[ = P_{t-1}^{-1} + W(t) Y_1^T(t) + Y_1(t) W^T(t) \]  
\[ + Y_1(t) K_1(t) Y_1^T(t) \]  
\tag{13} 

where
\[ W(t)_{n \times 1} = R^T(t-1) Y_1(t) \]  
\tag{14} 

\[ K_1(t)_{1 \times 1} = Y_1^T(t) Y_1(t) \]  
\tag{15} 

This can be written more compactly as
\[ P_{t}^{-1} = P_{t-1}^{-1} + X(t) \begin{bmatrix} 0 & 1 \\ 1 & K_1(t) \end{bmatrix} X^T(t) \]  
\tag{16} 

Inversion of eq.(13) yields
\[ P(t) = P(t-1) - P(t-1) X(t) \begin{bmatrix} 0 & 1 \\ 1 & K_1(t) \end{bmatrix} X^T(t) P(t-1) \]  
\tag{17} 

From eq.(10), we note that
\[ \dot{\theta}(t) = P(t) \left( R^T(t-1) + Y_1(t) Y_1^T(t) \right) \]  
\[ \left( z(t-1) + Y_1(t) y_{t-1} \right) \]  
\[ = P(t) R^T(t-1) z(t-1) + P(t) \left( Y_1(t) Y_1^T(t) \right) z(t-1) \]  
\[ + R^T(t-1) Y_1(t) y_{t-1} + Y_1(t) K_1(t) y_{t-1} \]  
\[ = P(t) P^{-1}(t-1) \dot{\theta}(t-1) + P(t) \left( Y_1(t) Y_1^T(t) \right) z(t-1) \]  
\[ + R^T(t-1) Y_1(t) y_{t-1} + Y_1(t) K_1(t) y_{t-1} \]  
\tag{18} 

From eq.(13), it follows that:
\[ P(t) P^{-1}(t-1) = \left( I - P(t) \left( W(t) Y_1^T(t) \right) + Y_1(t) W^T(t) + Y_1(t) K_1(t) Y_1^T(t) \right) \]  
\tag{19} 

Substituting eq.(19) in eq.(18), we obtain
\[ \dot{\theta}(t) = \dot{\theta}(t-1) - P(t) \left( W(t) Y_1^T(t) + Y_1(t) W^T(t) + Y_1(t) K_1(t) Y_1^T(t) \right) \dot{\theta}(t-1) \]  
\[ + P(t) \left( Y_1(t) Y_1^T(t) \right) z(t-1) + R^T(t-1) Y_1(t) y_{t-1} + Y_1(t) K_1(t) y_{t-1} \]  
\tag{20} 

Eq.(20) can also be written as:
\[ \dot{\theta}(t) = \dot{\theta}(t-1) + P(t) X(t) \begin{bmatrix} 0 & 1 \\ 1 & K_1(t) \end{bmatrix} X^T(t) \dot{\theta}(t-1) \]  
\tag{21} 

where
\[ S(t) = X(t) \begin{bmatrix} 0 & 1 \\ 1 & K_1(t) \end{bmatrix} X^T(t) \]  
\tag{22} 

Now we shall use the above result for developing an algorithm for the estimation of ARMA process parameters.

3. A COMBINED ADAPTIVE ALGORITHM FOR ESTIMATION OF ARMA PROCESS PARAMETERS

Let the ARMA process be represented by the eq:
\[ z_t = - \sum_{k=1}^{p} a_k z_{t-k} + u_t + \sum_{k=1}^{q} b_k u_{t-k} \]  
\tag{23} 

where \( \{u_t\} \) is a zero mean, non-Gaussian white noise sequence. Now assuming that \( \{b_k \leq k \leq q\} \) and \( \{u_t, s < t\} \) are known exactly, eq.(24) may be written as
\[ y_t = z_t + \sum_{k=1}^{p} \hat{a}_k z_{t-k} \]  
\tag{24} 

where \( \hat{a}_k \)’s are the estimates of \( \{a_k\} \) Next define the error:
\[ e_t = z_t + \sum_{k=1}^{p} \hat{a}_k z_{t-k} - \sum_{k=1}^{q} b_k u_{t-k} \]  
\tag{25} 

The parameters \( \hat{a}_k \) can then be estimated by minimizing the total square error \( \sum e_t^2 \) using the RLS algorithm [5]. Since in practice, the parameters \( \{b_k\} \) and \( \{u_k\} \) are unknown, replacing \( \{b_k\} \) by its estimate, \( \{\hat{b}_k\} \) we generate an approximate \( \hat{e}_t \) by recursively computing it as:
\[ \hat{e}_t = z_t + \sum_{k=1}^{p} \hat{a}_k z_{t-k} - \sum_{k=1}^{q} \hat{b}_k e_{t-k} = y_t - \sum_{k=1}^{q} \hat{b}_k e_{t-k} \] (26)

This algorithm is as shown in fig.(2).

4. SIMULATION STUDIES

The performance of the adaptive ORLS algorithm for an MA process was tested through a computer simulation of the Hammerstein model given by the following equation:

\[ y_t = u_t - 1.663u_{t-1} + 0.542u_{t-2} \] (27)

\[ u_t = v_t^2 \] (28)

where \( v_t \) is zero mean, Gaussian white noise sequence of variance 0.2045. The time histories of \( \hat{e}, \hat{b}_1 \) and \( \hat{b}_2 \) up to 3500 recursions are as given in fig.3 where it may be observed that the parameters to be identified converged to the values -1.6631 and 0.6922. Off line simulation of this process notably yielded -1.6115 and 0.5347. Here the sequence length was 10240 which was divided into 80 records of length 128 samples.

5. CONCLUSIONS

A new on-line technique, that uses ORLS in conjunction with RLS is proposed for the identification of non-linear Hammerstein models driven by Gaussian input (this is equivalent to the identification of linear ARMA models driven by non-Gaussian inputs). Simulation studies are carried out in respect of an MA process using ORLS only. The results obtained are found to be quite satisfactory. Work relating to the use of ORLS in conjunction with RLS is currently in progress.

REFERENCES


BOOTSTRAPPED SPATIAL SEPARATION OF WIDEBAND
SUPERIMPOSED SIGNALS

Hagit Messer and Yeheskel Bar-Ness
Dept. of ECE
New Jersey Institute of Technology
University heights
Newark, NJ 07102
U.S.A.

Abstract

Bootstrapped algorithms were developed and used for separation of two signals, when two versions of their weighted sum is given. In this paper we apply the bootstrap principle to the separation of \( N \) signals transmitted from point sources at different, unknown locations, when received by an array of \( M \) sensors. We present a general structure of the separation scheme which consists of delay elements and summation only. Its input is the \( M \)-sensor output signals and its output is the estimates of the \( N \) source signals. We show that if the source locations are known, this system provides a least squares estimate of the source signals. If not, it can adaptively converge to the least squares solution, provided that some prior information about the source signals is available. In particular, we present a detailed study of the bootstrapped algorithm for the separation of two sources received by two sensors. A simplified version of the algorithm is presented and the idea of adaptively controlling the unknown delays is discussed. We show that the proposed algorithm is a powerful tool for the decomposition of spatially mixed wideband signals.

I. Introduction and Background

The paper deals with the scenario in which \( N \) point sources are received by \( M \) omni-directional sensors. The received signal at the output of each of the \( M \) sensors can be modeled by:

\[
x_m(t) = \sum_{n=1}^{N} s_n(t - \tau_n(\theta_n)) + e_m(t)
\]

where: \( s_n(t) \) is the signal radiated from the \( n \)-th source; \( \theta_n \) represents the coordinates (location) of the \( n \)-th source; and \( e_m(t) \) is the additive noise at the \( m \)-th sensor. In the special case of a two dimensional array and far-field sources, \( \theta_n \) is the source bearing. \( \tau_m(\theta_n) \), the travel time of the \( n \)-th source from the array origin to the \( m \)-th sensor, is given by:

\[
\tau_m(\theta_n) = \frac{1}{c} \left[ x_m \sin \theta_n + y_m \cos \theta_n \right]
\]

where \( c \) is the propagation velocity of the signal wavefront and \( (x_m, y_m) \) are the Cartesian coordinates of the \( m \)-th sensor. If the array is an equally spaced linear array (ESLA) then \( y_m = 0 \) and \( x_m = (m - 1)d \), \( m = 1, ..., M \) where \( d \) is the separation between successive sensors and the plane origin is assumed at the coordinates of the first sensor. Thus, (2) becomes:

\[
\tau_m(\theta_n) = (m - 1) \frac{d}{c} \sin \theta_n
\]

In general there are \( M \times N \) \((M-1) \times N\) delays which are a function of the \( N \) source locations \( \theta_1, ..., \theta_N \). For an ESLA there are only \( N \) different delays which carry all the spatial information. The above model can match many practical applications in different fields. In passive sonar the source signals are wideband, noise-like random processes and the unknown source location vector, \( \theta = (\theta_1, ..., \theta_N)^T \), has to be estimated. In the active case the source signals are basically known, but the aim is the same - estimation of the source locations. However, in communication systems one is usually interested in the source signals themselves and not in their locations, which are sometimes known.

We assume broadband signals, which is more general than the usually assumed narrowband case, where a delay can be regarded as a phase shift. The received sensor data (1) can be represented in the frequency domain:

---

1This work was partially supported by grant from ROME (AFSC, Griffiss Air Force Base, NY) under contract F 30602-88-D-0025, Task C-0-2456.

2On sabatical leave from the Dept. of Electrical Engineering - Systems, Tel Aviv University, Tel Aviv 69978, Israel.
\[
Z_m(\omega_k) = \frac{1}{T} \int_{-T/2}^{T/2} z_m(t) e^{-j\omega_k t} dt
\]

\[m = 1, ..., M; \quad k = 1, ..., L\]  \hspace{1cm} (4)

\(Z_m(\omega_k)\) is the Fourier coefficient of the output from the \(m\)-th sensor at frequency \((2\pi/T)k\). The processing bandwidth is, therefore, \(B = L/T\). At each frequency, the array output is given by the \(m\)-dimensional vector:

\[
Z(\omega_k) = (Z_1(\omega_k), ..., Z_M(\omega_k))^T
\]

\hspace{1cm} (5)

Following the model of (1), \(Z(\omega_k)\) can be written as:

\[
Z(\omega_k) = \sum_{n=1}^{N} S_n(\omega_k)g(\omega_k, \theta_n) + E(\omega_k)
\]

\[= A(\omega_k)S(\omega_k) + E(\omega_k)\]  \hspace{1cm} (6)

where \(S(\omega_k) = (S_1(\omega_k), ..., S_N(\omega_k))^T\) and \(E(\omega_k) = (E_1(\omega_k), ..., E_M(\omega_k))^T\). \(S_n(\omega_k)\) and \(E_n(\omega_k)\) are the Fourier coefficients of the \(n\)-th source signal and the noise in the \(m\)-th sensor, respectively. Also,

\[
A(\omega_k, \ell) = [a_1(\omega_k) : ... : a_N(\omega_k)]
\]

\hspace{1cm} (7)

where

\[
a_n(\omega_k, \theta_n) = g_n(\omega_k) = (e^{j\omega_k \theta_n}, ..., e^{j\omega_k \theta_N})^T\]  \hspace{1cm} (8)

If \(M \geq N\) the least-squares estimate of the frequency domain vector of source signals, \(\hat{S}(\omega_k)\), given the data vector \(Z(\omega_k)\), is [1]:

\[
\hat{S}(\omega_k) = [A^*(\omega_k, \ell) A(\omega_k, \ell)]^{-1} A^*(\omega_k, \ell) Z(\omega_k)
\]

\hspace{1cm} (9)

For the special case of \(M = N = 2\) it easily can be verified that:

\[
[\begin{array}{c}
A^*(\omega_k, \ell) A(\omega_k, \ell)
\end{array}]^{-1}
\]

\[
\frac{1}{\sin^2 \omega_k \Delta}
\]

\[
\begin{bmatrix}
\cos \omega_k \Delta & e^{j\omega_k D_2} & e^{-j\omega_k D_1} & \cos \omega_k \Delta & e^{-j\omega_k D_2}
\cos \omega_k \Delta & e^{j\omega_k D_2} & e^{-j\omega_k D_1} & \cos \omega_k \Delta & e^{-j\omega_k D_2}
\end{bmatrix}
\]

\hspace{1cm} (10)

Where \(\Delta = D_1 - D_2\). In (8) we also assume that the array origin reference is the mid location between the elements, so that \(\tau_1(\ell) = -\tau_2(\ell) = D_1, \quad \tau_1(\ell) = -\tau_2(\ell) = D_2\). We see that, even if \(D_1\) and \(D_2\) are known, the implementation of the least squares solution of (9) requires filtering the array outputs \(z_1(t)\) using filters having transfer functions of the form: \(\cos \omega \Delta\) or \(1/\sin^2 \omega \Delta\), as well as pure delays (see Fig. 1). The implementation of the trigonometric filters is difficult, especially when \(\delta\) is unknown and is to be estimated adaptively. One possible approach to deal with this implementation problem is to approximate the trigonometric filters by FIR (or IIR) filters. In the sequel, we show that applying the bootstrap principle to this problem results in exact implementation of (9) which uses only delay elements and summations, in a feedback configuration.

II. The Bootstrapped Algorithm

Bootstrapped systems are multi-input multi-output feed-back systems in which each output "helps" to improve the other outputs. The idea is to assume that a system output is indeed the desired response and to use it, via feedback, to get other outputs. This approach has been successfully applied in satellite communication to improve separation of cross-pol signals [2]. In our problem, given any \(N\)-1 source signals one can get a good estimate of the remaining signal, say \(s_n(t)\) from any of the sensor outputs using:

\[
\hat{s}_n(t) = \delta_n(t - \tau_m(\ell_n))
\]

\[
= z_m(t) - \sum_{i=1, i \neq n}^{N} s_i(t - \tau_m(\ell_i))
\]

\hspace{1cm} (11)

If no noise exists, then the left hand side of (11) is indeed a delayed version of \(s_n(t)\). However, since noise is never zero, averaging over the estimated version of a certain signal from all sensor outputs will improve SNR output. Therefore, we have:

\[
\hat{s}_n(t) = \frac{1}{M} \sum_{m=1}^{M} \hat{s}_n(t + \tau_m(\ell_i))
\]

\hspace{1cm} (12)

The outputs of the proposed algorithm are \(N\) signals, \(y_1(t), ..., y_N(t)\) which are desired to be the best possible estimates of the \(N\) source signals \(s_1(t), ..., s_N(t)\). Following the bootstrapped approach, we replace the known source signals in (11) by their estimates \(\{\hat{s}_i(t)\}\). Therefore, the proposed scheme is described by the \(N \times M\) equations:
\[ y_{nm}(t) = y_n(t - \tau_m(\theta_n)) \]
\[ = z_n(t) - \sum_{i=1}^{N} y_i(t - \tau_m(\theta_i)) \]
\[ m = 1, \ldots, M; \quad n = 1, \ldots, N \quad (13) \]

\[ y_n(t) = \frac{1}{M} \sum_{m=1}^{M} y_{nm}(t + \tau_m(\theta_i)) \quad (14) \]

By transforming (13) to the frequency domain we get

\[ Y_n(\omega) = \frac{1}{M} \sum_{m=1}^{M} \sum_{n=1}^{N} e^{j\omega \tau_m(\theta_n)} Y_m(\omega) \]
\[ - \sum_{n=1}^{N} Y_i(\omega) e^{-j\omega \tau_m(\theta_n)} \]
\[ m = 1, \ldots, M; \quad n = 1, \ldots, N \quad (15) \]

In a matrix form this equation is equivalent to

\[ A^*(\omega, \theta) A(\omega, \theta) \chi(\omega) = A^*(\omega, \theta) Z(\omega) \quad (17) \]

where \( A(\omega, \theta) \) is given by (7) and (8). Therefore, the frequency domain representation of the output vector \( y(t) = (y_1(t), \ldots, y_N(t))^T \) is exactly the same as \( \tilde{S}(\omega) \) of (9). i.e., the system described by (12) is a realization of the least squares estimator of the source signals. In Fig. 2 we present a block diagram of this system for the special case of \( N = M = 2 \).

![Fig. 2: Bootstrapped implementation of the least-squares separator](image)

**III. Bootstrapped Separation of Sources at Unknown Locations**

If the source location is unknown, then \( \theta = (\theta_1, \ldots, \theta_N)^T \) is unknown, and \( \tau_m(\theta_i) \) in (9) is replaced by its estimate, \( \hat{\tau}_{mi} = \tau_m(\hat{\theta}_i) \), \( m = 1, \ldots, M, \quad i = 1, \ldots, N \). In that case, the frequency domain representation of the output vector is:

\[ \chi(\omega_k) = [A^*(\omega_k, \hat{\theta}) A(\omega_k, \hat{\theta})]^T A^*(\omega_k, \hat{\theta}) Z(\omega_k) \]
\[ = [A^* A ]^{-1} A^* Z \quad (18) \]

This is no longer the least square estimate of \( S(\omega_k) \), but only an approximation. In the sequel, we propose an adaptive algorithm by which an estimate of the delays \( D_i \) will be found. We show that, in case of no additive noise, the algorithm converges to the least squares solution of (9). We demonstrate our results for the special case where \( N = M = 2 \). However, generalization of the algorithm for any \( M \geq N \geq 2 \) is straightforward. Consider the system of Fig. 3.

It can be shown that for \( \tau_1(\theta_1) = -\tau_2(\theta_1) = D_1 \), \( \tau_2(\theta_2) = -D_2 \), the 2x2 transfer function matrix, \( H(\omega) \), which relates the two outputs, \( y_1(t) \) and \( y_2(t) \) to the two inputs \( z_1(t) \) and \( z_2(t) \) is exactly the same as those of the system of Figs. 1 and 2. That is, the system of Fig. 3 is another, alternative implementation of a least squares separator with \( H(\omega) \) given by (10). If initially \( \tau_1 \neq D_1 \) and/or \( \tau_2 \neq D_2 \) then we intend to adapt \( \tau_1 \) and \( \tau_2 \) so that in the steady state they reach these optimal values. We notice that the scheme of Fig. 3 is similar to the bootstrapped "power-power" separator of [2-7] which is applied to separate weighted sum of two uncorrelated signals. Then, the delays \( \tau_1 \) and \( \tau_2 \) are replaced by complex weights, say \( W_1 \) and \( W_2 \). In the frequency domain our unknown delays are represented by \( e^{j\omega \tau_1} \) and \( e^{-j\omega \tau_2} \) and the input signals are weighted sums of the uncorrelated frequency domain signals \( S_1(\omega) \) and \( S_2(\omega) \).

Therefore, the frequency domain representation of our problem is equivalent to the time domain separation problem of [2-10] and a similar approach can be considered. It was shown there that without noise \( \tau_1 = \tau_2 = 0 \), the power of the two output signals is minimal if and only if the controlled weights are equal to the unknown model parameters. It was also shown that for this configuration, the optimization criterion of minimum power is equivalent to the criterion of zero
correlation between the two outputs [3-4]. By analogy, in our problem the equivalent optimization criterion should be minimum power (or zero correlation) in the frequency domain. However, since power and correlation are preserved when transforming from time to frequency and vice versa (Parseval), this criterion can be applied to our problem either in the frequency domain, or in the time domain. Inspired by a possible hardware implementation using voltage controlled delay lines, we prefer the time domain approach. That is, we suggest to control the unknown delays \( \tau_1 \) and \( \tau_2 \) by an adaptive algorithm which seeks for the minimum power of both outputs, simultaneously (or, for the minimum power of their cross correlation signal). Notice, however, that this control procedure cannot be employed unless some information that distinguishes the signals to be separated is available. Mathematically, it can be shown that all possible optimization criteria (zero correlation\(^2\), minimum power) yield the same, or linearly dependent, control equations. Notice that to control both \( \tau_1 \) and \( \tau_2 \) one needs two independent equations. This difficulty can also be predicted by considering the separation problem as a multi-input multi-output identification problem, where it is well known that weighted sums cannot be separated if nothing is known about their components. In our application, it is also well known that the resolution capacity of an array of \( M \) sensors is bounded by the number of sensors (i.e., \( M \geq N \)) if absolutely nothing is known about the signals. Conversely, if prior information is available, it can be used to discriminate between the two signals in the control loops. As shown in [3,4], such dependency problem can be handled by a "discriminator" which uses the distinguishing information to emphasize \( s_1(t) \) in one of the control loops and another one which emphasizes \( s_2(t) \) in the other loop. This procedure yields two independent control equations, which guarantees convergence of the adaptive algorithm to the desired solution \( \tau_1 = D_1 \) and \( \tau_2 = D_2 \). The possible control loops are depicted in Fig. 4.

IV. Conclusions

For separation of signals radiated from point sources, we propose the system of Fig. 3, where the delays are controlled by any of the algorithms of Fig. 4 (there are 4 different combinations, for two unknown delays). We have shown that if the delays are adapted to the unknown model parameters exactly, then the outputs of the proposed bootstrapped system are the least-square estimates of the source signals. In a further study we investigate the adaptive algorithm with more details, we suggest alternative implementations of the separation configuration and we study the effect of the additive noise on the performance of the proposed separator.

References


\(^2\)Lately, the decorrelation approach is also used in [11]
Hybrid Mean/Median Wiener Filters

James L. Lansford
Department of Electrical and Computer Engineering
University of Colorado at Colorado Springs
1420 Austin Bluffs Parkway
Colorado Springs, CO 80903

Abstract
Wiener filters are used as whitening filters to remove correlation in the noise. Such filters, because they are based on a least squares model, exhibit poor performance in environments where noise exhibits burst or impulsive characteristics. The traditional formulation of the Wiener filter can be modified to allow a non-least squares criterion; this adaptive approach is more robust to transient events like burst noise.

Introduction
The traditional Wiener filter is based on an autoregressive model which can be described by the difference equation:

\[ y(n) = b_0 x(n) + a_1 y(n-1) + a_2 y(n-2) + \cdots \\
\hspace{1cm} + a_{N+1} y(n-N+1) + a_N y(n-N) \]  

(1)

If written in matrix form starting at \( n=N \) and assuming that \( M \) samples \( \{0...M-1\} \) of the sequence \( y \) are available, this results in the overdetermined system [1]:

\[
\begin{bmatrix}
  y(N-1) & y(N-2) & \cdots & y(0) \\
  y(N) & y(N-1) & \cdots & y(1) \\
  \vdots & \vdots & \ddots & \vdots \\
  y(M-2) & y(M-3) & \cdots & y(M-N-1)
\end{bmatrix}
\begin{bmatrix}
  \beta_0 \\
  \beta_1 \\
  \vdots \\
  \beta_{N-1}
\end{bmatrix}
=
\begin{bmatrix}
  y(N) \\
  y(N+1) \\
  \vdots \\
  y(M-1)
\end{bmatrix}
\]

(2)

of \( M-N \) equations in \( N \) unknowns, which could be written in matrix form as \( G a = f \). Recovering the \( a_i \) from a given sequence of outputs in this way is blind deconvolution with an assumed \( x(n) \) that is spectrally flat, such as an impulse or white noise. Even if this is not the case, effects of the source will be incorporated into the model [2].

A solution to the equations given in [2] can be developed by manipulating the matrices to obtain:

\[
a = \left[ G^T G \right]^{-1} G^T f
\]

(3)

which is the least squares solution.
This solution is not robust to impulses, bursts, or other transients because of the nature of the least squares solution [4]; furthermore, the model is maximum likelihood only if it can be shown that the residual sequence \( e = f - Ga \) is Gaussian [5].
In a communication system, the whitening process involves subtracting samples of the predicted values $G_a$ from the input sequence $f$ to yield the residual sequence $e$; Figure 1 shows this process in block diagram form.

In terms of frequency response, the spectral estimate derived from the autoregressive model given by (2) is:

$$ H(z) = \frac{b_0}{1 - \sum_{i=1}^{N} a_i z^{-i}} $$

(4)

Clearly, the matrix expression $G_a$ is an FIR filter of the form:

$$ H_{inv}(z) = 1 - \sum_{i=1}^{N} a_i z^{-i} $$

(5)

which is, of course, the inverse filter for the process $y(n)$. In the literature, little emphasis has been placed on the effect of the solution method for (2) on the spectral estimate.

**Robust Wiener Filter**

The least squares solution given by (3) is only one of an infinity of solutions to the overdetermined system in (2). In general, a p-normed error criterion can be defined:

$$ e = \sum_{i=1}^{N} |f_i - (G_a)_i|^p $$

(6)

where minimization of $e$ under the constraint $G_a = f$ when $p=2$ is the least squares solution. This generalization allows a “robust” solution, which is also known as the least absolute value ($p=1$) solution. In terms of maximum likelihood estimators, this formulation is quite comprehensive, encompassing Gaussian ($p=2$), Laplacian ($p=1$), and uniform ($p=\infty$) densities: the family of probability density functions is called the generalized Gaussian or p-Gaussian densities. The functional form is:

$$ f(x) = \frac{1}{\sigma \Gamma(1+1/p)^{1/p}} \exp \left\{ -\frac{1}{2} \left(\frac{x}{\sigma}\right)^p \right\} \quad -\infty < x < \infty $$

(7)

Solution to (2) with the objective function given by (6) is more complicated than the Cholesky decomposition typically used for the least squares formulation. As suggested by Yarlagadda et al., a modified residual steepest descent (RSD) method works well in this application. This algorithm implies a computational complexity of about 4 times that of the Cholesky decomposition, since the least squares problem must be solved in each iteration of the algorithm, and an average of around four iterations are required for convergence in typical situations.

**Robust Filtering of Communications Signals**

Using the approach described above, the technique was evaluated with a continuous sinusoid of the form:

$$ y(n) = \sin(\omega_0 n) + w(n) + \delta(n-k) $$

(8)

where $w(n)$ is a white Gaussian noise process whose variance dictates the SNR, and $\delta(n-k)$ is a delta function that occurs sometime during the analysis interval. If the p-normed algorithms are applied to this signal and the spectral estimate is computed, the sequence of spectra shown in Figure 2 results. It is interesting to note that if the noise had been purely impulsive, the spectrum for $p=1$ would have been maximum likelihood, while Gaussian noise alone would have had $p=2$ as its maximum likelihood estimator. The fact that the noise is a blend of the two types is reflected in the peak in the spectral estimate for a value of $p$ between one and two.
The effects of impulsive noise on the spectral estimate can be further observed to cause a bias in the spectral estimate. Figure 3 shows the $L_1$ and $L_2$ estimates for the case of $a_0=1.5$. As this figure shows, the least squares estimate is biased away from the "true" value because of the impulse, while the least absolute value estimator more closely represented the spectrum.

Finally, the output of the inverse filter is an indicator of how well the signal can be smoothed; Figure 4 shows the signal $y(t)$ and the smoothed estimate $Ga$ with an $L_1$ estimator. This figure clearly shows that the sinusoid's frequency is preserved (but not its phase) and that the impulse has been excised.

Summary

This paper has shown a generalized Wiener filter that can blend the properties of a median filter and an averager to generate a filter (and a spectral estimate) that is maximum likelihood for a wide variety of distributions, including Laplacian and Gaussian.

References

Figure 2: Spectral estimate as a function of $p$

Figure 3: Comparison of spectra for $p=1$ and $p=2$

Figure 4: Smoothing filter output
STUDY OF MA MODEL IDENTIFICATION USING ORDER STATISTICS IN THE PRESENCE OF NOISE

J.Y TOURNERET and B LACAZE
ENSEEIH / GAPSE, 2 Rue Camichel, 31 071 Toulouse, FRANCE

A new estimator of Moving Average (MA) parameters was presented in the Twenty-Sixth Annual Conference on Information Sciences and Systems [1]. This new estimator was studied for a model driven by a bounded input having a non zero probability of reaching its maximum. Such input laws are obtained in the case of binary, discrete or clipped inputs. The most frequently used MA parameter estimators are approximations of the Maximum Likelihood Method. These Maximum Likelihood Estimators (M.L.E.) have a zero error asymptotically. Our estimator is based on Order Statistics and will be called O.S. Estimator in the following. In absence of noise, its main property is to give a zero error for a finite signal point number with a high known probability. The aim of this paper is to study O.S estimator performances in the case of a model embedded in an additive noise. The study is done in the case of a binary input but can be extended to discrete or clipped inputs. In the first section, our new method of MA identification using Order Statistics is briefly described. The second section deals with the behaviour of the O.S. estimator in the presence of noise.

1 INTRODUCTION

Most commonly used Moving Average (MA) parameter estimators are approximations to the Maximum Likelihood Estimator (M.L.E.) [2],[3]. The use of the M.L.E. for parameter estimation can be justified for large data records on the basis of its asymptotic unbiased and minimum variance properties. In particular, the M.L.E. has a zero error asymptotically. A new estimator of MA parameters based on Order Statistics, which is denoted by O.S. estimator, was presented in the 26th C.I.S.S. [1]. The main property of this estimator is to give, in absence of noise, a zero error for a finite signal point number with a high known probability. In the first section, MA parameter identification using Order Statistics is briefly described. The second section deals with the estimator performances in the case of a MA model embedded with an additive gaussian noise. The study is carried out for a binary input but can be extended to discrete or clipped inputs.

2 MA IDENTIFICATION

Let $e(k)$ be the input noise of an order $q$ MA model, the parameters of which are denoted by $b_0, b_1, \ldots, b_q$. Let us denote $x(k)$ the model output, $n(k)$ the additive noise and $z(k) = x(k) + n(k)$. The model is then defined by the following recursive equations:

\[ k = 0, 1, \ldots, K - 1 \quad z(k) = n(k) + \sum_{i=0}^{q} b_i e(k-i) \quad (1) \]

The noise $n(k)$ is assumed to be Independent Identically Distributed (i.i.d.) and statistically independent of $x(k)$. In what follows, we propose a new method to estimate the parameters $(b_i)$ for a special class of input white noise : $e(n)$ is bounded by its maximum $E = \max \{ e(k) \}$ such that:

\[ P_{\max} = P[e(k) = E] + P[e(k) = -E] > 0 \quad (2) \]

Let us cascade at the output $z(k)$ a set of $q+1$ order $I$ MA models with respective parameters $c_j$, for $j=0,1,\ldots,q$, which will be referred to as "Analysis Filters". The analysis filter outputs, shown in Fig 1 and denoted by $y_j(k)$ for $j=0,1,\ldots,q$ are given by:

\[ y_j(k) = b_0 e(k) + \sum_{i=1}^{q} (b_i + c_j b_{i-1}) e(k-i) \]

\[ + c_j b_q e(k-q-1) + n(k) + c_j n(k-1) \quad (3) \]

Fig 1: Single-channel system with Analysis Filters
When there is no additive noise \((n(k)=0)\), the maximum of \(y_j(k)\) for \(k \in N\), denoted by \(\text{Max}(y_j(k))\), is given by:

\[
\text{Max}(y_j(k)) = E \left\{ |b_0| + \sum_{i=1}^{q} |b_i + c_i b_{i-1}| + |c_q| \right\}
\]  (4)

By denoting \(S_j(i)\) for \(i=1,2,\ldots,q\), the sign of \((b_i + c_i b_{i-1})\) and respectively \(S_j(0), S_j(q+1)\), the signs of \(b_0\) and \(c_j b_q\), Eq.(4) can be written in the following way:

\[
\text{Max}(y_j(k)) = E = b_0 S_j(0) + \sum_{t=1}^{q} (b_i + c_i b_{i-1}) S_j(i) + c_j b_q S_j(q+1) = \sum_{t=0}^{q} b_t [S(t) + c_j S(t+1)]
\]  (5)

The concatenation of Eq.(5) for \(j=0,1,\ldots,q\) leads to a system of \(q+1\) equations with \(q+1\) unknowns which can be written:

\[
\pmb{M} = \pmb{A} \pmb{b}
\]  (6)

with:

\[
\pmb{M} = \frac{1}{E} [\text{Max}(y_0(k)), \ldots, \text{Max}(y_q(k))]^T
\]

\[
\pmb{b} = [b_0, b_1, \ldots, b_q]^T
\]

\[
\pmb{A} = (\alpha_{hi}) = [S(i) + c_j S(i+1)]
\]

The matrix equation (6) will yield unique MA parameters provided that the \((q+1) \times (q+1)\) matrix \(\pmb{A}\) is of full rank \(q+1\). In [1], a method for choosing the parameters \(c_j\), in order to have a full rank matrix \(\pmb{A}\) is studied.

For practical implementation, \(\text{Max}(y_j(k))\) values are estimated with \(K\) Analysis Filter outputs \(y_j(0), y_j(1), \ldots, y_j(K-1)\) and will be then denoted by \(\text{Max}_K(y_j(k))\). The different \(S_j(i)\) sign estimations, obtained with rough estimates of MA parameters, are supposed to be rigorously exact. The matrix system (6) leads to:

\[
\pmb{A} \pmb{b} = \pmb{M}_K
\]  (7)

with:

\[
\pmb{b} = [b_0, b_1, \ldots, b_q]^T
\]

\[
\pmb{M}_K = \frac{1}{E} [\text{Max}_K(y_0(k)), \ldots, \text{Max}_K(y_q(k))]^T
\]  (8)

The solution for this system gives MA parameter estimators \(\delta_0, \delta_1, \ldots, \delta_q\). When the \(q+1\) absolute maxima \(\text{Max}_K(y_j(k))\), for \(j=0,1,\ldots,q\) are reached, that is to say \(\text{Max}_K(y_j(k)) = \text{Max}(y_j(k))\) for \(j=0,1,\ldots,q\), these MA parameter estimators are exactly equal to \(b_0, b_1, \ldots, b_q\) and then have a zero error. The aim of this paper is to study the performances of the O.S. estimator in the case of a model embedded with an additive gaussian noise.

### 3 NOISE EFFECTS

In this section, the model output is embedded with an additive noise which is assumed to be continuous. The probability density of \(y_j(k)\) (see Eq.(3)) and then of \(\text{Max}_K(y_j(k))\) depends on that of the noise. The effects of noise are then studied in the case of a binary input \(e(n)\), taking on equally likely values \(+1\) and \(-1\), but this study could be extended to discrete or clipped inputs.

Then the variable \(y_j(k)\) is the sum of a discrete variable \(e_j(k) = b_0 e(n) + \sum_{t=1}^{q} (b_i + c_i b_{i-1}) e(n-t-1) + c_j b_q e(n-q-1)\) taking on \(2^{q+2}\) different values denoted by \(u_i\) for \(i=1,2,\ldots,2^{q+2}\) and of a continuous variable \(n_j(k) = n(k) + c_j n(k-1)\) (see Eq.(3)). The probability density function (p.d.f.) of \(n_j(k)\) is denoted by \(f_j(t)\). With these notations, the p.d.f. of \(y_j(k)\) is given by:

\[
f_{y_j}(t) = \frac{1}{2^{q+2}} \sum_{i=1}^{2^{q+2}} f_j(t-u_i)
\]  (9)

\(n(k)\) is assumed to be a white gaussian noise with zero mean and variance \(\sigma^2\). Then \(n_j(k) = n(k) + c_j n(k-1)\) is gaussian with zero mean and variance \(\sigma_j^2 = (1+c_j^2)\sigma^2\). The \(y_j(k)\) p.d.f. is then given by:

\[
f_{y_j}(t) = \frac{1}{2^{q+2}} \sum_{i=1}^{2^{q+2}} \frac{1}{\sqrt{2\pi \sigma_j^2}} \exp \left[-\frac{(t-u_i)^2}{2\sigma_j^2}\right]
\]  (10)

The determination of \(\text{Max}_K(y_j(k))\) distribution is very difficult because the variables are not independent. A lot of work has been done in the extreme value field. Gumbel [4] has shown the existence of three asymptotic distributions of largest values. Each assumes a specific behaviour for absolute large values of the variable. From (10), the asymptotic \(\text{Max}_K(y_j(k))\) distribution can be determined with the help of the following:

1. The distribution of the largest normal values converges towards the Gumbel first asymptote [4].
2. Given \(N\) p.d.f. \(f_i(t)\) (for \(i=1,\ldots,N\)) of \(N\) variables whose largest value distributions converge towards the Gumbel first asymptote, it can be shown that the variable whose p.d.f. is \(\frac{1}{N} \sum_{i=1}^{N} f_i(t)\) has a largest value distribution converging towards the same asymptote.

Then, for a high signal point number \(K\), the \(\text{Max}_K(y_j(k))\) distribution is given by the first asymptote:

\[
f(t) = \alpha e^{-\alpha(t-u)} \exp(-e^{-\alpha(t-u)})\] with \(\alpha > 0\).


The parameters $\alpha$ and $u$ have then to be estimated. Numerous estimation methods of these parameters are available in the literature [4]. They lead to an estimated $\max_k(y_i(k))$ probability density function which has been compared with the $\max_k(y_i(k))$ histogram. For a $K=10,000$ signal point number, simulations show the good agreement between the two estimations. The results that follow deal with an order 2 MA Model with parameters $b_0 = 1$, $b_1 = 0.2$ and $b_2 = 0.5$:

![Fig 2a: $\max_k(y_i(k))$ p.d.f.](image)

![Fig 2b: $\max_k(y_i(k))$ p.d.f.](image)

![Fig 2c: $\max_k(y_i(k))$ p.d.f.](image)

**Fig 2a, 2b, 2c:** Comparison between theoretical and experimental $\max_k(y_i(n))$ p.d.f.

For an order 2 MA model with parameters $b_0 = 1$, $b_1 = 0.2$ and $b_2 = 0.5$.

The $\max_k(y_i(n))$ asymptotic distribution moments allow us to estimate the bias of the O.S. estimator. We thus obtain:

$$E[\hat{b}] - \bar{b} = A^{-1}(E[\hat{M}_k] - M)$$

with $E[\max_k(y_i(k))] = u + \frac{\gamma}{\alpha}$, $\gamma$ being the Euler constant ($\gamma = 0.577$) [4].

In order to determine the O.S. estimator covariance matrix, that of $\hat{M}_k$ has to be computed. But this is a very difficult task because the variables $\max_k(y_i(n))$ are not independent. For the sake of simplicity, let us consider a simpler model:

![Fig 3: $q+1$ single-channel model.](image)

The variables $\max_k(y_i(k))$ are then independent and the corresponding covariance matrix, denoted by $C_{a(i)}$, is diagonal. The $C_{a(i)}$ diagonal terms are equal to the $\max_k(y_i(k))$ variances that is to say to the first asymptote order 2 moment ($\frac{\pi^2}{6\alpha^2}$) [4]. The O.S. estimator covariance matrix is then given by:

$$E[(\hat{b} - \bar{b})(\hat{b} - \bar{b})^T] = A^{-1}C_{a(i)}(A^{-1})^T$$

The independence assumption is clearly not valid for the application to an actual MA model. But we show here-under that, although this theoretical approach is not realistic, it yields experimental results in satisfactory agreement with the model of Fig.3, then, the covariance matrices of $\hat{M}_k$ and $\hat{b}$ have been computed as the by-products of numerous simulations (2,000 estimations of $\max_k(y_i(k))$ and of $\hat{b}$). For the previous order 2 MA model, with a Signal to Noise Ratio (SNR) equal to 20dB, the following results are obtained:

$$\begin{pmatrix}
1.8e-3 & 7.3e-6 & 2.3e-6 \\
7.3e-6 & 4.4e-4 & -1.6e-6 \\
2.3e-6 & -1.6e-6 & 1.3e-4
\end{pmatrix}$$

$\hat{M}$ simulated covariance matrix

The $M_k$ covariance matrix extra-diagonal terms are negligible. The variables $\max_k(y_i(n))$ can then be assumed to be independent and the theoretical $\hat{b}$
covariance matrix can be determined. We then obtain:
\[
\begin{pmatrix}
5.2e-4 & -3.2e-5 & -5.5e-4 \\
-3.2e-5 & 6.5e-5 & 3.9e-5 \\
-5.5e-4 & 3.8e-5 & 6.2e-4
\end{pmatrix}
\]
\[\delta\text{ theoretical covariance matrix}\]
which can be compared with the experimental covariance matrix:
\[
\begin{pmatrix}
4.3e-4 & -2.9e-5 & -4.5e-4 \\
-2.9e-5 & 4.8e-5 & 3.4e-5 \\
-4.5e-4 & 3.4e-5 & 5.1e-4
\end{pmatrix}
\]
\[\delta\text{ experimental covariance matrix}\]

The matrix norm chosen for the comparison with the simulated and theoretical covariance matrix is the Trace of the difference of these two matrices. The relative error between the two covariance matrices is then approximately equal to 18%, which is a quite acceptable result. This shows the good agreement between simulations and theory.

As a matter of illustration, the MA parameter O.S. estimator variance is compared, for different Signal to Noise ratio, to that obtained using the conventional Durbin algorithm [2],[3]. The choice of the signal point number K in terms of the MA model order is studied in [1]. For our simulations, it has been fixed to \(K = 10,000\).

The following results deal with an order 2 MA Model with parameters \(a_0 = 1\), \(b_1 = 0.2\) and \(b_2 = 0.5\). M.L.E. and O.S.E. stand respectively for Maximum Likelihood Estimator and Order Statistic Estimator.

For Signal to Noise Ratio greater than 15 dB, the M.L.E. and O.S.E. variances have the same order of magnitude but for S.N.R. lower than 15dB, the Maximum Likelihood Estimator performs better.

The main drawback of the O.S. Estimator, in presence of gaussian noise (more generally unbounded noise), lies in its bias. Namely, when \(n(k)\) is not bounded, the following property is obviously verified:

\[\lim_{K \to \infty} \text{Max}_K[n(k)] = \lim_{K \to \infty} \text{Max}_K[y_j(k)] = +\infty\]

In order to solve this problem, an unbiased Order Statistic Estimator will be presented in a forthcoming publication.

4 CONCLUSION

The behaviour of a new estimator of Moving Average Model parameters, based on Order Statistics, has been studied in presence of an additive gaussian noise. An expression of the theoretical covariance matrix of this estimator has been computed under a simplifying assumption and compared with an experimental one. Then, M.L.E. And O.S.E. variances have been compared for various Signal to Noise Ratio. For high Signal to Noise Ratios, the O.S. estimator performs well, that is to say the MA parameter variances have the same order of magnitude as those obtained with the Maximum Likelihood Estimator. For lower Signal to Noise ratio, the classical Maximum Likelihood estimator will be preferred. On the other hand, the main effect of additive gaussian noise on the Order Statistic Estimator is its bias. For the case of an amplitude bounded noise with an approximately gaussian shape, an unbiased version of this estimator has been designed and will be published in a forthcoming publication.

5 REFERENCES


A NEURAL NETWORK FOR ONLINE ESTIMATION OF PARAMETERS
OF NOISY SINUSOIDAL SIGNALS

Andrzej CICHOCKI
Technical University of Warsaw
PL-00681 Warszawa, Poland
ul. Koszykowa 75

Tadeusz ŁOBOS
Technical University of Wrocław
PL-50370 Wrocław, Poland
pl. Grunwaldzki 13

New massively parallel algorithms for estimation of parameters of
sinewave contaminated by noise are proposed. The approach can be
viewed as an extension and generalization of the standard least-squares
minimization method. The implementation of the algorithms by an ap-
propriate neural network is also given. Illustrative computer simu-
lations results confirm validity and high performance of the proposed
solution.

1. INTRODUCTION

The problem of estimating the amplitudes, phase
angles and frequencies of sinusoidal signals
from noisy and distorted data have received con-
siderable attention recently. For control and
protection of electrical power systems it is de-
sired to estimate in real-time parameters of ba-
sic waveform (fundamental harmonic) of voltages
and currents. For this purpose different numeri-
cal algorithms have been developed e.g. based
on the Fourier and Kalman filtering [1].

In this paper we shall present new algorithms
associated with system of differential or dif-
ference equations which can be implemented (sim-
ulated) by analog VLSI circuits employing some
neural network principles [2,3,4,5]. Neural net-
works may provide a new kind of solution for
many standard signal processing problems, e.g.
recovering of analog signal corrupted by noise.

The main purpose of this paper is to present a
new algorithm and associated new architectures
of analog artificial neuron-like processors for
online estimation of parameters of sinusoidal
signal distorted by higher harmonics and cor-
rupted by noise. The problem of estimation is
formulated as an optimization problem [4].

2. STATEMENT OF THE PROBLEM

Consider the following sinusoidal signal
\[ x(t) = a \sin(\omega t) + b \cos(\omega t) \]  
(1)
in which \( a, b \) are unknown amplitudes,
\[ \omega = 2\pi f \] is unknown angular frequency.

Let \( y(t) \) denote the noise corrupted measurement
of \( x(t) \) i.e.
\[ y(t) = x(t) + e(t) \]  
(2)
where \( e(t) \) is unknown error. This error includes
random noise and distortion caused e.g. by
measurement instruments, or higher harmonics.

Consider the practical case where the signal of
interest \( y(t) \) is observed (measured) during a
finite time and only \( N \) samples of this signal
are available i.e. the discrete-time signal can
be specified as
\[ y(t) \bigg|_{t=mT} = y_n = X_a \sin(m\omega T) + X_b \cos(m\omega T) + e_n(T) \]
(3)
for \( m = 1, 2, \ldots, N \).

Hence, the error at the moment \( t = mT \) can be
expressed as
\[ e_n = y_n - x_n \]  
(4)

There exist a need for an online algorithm which
can satisfy two simultaneous requirements:
a) directly providing estimates of the param-
eters \( (X_a, X_b, \omega) \) on basis of data samples \( y_n \) ob-
tained from noisy measurement, and
b) providing automatically new estimates as each data sample
is received or providing improved estimate as
updated samples are received.

To formulate the above problem in terms of ar-
tificial neural networks the key step is to con-
struct an appropriate energy or cost function
E(\(\mathbf{x}\)) where \(\mathbf{x} = \left[ X_a X_b \omega \right]^T \), so the lowest energy state will correspond to desired solution \(\mathbf{x}^*\). Mathematically the problem can be formulated as follows: find a vector \(\mathbf{e}\) which minimizes the scalar energy function

\[
E(\mathbf{x}) = \left[ \sum_{n=1}^{N} |e_n|^p \right]^{1/p} = \| \mathbf{e} \|_p
\]

(6)

with \(1 \leq p \leq \infty\), where \(e = [e_1, e_2, \ldots, e_N]^T\).

For the above problem there are three important for practice special cases:
1) for \(p = 1\) the estimation problem is referred as least-absolute-deviations (L1-norm) signal model fitting;
2) for \(p = 2\) the problem is standard (L2-norm) least-squares model fitting;
3) for \(p = \infty\) the problem is minimax, Chebyshev or L\(\infty\)-norm model fitting.

The proper choice of the norm used depends on the distribution of the noise error in the sampled data. For normal distribution the best choice will be the least-squares norm. However, if there are outliers and wild noise the L\(\infty\)-norm is preferable. If the noise have uniform distribution the L\(\infty\)-norm may be the most suitable choice [4,8,8].

3. ESTIMATION OF PARAMETERS USING CHEBYSHEV (L\(\infty\)) NORM CRITERION

The problem can be formulated as the minimax (Chebyshev norm) optimization problem: find the parameters \(\mathbf{x}\) which minimize a scalar cost (error) function

\[
E_{\max} (\mathbf{x}, e, \omega_x^* x) = \max_{1 \leq n \leq N} \{|e_n(\mathbf{x}, e, \omega_x^* x)|\}.
\]

(6)

which can be written shortly as

\[
\min_{\mathbf{x}} \max_{1 \leq n \leq N} \{|e_n|\}
\]

(7)

In order to solve the problem we use dynamic gradient system [4,5], i.e. we apply the steepest descent continuous-time optimization algorithm which leads to set of nonlinear equations

\[
\frac{d\mathbf{x}}{dt} = \frac{1}{\tau_1} \sum_{n=1}^{N} S_n \text{sign}(e_n) \sin(\omega_n x T)
\]

(8)

\[
\frac{d\omega}{dt} = \frac{1}{\tau_3} \sum_{n=1}^{N} m T \text{sign}(e_n) I X \cos(\omega_n T) + X_b \sin(\omega_n T) + X_b \sin(\omega_n T)
\]

(10)

where \(\tau_1, \tau_2, \tau_3\) are the positive coefficients representing time constant of integrators and

\[
S_n = \begin{cases} 
1 & \text{if } |e_n| = \max_{1 \leq i \leq N} \{|e_i|\} \\
0 & \text{otherwise.}
\end{cases}
\]

Fig.1 shows a functional block diagram of an artificial neuron-like network which enables for parallel implementation of the above set of equations, and therefore estimation of the parameters in real-time. The network of Fig.1 consist of integrators, summers, multipliers, signum (hard limiter) activation functions and trigonometric (sin, cos) functions generators. The switches S are controlled by special subnetwork called Winner-Take-All (WTA) circuit. The function of the WTA is to select the strongest, i.e. the largest in absolute value instantaneous error [7]. The sign of selected error is transmitted for further processing while the other weaker error signals are completely inhibited by opening corresponding switches. The WTA circuit usually consist of mutually inhibitory links between processing units. Each unit inhibits all others. A collective dynamic competition take place; each unit compares the maximum of the incoming signals to its own strength. As a result of this competition the largest signal is selected. In the network shown in Fig.1 we have used a simple WTA circuit which employs an integrator acting as a maximum finder, i.e. its output signal tracks continuously in time the signal

\[
E(t) := \max_{1 \leq n \leq N} \{|e_n|\}
\]

(11)

This signal is compared with all actual error signals in voltage comparators. The switches are controlled by voltage comparators [5]. Extensive computer simulation experiments have confirmed that the neural network shown in Fig.1 allows to estimate in real-time desired parameters. Furthermore, it has been found that the network exhibits notable robustness since its functionality is not affected by parameter variations in a wide range. For example signum ac-
tivation functions can be replaced by sigmoid (tangent hyperbolic) or linear function without affecting the functionality of the network and its final accuracy.

4. ESTIMATION BASED ON OTHER CRITERIA

It is interesting to note that the neural network shown in Fig.1 can easily be modified to perform estimation of the parameters according to least-absolute values and least-squares criteria. For example by closing all the switches or simply by removing them and the associated WTA circuit the network will act according to the least-absolute-values criterion. Analogously, in order to estimate parameters according to the least-squares criterion all sigmoid activation functions must be replaced by linear functions or simply removed and all switches must be closed.

Summarizing, the proposed neural network is able to estimate desired parameters of the observed noisy sinusoidal signal according to $L_1$, $L_2$ or $L_\infty$ norm criterion. The choice of the criterion depends on distribution of noise in the measured signals.

5. COMPUTER SIMULATION EXPERIMENTS

In order to check validity and performance of the proposed algorithm, the associated network has been extensively simulated on computer. The simulations fully confirmed correctness of the presented approach and good agreement with theoretical considerations has been obtained. Due to limit of space we present here only one illustrative example.

Example

Let us consider sinusoidal signal contaminated by white Gaussian and impulsive noise. The observed signal has the form

$$y(t) = -120\sin(2\pi ft) + 40\cos(2\pi ft) + \epsilon(t)$$

where $f = 50$ Hz and $\epsilon(t)$ is a noise.

---

Fig. 1 Functional block diagram of neural network for estimating parameters of observed signal
It is required to find the parameters on basis of N=20 samples of the signal with sampling interval 0.8·10⁻⁶ seconds. The network was able to find in time less than 50 nanoseconds the following parameters:

\[ X_a = -119.73, \quad X_b = 39.81, \quad \omega = 314.12 \]

which are in good agreement with actual values of parameters of the measured signal. Exemplary trajectories of estimated parameters for another case (Fig. 2) are shown in Fig. 3.

Fig. 2 Observed noisy signal
\[ y(t) = 100 \sin(\omega t) + 75 \cos(\omega t) + e(t) \]
\[ \omega = 314.159 \text{ rad/s} \]

REFERENCES


Fig. 3 (a) Computer simulated trajectories of estimated parameters and error function, (b) trajectories for several initial conditions for \( \omega \) and zero initial conditions for \( X_a \) and \( X_b \).
BLIND IDENTIFICATION IN PRESENCE OF NOISE

Pierre COMON

THOMSON SINTRA, BP 138, F-06561 Valbonne Cedex, France

The problem of blind identification of noisy linear models of general form is addressed. Several aspects of the problem can be studied. The first concerns general statements that can be done independently of any algorithms, such as the definition of optimization criteria, or the determination of conditions of identifiability. The second aspect is more practical and includes the definition of sufficient assumptions, and the design of numerical algorithms. Lastly, some properties can only be accessed via simulations, namely the robustness of an algorithm in presence of non-gaussian unknown noise, or the effect of short data lengths. These questions are analyzed in the paper.

1. INTRODUCTION

In this framework, the following linear statistical model is assumed:

\[ y = M x + v, \]  

(1-1)

where \( x, y \) and \( v \) are random vectors with values in \( \mathbb{R} \) or \( \mathbb{C} \) and with zero mean and finite covariance, and \( M \) is a rectangular matrix with at most as many columns as rows (if \( M \) had more columns than rows, then some components of \( x \) should be incorporated in the noise part, \( v \)).

Given realizations of \( y \), it is desired to estimate both \( M \) and the corresponding realizations of \( x \), under the assumption that vector \( x \) has statistically independent components. However, because of the presence of the noise \( v \), it is in general impossible to recover exactly \( x \), especially if \( v \) is non-gaussian. Thus with the goal of delivering a "best estimate" \( \hat{M} \) of \( M \), we define a "contrast" function that serves as optimization criterion.

The qualifiers "blind" or "myopic" are often used when only the outputs of the system considered are observed, as it is the case in (1-1). Quantities to be estimated are either \( M \) or \( x \), or both. This blind identification problem is referred to as "sources separation" or "Independent Component Analysis" (ICA). When the matrix \( M \) in model (1-1) is Toeplitz triangular, it is actually dealt with a "Blind Deconvolution" (BD) problem, where vector \( x \) contains successive samples of a scalar white stochastic process. If in principle ICA contains BD as a particular case, general-purpose algorithms computing ICA are unmatched to BD precisely because the structure of the mixing matrix is not imposed. We shall be concerned in this paper only by the unconstrained ICA problem.

Since the early works on the subject drawing back to 1986 [1] [2] [3], see [12], more attention has been retained by ICA during the last years. It has been utilized for instance for localization problems in narrow-band antenna array processing [4] [5]. ICA-based localization techniques are of High-Resolution (HR) nature, but can be compared to the so-called Esprit algorithm where the mixing matrix is first estimated without taking advantage of the antenna manifold, the directions of arrival being estimated afterwards in a second stage. For wide-band signals, it is probably more appropriate to resort to blind identification techniques in time domain [6] [7] [8], especially for short data lengths.

2. GENERAL STATEMENTS

DEFINITION (2-1)

The ICA of a random vector \( y \) of size \( N \) with finite covariance \( V_y = F \Delta^2 F^t \), where \( \Delta \) is \( p \times p \) diagonal real positive and \( F \) is \( p \times N \) full rank;

(a) \( \Delta^2 \) is the covariance of a \( p \times 1 \) random vector, \( x \), whose components are "the most independent possible", in the sense of the maximization of a given "contrast function".

DEFINITION (2-2)

A contrast is a mapping \( \Psi \) from a space of random vectors, \( \mathcal{E} \), to \( \mathbb{R} \) satisfying the 3 requirements:

- \( \Psi(x) \) depends only on \( p_x \) \( \forall \ x \in \mathcal{E} \), each component \( x_i \) playing the same role.
- \( \Psi \) is invariant by scale change, that is:
  \[ \Psi(Ax) = \Psi(x), \quad \forall \ A \text{ diagonal regular}. \]
- If \( x \) has independent components, then:
  \[ \Psi(Ax) \leq \Psi(x), \quad \forall \ A \text{ regular}. \]

In order for a contrast to be of practical interest, it must also be discriminating:

DEFINITION (2-3)

A contrast will be said discriminating over a set \( \mathcal{E} \) if, for any random vector \( x \in \mathcal{E} \) having independent components, the equality \( \Psi(Ax) = \Psi(x) \) holds only when \( A \) is of the form \( \Lambda F \), where \( \Lambda \) is diagonal and \( F \) a permutation.

If a contrast is discriminating over a set \( \mathcal{E} \), then the
condition \( x \in \mathbb{E} \) determines identifiability conditions in the noiseless case \([10]\ [12]\). Identifiability conditions in the noisy case can hardly be investigated.

Now remark that if a pair \( (F, \Lambda) \) is an ICA of a random vector, \( y \), then so is the pair \( (F', \Lambda') \) with
\[
F' = F \Lambda D P, \quad \Lambda' = P^T \Lambda^{-1} \Lambda P,
\]
where \( \Lambda \) is a scaling \( pxp \) regular diagonal real positive matrix, \( D \) is a \( pxp \) diagonal matrix with entries of unit modulus, and \( P \) is a \( pxp \) permutation. So (2-1) defines an equivalence class of decompositions rather than a single one. It is possible to define a unique representative by adding the following assumptions:

(c) the columns of \( F \) have unit norm;
(d) the entries of \( A \) are sorted in decreasing order;
(e) the entry of largest modulus in each column of \( F \) is positive real.

Each of the constraint above removes one of the degree of freedom exhibited in (2-4). More precisely, (c), (d), and (e) determine \( \Lambda, P, \) and \( D, \) respectively.

Now, for any zero-mean random vector \( y \) with finite variance \( V_y \), we define the standardized variable \( \tilde{y} \) as follows. Denote
\[
V_y = U S^2 U^T
\]
the eigenvalue decomposition of \( V_y \), where \( U \) is \( Nxp \) and \( S \) is \( pxp \) full rank. Then \( \tilde{y} \) is the \( p \)-dimensional random variable with unit variance:
\[
\tilde{y} = S^{-1}U^T y.
\]

It may be shown [10] [12] that the opposite of the mutual information is a contrast, discriminating over the set of non-deterministic standardized random variables having at most one gaussian component:
\[
\Psi(z) = -\int_0^1 \log p_z(u) \frac{p_z(u)}{p_z(u)} du.
\]

3. PRACTICAL SUFFICIENT ASSUMPTIONS

In practice, the mutual information cannot be used because the probability density of observations \( y \) is generally unknown. Therefore, it is proposed to resort to contrasts based on moments of \( y \), that we can easily estimate. We look for a matrix \( F \) and a random vector \( z \) such that \( \Psi(p_x) \) is maximized with \( y = Fz \). With \( S \) and \( U \) defined in (2-7), we can set \( F = USQ^T \) and look for the unitary part \( Q \), bearing in mind that \( \tilde{y} = Q^T z \). This can be done by searching for the unitary matrix \( Q \) that maximizes the criterion \( \Psi(Q) = \Psi(p_x) \) [10] [12].

Denote \( \Gamma_{i,j} \) and \( K_{i,j} \), the joint cumulants of \( (\tilde{y}_i, \tilde{y}_j, \ldots, \tilde{y}_i) \) and \( (\tilde{z}_1, \tilde{z}_2, \ldots, \tilde{z}_i) \), respectively, and assume \( \tilde{Z} = Q^T \tilde{Y} \), where \( Q \) is a \( pxp \) unitary matrix. Then because of well-known properties [11] they are related by a multilinear relation. For instance, at order 4 we have:
\[
K_{ijkl} = \sum_{q,r,s} Q_{i,q} Q_{j,r} Q_{k,s} Q_{l,q} \Gamma_{qrsp} (3-1)
\]
A key result is given by the theorem below:

**THEOREM**

The sum of squares of marginal standardized cumulants of order \( r \), denoted
\[
\psi(Q) = \sum_{i=1}^p K_{ii,i,i,i}^2 (3-3)
\]
is a contrast for any \( r \geq 2 \). Moreover this contrast is discriminating over the set of random vectors having at most one null marginal cumulant of order \( r \), for any \( r > 2 \).

**Proof**

Let \( Q \) be \( pxp \) unitary. Denote \( \tilde{Q} \) the matrix defined by \( \tilde{Q}_{ij} = iQ_{ij \cdot} \) and \( \tilde{Q} \) the matrix defined by \( \tilde{Q}_{ij} = Q_{ij} \). Since \( Q \) is unitary, the matrix \( \tilde{Q} \) is doubly stochastic, i.e. the sum of its entries in any row or column is equal to 1. Yet from Birkhoff theorem, the set of doubly stochastic matrices is a convex polyhedron whose vertices are permutations. In other words \( \tilde{Q} \) can be decomposed as
\[
\tilde{Q} = \sum_{i=1}^p \alpha_i P_i, \quad \alpha_i \geq 0, \quad \sum \alpha_i = 1 \quad (3-4)
\]
where \( P_i \) are permutation matrices. Thus for any vector \( u \), if we denote \( \tilde{u} \) the vector with components \( u_i \), we have the inequality
\[
\|\tilde{Q} \cdot u\| \leq \sum \alpha_i \|P_i \cdot \tilde{u}\| = \|\tilde{u}\|. \quad (3-5)
\]
On the other hand, since \( \tilde{Q}_{ij} \leq 1 \), we also have for \( r \geq 2 \):
\[
\|\tilde{Q}_{ij} \cdot u\| \leq \|\tilde{Q}\| \|u\| \quad (3-6)
\]
Applying the triangular inequality yields, using (3-6):
\[
\sum_{i,j,k} \tilde{Q}_{ik} \tilde{Q}_{jk} \tilde{u}_i \tilde{u}_j \leq \|\tilde{Q}\| \|\tilde{u}\| \|\tilde{u}\| \leq \|\tilde{Q}\| \|\tilde{u}\|^2 \leq \|\tilde{Q}\| \|\tilde{u}\|^2 \leq \|\tilde{u}\|^2 = \|u\|^2. \quad (3-7)
\]
Now consider a standardized random vector \( \tilde{y} \) having zero cross-cumulants of order \( r \) and denote \( u_i \) its marginal cumulants of order \( r \). Because of the multilinearity of cumulants (3-1), the very left hand side of (3-7) is nothing else than \( \Psi(Q, \tilde{y}) \). So we have just proved with (3-7) that \( \Psi(Q, \tilde{y}) \leq \Psi(\tilde{y}) \) for any unitary matrix, \( Q \). This shows that (3-3) is indeed a contrast, as soon as \( r \geq 2 \).

Note that for \( r=2 \), the equality \( \|\tilde{Q} \cdot u\|^2 = \|u\|^2 \) always holds, so that the contrast is actually degenerated. It remains to prove that (3-3) is discriminating for any \( r>2 \).

Assume equality in (3-7). Then in particular
\[
\|\tilde{Q} \tilde{u}\|^2 = \|\tilde{u}\|^2 \quad (3-8)
\]
for some vector \( u \) having at most one null component. But because of (3-6), this implies that all the differences involved in (3-8) individually vanish:
\[
\tilde{Q} \tilde{u}_i^2 - \|\tilde{u}\|^2 \tilde{u}_i^2 = 0, \quad \forall i. \quad (3-8)
\]
Again, because \( \tilde{u}_i, \tilde{Q}_{ij}, \) and \( \tilde{Q}_{ij} \) are positive, this yields
\[ (\tilde{Q} \tilde{u}) \_j = \tilde{Q} \_j \tilde{u} \_j = 0, \forall i, \quad (3-9) \]
and next
\[ (\tilde{Q} \tilde{q} \_j) = (\tilde{Q} \_j \tilde{q} \_j) = 0, \forall i,j. \quad (3-10) \]
Consequently, for all values of \( i \), and for at least \( \rho - 1 \) values of \( j \), we have \( \tilde{Q} \_j = \tilde{Q} \_j \). Since \( 0 \leq j \leq 1 \), \( \tilde{Q} \_j \) is necessarily either zero or one for those pairs \((i,j)\). Now, a \( \rho \) by \( \rho \) doubly stochastic matrix \( \tilde{Q} \) containing only zeros or ones in a \( \rho \) by \( \rho - 1 \) submatrix is a permutation. The matrix \( Q \) is thus equal to a permutation up to multiplication of complex numbers of unit modulus. This may be written as \( Q = D \tilde{P} \), where \( D \) needs only be diagonal with unit modulus entries. This proves the second part of the theorem. \( \square \)

It can be shown [10] [12] that the practical contrast \((3-3)\) is actually an approximation of \((2-8)\) in the Edgerton sense.

5. EFFECT OF NON GAUSSIAN NOISE

In this section, matrix \( M \) is a Toeplitz circulant matrix with as first row
\[ m = [3 \ 0 \ 2 \ 1 \ -1 \ 1 \ 0 \ 1 \ -1 \ -2]. \]
In order to observe the effect of additive noise, consider the following observation model:
\[ y = (1 - \mu) M x + \mu \eta \ll m \ll w, \quad (5-1) \]
where \( \mu \) is the noise rate, \( 0 \leq \mu \leq 1 \). In order to facilitate the definition of signal to noise ratio, it is assumed that signal and noise kurtosis are equal in modulus. Thanks to the scaling factor \( \ll m \ll \), a signal to noise ratio of 0 dB is thus obtained for \( \mu = 0.5 \).

There are several ways to access performances of ICA. It has been chosen to use a distance measure between \( M \) and \( F \) that is invariant by postmultiplication by a factor \( AP \), where \( A \) is diagonal regular and \( P \) a permutation. This measure has been called a "gap" because it is not a distance in the strict sense, and is defined as follows:
\[ e(\tilde{\Theta}, \tilde{\Theta}) = \sum_i \sum_j | \tilde{\Theta}_{ij} - 1 |^p + \sum_i \sum_j | \tilde{\Theta}_{ij} - 1 |^q \]
\[ + \sum_i \sum_j | \tilde{\Theta}_{ij} - 1 |^r + \sum_i \sum_j | \tilde{\Theta}_{ij} - 1 |^r, \quad (5-2) \]
where \( \tilde{\Theta} = \tilde{\Theta}^{-1}, \tilde{\Theta} = \tilde{\Theta}^{-1} \), with \( \tilde{\Theta}_{kk} = \| \tilde{\Theta} \|, \tilde{\Theta}_{kk} = \| \tilde{\Theta} \|, \) and
\[ D = \tilde{\Theta}^{-1} \tilde{\Theta}. \]
This gap vanishes if and only if \( \tilde{\Theta} \) and \( \tilde{\Theta} \) are linked by a relation [10] [12] of the form \( \tilde{\Theta} = \tilde{\Theta} A \); in other words, \( (5-2) \) defines a distance between equivalence classes.

In section 6, the algorithm \((4-1)\) is utilized to compute an expectation of \( e(M,F) \) for various values of \( \mu \) and data length \( T \). But in the present section, asymptotic performances are analyzed (i.e. for infinite \( T \)). For this purpose, it is not necessary to generate real data, and simulations can be carried out directly from cumulants. This procedure is referred to as "validation algorithm" in this paper. Let the matrix
\[ A_\mu = [ (1-\mu) M \mu \text{limit} \ I]. \quad (5-3) \]
Denote \( \Gamma_x \) and \( \Gamma_w \) the vectors containing the marginal standardized cumulants of \( x \) and \( w \), respectively, and denote \( \Gamma \) the vector \( \Gamma = [\Gamma_x \Gamma_w] \). Then the validation algorithm can be designed [12] by replacing steps 1 and 4a of the algorithm \((4-1)\) described in section 4 by the following:

**Validation algorithm**

\[ (5-4) \]
1) Compute the SVD of \( A_\mu \) as \( A_\mu = U SV^* \), where \( V \) is \( N \times N \), \( S \) is \( p \times p \), and \( U \) is \( 2Np \times Np \). Set \( A = U^* \) and \( L = VS \).
2) Calculate the required cumulants using the multilinearly relation \((3-1)\).

Figure 1 represents the gap \( e(M,F) \) obtained when running the validation algorithm for 2 values of \( \Gamma_x \):
\[ \Gamma_x = -1 \{ 1 \ 1 \ 1 \ 1 \ -1 \ -1 \ -1 \ -1 \}; \] solid line;
\[ \Gamma_x = -1 \{ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \}; \] dashed line;
\[ \Gamma_x = -1 \{ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \} \text{ in both cases.} \]
It can be noticed that the variation around 0 dB is very steep, showing that ICA is very robust against non-gaussian noise.

Figure 2 shows initial and final values of the contrast for the same two simulations. The values obtained can be compared to the contrast of sources alone, $||\Gamma_x||^2=14.4$, or noise alone, $||\Gamma_w||^2=14.4$. It can be checked out that these values are indeed reached for $\mu=0$ and $\mu=1$.

![Figure 1](image1.png)

![Figure 2](image2.png)

6. ADDITIONAL EFFECT OF DATA LENGTH

When cumulants are estimated from data of limited size, estimation errors enter in the process exactly as an extraneous noise. The shorter the data length, the larger the extraneous noise. This is observed in figure 3, where the effect of data length becomes very sensitive below T=300. For instance, the computation of the ICA in the noiseless case (curve $\mu=0$) for $T=100$ is equivalent to the computation of the ICA with $T=300$ with an additive uniform noise of 40% (curve $\mu=0.4$).

For convenience, simulations have been run for N=2 in order to limit the computational burden. In fact, many trials are necessary for statistical averaging purposes (the product between $T$ and the number of averages has been chosen to be 24000 in this simulation). The mixing matrix was $M = \begin{bmatrix} 1 & 3 \\ -3 & 1 \end{bmatrix}$, with $y = (1-\mu)Mx + \mu w$.

Sources $x_i$ and noises $w_i$ were independent zero-mean standardized random variables uniformly distributed in $[-\sqrt{3}, \sqrt{3}]$; they have thus all a kurtosis of $-1.2$.

![Figure 3](image3.png)

REFERENCES


1: This work has been supported in part by the D.R.E.T., Paris.
THE $L^\infty$ WIENER PROBLEM: AN APPLICATION OF THE $L^\infty$ EXTENSION PROBLEM

Cristian J. Fajre and Clifford T. Mallis

Department of Electrical and Computer Engineering, University of Colorado at Boulder, Boulder, CO 80309-0425, U.S.A.

This work presents a solution to the $L^\infty$ Wiener problem, $\min_H \|F - GH\|_\infty$, where $F(x), G(x)$ and $H(z)$ are rational transfer functions. The norm used is $\|F\|_\infty = \sup_x |F(ze^j\omega)|$. The solution is computed by reducing this problem to an $L^\infty$ Extension Problem, which can be solved by applying the properties of Hankel operators and their Schmidt pairs. A closed form solution is computed, using efficient techniques for the spectral factorization of polynomials.

1. INTRODUCTION

The problem of minimizing $\|F - GH\|$ when the norm is the 2-norm is a typical equalization problem, when we want to minimize the energy of the error. The approach we now present corresponds to the minimization of the maximum absolute value of the error in our approximation.

Introducing some properties of All-Pass functions, we can express the minimization equation as

$$\min_H \|F - GH\|_\infty = \min \|\tilde{F}_2 + H_2\|_\infty,$$

where $\tilde{F}_2(z)$ is a known anticausal rational function and $H_2(x)$ is a causal rational function. The left-hand side of the previous equation corresponds to the family of $L^\infty$ Extension Problems, as stated in [1]. This problem has been solved in [2] using the state variable description of the system, and it is considered as an approximation of anticausal functions by causal functions. Throughout this work, $F(x)$ is defined as in [3],

$$F(x) = \sum_{k=-\infty}^{\infty} h_k x^{-k}.$$

2. THE APPROXIMATION PROBLEM

The statement of our approximation problem is,

- given: $F(x), G(x),$
- find: $H(x)$, causal,
- such that: $\|F - GH\|_\infty$ is minimized, \hspace{1cm} \hspace{1cm} (1)

where $F$ and $G$ are given as general transfer functions. In order to solve this problem, consider the factorization of $G(x)$ as

$$G(x) = G_0(x) E(x),$$

where $G_0$ is completely causal and minimum phase, and $E(x)$ is an all-pass transfer function, such that $E(x)E(x^{-1}) = 1$. Now set

$$H(x) = \frac{H_0(x)}{G_0(x)},$$

with $H_0$ causal, hence $H$ will also be causal. Replacing these two equations back into the norm equation, we have

$$\|F - GH\|_\infty = \|F \tilde{E} - H_0\|_\infty,$$

where $\tilde{E}(z) = E(z^{-1})$ and then $E \tilde{E} = 1$. The function $F \tilde{E}$ can be represented as

$$F(x) \tilde{E}(x) = F_1(x) + \tilde{F}_2(x),$$

where $F_1$ is causal and $\tilde{F}_2$ is anticausal. If we now set $H_0(x) = F_1(x) - H_2(x)$, with $H_2$ a causal function, we will have

$$\|F - GH\|_\infty = \|\tilde{F}_2 + H_2\|_\infty.$$

In order to solve the causal approximation problem, we have to find a causal extension function $H_2$ to the anticausal tail $\tilde{F}_2$. Thus, the solution for the approximation problem stated in (1) is

$$H(x) = \frac{F_1(x) - H_2(x)}{G_0(x)}.$$

We have to consider two cases for eqn. (4). We will talk about the Causal Extension when $F_2$ is an FIR, and about the Semi-infinite Interval Extension when $F_2$ is a rational IIR.

3. THE CAUSAL EXTENSION

In this case we have $\tilde{F}_2(z) = \sum_{k=-n}^{1} f_k z^{-k}$, which means that we are looking for a function $H(z)$ that satisfies

$$H(z) = \tilde{F}_2(z) + H_2(z) = \sum_{k=-n}^{1} f_k z^{-k} + \sum_{k=0}^{\infty} h_k z^{-k},$$

yielding the minimum $\infty$-norm. This can be accomplished if we consider a theorem by Carathéodory and Fejér, which states...
Theorem 3.1 Given the sequence \( \{f_0, \ldots, f_n\} \), construct the matrix \( H(f) \) as
\[
H(f) = \begin{bmatrix} f_n & \cdots & f_0 \\ \vdots & \ddots & \vdots \\ f_0 & & 0 \end{bmatrix}.
\] (7)

Now let \( \mu_0^n \) be the largest eigenvalue of \( HH^* \), then, every \( F(x) = \sum_{k=0}^{n} f_k x^{-k} \) satisfies \( \|F\|_\infty \geq \mu_0 \). Equality holds if, and only if,
\[
F(x) = \mu_0 e^{\gamma r} \prod_{k=1}^{r} \frac{x^{-1} - u_k}{1 - u_k x^{-1}},
\]
for \( \gamma \) a real constant, \( r \leq m \), and \( |u_k| < 1 \).

This minimizing function can be constructed as
\[
F(x) = \mu_0 \frac{u_0 + \cdots + u_m x^{-m}}{u_0 + \cdots + u_m x^{-m}} = \mu_0 \frac{U(x)}{U(z)},
\] (8)
where \( \{u_0, \ldots, u_m\} \) is an eigenvector corresponding to \( \mu_0 \). Clearly, the solution function is an all-pass transfer function, since it has a constant value over the unit circle. This result and the proof of the theorem can be found in [1, 4, 6, 8].

With this in mind, we can see that the solution for the extension problem in eqn. (6), is \( H(x) = x^n F(x) \), using the sequence \( \{f_0, \ldots, f_{n-1}\} \) to construct \( H \) and \( F(x) \), as shown in equations (7) and (8).

4. SEMI-INFINITE INTERVAL EXTENSION

Suppose now that \( \tilde{F}_2 \) is a rational function, with denominator degree greater than zero. Repeating the analysis done for eqn. (6), we are now looking for a function \( H \) that satisfies
\[
H(x) = \tilde{F}_2(x) + H_2(x) = \sum_{k=-\infty}^{-1} f_k x^{-k} + \sum_{k=0}^{\infty} h_k x^{-k},
\] (9)
and yields minimum \( \infty \)-norm. This shows clearly that in order to use the same procedure as in Section 3., we would need the eigenvalue decomposition of the infinite Hankel matrix constructed from the sequence \( \{\ldots, f_2, f_1\} \). Instead, we can make use of the theory of Hankel Operators presented in [7, 8], and consider Nehari's theorem, which shows that the solution of this problem is of the form
\[
H(x) = \mu_0 \frac{z V(x^{-1})}{U(z)},
\] (10)
where the functions \( V \) and \( U \) are constructed from the pair of infinite dimensional vectors \( (u, v) \), which constitute the Schmidt Paill corresponds to the largest singular value of the Hankel operator, \( \mu_0 \), as defined in [7, 8].

If we now apply this to eqn. (9), and considering real coefficients for the Hankel operator, we have that
\[
\tilde{H}(x) = \tilde{F}_2(x) + H_2(x) = \mu_0 \frac{z A(x^{-1})}{A(z)},
\] (11)
with \( A(z) = \sum_{k=0}^{\infty} a_k z^{-k} \). Recalling that \( \tilde{F}_2(x) = F_2(x^{-1}) \), and that \( H_2(x) \) is completely causal, we get that
\[
z F_2(z) A(x^{-1}) = \mu_0 A(z), \text{ in } [0, \infty).
\] (12)

If we now set
\[
F_2(z) = \tilde{A}(z), \quad A(z) = \tilde{A}(z), \quad H_2(z) = \tilde{A}(z),
\] (13)
where \( \tilde{A}(z) \) is of degree \( n \), and \( \tilde{A}(z) \) and \( \tilde{A}(z) \) are of degree \( n - 1 \). Equation (12) becomes
\[
\mu_0 \frac{\tilde{A}(z)}{\tilde{A}(z)} = \tilde{A}(z) \tilde{A}(z) = \tilde{A}(z) Y(z), \quad \text{in } [0, \infty),
\] (14)
where \( \tilde{A}(z) = z^{-n} \tilde{A}(z^{-1}) \), and \( \tilde{A}(z) = z^{-n+2} \tilde{A}(z^{-1}) \). This yields the matrix equation
\[
\mu_0 \gamma \gamma = \frac{A}{U} = \frac{M}{Z},
\] (15)
where
\[
\gamma = \begin{bmatrix} \gamma_0 \\ \vdots \\ \gamma_{n-1} \end{bmatrix},
\]
\[
A = \begin{bmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ \gamma_{n-1} & \cdots & \gamma_{n-1} \end{bmatrix},
\]
\[
Y = \begin{bmatrix} \gamma_0 & \cdots & \gamma_{n-1} \\ \vdots & \ddots & \vdots \\ \gamma_{n-1} & \cdots & \gamma_0 \end{bmatrix}, \quad \text{and}
\]
\[
J = \begin{bmatrix} 0 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 0 \end{bmatrix}.
\]

Equation (15) asks for the solution of an eigenvalue problem, but this time the matrices involved are of finite dimension. Once we have calculated \( \mu_0 \) and \( \gamma \), we get
\[
\tilde{f}(z) = \mu_0 \frac{z A(x^{-1})}{A(z)} = \mu_0 \frac{\tilde{A}(z)}{\tilde{A}(z)} \text{ and}
\]
\[
H_2(z) = \tilde{f}(z) - \tilde{F}_2(z),
\] (16)
which is our solution.

Figure 1: Impulse Response of \( H(z) \). The given points are marked in the picture with a \((\times)\).
5. RESULTS

5.1. Causal Extension

In this case we take a sequence of values and compute the rational function with impulse response starting with that sequence, as shown in Figure 1. This is done by computing the eigenvalue with maximum absolute value and its corresponding eigenvector. This case is interesting in order to observe what happens when the eigenvalue has multiplicity greater than one. As shown in Figure 2, the polynomials corresponding to these eigenvalues only differ on roots located on the unit circle, as presented in [9, 1]. This property shows that the final order of the resulting all-pass function will be \( n - p + 1 \), when the length of the sequence is \( n + 1 \) and the multiplicity of the eigenvalue is \( p \).

![Figure 2: Roots of \( U_0(x) \): (x), and \( U_1(x) \): (+).](image)

5.2. Semi-Infinite Interval Extension

In order to actually compute the coefficients of the rational solution we apply the techniques for factorization of polynomials presented in [10]. Based on this, we could implement a computational toolkit oriented to operations involving rational functions. This toolkit is also used to compute the \( L^1 \) approximation, which is shown in Figure 3, along with the \( L^\infty \) result and the desired function. In this case we took \( G(x) = 1 \).

The characterization of the best approximation in \( L^\infty \) is that the error curve is perfectly circular, as stated in [11, 12, 13], and this can be observed in Figure 4.

![Figure 3: Amplitude response of \( F(x) \): (---), \( H_{inf}(x) \): (---), and \( H_W(x) \): (---).](image)

![Figure 4: Error Curves for \( H_{inf}(x) \): (---), and \( H_W(x) \): (---).](image)

REFERENCES


A Non-Parametric Approach for Detecting Changes in the Autocorrelation Structure

Michail K. Tsatsanis and Georgios B. Giannakis

Department of Electrical Engineering
University of Virginia
Charlottesville, VA 22903-2442

Abstract — In this paper we derive a new test for detecting a jump in the autocorrelation function of a signal. The new test is simple, non-parametric, and does not assume knowledge of the signal’s p.d.f. We show consistency and asymptotic normality of the statistic and we derive expressions for the asymptotic variance. In this way we are able to assess the performance of the test and set the threshold in an optimal way.

I. INTRODUCTION

The detection of an abrupt change in a time series is an important engineering problem and has attracted the interest of many researchers over the past 30 years. It has numerous applications in quality control procedures and failure detection, (e.g. in ECG and econometric signals). It is also used for segmentation of non-stationary signals, (e.g. speech and seismic signals). Finally, in image processing, it can be used for texture segmentation, which is a basic step in every image understanding procedure.

Early work on this problem has focused on detecting the jumps in the mean of i.i.d. Gaussian observations (e.g. see [1] for a review). This formulation is rather restrictive and does not address the more interesting problem of changes in the spectral characteristics of a time series. Parametric techniques have been employed to solve this more general problem. Changes in the parameters of ARMA or state-space models can be detected using tests on the innovations of the associated whitening filter. If the alternative model is a-priori known, then a dynamic programming algorithm can be used to determine the transition times, [2]. If not, local tests must be used considering small jumps, [3]. These parametric tests are rather involved, assume Gaussian data, lack the generality of the non-parametric ones and require knowledge of the underlying model.

Non-parametric tests have been derived in [5] using evolutionary spectra. Unfortunately, these tests can not be used to detect the exact transition time, due to the windowing inherent in the estimation of the evolutionary spectra.

In this paper we propose time-domain, non-parametric tests for detecting jumps in the autocorrelation function of a time series. These tests can be easily generalized to detect changes in higher-order statistics of the process. They require no knowledge of the signal’s p.d.f. and are computationally simple.

In the next section we will state the problem and derive the test from a mean square criterion. We will show consistency of the test as the data length goes to infinity. In Section III we will consider the threshold selection problem and we will discuss the performance of the test. Finally, in Section IV we will show some preliminary simulations and in Section V we will draw the conclusions.

II. Problem Statement-Results

Let \( \{x(t)\}_{t=0}^{T} \) be a given record of \( T + 1 \) samples from the process \( x(t) \) with time-varying autocorrelation \( r(t; \tau) = E[x(t)x(t+\tau)] \). More specifically we wish to discriminate between the following hypotheses

\[
H_0 : r(t; \tau) = c , \quad t = 0, \ldots, T - \tau
\]

\[
H_1 : r(t; \tau) = \left\{ \begin{array}{ll}
  c & t = 0, \ldots, s - \tau - 1 \\
  c + \nu & t = s, \ldots, T - \tau
\end{array} \right.
\]

so that \( H_0 \) corresponds to a constant autocorrelation, while \( H_1 \) corresponds to a jump in \( r(t; \tau) \) at time point \( s \).

Notice that this setup considers only one lag of the autocorrelation function. As we will see later though, the extension to include more lags is rather straightforward. Also, notice that we have not specified the value of \( r(t; \tau) \) for \( t = s - \tau, \ldots, s - 1 \), (the transition period). There is no general rule about how \( r(t; \tau) \) goes from \( c \) to \( c + \nu \) as this depends on the specific application. A linear slope, or even a constant value between \( c \) and \( c + \nu \) may, in general, provide a good approximation. To simplify computations, we will assume \( r(t; \tau) = c , \quad t = s - \tau, \ldots, s - 1 \) in the sequel. The error introduced is anyway small, since both \( \nu \) and \( \tau \) are small numbers.
In order to estimate the time varying autocorrelation \( r(t; \tau) \) and test the hypotheses we consider the mean square criterion

\[
J(t, s) = E\{ [x(t)x(t + \tau) - \theta_s(t; \tau)]^2 \}
\]

(3)
depending on the parameter \( \theta_s(t; \tau) \). Since we are not interested on the precise value of the autocorrelation but rather on the jump, we modify (3) to a mean compensated criterion

\[
\hat{J}(t, s) = E\{ [x(t)x(t + \tau) - \hat{r}(\tau) - \theta_s(t; \tau)]^2 \}
\]

(4)

where

\[
\hat{r}(\tau) = \frac{1}{T - \tau + 1} \sum_{t=0}^{T-\tau} r(t; \tau)
\]

(5)

Further, we will try to minimize

\[
J_T(s) = \frac{1}{T - \tau + 1} \sum_{t=0}^{T-\tau} J(t, s)
\]

(6)

Assume for the time being that \( c \) and \( \nu \) are known and only \( s \) is to be determined. After substituting (4) to (6), expanding the square and dropping terms which do not depend on \( s \) we get

\[
J_T(s) = \frac{1}{T - \tau + 1} \sum_{t=0}^{T-\tau} [\theta_s(t; \tau)]^2 \]

(7)

\[
- \frac{2}{T - \tau + 1} \sum_{t=0}^{T-\tau} (r(t; \tau) - \hat{r}) \theta_s(t; \tau)
\]

(8)

(7) represents a matched filter based detector in the autocorrelation domain. It can be further simplified by scaling the \( \theta_s(t; \tau) \) to have the same energy for all \( s \), and omitting the first term. Also \( \hat{r} \) does not really have to be subtracted, since now \( \theta_s(t; \tau) \) has zero-mean.

After these simplifications the criterion to be maximized is

\[
J_T(s) = \frac{1}{T - \tau + 1} \sum_{t=0}^{T-\tau} r(t; \tau) \theta_s(t; \tau)
\]

(9)

where

\[
\theta_s(t; \tau) = \begin{cases} 
-A_s, & t = 0, \ldots, s-1 \\
B_s, & t = s, \ldots, T - \tau
\end{cases}
\]

(10)

and \( A_s, B_s > 0 \) are such that \( \frac{1}{(T - \tau + 1)} \sum_{t=0}^{T-\tau} \theta_s(t; \tau) = 0 \), \( \frac{1}{(T - \tau + 1)} \sum_{t=0}^{T-\tau} [\theta_s(t; \tau)]^2 = 1 \), i.e.,

\[
A_s = \sqrt{\frac{(T - s + \tau + 1)}{s}}, \quad B_s = \sqrt{\frac{s}{(T - s + \tau + 1)}}
\]

(11)

Notice that under \( H_0 \), \( J_T(s) = 0 \), for all \( s \), so that (9) can indicate the presence of a change.

We estimate \( J_T(s) \) by

\[
\hat{J}_T(s) = \frac{1}{T - \tau + 1} \sum_{t=0}^{T-\tau} \hat{x}(t)x(t + \tau) \theta_s(t; \tau)
\]

and select \( s \) as

\[
\hat{s} = \arg\max_{0 < s < T - \tau} \hat{J}_T(s)
\]

(12)

IIIA. Consistency

Next we show that \( J_T(s) \) is a meaningful estimator of \( J_T(s) \) which converges as \( T \to \infty \). We follow the approach taken in [3] to prove ergodicity.

Proposition 1: If \( x(t) \) is a general non-linear process of the Volterra type, with absolutely summable cumulants of all orders, then

\[
|J_T(s) - J_T(s)| \to 0
\]

(13)
as \( T \to \infty \) and \( s/T \to a \in (0, 1) \). □

The proof follows the steps of [3] although in a somehow different setting. We skip the proof here.

IV. Performance Analysis

Up to this point we have seen that \( J_T(s) = 0 \) under \( H_0 \), while it peaks at the transition time under \( H_1 \). This suggests that we should compare \( J_T(s) \) to some threshold \( \lambda \), to decide in favor of one hypothesis. In this section we will discuss the threshold selection as well as the performance of the test.

The threshold can be determined using a Neumann-Pearson approach, (for a given probability of false alarms), if the distribution of the statistic under \( H_0 \) is known. Unfortunately, in this case the distribution of \( x(t) \) is unknown and hence, only asymptotic analysis seems to be feasible. Our analysis is based on the following result, which establishes asymptotic normality of the estimate \( J_T(s) \).

Proposition 2: Under the assumptions of proposition 1,

\[
\sqrt{T - \tau + 1} [J_T(s) - J_T(s)] \to \mathcal{N}(0, \sigma^2)
\]

(14)
as \( T \to \infty \), where \( \mathcal{N} \) denotes convergence in distribution. Furthermore, if \( x(t) \) is finite dependent the variance \( \sigma^2 \) can be computed as

\[
\sigma^2 = \lim_{T \to \infty} E\{ [\sqrt{T - \tau + 1} [J_T(s) - J_T(s)]]^2 \}
\]

(15)

The proof is again along the lines of [3] and is omitted here.

Thus, we need to derive expressions for the mean and the variance of \( J_T(s) \) under the different hypotheses to conclude our discussion on the performance analysis.

We again present only the final results.

1) Under \( H_0 \), \( E[J_T(s)] = 0 \) and

\[
E\{[J_T(s)]^2\} = \left( \frac{A_s}{T - \tau + 1} \right)^2 \sum_{t=-\infty}^{T-1} (s - |t|) \mu_4(\tau, t, t + \tau)
\]

(16)
for \( s > (T - \tau)/2 \), with a similar expression for \( s < (T - \tau)/2 \). Here \( \mu_\delta(\tau_1, \tau_2, \tau_3) = E\{x(t)\bar{x}(t + \tau_1)\bar{x}(t + \tau_2)\bar{x}(t + \tau_3)\} \). By using (18) to compute the variance and exploiting the asymptotic normality of \( J_T(s) \), we can set the threshold, given a false alarm rate, from the tables of the normal distribution.

To compute the power of the test, we need the mean and variance of the statistic under \( H_0 \).

ii) Under \( H_0 \),

\[
E[\hat{J}_T(s)] = \nu(\sqrt{s(T - \tau - s + 1)})(T - \tau + 1) \tag{17}
\]

and

\[
E[[\hat{J}_T(s)]^2] = \left( \frac{A_s}{T-t+1} \right)^2 \sum_{t=T-t+1}^{T-t-s} \mu_\delta^2(\tau, t + \tau) \tag{18}
\]

\[
+ \left( \frac{B_s}{T-r+1} \right)^2 \sum_{t=T-r+1}^{T-r-s} \mu_\delta^2(\tau, t + \tau) - \frac{2A_sB_s}{(T-t+1)(T-r+1)}(s)(c + \nu) .
\]

The superscript (1) above refers to the statistics of the process before the jump while (2) refers to the process after the jump. The expression in (18) is derived assuming that the processes before and after the jump are independent.

IV. Extensions

The test can be easily extended to check for change in more than one autocorrelation lag. Let \( \hat{J}(s) \) be a vector containing \( p \) statistics of the form (11) for \( p \) different autocorrelation lags. Let \( \hat{\Sigma} = \frac{1}{T} \Sigma \hat{J}(s) \hat{J}(s)^T \). Then the quadratic form \( q(s) = \hat{J}(s)^T \Sigma \hat{J}(s) \) is a chi-square r.v. with \( p \) degrees of freedom under \( H_0 \), while it is a non-central chi-square under \( H_1 \). Hence, probabilities of detection and false alarms can be established.

Also, the criterion of (8) can clearly be extended to test jumps in higher-order statistics of the process. For example, for third-order cumulants we should use

\[
\hat{J}_2^3(s) = \frac{1}{T-t+1} \sum_{t=0}^{T-t} E\{x(t)\bar{x}(t + \tau_1)\bar{x}(t + \tau_2)\} \theta_3(t; \tau)
\]

where \( \tau_1, \tau_2 > 0, \tau = \max(\tau_1, \tau_2) \). Other extensions may include deriving a similar test in the frequency domain.

V. Simulations

We present two examples to illustrate the implementation issues. The first is a synthetic example with computer generated data, while the second shows an application on real ECG data.

In Fig. 1 we show a realization of an MA (2) Gaussian process generated by the non-minimum phase filter \( B_2(z) = 1 - z^{-1} + 2z^{-2} \) driven by standard normal i.i.d. r.v.'s. In Fig. 2 we show a realization of the same process for \( t = 0, \ldots, 500 \), and of a different MA process with \( B_2(z) = 1 - z^{-1} + 3z^{-2} \) for \( t > 500 \).

In Fig. 3 we see the statistic of (11) for \( \tau = 0 \), computed from the signal of Fig. 1. The statistic is close to zero, since the process is stationary. In the dashed line, we see the threshold that we should use, for a false alarm rate of 0.01. This threshold is computed using the variance expression (18). In Fig. 4 we see the statistic for the non-stationary signal of Fig. 2, which peaks at the jump point. We also show the threshold. In the second example we apply the non-parametric test on ECG data to monitor the variability of the heart beat rate. It is well known in the medical community that, if an infarction is developing in the body, the heart beat rate becomes fairly constant, compared to the (more erratic) healthy one. This is very important in on-line monitoring of the patient's condition, especially in neo-natal intensive care units, where infants cannot express their symptoms, yet early diagnosis is crucial for their survival.

The heart rate variability is measured by the covariance of the beat-to-beat time interval (or R-R interval). In Fig. 5 we see the ECG signal of an (initially) healthy baby, that becomes sick. In Fig. 6, we plot the R-R intervals (extracted from the signal of Fig. 5) as a time-series. The test was applied to the time-series using a moving window of length 200 samples. Fig. 7 shows the statistic's value that peaks at the jump point.

V. Conclusions

We have derived a new test for detecting a jump in the autocorrelation function of a time series and we have analyzed its performance. We have proved consistency and asymptotic normality of the statistic. We have also shown its use through simulations.

References


Figure 1: A stationary signal, (no jump)

Figure 2: A non-stationary signal, (jump at t = 500)

Figure 3: $J_T(s)$ for the signal in Fig. 1

Figure 4: $J_T(s)$ for the signal in Fig. 2

Figure 5: A baby's ECG signal

Figure 6: R-R intervals as a time series

Figure 7: The test statistic for $\tau = 0$
Signal Representation using Operators

Klas Nordberg  Hans Knutsson  Gösta Graulund
Computer Vision Laboratory, Department of Electrical Engineering
Linköping University, S-581 83 LINKÖPING, SWEDEN
Phone: +46-13-28 16 34, Telefax: +46-13-28 16 34

Abstract

This paper presents a new way of representing a signal as well as an algorithm for extracting parameters of a specific class of features. The signal representation is based on the observation that we are often able to associate features with operators transforming the signal. The novel results presented in this paper are derived by restricting the operators to continuous one-parameter unitary operators. Under this restriction, it is shown that there exists a simple algorithm for extracting the parameters of the operators. The representation presented in this paper is intended for hierarchical processing structures.

1 Introduction

There are various ways of representing signals and the representation chosen is a reflection of assumptions and intentions regarding the signal. One common representation is the ordered representation, which implies that the signal elements can be ordered relative to each other. Take a sampled audio signal as an example. The elements of this signal correspond to a sequence of samples and each element is ordered according to when it was sampled. In image processing, the signals are often image neighbourhoods and it is therefore natural to assign spatial coordinates to the elements of the signal. Here, each element is ordered according to what spatial position is relative to the others, which implies a two dimensional ordering. The ordered representation allows us to use Fourier analysis as a way of describing features of the signal. In the case of image processing, features such as local orientation and spatial frequency, curvature, etc. can be defined this way.

An other way of representing a signal is to consider it as a vector in a vector space, usually $\mathbb{R}^n$. The signal is then expressed as a linear combination of a set of basis vectors. We will refer to this type of representation as a linear representation. The choice of basis vectors is of course the main issue for this type of representation and at least two strategies for the choice deserve to be mentioned. The first is to minimize the number of basis vectors for the representation, which will allow us to describe a large part of the signal with just a few parameters, i.e. the coordinates of the signal relative to the basis. The second approach results in us being able to interpret the coordinates of the signal in some specific way, for example as an indication of class membership. Template matching is an example of this strategy. Each basis vector will correspond to a template and if the signals coordinate relative to one of the basis vectors is large in magnitude, it is taken as an indication of the similarity between the signal and the corresponding template.

Both ordered and linear representations are used in image processing as well as in other areas of signal processing. However, neither of them have proved fruitful tools in the context of hierarchical processing structures. In such a processing structure, features are represented and parameters extracted at each level and as we ascend through the hierarchy, the features become more and more abstract. In the higher parts of this hierarchy it is reasonable to expect the features detected to be relations between objects and perhaps also descriptions of possible responses the processing system can give for a certain event. As an example we might have to detect and represent the event "there is an obstacle on the road in front of the car, push the brake pedal and steer to the left of the obstacle". For signals describing such events, it will be difficult or even impossible to order its elements. The reason is that the features represented here simply do not correspond to spatial events like straight lines, curves, corners, etc. This may be one reason why it has proved difficult to define higher order features in most hierarchical processing systems.

The linear representation also seems to be a dead end. One reason is that even quite simple signals might have to be represented with a large number of basis vectors. Take as an example a cyclic image sequence showing a man waving his arms. We can construct this sequence so that each pixel of the image will be a continuous function of time. Let us consider the entire image as a signal vector in a vector space whose dimensionality is the same as the number of pixels in the image. As the image sequence is cyclic, the signal vector will move along a closed curve in the signal space. The signal is simple in the sense that we can characterize it using only one parameter which describes where it is on the curve. However, if we use a linear representation for this signal we
would most certainly need a large number of basis vectors since the curve is not embedded in a linear subspace of small dimensionality. From this we infer that the simplicity of a general signal, reflected in the number of parameters needed to characterize it in the signal space, does not necessarily correspond to a simple linear representation of it, i.e. by few coordinates. In the worst case a linear representation for a signal might have to include the same number of basis vectors as the dimensionality of the signal space. Furthermore, this representation exhibits a somewhat more subtle flaw. Even if the signal can be linearly represented with a reasonably small basis, it will in general not be the case that the coordinates of the signal will correspond to our intuitive concept of a feature. Almost all feature extracting algorithms use various non-linear combinations of the coordinates to represent the feature, and there is no general approach to how this should be done.

We can summarize the problems of hierarchical information processing in the following way. First, it may not be possible to define an ordering of the signal elements. Therefore signal analysis based on Fourier methods can not be used as a general approach for feature extraction. Secondly, there must be a uniform way of defining signal features at all levels of the hierarchy. This implies that the features must be encoded in the signal in a very simple way in order to be detected by the processing units. For a thorough discussion of this subject, see [Granlund and Knutsson, 1983]. In the next sections we will present a signal representation which enables a definition of features for the signal in an intuitive way and when mathematically restricted yields a simple algorithm for extraction of the feature parameters.

2 An operator representation

As a tool for developing a new signal model we will use as a signal an image neighbourhood. Since suggested in [Granlund, 1978], it has become a widely accepted hypothesis that a sufficiently small neighbourhood only contains linear structures. Such a structure is characterized by its orientation, frequency and phase. Let us consider a specific instance of this signal. If we change the orientation of the linear structure corresponding to the signal we will generate a second instance of the signal. Changing the frequency will result in a third, etc. Thus, we are able to generate the complete set of signals, observed in the neighbourhood, by continuously changing the three parameters determining the orientation, frequency and phase of the corresponding linear structure. Any change in one of these parameters can be regarded as an operator transforming the signal, i.e. if we want to change the value of the orientation parameter by some amount, one and the same operator can be applied to the signal, regardless of its initial orientation. These operators are functions only of how much we change the parameters and independent of the signal itself.

Let us call the fixed instance of the signal \( v_0 \). According to the previous discussion we can write any signal in the neighbourhood \( v(\theta, \omega, \varphi) \), with orientation \( \theta \), frequency \( \omega \) and phase \( \varphi \), as

\[
v(\theta, \omega, \varphi) = T_\theta(\theta)T_\omega(\omega)T_\varphi(\varphi)v_0,
\]

where \( T_\theta, T_\omega, \) and \( T_\varphi \) are operator changing the values of the orientation, frequency and phase respectively. It is natural to set

\[
T_\theta(0) = T_\omega(0) = T_\varphi(0) = I,
\]

where \( I \) is the identity operator. Hereby, the values of the orientation, frequency and phase of the signal \( v(\theta, \omega, \varphi) \) will be defined relative to the corresponding values of \( v_0 \).

We will now generalize the ideas presented so far. We represent the signal in terms of operators. Each operator corresponds to a feature of the signal. This allows us to see an operator not merely as something changing the value of the signal elements, but rather as something changing the value of a parameter measuring a feature. For a signal which has \( n \) features, which can all be described with operators and each feature being quantified by a real variable \( x_k \), we can then write

\[
v(x_1, x_2, \ldots, x_n) = T_1(x_1)T_2(x_2) \ldots T_n(x_n)v_0,
\]

where \( T_k \) is an operator changing the value of the \( k \)-th feature. We will refer to the signal representation, defined by equation (1), as an operator representation. So far nothing new has been presented. Ideas along these lines have been presented in [Wilson and Spann, 1988], [Kanatani, 1990] and [Lenz, 1990]. In the following sections we will see what will happen if we restrict the operators to a specific type. To be able to formulate the operator representation in a convenient way, we briefly present a few results from matrix algebra.

3 Matrix algebra

Many of the results presented in this section are without any proof, for a more thorough presentation see [Nordberg, 1992]. We begin with some properties regarding the eigenvectors and eigenvalues of anti-Hermitian matrices, i.e. matrices \( H \) for which \( H^* = -H \). Here, the * sign indicates transposition and complex conjugation.

- The eigenvalues of an anti-Hermitian matrix are imaginary, i.e. are of the type \( i\lambda, \lambda \in \mathbb{R} \).

- For an anti-Hermitian matrix \( H \) it is possible to find a set of orthogonal eigenvectors of \( H \), represented by the unitary matrix \( E \), containing the eigenvectors in its columns, such that

\[
H = EDE^*.
\]
where $D$ is a diagonal matrix containing the eigenvalues corresponding to the eigenvectors in $E$.

Here, a matrix $E$ is called unitary whenever $EE^* = I$, regardless of whether $E$ is real or complex. Note also that the matrix $E$ might not be unique, permutations of its columns disregarded. The matrix $D$ however, is unique, permutations disregarded.

Next we will define an exponential function for matrices. For our purposes it is enough to define this function only for anti-Hermitian matrices. For an anti-Hermitian matrix $H = EDE^*$, according to the above, we define

$$e^H = Ee^D E^*.$$ 

As was mentioned above, $E$ might not be unique. However, all matrices $E$ and $D$ satisfying $H = EDE^*$ will produce the same value of $e^H$.

If we denote the diagonal elements of $D$ as $i\lambda_k$, then $e^D$ is also a diagonal matrix and its diagonal elements are $e^{i\lambda_k}$. From this definition of the exponential function we immediately see that $e^H$ must be unitary as it can be factorized into three unitary matrices. Two more properties are now at hand.

- If $H$ is anti-Hermitian, then $e^H$ is unitary.
- If $H$ is real and anti-Hermitian, then $e^H$ is real and orthogonal.

If $H$ is anti-Hermitian, then so is $xH$ for any real $x$. Hence, we can define a continuous mapping from a real parameter $x$ to unitary matrices $T(x)$ in the following way

$$T(x) = e^{xH}.$$ 

In general, matrices, including unitary matrices, do not commute. For two arbitrary unitary matrices $T_1$ and $T_2$, it is thus in general the case that

$$T_1 T_2 \neq T_2 T_1.$$ 

However, if we can write $T_1 = e^{H_1}$ and $T_2 = e^{H_2}$, we have the following proposition.

- $T_1 T_2 = T_2 T_1$ if and only if $H_1 H_2 = H_2 H_1$. If this condition is met, then

$$e^{H_1} e^{H_2} = e^{H_2 + H_1}.$$ 

(2)

4 A restricted operator representation

We will now restrict the operators used in the operator representation of a signal according to equation (1). Two restrictions will be made without any discussion of their validity. For a thorough presentation of this subject, see [Nordberg, 1992]. The two restrictions imposed on the operator are

- The operators are continuous one-parameter unitary operators. To each operator $T_k(x_k)$ we associate an anti-Hermitian matrix $H_k$, such that

$$T_k(x_k) = e^{x_k H_k}.$$ 

(3)

- All operators commute.

The first restriction provides us with a neat mathematical formulation of the operators. We have already stated that features of the signal corresponds to operators which changes the values of the parameters quantifying the features. Both a feature and an operator are quite general and abstract concepts. Now we can associate a mathematical object, an anti-Hermitian matrix, to each feature. Properties of the feature will then be reflected in properties of the matrix. We can combine equations (1) and (3), giving

$$v(x_1, x_2, \ldots, x_n) =
\begin{align*}
e^{x_1 H_1} e^{x_2 H_2} \cdots e^{x_n H_n} v_0.
\end{align*}$$ 

(4)

The second restriction imposed on the operators allows us to use equation (2) to rewrite equation (4) as

$$v(x_1, x_2, \ldots, x_n) =
\begin{align*}
e^{x_1 H_1 + x_2 H_2 + \ldots + x_n H_n} v_0.
\end{align*}$$ 

(5)

According to equation (5) we have thus arrived to a linear representation of the signal. However, it is a linear representation of the operators transforming the signal, not of the signal itself. The parameters $x_k$ can therefore be interpreted as feature coordinates of the signal. They define a point in the linear space of anti-Hermitian matrices, which in turn corresponds to a unitary operator, transforming $v_0$ to $v(x_1, x_2, \ldots, x_n)$.

5 Extracting the features

This section will describe an algorithm for the extraction of feature parameters. It is a generalization of the algorithm for estimation of local orientation as defined by [Knutsson, 1982], see [Nordberg, 1992]. We assume that the signal can be represented according to equation (5). We will consider only a signal which can be described with one operator corresponding to one feature of the signal. However, the algorithm presented here can be extended to multiple features, see [Nordberg, 1992]. Thus, we express the signal as

$$v(x) = e^{Hv_0}.$$ 

(6)

Our goal now is to find a way of computing the real parameter $x$. There is of course no unique way to achieve this task and we will concentrate on one method. It is based on the assumption that we can find a matrix $X$, such that

$$\nabla v(x) X v(x) = re^{ixz},$$ 

(7)
where \( r > 0 \) and \( \kappa \) are not yet determined constants. Inserting equation (6) in (7) yields
\[
\nu e^{-\kappa r} \mathbf{X} e^{\kappa r} \nu_0 = r e^{i\kappa r}.
\]
Differentiating the left and right hand side of equation (8) with respect to \( z \) results in
\[
\nu e^{-\kappa r} (\mathbf{X} \mathbf{H} - \mathbf{H} \mathbf{X}) e^{i\kappa r} \nu_0 = \kappa i e^{i\kappa r},
\]
which according to equation (8) is the same as
\[
\nu e^{-\kappa r} (\mathbf{X} \mathbf{H} - \mathbf{H} \mathbf{X} - i \kappa r) e^{i\kappa r} \nu_0 = 0.
\]
We see that equation (9) is satisfied if
\[
\mathbf{X} \mathbf{H} - \mathbf{H} \mathbf{X} = i \kappa r.
\]
Note that equation (10) does not depend on the signal, only on the operator changing the signal, represented by the anti-Hermitian matrix \( \mathbf{H} \). If we can find an \( \mathbf{X} \) solving equation (10), we have proved the assumption equation (7) to be valid. Let \( \mathbf{e}_m \) and \( \mathbf{e}_n \) be eigenvectors of \( \mathbf{H} \) with eigenvalues \( i \lambda_m \) and \( i \lambda_n \) such that \( \lambda_m = \lambda_n = \kappa \). If we set \( \mathbf{X} = \mathbf{e}_m \mathbf{e}_n^* \), and insert this \( \mathbf{X} \) in the left hand side of equation (10), we get
\[
e_m e_n^* \mathbf{H} - e_m (\mathbf{H} e_n) - (\mathbf{H} e_n) e_n^* =
- e_m (i \lambda_m e_n) - (i \lambda_m e_n) e_n^* =
i \lambda_m e_m e_n^* - i \lambda_m e_m e_n^* = i (\lambda_m - \lambda_m) e_m e_n^* = i \kappa r.
\]
We have thus proved \( \mathbf{X} = \mathbf{e}_m \mathbf{e}_n^* \) to be a solution of equation (10) and therefore the assumption made in equation (7) is valid. It can be proved, see [Nordberg, 1992], that we can only choose \( i \kappa r \) to be a difference between eigenvalues of \( \mathbf{H} \), as indicated in the derivation above. There may of course be many pairs of eigenvalues having the same difference which leads us to the general solution of equation (10),
\[
\mathbf{X} = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij} \mathbf{e}_i \mathbf{e}_j^*,
\]
where \( c_{ij} \) is an arbitrary constant if \( \lambda_j - \lambda_i = \kappa \) and zero otherwise. These constants determine the value of \( r \) according to
\[
r = \sum_{i=1}^{n} \sum_{j=1}^{m} c_{ij}.
\]

6 Summary

This paper has introduced a new way of representing signals. The representation makes a connection between features of the signal and operators acting on the signal so as to change the feature parameters. By restricting the operators to continuous one-parameter unitary operators it is possible to write a signal as
\[
\nu(x_1, x_2, \ldots, x_n) = e^{i \kappa r} \mathbf{H}_1 + e^{i \kappa r} \mathbf{H}_2 + \ldots + e^{i \kappa r} \mathbf{H}_n \nu_0.
\]
Each term in the exponent is an anti-Hermitian matrix and corresponds to a feature of the signal and each real \( \kappa r \) corresponds to a feature coordinate. For a signal which can be expressed as
\[
\nu(x) = e^{i \kappa r} \nu_0,
\]
we have shown that it is possible to find a matrix \( \mathbf{X} \), such that
\[
\nu(x) \mathbf{X} \nu(x) = e^{i \kappa r},
\]
for certain values of \( \kappa \). This allows us to extract the feature coordinate \( x \) from the signal.

It is worth pointing out that both the representation and feature extraction algorithm presented in this paper is applicable to real as well as complex signals. In the case of complex signals we would have to use complex operators, defined by complex anti-Hermitian matrices.

7 References


INSTANTANEOUS FREQUENCY AND ITS MEASUREMENT USING DIGITAL FILTERS

Václav V. ČIŽEK

Institute of Radio Engineering and Electronics Czechoslovak
Academy of Sciences, 182 51 Praha 8, Chaberská 57, Czechoslovakia

Evaluation of instantaneous frequency defined by an analytic signal requires determination of its Hilbert transform, its derivative and the derivative of the Hilbert transform. These auxiliary signals can be obtained with the limited accuracy only. In the paper the effect of inaccurately determined auxiliary signals on determination of instantaneous frequency for a harmonic signal, sum of two harmonic signals and for an amplitude modulated signal, is analyzed. It turns out that the relative error of the instantaneous frequency is in the case of filters with linear phase about of the same order as the relative deviation in amplitude characteristics of the filters. In the case of harmonic signals the filter inaccuracies cause occurrence of periodic components in the envelope and in the instantaneous frequency. Results of the theoretical analysis are compared with the results obtained by computer simulation.

1. INTRODUCTION

Instantaneous frequency and envelope are finding their practical use in a number of applications (biomedicine, astronomy, speech processing, acoustics). Their definition based on an analytic signal requires calculation of Hilbert transform. In the literature, e.g. [1], [2], some devices for instantaneous frequency measurement have been described. As a rule, little requirements were placed on the accuracy of the instantaneous frequency evaluation, neither was the accuracy analyzed. In this contribution a method based on application of digital filters will be analyzed from the accuracy viewpoint. The often used method for instantaneous frequency determination based on calculation of the discrete Fourier transform is highly accurate, but only for periodic signals. In cases where instantaneous frequency is to be evaluated in real time or when we deal with nonperiodic signals, evaluation of instantaneous frequency by digital filtering is the only option we have. This also allows additional processing of instantaneous frequency signal by smoothing or filtering.

2. PRINCIPLE OF THE METHOD EmployING DIGITAL FILTERS

The instantaneous frequency evaluation employing digital filters is based on expressing the instantaneous frequency of a signal \( x(n) \) utilizing quadrature signal \( y(n) \) and the derivatives \( \dot{x}(n) \) and \( \dot{y}(n) \). It can be expressed as

\[
\omega(n) = (x(n)\dot{y}(n) - \dot{x}(n)y(n))A^2(n)
\]

where

\[
A^2(n) = x^2(n) + y^2(n)
\]

is the square of the signal envelope. In Fig. 1 we denote by HF a Hilbert filter, by DF a differentiating filter, by DHF a differentiating Hilbert filter performing differentiation of the Hilbert transform and arithmetic operations by AO. The scheme with DHF is substantially more advantageous than realization of \( \dot{x}(n) \) by a cascade connection of HF and DF as was shown in [3].

![Diagram](Figure 1)

From the accuracy point of view it is expedient to use filters with finite
impulse response and linear phase which introduce into the measurement only errors arising from approximation of amplitude characteristics of the ideal filters HF, DF and DHF. For error analysis it proved very advantageous to use filters with equal-ripple approximation in which a constant approximation deviation is known. Since HF, DF and DHF designed to have approximately equal deviation have not the same degree (length), they have to be completed by adding some delay elements.

The filter method makes the evaluation possible in a limited range of frequencies only, which is given by the approximation interval used during the filter design.

The ideal frequency characteristics of these filters are shown in Fig.2 where

\[ \text{H}(e^{j\omega}) \text{ is the frequency characteristic of filter HF, } \text{D}(e^{j\omega}) \text{ is the frequency characteristic of filter DF and } \text{G}(e^{j\omega}) \text{ is the frequency characteristic of filter DHF.} \]

\[ \text{Figure 2} \]

The scheme in Fig.1 was realized by computer simulation and at present its realization with the aid of a digital signal processor is being prepared.

3. EVALUATION ACCURACY

3.1. General solution

As a result of using filter with non-ideal characteristics, instead of signals \( y(n) \), \( x(n) \) and \( \tilde{y}(n) \) we only get their approximate values

\[ \tilde{y}(n) = y(n) + e_1(n), \]
\[ \tilde{x}(n) = x(n) + e_2(n), \]
\[ \tilde{y}(n) = y(n) + e_3(n). \]

In this case, the signal envelope is

\[ \tilde{A}^2(n) = A^2(n) + 2e_1(n)y(n) + e_1^2(n) \]

and its instantaneous frequency

\[ \tilde{\omega}(n) = \omega(n) + \frac{e_0(n)}{A^2(n) + \frac{e_0^2(n)}{A^2(n)}}, \]

where

\[ e_0(n) = x(n)e_3(n) - x(n)e_1(n) - y(n)e_2(n) + e_1(n)e_2(n). \]

3.2. Harmonic signal

The distorted signals corresponding to the harmonic signal \( x(n) = \cos \omega_0 n \) are

\[ \tilde{y}(n) = (1 - d_3) \sin \omega_0 n, \]
\[ \tilde{x}(n) = (1 + d_3) \omega_0 \sin \omega_0 n, \]
\[ \tilde{y}(n) = (1 - d_3) \omega_0 \cos \omega_0 n. \]

Assuming \( d_3 \ll 1 \) and after some manipulation of the above relations we get for the instantaneous frequency and envelope the following expressions

\[ \tilde{\omega}(n) = \omega_0 (1 + d_1(n)), \]
\[ d_1(n) = \frac{1}{2} (d_3 + d_2 - d_1) + \frac{1}{4} (d_3 - d_2 + d_1) \cos 2 \omega_0 n \]
\[ \tilde{A}(n) = \left( 1 + d_1(n) \right) \frac{1}{2} d_2 - \frac{1}{4} d_2 \cos 2 \omega_0 n. \]

In the special case when \( d_1 = d_2 - d_3 = d \) compensation of errors takes place and it holds

\[ \tilde{\omega}(n) = \omega_0 (1 + \frac{1}{2} d + \frac{1}{2} d \cos 2 \omega_0 n). \]

From relations (5) and (6) it follows that the error caused by inaccuracy of the auxiliary signals \( y(n), x(n) \) and \( y(n) \) manifests itself under the given conditions, distorting the envelope and frequency. The relative error of the envelope (6) and frequency (5) has a constant and a periodic component whose frequency is equal to the double of the signal frequency. Determination of the harmonic signal frequency with the aid of the filter method results in interpreting this signal as a signal with "amplitude and frequency modulation" by frequency \( 2 \omega_0 \). The maximum values of these "modulations" are given by deviations of the filters from ideal characteristics. One has to bear in mind that the quantities \( d_1, d_2 \) and \( d_3 \) are differ-
ent for different frequencies and, as a result, the compensation of error in \( \omega(n) \) can take place for some frequencies only. As a rule, the quantities \( d_1 \), \( d_2 \) and \( d_3 \) representing the filter inaccuracies are unknown. When the filters HF, DF and DHF have linear phase and have been designed with the aid of program [4], assuming equi-ripple approximation, we also obtain the values of the maximum relative approximation deviation. This program has to be slightly modified to be suitable for designing the DHF, as well. If we denote the maximum approximation deviations as \( D_1 \), \( D_2 \) and \( D_3 \), the following relations hold

\[
|d_1| \leq D_1, \quad |d_2| \leq D_2 \quad \text{and} \quad |d_3| \leq D_3
\]

and the relative error of the envelope

\[
\frac{|A(n) - A|}{A} \leq D_1
\]

and of instantaneous frequency

\[
\left| \frac{\omega(n) - \omega_0}{\omega_0} \right| \leq D_1 + D_2 + D_3
\]

can be estimated.

The filter method according to Fig.1 was simulated on a computer for various filter orders. For smaller accuracy the following set of filters was simulated, where \( f_L \) and \( f_H \) is the lower and higher cut-off frequency, respectively.

Filter Degree \( f_L \) \( f_H \) \( D \)
1. HF 39 500 4500 6.79 \times 10^{-4}
2. DF 39 500 4500 2.88 \times 10^{-5}
3. DHF 27 0 4500 2.14 \times 10^{-4}

From calculation of the relative errors

\[
\text{the range of instantaneous frequency } 1245.28 \text{Hz} \leq f \leq 1254.71 \text{Hz}
\]

follows for the signal frequency \( f_0 = 1250 \text{ Hz} \) and the sampling frequency was found to be \( 1250.2 \text{ Hz} \).

Another set of filters was simulated for high accuracy:

Filter Degree \( f_L \) \( f_H \) \( D \)
1. HF 127 410 7880 1.16 \times 10^{-5}
2. DF 127 410 7880 5.6 \times 10^{-6}
3. DHF 63 0 7880 5.6 \times 10^{-6}

The estimates give the range of instantaneous frequency \( 1023.8 \text{ Hz} \leq f \leq 1024.1 \text{ Hz} \)

for \( f_0 = 1024 \text{ Hz} \). From simulation the range \( 1023.92 \text{ Hz} \leq f \leq 1024.06 \text{ Hz} \) was obtained. A simulation for different values of \( f_0 \) in the passband provided frequencies in the range \( 1024 \text{ Hz} \leq 0.1 \text{ Hz} \).

In Fig.3 the variation of instantaneous frequency for the first set of filters is depicted showing a good agreement with theoretical results.

![Figure 3](image)

3.3. Two-frequency signal

The instantaneous envelope and frequency of a two-frequency signal

\[
x(t) = \cos \omega_1 t + K \cos(\omega_2 t + \varphi), \quad K > 0 \quad \text{is} \quad [3]
\]

\[
A^2(n) = 1 + K^2 + 2K \cos \nu, \quad \nu = (\omega_2 - \omega_1) t + \varphi,
\]

\[
\omega(n) = (\omega_1 + K \omega_2 + K (\omega_1 + \omega_2) \cos \nu) A^2(n).
\]

The instantaneous frequency then lie in the interval

\[
\frac{K \omega_2 - \omega_1}{K - 1}, \quad \frac{K \omega_2 + \omega_1}{K + 1}.
\]

A general analysis of the effects of inaccuracies would in this case involve 6 quantities characterizing the filter inaccuracies at frequencies \( \omega_1 \) and \( \omega_2 \).

We therefore introduce a simplifying assumption that the distortion takes place only at frequency \( \omega_2 \).

For \( d_1 = d_2 = d_3 = d \), \( K \gg 1 \) and \( K^2 \ll 1 \) we get for instantaneous frequency an estimation

\[
\tilde{\omega}(n) = \omega(n) \pm Kd \frac{1 + 2K \omega_2}{A^2(n)}
\]

In the special case when \( K = 1 \), the following estimation can be derived

\[
\tilde{\omega}(n) = \frac{\omega_1 + \omega_2}{2} \pm d(\omega_1 + 3\omega_2).
\]
4. A SIMPLIFIED MODEL FOR FILTER INACCURACY EFFECTS

4.1. Two-frequency signal

Since we assume that the filter inaccuracy is caused by deviations of amplitude characteristics of the filters, we try to approximate the effect of these deviations in the case of a two-frequency signal by changing the amplitude \( K \) so that we shall investigate a signal

\[
x(t) = \cos \omega_1 t + K(1+\delta_0)\cos(\omega_2 t + \gamma).
\]

From relations (9) and (10) we then obtain

\[
\hat{a}^2(n) = A^2(n) + 2d_0 K(1+\cos \gamma) + d^2 K^2, \quad (14)
\]

\[
\hat{b}(n) = 0(n) + 2d_0 K \omega_1 \cos \omega_2 K(1+\cos \gamma) + d^2 K^2 \omega_0^2
\]

\[
0(n) = \omega_1 K^2 \omega_2^2 + K(1+\omega_1 \cos \omega_2) \cos \omega_2, \quad \hat{a}(n) = \hat{b}(n) \hat{a}^2(n)
\]

Under the assumption that \( K \gg 1 \) but \( d^2 K^2 \ll 1 \) we obtain the following relation for instantaneous frequency

\[
\hat{b}(n) = \omega(n) + 2d_0 K(1+\cos(\omega_1 + \omega_2))^2 A^2(n).
\]

It follows from relation (11) that the instantaneous frequency would take on values lying in an interval

\[
\frac{K(1+\delta_0)\omega_2 - \omega_1}{K(1+\delta_0) - 1}, \quad \frac{K(1+\delta_0)\omega_2 + \omega_1}{K(1+\delta_0) + 1}.
\]

For the case \( K = 1 \) this estimation is true

\[
\hat{b}(n) = \frac{\omega_1 + \omega_2}{2} \pm 2d_0 (\omega_1 + \omega_2).
\]

From comparison with section 3.3. it follows that this simplified model is an usable approximation.

4.2. Amplitude modulated signal

In the case of an amplitude modulation we evaluate the influence of the change in amplitude of one sideband of the signal

\[
x(n) = \cos(\omega_0 t) n + \cos \omega_0 n + (a + \epsilon) \cos (\omega_0 + \Omega) n
\]

on the instantaneous frequency. Then, we get

\[
\hat{b}(n) = \omega(n) + \frac{e(2a + \cos \omega_0 n) + e^2}{(1 + (2a + \cos \omega_0 n)^2 e^2 \sin^2 \omega_0 n).}
\]

The inaccuracy of filters manifests itself as frequency modulation and for \( a \ll 1 \) the instantaneous frequency stays within the interval

\[
\omega_0 - \frac{\Delta \epsilon}{1-2a}, \quad \omega_0 + \frac{\Delta \epsilon}{1+2a}.
\]

5. CONCLUSIONS

The analysis has shown that approximation errors of the filter frequency characteristics result in approximately the same relative error in instantaneous frequency. The magnitude of this error depends on the order of the filters used in the procedure. The results obtained by simulation show that by using digital filters instantaneous frequency of the signals, whose spectrum lies within the frequency bandwidth of the filters, can be evaluated with sufficient accuracy.

6. REFERENCES


Multichannel Multirate Sampling of Low-pass Stochastic Signals

Takuro KIDA and Somsak SA-NGUANKOTCHAKORN

Department of Information Processing, Graduate School at Nagatsuta,
Tokyo Institute of Technology,
4259 Nagatsuta, Midori-ku, Yokohama-shi 227, Japan Tel.045-922-1111 Ext.2522
e-mail: somsak@ip.titech.ac.jp

Abstract

The extended form of the multichannel sampling theorem of a class of band-limited stochastic input signals is presented. The sample values used in this paper are nonuniform sample points repeating periodically. They are chosen from the output signals of a parallel bank of a finite number of linear shift invariant filters with the random process noise interfering with each channel prior to sampling. Firstly, we indicate that the interpolation functions can be derived by using the concept of the generating function for sampling theorem[4]-[6]. Next, under the condition that the signal treated here is not necessarily stationary, we establish a number of relations between the interpolation functions and the noises in the system by showing the covariance of the output signal and various kinds of power spectral density functions in terms of the interpolation functions.

1 Introduction

The sampling theorem for band-limited signal has been extended in a several directions, such as nonuniform sampling theorem and derivative sampling theorem[1]-[6]. In the papers of Papoulis[2] and Brown[3], for instances, they interpreted lucidly these sampling theorems and extended the domain of applicability to embrace multichannel sampling which resembles to the framework called filter banks. It was stated that one can restore the original signal by using the simultaneous sample points of the outputs of 2N linear time invariant filters having the input as a deterministic signal \( f(t) \) band limited to \( |\omega| \leq W \) with a uniform sampling rate of \( W/(2N) \) samples per second. Basically, the recovery of a signal from generalized samples is essentially a problem of designing appropriate linear filters called reconstruction (or interpolation) filters.

In this paper, the concept of multichannel sampling theorem is extended to the class of stochastic band-limited signal. We model the system as the multichannel sampling framework whose input is contained in the class of the signals mentioned above. We assume that the random process noise \( n_k(\nu,t) \) with a random variable \( \nu \), are added to each of the corresponding output signals of the analysis filters prior to sampling. Further, the sample points have the form of \( (nT + a_k) \ (0 \leq a_1 < a_2 < \ldots < T \leq 2N/W) \) \( (n = 0, \pm 1, \pm 2, \ldots) \)[5]. Hence, these are nonuniform sample points in general. Using this model, firstly we establish the general expression and derive the necessary and sufficient condition for the interpolation formula, by utilizing the concept of the generating function for sampling theorem[4]-[6], in order that the output signal always converges, in the statistical mean sense, to the input signal. In this discussion, all the interpolation functions can be derived directly by the Fourier series expansion of the generating function with respect to the angular frequency \( \omega \).

In addition, we also consider the covariances of the restored output signal and derive various kinds of power spectral density functions. It is important to note that the signal treated here is not necessarily wide sense stationary.

2 Notations and terminology

(a) The integers are expressed by the symbols \( N, k, m, n, p, q \) and \( i \). The real numbers are indicated by \( W, T \) and \( a_k \). Throughout this paper, \( k = 1 \sim 2N \).

(b) Let \( \Omega = [0, \pm 1, \pm 2, \ldots] \) be the closed intervals on the \( \omega \)-axis given by

\[
\Omega_m = [(N - m)W/N, (N - m + 1)W/N]
\]

The fact that \( \omega \) belongs to \( \Omega_m \) is written as \( \omega \in \Omega_m \).

(c) The notation \( f(t) \rightarrow F(\omega) \) indicates that \( f(t) \) and \( F(\omega) \) form a Fourier transform pair satisfying

\[
F(\omega) = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (2)
\]

\[
f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} F(\omega) e^{j\omega t} d\omega \quad (3)
\]

(d) Bold-faced quantities denote matrices and vectors as in \( A(\omega), \xi \) etc. The quantity \( z^* \) denotes the conjugate of \( z \).

(e) Let \( (\nu, t) \) be a random process with a random variable \( \nu \). The statistical mean or expected value of a random process is defined by

\[
f(\nu) = E[f(\nu, t)] \quad (4)
\]

where \( E[\cdot] \) is called the expected value operator. The variance of \( f(\nu, t) \) is defined as

\[
\sigma^2_f = E[(f(\nu, t) - f(t))^2] \quad (5)
\]
A random process \( f(\nu, t) \) is said to be wide-sense stationary if two of its statistics, its mean and autocorrelation function, do not vary with a shift in time origin. That is, the random processes \( f(t) \) satisfy
\[
E[f(s)] = a_s \quad \text{(constant)} \tag{6}
\]
\[
E[f(s) f(t) s] = \gamma_m(t - s) \tag{7}
\]
where \( \gamma_m(t - s) \) is the autocorrelation function depends only on \( t - s \).

(1) In the following discussion, we consider \( 2N \) linear time invariant analysis filters whose transfer functions are denoted by \( H_k(\omega) \) and the time domain impulse responses are written as \( h_k(t) \). The random processes corresponding to external noise or observation error being added to each channel of the outputs of each analysis filters are written as \( n_k(\nu, t) \). They are assumed to be independent of \( h_k(t) = f(\nu, t) \) and satisfying \( E[n_k(\nu, t)] = 0 \).

3 Interpolation of band-limited stochastic signal

In this section, the input signal \( f(\nu, t) \), assumed to be band-limited stochastic signal (in the region \( |\omega| \leq W \) with random variable \( \nu \), is taken as the common input to each of the linear shift invariant filters \( H_k(\omega) \). The corresponding signals are denoted by
\[
g_k(\nu, t) = h_k(t) \ast f(\nu, t) + n_k(\nu, t) \tag{8}
\]
where \( h_k(t) \longrightarrow H_k(\omega) \). We assume additionally that \( H_k(\omega) \) are square-integrable and bounded on the region \( |\omega| \leq W \).

Now, the output signal can be expressed as follows.
\[
y(\nu, t) = \sum_{k=1}^{2N} \sum_{n=-\infty}^{\infty} g_k(\nu, nt + a_k) \psi_k(t - nt - a_k) \tag{9}
\]
where the quantities \( T \) and \( a_k \) are the positive numbers satisfying
\[
0 < a_1 < a_2 < \ldots < a_{2N} < T = \frac{2N\pi}{W} \tag{10}
\]
and \( \psi_k(t) \) are the deterministic time functions, called the interpolation functions, whose Fourier transforms \( \Phi_k(\omega) \) are piece-wise continuous. We assume that the interpolation functions satisfy the next equation.
\[
\sum_{k=1}^{2N} \sum_{n=-\infty}^{\infty} |\psi_k(t - nt - a_k)|^2 < +\infty \tag{11}
\]

In this discussion, it is assumed that the Fourier transform of \( f(\nu, t) \) satisfies
\[
F(\omega) = 0 \quad (|\omega| > W) \tag{12}
\]
Firstly, we will discuss about the necessary and sufficient condition that the output signal always converges to the input signal in the sense of statistical mean.

As a preliminaries, we provide the following lemma.

Lemma 1 Let
\[
y(t) \longrightarrow Y(\omega) \tag{13}
\]
where \( y(t) = E[y(\nu, t)] \). Then, \( Y(\omega) \) is expressed in \( \Omega_m \) as
\[
Y(\omega) = W \sum_{m=-N}^{m+N} \sum_{k=1}^{2N} \sum_{s=-\infty}^{\infty} e^{j2\pi k s W} H_k(\omega - nW/N) \Phi_k(\omega) F(\omega - nW/N) \tag{14}
\]
where \( n = 0, \pm 1, \pm 2, \ldots \)

Next theorem and its corollary show how to choose \( \Phi_k(\omega) \) in order that \( y(t) \) always converges to \( f(t) \).

Theorem 1 The necessary and sufficient condition that \( y(t) \) always converges to \( f(t) \) is
\[
(A) \text{ in } \Omega_m \quad (m = 1 \sim 2N) \tag{15}
\]
\[
(b) \text{ in } \Omega_m \quad (m < 1 \text{ or } m > 2N) \tag{16}
\]
where
\[
\Phi_k(\omega) = \left[ \begin{array}{cc}
\phi_k(\omega) & 0 \\
0 & 0
\end{array} \right] \tag{17}
\]
\[
\begin{aligned}
\Phi_k(\omega) &= [\Phi_{1k}(\omega), \Phi_{2k}(\omega), \ldots, \Phi_{2Nk}(\omega)] \\
A_m(\omega) &= \left[ \begin{array}{c}
\phi_k(\omega) \\
\phi_k(\omega)
\end{array} \right] \\
A_m(\omega) &= \left[ \begin{array}{c}
\phi_k(\omega) \\
\phi_k(\omega)
\end{array} \right] \\
&= \left[ \begin{array}{c}
\phi_k(\omega) \\
\phi_k(\omega)
\end{array} \right] \\
&= \left[ \begin{array}{cc}
\phi_k(\omega) & 0 \\
0 & 0
\end{array} \right] \tag{18}
\end{aligned}
\]
and \( H_k(\omega) \) are the transfer functions of the analysis filters.

This theorem can be proved easily by using the concept of the generating function for sampling theorem which will be mentioned in the next section.

As is evident in theorem 1, if \( \Phi_k(\omega) \) \( (k, m = 1 \sim 2N) \) are chosen so as to satisfy the conditions (a) and (b) in the theorem, the next identity holds.
\[
f(t) = \sum_{k=1}^{2N} \sum_{n=-\infty}^{\infty} g_k(\nu, nt + a_k) \psi_k(t - nt - a_k) \tag{20}
\]

Eq.(20) provides a series of sampling theorems if appropriate sets of \( a_k \) and \( H_k(\omega) \) are selected.

Let us now determine the closed form expressions of \( \Phi_k(\omega) \) \( (k, m = 1 \sim 2N) \) under the condition that only \( A_1(\omega) \) is nonsingular.

Corollary 1 Suppose that the matrix
\[
A_1(\omega) = \left[ \begin{array}{c}
\phi_k(\omega) \\
\phi_k(\omega)
\end{array} \right] (|\omega| > W) \tag{21}
\]

is nonsingular for all \( \omega \). Further, let
\[
A_1(\omega)^{-1} = [\Theta_{mn}(\omega)] \quad (m, n = 1 \sim 2N) \tag{22}
\]
Then the solutions of Eq.(15) and Eq.(16) are given by
\[
(a) \text{ in } \Omega_m \quad (m = 1 \sim 2N) \tag{23}
\]
\[
\Phi_k(\omega) = \frac{2\pi N}{W} e^{-j(m-1)W} \Theta_{km}(\omega + (m - 1)W/N) \tag{24}
\]
\[
(b) \text{ in } \Omega_m \quad (m < 1 \text{ or } m > 2N) \tag{25}
\]
\[
\Phi_k(\omega) = 0 \quad (k = 1 \sim 2N; m < 1 \text{ or } m > 2N) \tag{26}
\]
Corollary 1 can be proved easily if we notice the relation
\[ A_m(\omega) = A_1(\omega + (m - 1)W/N)\Delta - i W \frac{\alpha_m}{N} \]  (25)
where \( \Delta \) is the diagonal matrix with the \( n \)-th diagonal element of which is \( \beta_n \) \((n = 1 \sim 2N)\).

The significant aspect of Corollary 1 is that all the \( \Phi_m(\omega) \) can be obtained by evaluating only one matrix \( A_1(\omega)^{-1} \).

As a direct consequence of Corollary 1, we have

**Theorem 2** If (a) and (b) in Corollary 1 are satisfied, then the interpolation functions \( \psi_0(t) \) in Eq. (30) are given by
\[ \psi_0(t) = \frac{N}{W} \sum_{\nu = 1}^{2N} e^{-i(\nu - 1)(\nu + \frac{1}{2})W/N} \int_{-W/2}^W \Theta_m(\omega)e^{i\omega t} \, d\omega \]  (26)
where \( \psi_0(t) \rightarrow \Phi_m(\omega) \) and \( [\Theta_m(\omega)] = A_1(\omega)^{-1} \).

The proof is omitted.

4 Calculation of interpolation function by generating function

In this section, we show how to use the concept of the generating function for sampling theorem [4]-[6] to derive the interpolation functions.

Here, we assume again that the interpolation functions \( \psi(t) \) satisfy Ineq.(11).

Now, we consider the 2-variable functions \( \Psi_k(\omega, t) \) \((k = 1 \sim 2N)\) which satisfy the next 3 conditions at the same time.

(a) \( \Psi_k(\omega, t) \) are periodic functions with period \( W \).

\[ \Psi_k(\omega, t) = \Psi_k(\omega - pW/N, t) \quad (p = 0, 1, 2, \ldots) \]  (27)

(b) For any finite \( t \), the following inequality holds.
\[ \int_{\Omega_m} |\Psi_k(\omega, t)|^2 \, d\omega < +\infty \quad \omega \in \Omega_m \]  (28)

(c) There exist the solutions \( \Psi_k(\omega, t) \) of the next linear equations.
\[ \frac{2N}{W} \int_{-W/2}^W H_k(\omega - pW/N)\Psi_k(\omega, t) = e^{i\omega t} \]  (29)
where the 2-variable functions \( \Psi_k(\omega, t) \) are called the generating functions. Then the next theorem holds.

**Theorem 3** The necessary and sufficient condition that \( y(t) \) always converges to \( f(t) \) is that there exist the functions \( \Psi_k(\omega, t) \) satisfying conditions (a) and (b) simultaneously.

In proving the necessity of the theorem, Riesz-Fischer's theorem is necessary. Hence, it is necessary to assume that Ineq.(11) is valid. Therefore, there exist \( \Psi_k(\omega, t) \) \((k = 1 \sim 2N)\) satisfying
\[ \Psi_k(\omega, t) = \sum_{n = -\infty}^{\infty} \psi_{kn}(t)e^{-i\omega nt} \]  (30)
By Parseval's formula, the conditions (a) and (b) hold obviously. With respect to the necessity of the condition
\( (c) \), see [3] for justification. we will show here only the guideline of the sufficiency of the proof. The similar proof in detail was shown in [9] of the case of uniform sampling.

Now, let the error between the statistical mean of input and output signal be defined as
\[ e(t) = \int_t f(t) - y(t) \]  (31)
\[ = \int_t f(t) - \sum_{k = 1}^{2N} \sum_{n = -\infty}^{\infty} g_k(nT - a_k)\psi_k(t - nT - a_k) \]  (32)
Next, the generating functions \( \Psi_k(\omega, t) \) are defined by
\[ e^{j\omega t} = \sum_{k = 1}^{2N} e^{j\omega a_k} H_k(\omega)\Psi_k(\omega, t) \]  (33)
Substituting Eq.(33) into Eq.(32), we have
\[ e(t) = \int_t \left[ \frac{1}{2\pi} \int_{-W}^{W} F(\omega) e^{j\omega t} \sum_{k = 1}^{2N} e^{j\omega a_k} H_k(\omega) \Psi_k(\omega, t) \right] \]  (34)
From the assumption of the sufficiency, we can use the conditions (a) and (b). Hence, Eq.(30) is obvious. Further, although the detail discussion is omitted, if the average rate of the sample points contains no redundancy, we can prove that
\[ \psi_m(t) = \psi_m(t - nT - a_k) \]  (35)
Substituting Eq.(30) and Eq.(35) into Eq.(34), we can easily prove that \( e(t) = 0 \).

Replacing \( \omega \) in Eq.(33) by \( \omega - pW/N \) \((p = 1, 2, \ldots, 2N - 1)\), we get
\[ e^{(\omega - pW)/N} = \sum_{k = 1}^{2N} e^{(\omega - pW)/N} H_k(\omega - pW/N)\Psi_k(\omega, t) \]  (36)
That is, the condition (c) holds obviously. Theorem 1 can be proved easily from Eq.(30), Eq.(35) and Eq.(36).

Actually, the concept of the generating function is usable even in the case of oversampling. However, in this case, Eq.(35) is not always valid.

5 Covariance and variance of output signal \( y(\nu, t) \)

In this section, we wish to determine the covariance and the variance of the output signal \( y(\nu, t) \).

Now, the error of the output signal is defined as
\[ E_{\nu}(\nu, t) = y(\nu, t) - y(t) \]
\[ = \sum_{k = 1}^{2N} \sum_{n = -\infty}^{\infty} \epsilon_k(\nu, nT + a_k)\psi_k(t - nT - a_k) \]  (37)
where \( \epsilon_k(\nu, t) = g_k(\nu, t) - g_k(\nu, t) \) and \( y(t) = E[y(\nu, t)] \) is the statistical mean of the output signal.
Here, we assume that $e_k(t, t)$ \((k = 1 \sim 2N)\) satisfy

\[
\begin{align*}
E[e_k(t, t)] &= 0 \\
E[e_k(t, s)e_m(s)] &= \gamma_{km}(t-s)r_{m}(s)r_{m}(s)
\end{align*}
\] (38)

where \(r_{k}(t) \rightarrow R_{k}(\omega) \quad (i = 1 \sim 2N)\) are deterministic functions of \(t\), for example, the output signals of \(R_{k}(\omega)\) for a typical deterministic input signal \(f(t)\). We suppose that \(|R_{k}(\omega)| = 0 \quad (|\omega| > W)\).

In this case, the covariance of \(y(t, t)\) is given by

\[
\begin{align*}
\gamma(t, \tau) &= E[y(t, t + \tau)|E[y(t,t)]] \\
&= \sum_{k=1}^{2N} \sum_{m=1}^{2N} \sum_{n=-\infty}^{\infty} \sum_{p=\infty}^{\infty} \gamma_{km}((n-p)T + a_k - a_m) \\
& \quad \times r_{k}(nT + a_k)r_{m}(pT + a_m) \\
& \quad \times \phi_{k}(t + \tau - nT - a_k)\phi_{m}(t - pT - a_m)
\end{align*}
\] (39)

Further, let the power spectrum density function be written as \(K(\omega, t)\), where \(c(t, \tau) \rightarrow K(\omega, t)\). Then the following theorem is obtained directly.

**Theorem 4** Let \(\gamma_{km}(t) \rightarrow \Gamma_{km}(\omega)\). Further, let

\[
\begin{align*}
\hat{K}_{km}(\omega) &= \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \Gamma_{km}(w - nW/N)e^{-j\omega(n+a_m)W/N} \\
\hat{R}_{k}(\omega) &= \sum_{p=\infty}^{\infty} R_{k}(w + pW/N)e^{jp\omega W/N}
\end{align*}
\] (40)

Then, \(K(\omega, t)\) can be expressed as

\[
K(\omega, t) = \left(\frac{W}{4\pi^2N^2}\right)^{\frac{2N}{2}} \sum_{k=1}^{2N} \sum_{m=1}^{2N} \sum_{n=-\infty}^{\infty} \sum_{p=\infty}^{\infty} \hat{K}_{km}(\omega) \hat{R}_{k}(\omega) e^{-j\omega(n+a_m)W/N} \\
& \quad \times \phi_{k}(w - nW/N)e^{j\omega(t-a_k)W/N} \\
& \quad \times \phi_{m}(w - pW/N)e^{j\omega(t-a_m)W/N}
\] (41)

The proof is omitted.

If $e_k(t, t)$ are wide sense stationary signals, that is

\[
\begin{align*}
E[e_{k}(t, t)] &= 0 \\
E[e_{k}(t, s)e_{m}(s)] &= \gamma_{km}(t-s)
\end{align*}
\] (43)

holds, then the following theorem is immediate.

**Theorem 5** Under the condition that Eq. (43) holds, we have

\[
K(\omega, t) = \left(\frac{W}{4\pi^2N^2}\right)^{\frac{2N}{2}} \sum_{k=1}^{2N} \sum_{m=1}^{2N} \sum_{n=-\infty}^{\infty} \sum_{p=\infty}^{\infty} \Gamma_{km}(\omega) \Phi_{k}(w - nW/N)e^{-j\omega(n+a_m)W/N} \\
& \quad \times \phi_{k}(w - nW/N)e^{j\omega(t-a_k)W/N} \\
& \quad \times \phi_{m}(w - pW/N)e^{j\omega(t-a_m)W/N}
\] (44)

Substituting $R_{k}(\omega) = 2\pi \delta(\omega)$ into Eq. (42), we may easily derive Eq. (44).

Further, if $e_{k}(t, t)$ and $e_{m}(t, t)$ are independent of each other, we may obtain the next corollary.

**Corollary 1** If $E[e_{k}(t, t)e_{m}(t, s)] = 0 \quad (k \neq m; k, m = 1 \sim 2N)$

\[
K(\omega, t) = \left(\frac{W}{4\pi^2N^2}\right)^{\frac{2N}{2}} \sum_{k=1}^{2N} \Phi_{k}(\omega) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \Gamma_{k}(\omega - u) \\
& \quad \times \hat{R}_{k}(u)\hat{R}_{k}(v)\phi_{k}(\omega - u + v)e^{j\omega(u+v)W/N}dudv
\] (45)

where $\hat{R}_{k}(\omega) = Eq.(41)$ and

\[
\Gamma_{k}(\omega) = \sum_{n=-\infty}^{\infty} \Gamma_{nk}(\omega - nW/N)
\] (46)

Moreover, if $e_{k}(t, t)$ are wide sense stationary signals,

\[
K(\omega, t) = \frac{W^2}{4\pi^2N^2} \sum_{k=1}^{2N} \sum_{m=1}^{2N} \sum_{n=-\infty}^{\infty} \sum_{p=\infty}^{\infty} \Gamma_{km}(\omega) \Phi_{k}(\omega - nW/N) \Phi_{m}(\omega - pW/N)e^{j\omega(t-a_k+a_m)W/N}
\] (47)

\[
W \int_{0}^{2\pi N W} K(\omega, t) d\omega = W^2 \sum_{k=1}^{2N} \gamma_{k}(\omega) \Phi_{k}(\omega) \Phi_{k}(\omega) d\omega
\] (48)

Using this $K(\omega, t)$ function as a measure, we may achieve the optimum arrangement of the sample points, by the ordinary numerical optimization procedure, that minimize the error of output signal.

**Conclusion**

The extended form of the multichannel sampling of the class of stochastic band-limited signal with nonuniform sampling points has been investigated. Our discussion covered many kinds of errors, such as quantization error, round-off error, jitter etc., which occur in the physical applications of this transmission system. We study the effect of interpolation functions on such these kinds of errors by deriving various power spectral density functions.

**Acknowledgement**

The authors wish to thank Prof. A. Pettwee of Rohr University, Prof. Kenneth Jenkins of Coordinated Science Laboratory, University of Illinois, and my colleague Mr. Hiroshi Mochizuki of Tokyo Institute of Technology for helping comments.

**References**


OPTIMIZATION OF MULTIRESOLUTION ANALYSIS FOR FINITE SEQUENCES

Jean-Christophe PESQUET

Laboratoire des Signaux et Systèmes, CNRS/UPS, Ecole Supérieure d’Electricité, Plateau de Moulon, 91192 Gif sur Yvette Cédex, France, pesquet@frese51.bitnet

At first, this paper presents a matrix formula of the periodic wavelet transform. Then, the problem of the optimal choice of orthonormal wavelets is considered. A statistical approach is adopted in order to obtain a compact representation of a given set of input signals. Moreover, a parametrization of wavelets is introduced in the frequency domain and a multiscale method is proposed to estimate the frequency responses of the filters used at each resolution level.

1 INTRODUCTION

In the use of wavelet transforms [1] [2], a crucial step is to choose the wavelet which is best matched to the signal to be processed. The problem can also be solved in trying to find the set of wavelets which best approximates the Karhunen-Loève basis of the signal. A possible approach is to use a “dictionary” of wavelets and to look for the one which minimizes an entropy criterion [3]. However, the computational cost of this search is high, when the dictionary is large. Moreover, this method can only be successful if the dictionary contains a “good” wavelet. Another way to approach the problem, which is adopted here, is to design the wavelet through an optimization procedure. Because of the strong connections existing between wavelet transforms and octaveband filter banks [4] [5], this is equivalent to synthesizing optimal QMF filters. Such an approach has been used in [5]. The impulse responses of the filters characterizing the wavelet decomposition are then found by minimizing an upper bound of the distance between a deterministic signal and its approximation up to a given scale.

In this paper, we consider the multiresolution analyses of $2^K$-periodic sequences [7]. More generally, the corresponding transform may be applied to finite sequences of $2^K$ samples to avoid the boundary effects occurring with the usual discrete wavelets [8]. We focus attention on the frequency domain implementation of this transform which leads to fast algorithms. At first, the principles of the discrete periodic wavelet transform are recalled and a matrix formulation of the transform is given. Then, a multiscale method is proposed to find the optimal transform.

2 WAVELET TRANSFORM FOR PERIODIC SEQUENCES

2.1 Reminders and notations

A multiresolution analysis of the space $\ell^2(\mathbb{Z}/2^K\mathbb{Z})$ of $2^K$-periodic sequences is defined by vector subspaces $(V_k)_{0 \leq k \leq K}$ and $(O_k)_{1 \leq k \leq K}$ such that $\dim V_k = 2^{K-k}$, $V_k \oplus O_k = V_{k+1}$ (with $V_0 = \ell^2(\mathbb{Z}/2^K\mathbb{Z})$) and $O_k$ is orthogonal to $V_k$. The spaces $V_k$ and $O_k$ are spanned by orthonormal bases, denoted respectively by $[\phi_{s,m}(n)]_{0 \leq m < 2^{k-s}}$ and $[\psi_{s,m}(n)]_{0 \leq m < 2^{k-s}}$ which satisfy the following translation property:

$$
\phi_{s,m}(n) = \phi_{s,0}(n-2^k m), \quad \psi_{s,m}(n) = \psi_{s,0}(n-2^k m), \quad (1)
$$

for $m \in \{0, \ldots, 2^{K-k}-1\}$. This definition implies that there exist sequences $[h_k(m)]_{0 \leq m < 2^{k-s}}$ and $[g_k(m)]_{0 \leq m < 2^{k-s}}$ such that

$$
\phi_{s+1,0}(n) = \sum_{m=0}^{2^{K-s}-1} h_k(m) \phi_{s,0}(n-2^k m), \quad (2)
$$

$$
\psi_{s+1,0}(n) = \sum_{m=0}^{2^{K-s}-1} g_k(m) \psi_{s,0}(n-2^k m). \quad (3)
$$

Furthermore, the wavelets $\psi_{s,m}(n)$ with $k \in \{1, \ldots, K\}$ and $m \in \{0, \ldots, 2^{K-k}-1\}$ and the approximation sequence $\phi_{0,0}(n)$ at the lowest resolution level form an orthonormal basis of $\ell^2(\mathbb{Z}/2^K\mathbb{Z})$.

Note that we do not impose any relations between the sequences $h_k(m)$ or $g_k(m)$ at different scales. Indeed, the above definition does not introduce any kind of “dilatation” property as usually required in the construction of wavelets. It must be emphasized that this constraint would restrict the set of possible bases and, at the same time, make their optimal choice more difficult.
The approximation coefficients \( c_k(m) \) and the wavelet coefficients \( d_k(m) \) of a sequence \( x(n) \) of \( \ell^2(\mathbb{Z}/2^K \mathbb{Z}) \) are

\[
\begin{align*}
    c_k(m) &= \sum_{n=0}^{2^K-1} x(n) \phi_{k,m}(n)^*, \quad (4) \\
    d_k(m) &= \sum_{n=0}^{2^K-1} x(n) \psi_{k,m}(n)^*. \quad (5)
\end{align*}
\]

A classical result is that \( c_{k+1}(n) \) and \( d_{k+1}(n) \) may be computed from \( c_k(n) \) (with \( c_0(n) = x(n) \)) by the following recursions:

\[
\begin{align*}
    c_{k+1}(n) &= \sum_{m=0}^{2^K-1} h_k(m)^* c_k(2n + m), \quad (6) \\
    d_{k+1}(n) &= \sum_{m=0}^{2^K-1} g_k(m)^* c_k(2n + m). \quad (7)
\end{align*}
\]

Subsequently, the normalized DFTs of \( [c_k(n)]_{0 \leq n < 2^K} \) and \( [d_k(n)]_{0 \leq n < 2^K} \) will be denoted respectively by \( [C_k(p)]_{0 \leq p < 2^K} \) and \( [D_k(p)]_{0 \leq p < 2^K} \), and the DFTs of \( \sqrt{2} h_k(m) g_k(m) \) will be denoted respectively by \( [H_k(p)]_{0 \leq p < 2^K} \) and \( [G_k(p)]_{0 \leq p < 2^K} \).

2.2 Matrix formulation

For \( k \in \{1, \ldots, K\} \), let us define the vectors

\[
\begin{align*}
    c_k &= [c_k(0), \ldots, c_k(2^K - k - 1)]^t, \quad (8) \\
    d_k &= [d_k(0), \ldots, d_k(2^K - k - 1)]^t. \quad (9)
\end{align*}
\]

A matrix form of equations (6) and (7) may be given:

\[
\begin{align*}
    \begin{bmatrix}
    C_{k+1} \\
    D_{k+1}
    \end{bmatrix}
    &=
    \begin{bmatrix}
    H_k^t \\
    G_k^t
    \end{bmatrix}
    \begin{bmatrix}
    C_k \\
    D_k
    \end{bmatrix}, \quad (10)
    \\
    c_{k+1} &= T_{k+1}^t c_k, \quad (11) \\
    d_{k+1} &= T_{k+1}^t d_k, \quad (12)
\end{align*}
\]

where \(^t\) represents the Hermitian transpose operation, \( T_k \) denotes the square matrix corresponding to the normalized DFT of length \( 2^k \) and \( H_k \) is the following diagonal block matrix,

\[
H_k = \begin{bmatrix}
    D H_k(0, 2^k - k - 1) \\
    D H_k(2^k - k - 1, 2^k - 1)
\end{bmatrix}. \quad (13)
\]

\[
D H_k(n, m) = \text{Diag}(H_k(n), \ldots, H_k(m)), \quad n \leq m. \quad (14)
\]

The matrix \( G_k \) is expressed from \( [G_k(p)]_{0 \leq p < 2^K} \), in a similar way. Let us note that the transform between \( c_k \) and \( c_{k+1}d_{k+1} \) is unitary if the matrices

\[
M_k(p) = \begin{bmatrix}
    H_k(p) & G_k(p) \\
    H_k(2^k - k - 1 + p) & G_k(2^k - k - 1 + p)
\end{bmatrix} \quad (15)
\]

are unitary, for all \( p \in \{0, \ldots, 2^k - k - 1\} \). This result is known as the orthonormality or perfect reconstruction condition.

In the expression of the whole wavelet decomposition, a further simplification occurs as all the DFTs, except the first, and half the inverse DFTs may be omitted:

\[
\begin{bmatrix}
    c_0 \\
    d_0 \\
    \vdots \\
    d_{k-1} \\
    z(0) \\
    z(1) \\
    \vdots \\
    z(2^{k-1} - 1)
\end{bmatrix} = P \begin{bmatrix}
    H_{k-1} \\
    H_{k-2} \\
    \vdots \\
    H_0
\end{bmatrix} T_k. \quad (17)
\]

2.3 Zeros of the basis sequences

Equations (2) and (3) show that the DFTs \( \Phi_\omega(p) \) of \( \phi_\omega(n) \) and \( \Psi_\omega(p) \) of \( \psi_\omega(n) \) are such that, for \( p \in \{0, \ldots, 2^k - 1\} \),

\[
\Phi_\omega(p) = 2^{k-1} \prod_{i=0}^{k-1} H_i(p), \quad (18)
\]

\[
\Psi_\omega(p) = 2^k G_{k-1}(p) \prod_{i=0}^{k-2} H_i(p), \quad k > 1, \quad (19)
\]

\[
\Psi_\omega(p) = 2^k G_0(p), \quad (20)
\]

where the sequences \( H_k(p) \) and \( G_k(p) \) now designate periodic extensions with period \( 2^k \). Generally, \( H_k(p) \) is the frequency response of a low-pass filter such that

\[
H_k(2^k - k - 1) = 0, \quad G_k(0) = 0. \quad (21)
\]

So, according to (18), (19) and (20), \( \Phi_\omega(p) \) has at least \( 2^k - 1 \) zeros on the interval \( \{0, \ldots, 2^k - 1\} \), for

\[
p = 2^{k-1} q, \quad q \in \{1, \ldots, 2^k - 1\}, \quad (22)
\]

and \( \Psi_\omega(p) \) has at least \( 2^{k-1} - 1 \) zeros, for

\[
p = 2^{k-1-1} q, \quad q \in \{0, \ldots, 2^{k-1} - 1\}. \quad (23)
\]

This implies, in particular, that \( \phi_\omega(n) = 2^{-k} \) so that \( c_0(n) / 2^\frac{k}{2} \) is the mean value of the signal \( x(n) \).

3 OPTIMIZATION OF THE WAVELET TRANSFORM

We will now try to choose the coefficients \( H_k(p) \) and \( G_k(p) \) in order to get the "best" orthonormal wavelet basis. To solve this problem, we will try to optimally concentrate the energy of the components of the transformed vector. The optimization procedure takes advantage of the structure of multiresolution analysis.
3.1 Parametrization of orthonormal wavelets

For orthonormal wavelets, the matrix $M_k(p)$ defined by (15) is unitary. So, it is straightforward to show that we have the following parametrization, for $k \in \{0, \ldots, K - 1\}$ and $p \in \{0, \ldots, 2^k - k - 1\}$,

$$H_k(p) = H_k^*(p) \exp[j\alpha_k(p)],$$

$$H_k(2^k - k - 1 + p) = H_k^*(2^k - k - 1 + p) \exp[j\alpha_k(p)],$$

$$G_k(p) = G_k^*(p) \exp[j\beta_k(p)],$$

$$G_k(2^k - k - 1 + p) = G_k^*(2^k - k - 1 + p) \exp[j\beta_k(p)],$$

where

$$H_k^*(p) = \cos[\varphi_k(p)] \exp[j\Delta_k(p)],$$

$$H_k^*(2^k - k - 1 + p) = \cos[\varphi_k(p)] \exp[-j\Delta_k(p)],$$

$$G_k^*(p) = -\sin[\varphi_k(p)] \exp[j\Delta_k(p)],$$

$$G_k^*(2^k - k - 1 + p) = \cos[\varphi_k(p)] \exp[-j\Delta_k(p)],$$

and $(\alpha_k(p), \beta_k(p), \varphi_k(p), \Delta_k(p)) \in \mathbb{R}^4$.

Then, by using equations (18) and (19), it can be proved that the wavelets may be parametrized as follows:

$$\Psi_{K-1}(p) = 2^\frac{K}{2} \exp[j\sigma_{K-1}(0)] \prod_{n=0}^{K-1} H_n(p),$$

$$\Psi_0(p) = 2^\frac{K}{2} \exp[j\beta_{K-1}(p)] \prod_{n=0}^{K-1} H_n(p),$$

where $\sigma_{K-1}(0) \in \mathbb{R}$, $\beta_k(p) \in \mathbb{F}(2^k, 2^k)$, and $(H_k(p), G_k(p))(\varphi_k(p), \Delta_k(p)) \in \mathbb{F}(2^k, 2^k)$ is linked to $\varphi_k(p)$ and some variable $\Delta_k(p)$ by relations similar to (28)-(31). Therefore, the wavelet transform may be computed by replacing $[H_k(p)]_{k \leq K-1}$ by $[H_k(p)]_{k \leq K-1}$ and $[G_k(p)]_{k \leq K-1}$ by $[G_k(p)]_{k \leq K-1}$ and $[\exp[j\beta_k(p)]G_k(p)]_{k \leq K-1}$. Moreover, a unique set of parameters may be found by taking

$$0 \leq \varphi_k(p) \leq \frac{\pi}{2},$$

$$0 \leq \Delta_k(p) \leq \frac{\pi}{2},$$

$$\alpha_k(p) = 0,$$

$$0 \leq \alpha_k(0) < 2\pi, 0 \leq \beta_k(p) < 2\pi.$$

3.2 One-scale optimization

Let us assume that $a_k(p) \equiv a_k$ and $d_k(p) \equiv d_k$ are known and let us optimize the choice of $(H_k(p), \varphi_k(p), \Delta_k(p)).$

To do so, it may seem natural to maximize $\eta_k = \max ||\epsilon_k + ||d_k + ||$ in order to obtain a concentration of the energy. As a result of relations (12), $\eta_k$ may also be written as $\max(||C_k + ||D_k + ||).$

To find $\varphi_k(p)$ and $\Delta_k(p)$, it is convenient to classify sequences $C_k(p)$ into two categories: a class $E_k^+$ such that $||C_k|| < ||D_k + ||$ and a class $E_k^-$ such that $||C_k|| \geq ||D_k + ||$. It is then straightforward to show that, if $C_k$ belongs to $E_k^+$,$$

$$\cos[\varphi_k(p)] = \sqrt{\frac{||C_k||}{\sqrt{||C_k||^2 + ||C_k||^2}}} = \cos[\varphi_k(p)],$$

$$\Delta_k(p) = \frac{1}{2} \arg[C_k(p)C_k^*(2^k - k - 1 + p)](\pi),$$

whereas, if $C_k$ belongs to $E_k^-$$$

$$\cos[\varphi_k(p)] = \sqrt{\frac{||C_k||}{\sqrt{||C_k||^2 + ||C_k||^2}}} = \cos[\varphi_k(p)],$$

$$\Delta_k(p) = \frac{1}{2} \arg[C_k(p)C_k^*(2^k - k - 1 + p)](\pi),$$

To eliminate this ambiguity, it may be assumed that

$$\sum_{p=0}^{2^k - k - 1} \cos[\varphi_k(p)]^2 + \sum_{p=2^k - k - 2}^{2^k - k - 1} \sin[\varphi_k(p)]^2 > \sigma_k,$$

with $\sigma_k = 2^k - k - 2$. This assumption allows us to specify $H_k(p)$ as the frequency response of a low-pass filter. Then, the classification is simply achieved by checking the above condition. Note that formula (38) and (40) are not valid when $C_k(p) = C_k(2^k - k - 1 + p)$ = 0. The corresponding values of $p$ must be discarded (and $\sigma_k$ must be decreased). According to section 2.3, this case may occur when the input signal is a wavelet. In a similar way, formula (39) and (41) cannot be used when $C_k(p) = 0$ or $C_k(2^k - k - 1 + p) = 0$. However, if $C_k(p) \neq 0$ and $C_k(2^k - k - 1 + p) \neq 0$ or if $C_k(p) \neq 0$ and $C_k(2^k - k - 1 + p) = 0$, (36) yields $\Delta_k(p) = 0$.

By defining

$$C_k + 1(p) = H_k(p)^*C_k(p) + H_k(2^k - k - 1 + p)^*C_k(2^k - k - 1 + p),$$

$$D_k + 1(p) = G_k(p)^*C_k(p) + G_k(2^k - k - 1 + p)^*C_k(2^k - k - 1 + p),$$

we have

$$D_k + (p) = \exp[-j\beta_k(p)]D_k(p).$$

An optimal decomposition is obtained by making $\{D_k(m)\}_{m}$ as close as possible to a Kronecker sequence, which is equivalent to minimize $||D_k + (p) = \exp[-j(2\pi \frac{m}{2^k - k - 1} + \theta_k)]||$, where $\theta_k \in [0, 2\pi]$ and $m_k \in \{0, \ldots, 2^k - k - 1\}$. In this approach, the choice of $m_k$ and $\theta_k$ is free and, by taking, $m_k = \theta_k = 0$,

$$\beta_k(p) = \arg[D_k + (p)].$$

This formula can only be applied if $D_k + (p) \neq 0$, which is not satisfied when $C_k$ belongs to $E_k^-$. To determine $\sigma_k(0)$, the same approach may be used for the low-pass output, at the lowest resolution level.
3.3 Statistical estimations

In the previous method, the values of the parameters are undetermined in several cases. To solve this problem, several realizations of $C_k$ must be considered. Moreover, this situation is even more interesting from a practical point of view.

To estimate $\varphi_k(p)$ and $\Delta_k(p)$, we propose to maximize the criterion

$$\eta_k(p) = E_+ \{ | \bar{C}_{k+1}(p) |^2 \} P_+ + E_- \{ | \bar{D}_{k+1}(p) |^2 \} P_-,$$

where $\bar{C}_{k+1}(p)$ and $\bar{D}_{k+1}(p)$ are given by relations (43) and (44), $P_+$ (resp. $P_-$) is the probability measure of $E_+^+$ (resp. $E_+^-$) and $P_+$ (resp. $P_-$) is the probability measure of $E_-^+$ (resp. $E_-^-$). At some tractations, the expressions of the estimates may be found:

$$\varphi_k(p) = \frac{1}{2} \arctan \left( \frac{2 | A_k(p) |}{B_k(p)} \right),$$

$$\Delta_k(p) = \frac{1}{2} \arg[A_k(p) | \sigma],$$

with

$$A_k(p) = E_+ \{ C_k(p) \bar{C}_k(2^{K-k-1} + p)^* \} P_+ - E_- \{ C_k(p) \bar{C}_k(2^{K-k-1} + p)^* \} P_-,$$

$$B_k(p) = E_+ \{ | C_k(p) |^2 - | C_k(2^{K-k-1} + p)^* |^2 \} P_+ - E_- \{ | C_k(p) |^2 - | C_k(2^{K-k-1} + p)^* |^2 \} P_-.$$

Estimating $\beta_k(p)$ is a more difficult problem. Indeed, the different realizations of $C_k$ may lead to sequences $\{d_{k+1}(n)\}_n$ localized at different times. So, a null value can no longer be taken for the variables $m_k$ and $\theta_k$ which have been defined in the previous section. An iterative method may be used which requires i) an a priori estimation of $\beta_k(p)$, ii) the determination of $m_k$ and $\theta_k$, for each realization of $\{C_k(p)\}_n$, and iii) an a posteriori statistical estimation of $\beta_k(p)$.

3.4 Multiscal optimization

To obtain the entire set $\varphi_k(p)$, $\Delta_k(p)$ and $\beta_k(p)$ when $k \in \{0, \ldots, K-1\}$ and $p \in \{0, \ldots, 2^{K-k-1} - 1\}$, the procedure is repeated recursively for each resolution level by starting from the highest resolution ($k = 0$).

Since this method has been introduced in a rather intuitive way, one may wonder about its reliability. In fact, we have the following result which shows the validity of the algorithm: the parameters $\varphi_k(p)$, $\Delta_k(p)$, $\beta_k(p)$ and $\alpha_{K-k-1}(0)$ characterizing a wavelet transform are correctly identified by the method if the sequences $\psi_{10}(n), \ldots, \psi_{K0}(n)$ and $\phi_{K0}(n)$ are taken as realizations of the input signal. This result is still valid if the sequences $\{\psi_{m}(n)_k\}_{0 \leq k \leq K, 0 \leq m < 2^k}$ and $\phi_{K0}(n)$ are chosen as input sequences. However, $\beta_k(p)$ is identified up to a linear phase, which corresponds to reindexing $\{\psi_{m}(n)\}_m$.

4 CONCLUSION

In this paper, some methods have been proposed to optimize the choice of the coefficients characterizing the periodic discrete wavelet transforms. Our approach relies upon a frequency domain parametrization and uses some estimates of the parameters.

Several points need further studies. At first, it could be interesting to introduce further constraints in the optimization procedure to impose some regularity conditions. Secondly, more elaborate statistical estimations might be performed.

ACKNOWLEDGEMENTS

The author would like to thank Dr. H. Krim for many interesting discussions.

References


NOISE CANCELLATION WITH UNCORRELATED REFERENCE
— A NON LINEAR FILTER APPROACH

Min CAI* John BALL**

* Electricité De France, DER/SDM, 6, quai Watier 78401 Chatou, GdR, CNRS France
** Georgia Tech Lorraine, 2 et 3 rue Marconi, 57070 Metz, France

This paper investigates the situations of noise cancellation in which the useful and the reference signals are not necessarily correlated. We propose a method based upon a nonlinear structure which requires only that the signals be dependent, not necessarily correlated. We give the proof and the simulation results which confirm the previous proposition.

1. INTRODUCTION

The Wiener filter and the linear adaptive filter have been largely used in noise cancellation where it is assumed that a reference signal is correlated with the useful signal but uncorrelated with this noise. In other words, the reference and the useful signals must have overlapping spectral components. In many applications, this condition is not satisfied; one example is shock characterization in nuclear power process monitoring. The useful signal is a shock wave received by an accelerometer far from an incidence of collision and the reference is the same wave received by a nearer sensor. These sensor signals propagate through a metallic structure and are therefore filtered by a resonance response. They generally have separated spectral bands, thus classical methods do not work.

We propose a new method of noise cancellation which is based on a nonlinear structure. This method requires only that the signals are dependent, and thus operates well even when the reference and the useful signal are uncorrelated.

After giving the definition of the method, we show how the proposed method resolves the non-overlapping spectra problem. We detail the explanation in the simple sinusoid case. The result will be extended to the general case. Some simulation results will also be presented.

2. PRINCIPLE OF THE METHOD

Noise Cancellation

We use a sensor signal \( u \) which contains a useful signal \( s \) and an additive noise \( w \):

\[
    u = s + w.
\]

We have also a reference signal \( x \) of the useful signal. It is assumed that \( x \) is independent of \( w \), but dependent on \( s \). The noise cancellation problem intends to estimate \( s \) by filtering \( x \).

When \( x \) is correlated to \( s \), one solution is to use Wiener linear filter \( h \) which can be obtained by minimizing the mean separated error (difference between \( u \) and the filter output \( y = h \ast x \)):

\[
    \min_h E \left[ \left| h \ast x - u \right| ^2 \right].
\]

In this paper, we consider the case where \( x \) is not necessarily correlated to \( s \) (classical second order sense). In other words, components of their spectra do not necessarily overlap:

\[
    |X(f)| \cdot |S(f)| = 0.
\]

Non-Linear Filtering

We propose to use the following non-linear filtering defined by:

\[
    y = h_n \ast x^n + h_{n-1} \ast x^{n-1},
\]

where \( x^n \) is the \( n \)th power of \( x \) and \( h_n \) a coefficient sequence. \( h_n \) is called \( n \)th order impulse response. Its memory length is denoted by \( M_n \).

The previous non-linear filter is just a reduced form of the Volterra filter: all cross terms \( x_{m \cdot n} \) and all terms of order lower than \( N-1 \) are omitted. Such a filter can be decomposed into two parts:

1) a fixed non-linear transformation from \( x \) to \( x^n, x^{n-1} \);

2) a coefficient dependent linear filtering. Although the filter is non-linear, the relation between the output and the filter coefficients is linear. As in the linear filter case, \( s \) can be estimated by minimizing the mean square error:

\[
    \min_{h_n, h_{n-1}} E \left[ \left| u - h_n \ast x^n - h_{n-1} \ast x^{n-1} \right|^2 \right].
\]
Algorithms such as LMS and RLS can therefore be used to calculate adaptively the optimal filter.

**For the Nonoverlapping Spectra**

If the non-linear part is missing, the band of a linear filter output \( B_s \) stays in the band of the input \( B_s \). When the band of the useful signal \( B_s \) is separated from \( B_s \): \( B_s \cap B_s = \emptyset \), it is also separated from \( B_s \). Thus, no linear filter can generate an output spectrum to estimate the useful spectrum.

Why can this problem be solved by using non-linearity? The basic idea is to use the harmonics of \( B_s \) to approach \( B_s \). It is easy to see in the previous decomposition of the filter that

1) By the nonlinear transformation, we generate in \( x^n \) and \( x^{n-1} \) the harmonics of the frequencies of \( B_s \). When \( n \) is suitably chosen, the harmonic bands will contain \( B_s \);
2) A linear filtering on \( x^n \) and \( x^{n-1} \) only weights the harmonic spectrum. The error minimization generates a filter whose output spectrum approaches the useful one in \( B_s \), and cancels the harmonics outside \( B_s \). The mathematical formulation of this investigation will be given in the following sections.

**3. SINUSOID CASE**

Consider at first the following simple case where \( x \) and \( s \) are both sinusoids:

\[
\begin{align*}
x_n &= \alpha \sin(2\pi f_1 n + \phi) \\
s_n &= \alpha' \sin(2\pi f_1 n + \phi')
\end{align*}
\]

We shall prove the existence of a reduced Volterra filter whose output \( y \) is exactly equal to \( s \).

**Harmonic Generation**

When the input of the filter (the reference \( x \)) is a sinusoid of the frequency \( f_1 \), what is the output like? By using a simple calculation (see Appendix 1), one obtains the Fourier transform of \( x^n \) written as:

\[
\text{FT}(x^n) = \sum_{k=-N}^{N} G(k/2) \delta_k n e^{j(2\pi/k) f_1 (k-2m)}
\]

If \( n \) is even, \( x^n \) contains all even harmonics \( 2mf \) from \( -Nf \) to \( Nf \), so as with \( x^{n-1} \) the odd harmonics \( (2m+1)f \). The situation is analogous if \( N \) is odd. In any case, all harmonics of \( f_1 \) from \( -Nf \) to \( Nf \) are generated by \( x^n \) and \( x^{n-1} \). That is why we use only the two highest order terms.

With the same parities as \( x^n \) and \( x^{n-1} \), the linear filters \( h_n \) and \( h_{n-1} \) control the gain of the even or odd harmonics. The Fourier transform of the output \( y(\tau) \) depends on the filter transfer function \( H_n(\tau) \), \( H_{n-1}(\tau) \) by:

\[
y(\tau) = \sum_{n=-N}^{N} H_n(\tau) e^{-j \omega \tau} \delta(\tau - n\tau)
\]

\[
A(nf_1) = \begin{cases} 
(\alpha/2j)^{n} & \text{if } n = N/2 \\
(\alpha/2j)^{n} & \text{if } n = N + 1/2
\end{cases}
\]

**Frequency equivalence**

We know that the spectrum of a digital signal is periodic of period 1: \( x(\tau) \equiv X(\tau + 1) \). For spectrum behavior; two frequencies \( \tau \), \( \tau' \) are equivalent, if they are equivalent modulo 1: \( \tau \equiv \tau' \pmod{1} \) (or \( \tau, \tau' \in \mathbb{Z} \)). In the following, we define all spectra in the interval \([0,1]\). By aliasing, the harmonics \( nf \) fold back into this interval, if it exceeds 1. Their equivalent frequencies in \([0,1]\) are called basic frequencies.

**Basic Frequencies of the Harmonics**

We define the mapping function \( g \) from \( \mathbb{N} \) to \([0,1]\) which corresponds to the harmonic number to the basic frequency:

\[
g(n) = nf_1 - [nf_1] \quad [x] \text{ is the integer part of } x
\]

We can see that \( \text{Image}(g) \) has a finite number of elements, iff \( f_1 \) is a rational number.

Now, let's consider the case of the rational frequency. Denote \( f_1 = p/q \) its irreducible form. It is easy to verify that

1) \( g \) is periodic of period \( q \): \( g(n+q) = g(n) \);
2) \( g(n) \) is a rational number with the same denominator: \( g(n) = x/q \), \( x \in \mathbb{Z} \).

So we can reduce the definition domain into:

\[
\{(1,2,\ldots,q) \} \quad \rightarrow \quad \{0,1/q,\ldots,(q-1)/q\}
\]

It follows from the Appendix 2 that \( g \) defined such, is a bijection. So there are \( q \) basic frequencies of the harmonics and they are uniformly distributed from 0 to 1. \( \text{Image}(g) = \{0,1/q,\ldots,(q-1)/q\} \).

**Estimation of the useful signal**
Assume that the useful signal \( s \) is also at a rational frequency: \( f_s = p'/q' \) and \( q' \) is a divisor of \( q; q'/q = b \in N \).

In this case, \( e_s \) is in \( \text{Imag}(g) \). There exists an integer \( N \) for which \( g(n) = e_s \). If we use a \( N \) order filter such that

\[
\begin{align*}
&h_{0-1} = 0 \\
&h_{n}(n\xi_s) = 0, \text{ for } n \in \{1, 2, \ldots, N-1\} \\
&h_{n}(N\xi_s) = (2j/\alpha) a^b e^{i\phi(n-m)}
\end{align*}
\]

all harmonics will be cancelled, except \( N\xi_s \). The output will be:

\[
Y(f) = a^{n'}(e^{i\phi_s} \delta_{f-f_s} + e^{-i\phi} \delta_{f+f_s}),
\]

and equal to the useful signal: \( Y_n = \delta_n \).

4. NON-ZERO BANDWIDTH CASE

When the signal is not a pure sinusoid but a non-zero bandwidth signal, it can be written as a sinusoid modulated by a lower frequency base band signal:

\[
\begin{align*}
X_n &= a_n \sin(2\pi f_s n) \\
Y_n &= b_n \sin(2\pi f_s n)
\end{align*}
\]

with \( a_n, b_n \) the base band signals. The \( k \)th power of \( X_n \) can be seen as \( a_n^k \) modulated by the \( k \)th power of the sinusoid:

\[
x_{n}^k = a_n^k \sin^k(2\pi f_s n).
\]

It is easy to prove that \( a_n^k \) is also a base band signal, but the bandwidth is larger than that of \( a_n \). Since the power of the sinusoid generates the harmonics at \( k\xi_s \), \( X^k \) is composed of harmonic bands which are centered at \( k\xi_s \). The useful signal band \( B_n \), centered at \( f_s \), will be included in one of the harmonic bands. By using a suitable filtering on these “power” signals, we can preserve the band \( B_n \) and attenuate all others. We then obtain a filtered signal \( y_n \) quite near to the useful signal \( s_n \).

5. SIMULATION RESULTS

Simulation results are shown in figures 1, 2, 3, and 4. The case of pure sinusoids is shown in figure 1, where an error convergence curve is given for an input signal at frequency \( f_s = 19/32 \), and a useful signal at \( f_s = 25/32 \). The filter of third order was calculated using LMS and was of memory \( \nu_m = 5 \). The filter converged after 1000 samples with an error residue on the order of 30 dB.

The results for the general case are shown in figures 2, 3, and 4. The signals involved are random, generated by a low pass filtered gaussian noise which is then sinusoid modulated to frequencies \( f_1 = 29/32 \) and \( f_2 = 26/32 \). The second order filter was again calculated by LMS with a memory length \( \nu_m = 10 \). The error residue was on the order of 20 dB after 1000 samples. Figures 2 and 3 are the spectra of the reference and useful signals respectively showing clearly that they do not overlap. Figure 4 shows the spectrum of the filter output and one sees clearly that the spectrum of the filter output is a good approximation of the useful signal spectrum. It is seen in both cases that the reduced order Volterra filters produce useful results, justifying the theory developed earlier.

CONCLUSION

We have investigated the situations of noise cancellation in which the useful and reference signals are not necessarily correlated (or having nonoverlapping spectral components). A nonlinear method has been proposed and requires only that the signals be dependent, not necessarily correlated. The mathematical investigation and the simulation results have confirmed the previous proposition.

REFERENCES

"Le filtrage de Volterra transverse rã©el et complexe en traitement du signal"
Traitement du signal, vol 7, No 5.

"Optimal Linear-Quadratic Systems for Detection and Estimation"

"A nonlinear adaptive noise canceller"
Proc. ICASSP 1980

APPENDIX I

We try to calculate the output of the reduced Volterra filter defined as:

\[
Y = h_{-1} * x_{-1} + h_{-2} * x_{-2} \\
\]

when the input signal is a sinusoid at the frequency \( f_s \): \( x_n = \alpha \sin(2\pi f_s n + \phi) \).

\[
x_n^{(k)} = (\alpha/2j)^{k} \sum_{n=0}^{k} \binom{n}{k} \alpha^{n-k} e^{i(2\pi f_s n + \phi)}
\]

\(
\text{FT}(x^k) = \sum_{n=0}^{k} \binom{n}{k} \alpha^{n-k} e^{i(2\pi f_s n + \phi)}
\)

\[
\text{FT}(x^k) = (\alpha/2j)^{k} \sum_{n=0}^{k} \binom{n}{k} \alpha^{n-k} e^{i(2\pi f_s n + \phi)}
\]
Denote $\gamma(f)$, $H_n(f)$ and $H_{n-1}(f)$ the Fourier transform. So,
\[
\gamma(f) = H_0(f) + H_{n-1}(f) \sum_{n=0, n \neq 0} (a/2j)^{nC_n} \delta f - (N-2m)f1 \\
+ H_n(f) \sum_{n=0, n \neq 0} (a/2j)^{nC_{n-1}} \delta f - (N-1-2m)f1 \\
- \sum_{n=0, n \neq 0} A_n(nf_1) e^{-2\pi i \delta f - nf_1},
\]
with
\[
A_n(nf_1) = \begin{cases} 
(\alpha/2j)^{nC_n} H_n(nf_1) & \text{if } n=N-2m \\
(\alpha/2j)^{nC_{n-1}} H_{n-1}(nf_1) & \text{if } n=N-1-2m
\end{cases}
\]

**APPENDIX 2**

We want to prove the following proposition:

If $f_1$ is a rational number and $p/q$ is its irreducible form, the function
\[
g(n) = nf_1 - [nf_1],
\]
with $[x]$ the integer part of $x$, is a bijection from $(1, 2, \ldots, q)$ to $(0, 1/q, \ldots, (q-1)/q)$.

a) $\text{Image}(g) \in (0, 1/q, \ldots, (q-1)/q)$:
\[
g(n) = nf_1 - [nf_1] = np/q - [np/q]
\]
and $[np/q] \leq np/q < [np/q]+1$

\[
\implies 0 \leq np-[np/q]q < q
\]

b) $g(n) = g(n') \implies n=n'$:

If $n>n'$, $g(n) = g(n')$
\[
\implies nf_1 - [nf_1] = n'f_1 - [n'f_1]
\]
\[
\implies f_1 = ([nf_1] - [n'f_1]) / (n-n').
\]
Since $p/q$ is the irreducible form of $f_1$, $(n-n')$ has to be divisible by $q$. But,
\[
nsq, \ n' \geq 1 \implies n-n' \leq sq-1,
\]
$(n-n')$ can not be divisible by $q$. That is the contradiction.

It follows from b) that Image$(g)$ has the same number of elements as the definition domain: Card(Image$(g)$)$=q$. This implies with a) that: Image$(g) = (0, 1/q, \ldots, (q-1)/q)$, and $g$ is a bijection.
DETECTION OF DETERMINISTIC TRANSIENT SIGNALS

A.MAGUER

SAACLANT Undersea Research Centre - 19138 - La Spezia - ITALY

Abstract: An overview of receivers (matched filter, matched subspace filter, CFAR matched subspace filter, energy detector and the adaptive receiver proposed by Porat and Friedlander) for detection of deterministic transient signals is performed. The adaptive receiver is implemented and its empirical performance (computed by Monte-Carlo simulations) is compared to the other theoretical receiver ones. Its performance is shown to be lower than the CFAR matched subspace filter ones (to which it theoretically approximately tends) but in most cases well over than the energy detector ones.

1. Introduction

Several methods have been used for the detection of transients [1,2,3]. In this paper we deal with classical receivers (matched filter, energy detector, matched subspace filter, CFAR matched subspace filter) and the adaptive receiver (proposed in [4] by Porat and Friedlander) for detection of deterministic transient signals. First, theoretical computation of each receiver is performed and visualised on ROC curves. It is shown (as in [4]) that the performance of the adaptive receiver tends asymptotically towards those of the matched subspace filter for some appropriate choice of parameter values, which is a very interesting result. Then, empirical computation of the adaptive receiver ROC curve is performed for a specific exponentially damped sinusoid in order to verify the approximations made by Porat and Friedlander for the computation of the adaptive receiver. Several duration and several values of the constraint parameter of the adaptive receiver have been tested. The last part of this paper is the comparison of the theoretical ROC curves of the classical receivers with the empirical ROC curve obtained with the adaptive receiver.

2. Computation of the receivers and their performance

The signal \( r \) observed at the receiver is the sum of signal \( s \) and noise \( n \) under \( H_1 \) and equal to noise \( n \) under \( H_0 \).

\[
E: \begin{cases} g \text{, under } H_0 \\ \xi + n \text{, under } H_1 \end{cases}
\]

where \( g \) and \( n \) are \( N \)-dimensional vectors, \( n \) is gaussian, centered, decorrelated, with variance \( \sigma_n^2 \), and \( E_s = \sum_{i=1}^{N} s_i^2 \) is the energy of the signal \( s \).

2.1 Matched filter

The matched filter corresponds to a situation where the signal waveform is known and represents an upper bound on the performance of any receiver. It has the structure:

\[
r_{MF}(r) = \frac{H_1}{\sigma_n^2} \lesssim H_0 \tag{2}
\]

Under \( H_0 \), \( r_{MF} \) is gaussian with mean and variance equal to 0 and \( \sigma_s^2 E_s \). Under \( H_1 \), \( r_{MF} \) is gaussian with mean and variance equal to \( E_s \) and \( \sigma_s^2 E_s \).

2.2 Energy detector

The energy detector makes no assumptions about the signal. Hence, it represents a lower bound on the performance for any receiver. It is given by:

\[
r_{ED}(r) = \frac{1}{\sigma^2 E_s} \lesssim H_1 \tag{3}
\]

Under \( H_0, r_{ED}(r) \) is distributed as a \( \chi^2 \) law with \( N \) degrees of freedom. Under \( H_1, r_{ED}(r) \) is distributed as a centered \( \chi^2 \) law with \( N \) degrees of freedom. The decentralized parameter is equal to \( \frac{E_s}{\sigma^2} \).

2.3 Receivers for partially unknown signals

In this part, the signal to be detected is supposed to be characterized by the impulse response of a rational transfer function. So, we have:

\[
s_k = -\sum_{i=1}^{p} a_i s_{k-i} + \sum_{j=1}^{q} b_j \delta(k-j) \tag{4}
\]

where \( \delta(k) \) is the Dirac function, \( a_i \) and \( b_j \) are the coefficients of the transfer function (respectively poles and zeros).

In matrix form, we have:

\[
A \cdot x = b \text{ or } x = A^{-1} \cdot b \tag{5}
\]

Then, by computing the singular value decomposition SVD of the matrix \( X_{(N,p)} \) deduced from matrix \( A^{-1} \) (by conserving the first \( q \) columns) we obtain from (5):

\[
\bar{x} = U \cdot \Lambda \cdot V^T \cdot b = U \cdot g \tag{6}
\]

where \( \Lambda \) is a diagonal matrix composed by the eigenvalues of \( X_{(N,p)} \cdot U_q \) and \( V \) are orthogonal matrices.

2.3.1 Matched subspace filter: \( a_i(\omega, \alpha) \) known, \( b_i \) unknown, \( \sigma \) known

For this receiver \( \sigma^2 \) and the \( a_i \) are supposed to be known. So, the problem in this case is the detection of signals with known modes but with unknown initial conditions.

The generalised likelihood ratio is equal to:

\[
G(r) = \frac{\sup_{\theta} p[S_r(\theta) \mid H_1]}{p[S_r \mid H_0]} \tag{7}
\]
Setting $\mathbf{t} = U'Z$, $Z$ is a Gaussian, where $U = [U, V]$ is an orthogonal matrix $(U'U = I_p)$ with $V$ an $(N, N-p)$ arbitrary matrix, the matched subspace filter can be written:

$$ r_{MSP}(r) = \frac{1}{\sigma^2} \mathbf{e}' P_s \mathbf{e} = \sum_{i=1}^{p} \left( \frac{t_i}{\sigma} \right)^2 H_i \geq \eta \tag{8} $$

$P_s = U_s U_s'$ is the orthogonal projector on the signal subspace.

Under $H_0$, $r_{MSP}(r)$ is distributed as a centered $\chi^2$ law with $p$ degrees of freedom. Under $H_1$, $r_{MSP}(r)$ is distributed as a decentered $\chi^2$ law with $p$ degrees of freedom. The decentralized parameter is the same as before.

2.3.2 CFAR matched subspace filter: $a_l(\omega, a)$ known, $b_l$ unknown, $\sigma$ unknown.

For this receiver the parameters $a_l$ are supposed to be known as for the previous receiver. However, $\sigma^2$ is unknown.

The generalized likelihood ratio is equal to:

$$ G(r) = \frac{\sup_{\psi} \left| Pr / \left( \psi, \sigma, H_1 \right) \right|}{\sup_{\psi} \left| Pr / \left( \psi, \sigma, H_0 \right) \right|} \tag{9} $$

After easy computations, the CFAR matched subspace filter defined as the function $(G(r))^{\frac{1}{2}} - 1$ can be expressed by:

$$ r_{CMSF}(r) = \frac{\|U'_{1} x\|^2}{\|U'_{1} z\|^2} = \sum_{i=p+1}^{p} \frac{e_i^2}{\lambda_i} H_i \tag{10} $$

Then, under $H_0$, $(N-p) r_{CMSF}(r)$ is distributed as a centered $F$ law with $p$ and $N-p$ degrees of freedom. Under $H_1$, $(N-p) r_{CMSF}(r)$ is distributed as a decentered $F$ law with $p$ and $N-p$ degrees of freedom. The decentralized parameter is the same as for the previous receiver.

2.3.3 Adaptive receiver: $a_l$ and $b_l$ unknown.

For this receiver, neither $a_l$ or the $b_l$'s are known. So, this is a more general case than the two previous receivers.

In [4], Porat and Friedlander have proposed the replacement of the maximum likelihood estimator $\hat{\theta}$ by a new estimator $\hat{\theta}_F$ which minimises:

$$ G_{PF}(\theta) = \frac{1}{\sigma^2} \left( \mathbf{e}^\prime - \mathbf{e}(\theta) \right)^\prime \left( \mathbf{e}^\prime - \mathbf{e}(\theta) \right) + \mu^2 \mathbf{e}' \mathbf{e} \tag{11} $$

where $\mu^2$ is a constraint parameter which allows to regularize the Jacobian matrix $J(\theta)$ (rank $m$) and to avoid numerical problems when only noise is present (no signal).

Then, Porat and Friedlander have set:

$$ \hat{\theta}_F = \hat{\theta} + \tilde{\theta} \tag{12} $$

and have also made the following hypotheses for performing approximate analysis of their receiver:

$$ d(\hat{\theta}_F) \sim d(\hat{\theta}) + J(\hat{\theta}) \tilde{\theta} \quad \text{and} \quad J(\hat{\theta}_F) \sim J(\hat{\theta}) \tag{13} $$

Here, $\sigma^2$ is supposed to be known (remark: In [4], Porat and Friedlander have also analyzed the case where $\sigma$ is unknown). The main results are summarized below.

Setting $g = U_{1} \mathbf{d}$, the adaptive receiver is given by:

$$ r_{AA}(r)_{H_0} = \sum_{k=1}^{m} \frac{\left( \lambda_k^2 + 2 \sigma^2 \mu^2 \right) \lambda_k^2}{\left( \lambda_k^2 + \sigma^2 \mu^2 \right)^2} \left( \frac{m_k}{\sigma^2} \right)^2 \tag{14} $$

$$ r_{AA}(r)_{H_1} = \beta_k \sum_{k=1}^{m} \left( \frac{m_k}{\sigma^2} + \frac{1}{\sigma^2 (\sigma - \alpha_k (V_j, d))} \right)^2 \tag{15} $$

with,

$$ g = U_{1} \mathbf{d}, \quad J(\mathbf{d}) = U_{1} \left( \begin{array}{c} \Lambda \\ 0 \\ 0 \end{array} \right) \tag{16} $$

$$ \beta_k = \frac{\left( \lambda_k^2 + 2 \sigma^2 \mu^2 \right) \lambda_k^2}{\left( \lambda_k^2 + \sigma^2 \mu^2 \right)^2}, \quad \alpha_k = \frac{\left( \sigma^2 \mu^2 \right)^2 \lambda_k}{\left( \lambda_k^2 + \sigma^2 \mu^2 \right)^2} \tag{17} $$

$$ \gamma = \frac{1}{\sigma^2} \left( g^\prime - d^\prime \right) \left( g - d \right) - \sum_{k=1}^{m} \frac{1}{\sigma^2 \beta_k} \left( m_k - \alpha_k (V_j, d) \right) \tag{18} $$

For rational signal all the $\lambda_k$ are equal to $1$. The rank $m$ is exactly equal to half the number of unknown parameters.

particular case: $\sigma \mu << 1$ (that means, as $\sigma$ is known, we must choose $\mu$ which verifies the previous conditions).

Then $\beta_k \approx 1, \alpha_k \approx 0, \text{and} \gamma \approx 0$. So, we deduce:

$$ r_{AA}(r)_{H_0} = \sum_{k=1}^{m} \left( \frac{m_k}{\sigma^2} \right)^2 \tag{19} $$

$$ r_{AA}(r)_{H_1} = \sum_{k=1}^{m} \left( \frac{m_k}{\sigma^2} + \frac{g_k}{\sigma} \right)^2 \tag{20} $$

So, under $H_0$, $r_{AA}(r)$ is distributed, as a central $\chi^2$ law with $m$ degrees of freedom and under $H_1$, as a non central $\chi^2$ law with $m$ degrees of freedom. $\mathbf{d}' \mathbf{d}$ is the decentralized parameter. As $g^\prime - d^\prime$ the decentralized parameter is the same as for the matched subspace filter.

Finally, when $\sigma \mu << 1$ the performance of the adaptive receiver (for $\sigma$ known) tends to the matched subspace filter one. So, the adaptive receiver will have the same performances as if the modes of the signal to be detected were perfectly known, which is a very interesting result and of course, the main interest of this detector.

3. Theoretical ROC curves

The theoretical ROC curves have been determined, for different given probability of false alarm $P_{fa}$, for different values of signal-to-noise ratio (from -10 dB to 20 dB). Some of the results are shown in figures 1 to 4, corresponding to $2$ values of length $N$ of the signal to be detected ($N=10$ or $N=60$) and two probability of false alarm $P_{fa}$ (0.1 and 0.001).
On each figure, 5 curves are plotted. Four of them correspond to the theoretical ROC curves of the energy detector, the matched subspace filter (and also asymptotically the adaptive receiver), the CFAR matched subspace filter and finally the matched filter. The fifth one is the empirical ROC curve of the adaptive receiver which will be discussed later in the paper.

The performance of the matched subspace filter (and so theoretically of the adaptive receiver) are always well upper than those of the energy detector. Sometimes, they are not very far from the matched filter performance, specially for long duration signal and low $P_{fa}$. So, it encourages the use of the adaptive receiver for the detection of deterministic transients.

4. Empirical ROC curves of the adaptive receiver

In order to validate the approximate results obtained by Porat and Friedlander, the adaptive receiver has been implemented by determining $\hat{\theta}_P$ which minimizes $C_{PP}(\theta)$. This is equivalent to a non-linear least squares problem for which a modified Gauss-Newton algorithm has been used.

4.1. Performance under $H_0$

The empirical cumulative distribution of $r_{AA}(r)$ under $H_0$ has been determined (1000 Monte-Carlo simulations were run) and compared to the theoretical one. Several values of the product $\sigma \mu$ and also different values of signal duration $N$ have been tested. Some of the results are visualized on figures 5 to 6 for values of the product $\sigma \mu$ equal respectively to 0.000001 and 5. In any case this distribution does not match any known statistical law and the differences between the theoretical and the empirical distribution are always great except for $\sigma \mu$ equal to 5 for which the two distributions are almost equal.

4.2. Performance under $H_1$

The computation of the probability of detection $P_d$ under $H_1$ has been computed with Monte-Carlo simulation (1000 runs) for an exponentially damped sinusoid with 0.05 as damping coefficient and 0.35 as normalized frequency. The phase $\phi$ of the signal is fixed to 0. The amplitude of the signal has been chosen to generate, according a given white noise of known variance $\sigma^2$, a variation of signal-to-noise ratio from -10 dB to 20 dB. The best results in term of detection are always achieved for the product $\sigma \mu$ equal to 1, and this whatever the duration of the signal, and the probability of false alarm $P_{fa}$. Some of the results are shown on figures 1 to 4 (see adaptive receiver curve).

5. Comparison theoretical-empirical

The last part of this paper is the comparison of the empirical ROC curve of the adaptive receiver (for $\sigma \mu = 1$ which gives the best results) with the theoretical one of the energy detector, the matched subspace and CFAR matched subspace filters and finally the matched filter (see figures 1 to 4).

For any value of $N$, differences between empirical distribution of the adaptive receiver and distribution of the matched subspace filter (which should be equal asymptotically) exist and are greater for small values of $N$. Moreover, the adaptive receiver is always greater than the energy detector, for great values of $N$. However, for small values of $N$ (see figure 1, $N=10, P_{fa} = 0.1$), the differences are not very great and even null and then can limit the interest of this detector for medium band or large band signals.

6. Conclusions

In this paper performance of classical receivers and adaptive receivers have been compared for the detection of deterministic rational transient signals. From a theoretical point of view the performance of the adaptive receiver has been shown to tend asymptotically towards the performance of the matched subspace filter which is very interesting. However, the empirical performance of the adaptive receiver developed by Porat and Friedlander has been shown, depending on the signal you try to detect and on the $P_{fa}$ you want, to be lower than the theoretical one. However, in passive sonar it will probably always be difficult, even if the frequency (but never the damping coefficient) of the signal to be detected is precisely known, to set one of the sonar frequency bins accurately to the particular frequency of the signal. Therefore, loss of detection performance (due to mismatch) will be observed with the matched subspace and the CFAR matched subspace filters, which will decrease their potential superiority against the adaptive receiver. Moreover, the adaptive receiver has been shown to be well better than the energy detector for narrow band signals. So, the adaptive receiver seems to be a good detector to use for detection of some deterministic transients.

7. References

[1] Invariant detection of transient ARMA signals with unknown initial conditions.
S.KAY-LL SCHARF.38.4.IICASSP 1984

B.FRIEDLANDER-B.FORAT.ASP. 37.2.1989

using sequenced short-time power spectra.
J.J.WOLCIN. NUSC report TMS31133.1983

B.FORAT-B.FRIEDLANDER.ASP. 34,6,1986
figure 1: Performance curves. N=10 and $P_{fa} = 0.1$

figure 2: Performance curves. N=10 and $P_{fa} = 0.001$

figure 3: Performance curves. N=60 and $P_{fa} = 0.1$

figure 4: Performance curves. N=60 and $P_{fa} = 0.001$

figure 5: Cumulative distribution under $H_0: \sigma = 0.000001$

figure 6: Cumulative distribution under $H_0: \sigma = 5$. 
A Class of Recursive Newton-Type Polynomial Extrapolation Filters

Seppo J. Ovaska
KONE Elevators
Research Center
P. O. Box 6, SF-005801 Hyvinkää, Finland

Olli Vainio
Tampere University of Technology
Signal Processing Laboratory
P. O. Box 553, SF-33101 Tampere, Finland

Modifications and extensions of Newton's extrapolation formula are introduced for digital signal processing purposes. The proposed class of computationally efficient polynomial extrapolation filters is called Recursive Linear Smoothed Newton (RLSN) predictors. Unlike the basic Newton algorithm, the RLSN predictors behave well also with noisy input signals. The characteristics of RLSN predictors compare favorably with other low-order polynomial prediction approaches.

1. INTRODUCTION

Extrapolation of limited data sets is a subject that dates back to antiquity, and comes up frequently in all application areas of digital signal processing, e.g., in instrumentation and measurement systems. The common problem is to forecast future samples of a primary signal corrupted by additive white Gaussian noise or uniformly distributed noise [1], [2]. An essential ingredient of the extrapolation problem is the postulation of a physical model for the underlying process [1]. In this paper, we consider polynomial extrapolation that is a special case of forward prediction. The primary signal \( u(p) \) is assumed to be a polynomial of degree \( M \), i.e.,

\[
u(p) = \lambda_0 + \lambda_1 p + \ldots + \lambda_M p^M
\]  

(1)

The polynomial coefficients \( \lambda_i, i = 0, \ldots, M \), are unknown real constants. The noisy input sequence, available to the predictor, is

\[
i(p) = u(p) + e(p)
\]  

(2)

where \( e(p) \) is the disturbing additive noise component. The difference equation of a general \( n \)-step-ahead predictor can be written as

\[
u(p + n) = \sum_{j=1}^{V} \alpha_j i(p + n - j) + \sum_{k=0}^{W} \beta_k i(p - k)
\]  

(3)

The coefficients \( \alpha_j \) and \( \beta_k \) are real constants. In the case of FIR predictors, all the feedback coefficients \( \alpha_j \) are equal to zero. In a noise-free situation we require that the steady-state prediction error is zero, i.e., \( u(p + n) = i(p + n) \).

The Newton's prediction algorithm has been used by mathematicians to extrapolate tables of polynomials and transcendental functions. Newton predictors are attractive for extrapolation of polynomials because of their low computational complexity and the straightforward design process. A general \( n \)-step-ahead Newton predictor of an \( M \)th-order polynomial can be expressed as the \( z \)-domain transfer function [3]

\[
H_M(z) = \sum_{k=0}^{M} (1 - z^{-n})^k
\]  

(4)

Successful DSP applications of Newton predictors have been reported in the literature. Athani and others used them to sample trend monitoring that was used for process failure detection [4]. Ovaska proposed Newton predictors for transmission error correction in a measuring oriented communication protocol [5], [6]. However, the predictors based on this computationally efficient algorithm have considerable gain at the higher frequencies because of the high-pass nature, and upper-band amplification of the difference operators. This property reduces the applicability of the basic Newton predictors in practical signal processing where the narrow-band primary signal is often corrupted by additive wide-band noise.

In this paper, we develop modifications and extensions of the original Newton predictors, aiming at considerably lowered overall noise gain while preserving the prediction property. The resulting new class of polynomial predictors, called the Recursive Linear Smoothed Newton (RLSN) predictors, is presented in the next Section. In Section 3, we quantitatively compare the characteristics of the RLSN predictors with some other predictor approaches, such as the optimal FIR predictors and IIR predictors designed in the frequency domain. Additional practical considerations and implementation issues are discussed in Section 4.

2. THE RLSN PREDICTORS

In a recent paper [7], two smoothed versions of the original Newton predictor were proposed. Those predictors were called the Linear Smoothed Newton (LSN) and the Median Smoothed Newton (MSN) predictors. The LSN predictor is recommended to be used with white Gaussian and uniformly distributed additive noises, and the MSN counterpart with impulsive or doubly exponentially distributed noise. In both cases, the smoothing operation is performed solely for the highest-order difference \( (1 - z^{-n})^M \) of the Newton predictor of Eq. (4). The highest-order difference has the largest noise gain, and it approximates the highest-order non-zero derivative that attains a constant value if the polynomial is noiseless. All the lower-order derivatives are time varying, and all the higher-order derivatives are equal to zero. The general \( n \)-domain transfer function of LSN \( n \)-step-ahead predictors can be written as

\[
G_M(z) = S(z)(1 - z^{-n})^M + \sum_{k=0}^{M-1} (1 - z^{-n})^k
\]  

(5)

where \( S(z) \) is the transfer function of some low-pass filter, preferably a moving averager,
\[ S(z) = \frac{1}{N} \sum_{n=0}^{N-1} z^{-n}, \quad (6) \]

that smooths the noisy \( M \)-th-order difference sequence \( N \) is a positive integer). Insertion of a linear smoother into the original algorithm reduces the highpass nature and noise gain of the Newton predictors without adding any steady-state prediction error to the result. The MSN predictor is obtained by replacing the linear filter \( S(z) \) with a median filter.

In order to further enhance the applicability of the LSN and MSN predictors, we first propose a recursive extension. It is desirable, however, to maintain the simple principal structure of the original LSN predictor. In this new approach, we feed back the predicted estimate \( \hat{p}(p) \) of the current input sample \( u(p) \), add it with weighting \( (1 - a) \) to the weighted true input \( u(p) \), and further add this sum to the sum of the outputs of the difference operators, which get their inputs \( u(p) \) directly without pre-scaling. This kind of manipulation does not cause any steady-state error into the predicted output. The modified z-domain transfer function can be expressed as

\[ P_z^{(2)}(z) = \frac{a + S(z)(1 - z^{-M}) + \sum_{m=1}^{M-1} (1 - z^{-m})^2}{1 - (1 - a)z^{-1}} \quad (7) \]

This transfer function has only one additional multiplication and addition as compared to the nonrecursive LSN counterpart, and with appropriate values of the parameters \( a \) and \( N \), it has the desired lowpass nature. The width of the passband is mainly controlled by \( N \), and the stopband attenuation by \( a \).

For example, in the case of the one-step-ahead ramp predictor, where \( S(z) \) is an \( N \)-length moving average of Eq. (6), the transfer function (7) can be reduced to the form

\[ P_z^{(1)}(z) = \frac{(a + 1/N)z^{-N} - z^{-N}/N}{1 - (1 - a)z^{-1}} \quad (8) \]

Delayed cascading of the basic sections (8) offers a viable way to further increase the stopband attenuation. It is also possible to obtain ramp predictors with a complex conjugate pole pair by cascading two basic sections with \( a = a_{re} + ia_{im} \) and \( a = a_{re} - ia_{im} \), respectively. The resulting transfer function can be expressed as

\[ L_z^{(2)}(z) = \frac{c_0 + c_N z^{-N} + c_M z^{-2N}}{1 - (-2a_{re} + 2)c_{re} - (a_{re}^2 + 2a_{re} - a_{im}) z^{-2} - c_{re}}, \quad (9) \]

where \( c_{re} = a_{re} + 2a_{re}/N + a_{re} + 1/N^2, c_{im} = 2a_{re}/N - 2/N^2, \) and \( c_{im} = 1/N^2. \)

Although the ramp predictor of the form (8) may have the lowpass nature, the predictors (7) of higher-order polynomials have still gain greater than or equal to unity at the higher frequencies. The reason for this is that only the primary input signal and the highest-order successive differences are smoothed; the primary signal by an 'exponential' averager, and its highest-order non-zero derivative by a 'uniform' averager. All the other difference paths have disturbing gain at the higher frequencies. Smoothing of the approximations of time-varying derivatives cannot be carried out by a conventional lowpass filter or moving averager, because they all introduce lag into the primary signal and thus distort the prediction function (7). To overcome this problem, we propose an enhanced version of the basic recursive transfer function of Eq. (7). In the new transfer function, the lower-order successive differences, corresponding to the time-varying derivatives, are smoothed by polynomial fil-

ters, i.e., zero-step-ahead predictors, and so the polynomial nature of the signal is fully exploited. First, we need an additional ramp predictor to obtain a predictor for second-order polynomials; then we use both the ramp predictor and the second-order polynomial predictor to obtain the predictor for third-order polynomials, etc. Thus the general transfer function can be written in a form that contains a recursion

\[ K_0^z(z) = a + S(z)(1 - z^{-1})^M + z^{-1} \sum_{m=1}^{M-1} K_m^z(1 - z^{-m})^2, \quad (10) \]

We call this new class of polynomial extrapolation filters Recursive Linear Smoothed Newton (RLSN). The RLSN extrapolator, depicted in Fig. 1, offers great flexibility to the user to optimize its characteristics for different signal shapes, noise conditions, and computational environments.

![Fig. 1. Data flow diagram of the RLSN predictor.](image)

Typical frequency response characteristics can be seen, for example, in the case of the second-order RLSN predictor. The transfer function can be reduced to:

\[ K_2^z(z) = \frac{1}{(1 + a_{re}) + (a_{im}^2 + a - 1)z^{-1} - (a_{im} + a)z^{-2}} \]

\[ \frac{1}{(1 - (1 - a)z^{-1})^2} \]

\[ \frac{1}{(1 - z^{-1})^2} \]

It has a double pole at \( z = 1 - a \), and \( N + 2 \) zeros. The frequency response is shown in Fig. 2 as a 3-D plot for \( a = 0 \ldots 1 \) and \( N = 16 \). For higher-order transfer functions, the multiplicity of the pole causes practical upper limits for the parameter \( a \), if undesirable bands of elevated gain are to be avoided. The practical lower limit of \( a \) is about 0.05 \ldots 0.1.

3. NUMERICAL COMPARISONS

In this Section, we compare the computational complexities and noise gains of different polynomial predictors, taking the second- and fourth-order cases as examples. Assuming that the additive zero-mean input noise has a flat power spectrum, i.e., the noise samples are mutually uncorrelated, the noise gain (NG) can be computed either in the time domain or in the frequency domain as [9]
where \( h_k \) is the impulse response of the prediction filter and \( H(e^{j\omega}) \) is the corresponding frequency response.

3.1. Other Approaches to Polynomial Prediction

The coefficients of a direct form FIR predictor can be optimized to minimize the noise power at the predictor output for a given polynomial degree [2]. The drawback of this approach is the number of arithmetic operations required, although recently a computationally efficient implementation has been introduced [8].

Besides the time-domain approaches discussed above, it is possible to design predictors in the frequency domain. For this purpose, one can try to approximate the ideal predictor characteristics by requiring that the magnitude \( |H(e^{j\omega})| \approx 1 \) and the group delay \( \delta \approx |H(e^{j\omega})| \approx 1 \). Such an approximation is only possible on a relatively narrow band \( \omega = 0 \ldots \omega_p \), and the resulting predictors are not strictly polynomial predictors. For a second-order IIR section, optimized using the method of steepest descent, the zero locations are at \( r_1 = 0.552, \theta_1 = -0.851 \), and the poles at \( r_p = 0.393, \theta_p = 0.805 \). For a fourth-order IIR predictor, optimization places the zeroes at \( r_{p1} = r_{p2} = 0.616, \theta_{p1} = \theta_{p2} = \pm 0.870 \), and the poles at \( r_{p3} = r_{p4} = 0.819, \theta_{p1} = \theta_{p2} = \pm 0.596 \). The prediction bandwidth \( \omega_p \approx 0.36\omega \), which is slightly wider than that of the RLSN predictors.

3.2. Results of the Comparison

In Tables I and II, we compare different types of one-step-ahead predictors for the second- and fourth-order cases, respectively. The following predictors are included in the comparison: the optimal FIR predictor [2], [8], the IIR predictor, the basic Newton predictor (Eq. (4)), and the three types of smoothed Newton predictors discussed in this paper (Eqs. (5), (7), and (10)). Note that for the IIR predictor, 'order' refers to the filter order rather than the order of the polynomial to be predicted.

It is clearly visible that the RLSN predictor offers the lowest noise gain with low computational complexity. The RLSN is also the only predictor that has true noise attenuation in fourth-order prediction - all the other predictors are amplifying the input noise. The frequency-domain approach does not give useful results for polynomial prediction.

Finally, let's equalize the noise gains of the optimal FIR predictor and the RLSN predictor, and compare their computational complexities. Choosing, for example, the second-order case and requiring that \( NG = 0.5 \), we get \( k = 2 \), for the FIR predictor, and \( a = 0.27 \) with \( N = 10 \) for the RLSN predictor. The number of arithmetic operations (MUL + ADD) in the FIR case is 18 while in the RLSN case it is only 11 (\( \approx 40\% \) less). The number of unit delays is almost equal in both cases. We can conclude that the RLSN predictors have excellent noise attenuation characteristics and low computational complexity when compared to other commonly used low-order polynomial predictors.

4. IMPLEMENTATION CONSIDERATIONS

After expanding the s-domain transfer functions of some low-order one-step-ahead RLSN predictors, we can see that in their final substituted numerical forms the numerators have sparse impulse responses with several coefficients equal to zero. The relative sparsity increases as \( N \) increases. For the first- and second-order predictors, the most efficient implementation is based on the data flow graph of Fig. 1 directly. The flow graph implementation makes it possible to easily adjust the parameters \( a \) and \( N \) dynamically. For higher-order predictors with fixed parameters, the most practical approach is to expand the recursive transfer function (10) into a direct-form rational transfer function of the form

\[
K_N(z) = \frac{\sum_{k=0}^{N-1} b_k z^{-k}}{N(1 - (1 - a)z^{-1})^M}
\]

This form of the transfer function leads to the canonical direct form implementation. For example, the non-zero numerator coefficients of the third-order predictor are:

\[
\begin{align*}
b_0 &= 1 + aN \quad b_1 = 2aN + 2a - 2 \\
b_2 &= a^2N + a^2 - 3aN - 4a + 1 \quad b_3 = 2a + 2aN - 2a^2 - 3a^2N \\
b_4 &= a^3N + a^3 \\
b_{N+1} &= 2 - 2a \\
b_{N+2} &= 4a - a^2 - 1 \\
b_{N+3} &= 2a^3 - 2a \\
b_{N+4} &= -a^3
\end{align*}
\]

The drawback of this approach is that the expanded coefficients must be recomputed every time the parameters are changed, or the polynomial order \( M \) is increased. This would be difficult to manage efficiently in a fixed-point signal processor implementation because of the large range of possible values.

The basic data flow graph of Fig. 1 can be easily divided into compact computational modules. Thus the efficient signal processor implementation is based on using subroutines or macros and constructing the program in a modular way. The parameters \( a \) and \( N \) are updated in memory locations. The recursive structure of the RLSN predictors is managed by allocating dedicated memory areas to the data at each level of the hierarchy.

In integrated circuit (ASIC) implementations, the word-length can be adjusted according to the worst-case requirements. On the other hand, the chip area must be kept as small as possible, which limits the allocation of hardware resources. The microprogrammable VLSI architecture of Fig. 3 could provide the desired flexibility in a relatively compact structure. The on-chip RAM stores the data samples and internal delay lines as ring buffer structures. The multiplier-adder configuration allows computations using integer or fractional coefficients, or their combinations (multiple clock cycles are needed to implement coefficients outside of the range \(-2 \ldots 2\)). The accumulator accumulates intermediate and final results, and stores them into the RAM. The operations are sequentially activated by the micropro-
Table I. Comparison of different second-order predictors.
(Notation: MUL = number of multiplications, ADD = number of additions/subtractions, DEL = number of unit delays.)

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Param.</th>
<th>NG</th>
<th>MUL</th>
<th>ADD</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal FIR</td>
<td>$k = 10$</td>
<td>0.73</td>
<td>5</td>
<td>13</td>
<td>21(1)</td>
</tr>
<tr>
<td>IIR</td>
<td>-</td>
<td>2.77</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Newton (4)</td>
<td>-</td>
<td>10.00</td>
<td>4</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>LSN (5)</td>
<td>$N = 10$</td>
<td>5.39</td>
<td>4</td>
<td>17</td>
<td></td>
</tr>
<tr>
<td>LSN-FB (7)</td>
<td>$N = 16$</td>
<td>1.36</td>
<td>2</td>
<td>5</td>
<td>18</td>
</tr>
<tr>
<td>RLSN (10)</td>
<td>$N = 16$</td>
<td>0.23</td>
<td>4</td>
<td>7</td>
<td>20</td>
</tr>
</tbody>
</table>

(1) Campbell-Neuvo implementation [8]
(2) with $a = 0.1$.

Table II. Comparison of different fourth-order predictors

<table>
<thead>
<tr>
<th>Predictor</th>
<th>Param.</th>
<th>NG</th>
<th>MUL</th>
<th>ADD</th>
<th>DEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optimal FIR</td>
<td>$k = 10$</td>
<td>3.66</td>
<td>9</td>
<td>15</td>
<td>15(1)</td>
</tr>
<tr>
<td>IIR</td>
<td>-</td>
<td>2.10</td>
<td>8</td>
<td>8</td>
<td>4</td>
</tr>
<tr>
<td>Newton (4)</td>
<td>-</td>
<td>251.00</td>
<td>2</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>LSN (5)</td>
<td>$N = 16$</td>
<td>73.53</td>
<td>-</td>
<td>8</td>
<td>19</td>
</tr>
<tr>
<td>LSN-FB (7)</td>
<td>$N = 16$</td>
<td>22.58</td>
<td>2</td>
<td>9</td>
<td>20</td>
</tr>
<tr>
<td>RLSN (10)</td>
<td>$N = 16$</td>
<td>0.50</td>
<td>18</td>
<td>17</td>
<td>22</td>
</tr>
</tbody>
</table>

(1) Direct form implementation
(2) with $a = 0.1$.

gram controller. In order to avoid the control logic becoming overly complicated, only a limited number of parameter combinations are supported.

The speed of this architecture is limited by the access time of the RAM, which is typically about 30 ns for contemporary CMOS technologies. Sampling rates exceeding 1 MHz could be achieved for low-order polynomial predictors.

5. CONCLUSIONS

The new Recursive Linear Smoothed Newton predictor offers good noise attenuation characteristics with low-order polynomials. The computational complexity of the RLSN predictors is usually low because the numerator of the transfer function has a sparse impulse response, i.e., most of the coefficients of the tapped delay line are equal to zero, and the order of the denominator is typically no more than three.

Several applications can be found for the RLSN predictor. These include use as a stand-alone predictor and a real-time curve fitter. Velocity predict filtering, commonly needed in motion control systems, is one potential application area of RLSN predictors. An appropriately delayed RLSN predictor can be used as a 'delayless' prefilter, or an estimator of a noisy tachometer signal if the underlying primary velocity pattern approximates some polynomial form.

Our future research will focus on developing adaptive extensions to the basic LSN- and RLSN-structures, and performing corresponding studies with the Median Smoothed Newton predictor. Although we have analyzed solely the properties of RLSN predictors in the one-dimensional case, the proposed approach can be applied to two-dimensional signal processing with similar benefits.

REFERENCES

Gibbs random field texture model

Witold Czarnecki

Institute of Telecommunications, Warsaw Technical University
ul. Nowowiejska 15/19, 00-665 Warszawa, POLAND

The Gibbs random field model of a texture having a given marginal probability distribution and predefined correlation coefficients is proposed. An algorithm of computer generation of such textures based on the model is described and some results obtained in generation experiments are presented.

1. Introduction

Textures play a very important role in the image processing. They are "portions of natural scenes that are devoid of significant details over large area and exhibit a repetitive structure" [1]. The textures can be the visual representations of spatially homogeneous structures both natural and man-made in general sense. The characteristic features of textures can be found also in signals which do not form images in ordinary meaning of the word. It is useful to apply image processing techniques to "non image" signals too. The textures represent often noisy background on which elements with information content occur. The texture modelling is a particular case of noise signal modelling. The visual representation of noise signals have the structure of textures i.e. have no details and are quasi-repetitive.

In numerous applications such 2-d signals are very important and it is desired to have computer programmes for their fast generation. If they are not white noises then their modelling and computer simulation is not easy [2]. It is mainly due to the difficulty of independent shaping of the probability distributions and the correlation. The noise signals are modelled as stochastic processes. The complexity of an adopted model depends on such properties as stationarity, normality, correlation and others. A model is valuable if it makes possible the definition of a simple algorithm suitable for computer simulation of the signal. It is not always easy to meet this requirement especially when a stationary noise signal is non-gaussian and has non-white spectrum. That is why the creation of a model of any particular class of noise signals is interesting if it assures the effective computer simulation.

The proposed 2-d signal model is of Gibbs random field (GRF) class. It makes possible the formulation of a simple generation algorithm to be used in computer programmes for simulation studies of image processing problems. In many applications the simulation studies are inevitable and fast generation of the 2-d random signals becomes necessary.

2. Problem formulation

Let \( X \) be a discrete 2-d stationary and homogeneous random signal defined over a rectangular lattice \( L \)

\[
X = \{x_{ij} ; (i,j) \in \mathbb{L} \}
\]

\[
\mathbb{L} = \{(i,j) ; i = 1, 2, \ldots, I ; j = 1, 2, \ldots, J \}
\]

The elements \( x_{ij} \) form a structure of \( I \) rows and \( J \) columns. For probabilistic description of \( X \) it is necessary to know the probability \( P(x_{ij}=x) \) where a subset \( X_{ij} \) of \( X \) is defined over an arbitrary subset \( L_{ij} \) of \( L \). It is often supposed that marginal probability distribution and the moments of the second order probability distribution is sufficient.

The problem is to define a model of 2-d signal \( X \) of marginal distribution \( p(x) \) and correlation coefficients between the neighboring elements of rows and columns \( p_{ij} \) and \( p_{ij} \) respectively. It is desired that the model of \( X \) enable to define a procedure of sequential generation of its elements \( x_{ij} \).
3. Gibbs Random Field Model

If a GRF model for a discrete, 2-d random signal $X$ is defined over $L$ then the neighbourhood system $\mu$ associated with it is a collection of the following subsets of $L$:

$$\mu = \{ \mu_{i,j} \}_{(i,j) \in L}$$  (2)

where $(i,j) \in \mu_{i,j}$ and if $(k,1) \in \mu_{i,j}$ then $(i,j) \in \mu_{k,1}$ for any $(i,j) \in L$. The elements $\mu_{i,j}$ called cliques such that:
- $c$ is a single element of $L$;
- if $c$ is not a single element of $L$ then a pair of elements $(i,j) \neq (k,1)$ belongs to $c$ if $(i,j) \in \mu_{k,1}$ form the set $C(L,\mu)$. The cliques composed of one or two elements of $L$ will be called pointcliques or paircliques respectively. The signal $X$ is a GRF with respect to the neighbourhood system $\mu$ if:

$$P(X=x) = P(x) = \frac{1}{Z} e^{-q(x)}$$  (3)

where $q(x) = \sum_{c \in C} V_c(x)$, $Z = \sum_x e^{-q(x)}$.

$V_c(x)$, $U_k(x)$ and $Z$ are the potential associated with the clique $c$, the energy function of the realization $x$ and the partition function respectively. To define a GRF model of $X$ it is necessary to determine the neighbourhood system $\mu$, the clique potentials $V_c(x)$ and the energy function $U_k(x)$.

4. Properties of GRF

The probability $P_k(x)$ is large when the energy $U_k(x)$ is small. The energy depends on the set of cliques $C$ and the potentials of each clique $V_c(x)$. So they define the nature of the signal $X$ and in consequence the correlations. Signal processing which modifies in a desired way its correlation can rely on forming such a permutation of signal elements which minimizes the given energy function. In such a case the desired modification of the correlation will not be followed by any change of the marginal distribution.

A realization $x$ of the GRF $X$ defined by $\mu$, $V_c(x)$ and $U_k(x)$ can be a permutation of elements of some auxiliary signal $X_{\mu}$ having the same marginal probability as $X$. The permutation which minimizes the energy function $U_k(x)$ is the most probable realization of $X$ composed of the elements of $X_{\mu}$. The GRF model enables to compare particular realiza-

tion probability. It is natural then to permute the elements of a simple in sense auxiliary signal $X_{\mu}$ (a white noise for example) in order to find the most probable realization of $X$.

To generate a GRF correlated signal $X$ it is necessary to repeat many times, (like in the Metropolis algorithm), the following sequence of operations:
- choose at random a pair of elements $X_{i,j}$ and $X_{k,l}$;
- exchange their values if it takes the signal to the lower energy state.

The repetition of these operations will result in an enhancement of signal features defined by the energy function $U_k(x)$. In some cases its form could be expressed by horizontal and vertical correlation coefficients $\rho_x, \rho_y$.

The main inconvenience of the above procedure is its "globality". It needs prior generation of $X_{\mu}$. For great values of $I$ and $J$ it could be cumbersome. It is possible to modify the procedure to make it "local". If the lattice $L$ is divided in the following parts: $L^I$, where the lattice form already the generated signal $X$, $L-I^I-(i,j)$ where the elements form the auxiliary signal $X_{\mu}$, yet $(i,j)$ between them then it is possible to generate an elements $X_{i,j}$ as follows:
- for each $(i,j)$ taken in deterministic order of generation choose $K$ different elements $X_{k,\mu}$; $k=1,2,\ldots,K$ taken at random from among $L-I^I-(i,j)$ positions,
- choose one out of $K$ pairs $X_{i,j}, X_{k,\mu}$ for which the exchange of values takes the signal in $L^I+(i,j)$ part to the lowest energy state,
- exchange the values of the elements of the chosen pair $X_{i,j}$.

The sequence of $K$ random elements $X_{k,\mu}$ of $X_{\mu}$ can be arranged in a set $R$ not necessarily of 2-d nature. When the auxiliary signal $X_{\mu}$ is a white noise the set $R$ is composed of $K$ independent random numbers having the probability distribution $p(x)$. One element from among them is chosen as $X_{i,j}$, provided it minimizes the energy function $U_k(x)$ over the region $L^I+(i,j)$.

5. Energy function

The correlations of the signal $X$ depends on the energy function $U_k(x)$. The first order neighbourhood system $\mu$ defines only two types of 2-element cliques: horizontal (both elements of the same row) and vertical (both elements of the same column). Such a neighbourhood system was adopted in all numerical experiments. It is easy to foresee that the correlations will depend on the
fig.1. Rayleigh texture and its correlation function (lexicographic order of generation, K=20, $p_1=0.6$, $p_2=0.3$).

fig.2. Rayleigh texture and its correlation function (meanderlike order of generation, K=20, $p_1=0.6$, $p_2=0.3$).

fig.3. Rayleigh texture realizations for $p_1=0.9$, $p_2=0.9$ (upper row) and $p_1=0.9$, $p_2=0.9$ (lower row).

fig.4. Rayleigh texture realizations for $p_1=0.3$ and from upper left quadrant in clockwise manner: $p_1=0.2$, 0.4, 0.8, 0.6.

fig.5. Rayleigh texture realizations for $p_1=0.6$ and from upper left quadrant in clockwise manner: $p_1=0.1$, 0.3, 0.7, 0.5.

fig.6. Rayleigh texture and its correlation function (lexicographic order of generation, K=20, $p_1=0.99$, $p_2=0.99$).
paircliques potentials. The following energy function was proposed
\[ U_c(z) = | \sum_{c \in C_c} V_c(z) |^2 + | \sum_{c \in C_c} V_c(z) | \]  
with pairclique potentials
\[ V_{(i,j),(m,n)}(z) = (x_{i,j} - x_{m,n})^2 - 2(1 - \rho_c) \sigma_x^2 \]

In the preceding formulae the sets of horizontal and vertical paircliques of the neighbourhood system \( \mu \) are denoted by \( C_\mu \) and \( C_\mu' \) respectively. The marginal probability \( p(x) \) is of mean value \( m_\mu \) and variance \( \sigma_\mu^2 \). The correlation coefficient between the elements of pairclique \( c \) is \( \rho_c \). It is easy to show that the simultaneous minimization of the two positive terms of \( U_c(x) \) (4) will guarantee the desired correlation properties of \( X \).

6. Generation algorithm

The GRF model of the texture \( X \) enables to define the following generation algorithm. An element \( X_{i,j} \) can be taken as one out of \( K \) independent random numbers \( R_k \) of \( R \) which fits to the set \( X^{10} \) of all previously generated elements
\[ X^{10} = \{ X_{i,j} \} \]

If the elements of the signal are generated in a row after row manner and always from left to right (lexicographic order) then \( L^{10} \) has the following form
\[ L^{10} = \{ (k,l); 1 \leq k \leq l \leq n \} \]

The elements can also be generated in a row after row manner but from left to right in the odd numbered rows and from right to left in the even numbered rows (meanderlike order) \( [8] \). In this case the set of previously generated elements for \( (i,j) \) from odd numbered rows takes the form as follows
\[ L^{10} = \{ (k,l); 1 \leq k \leq l \leq n \} \]

The criterion of fitness is the value of the energy function of \( U_c(x^{10} + \tilde{x}_{10}) \). After each generation the chosen number \( R_k \) in \( R \) is replaced by a new random number. The generation algorithm takes the following form:
- for the given \( p(x) \), \( \rho_\mu \) and \( \rho_c \) define the potentials \( V_{(\mu)}(x) \) as in (5),
- for the fixed value of \( K \) create the set \( R \) of \( K \) random numbers \( R_k \),
- take for \( X_{i,j} \) a random number of probability distribution \( p(x) \),

\[ X_{i,j} = \text{random}_{p(x)} \]

- in a row after row manner generate the successive elements of the signal taking for \( X_{i,j} \) the value of \( R_k \) which minimizes the energy function \( U_c(x^{10} + \tilde{x}_{10}) \); replace \( R_k \) in \( R \) by a new random number,

\[ U_c(x^{10} + \tilde{x}_{10}) \leq U_c(x^{10} + \tilde{x}_{10} + R_k); k=1,2,\ldots,K \]

\[ X_{i,j} = R_k; R_k = \text{random}_{p(x)} \]

7. Simulation experiments

The computer simulations were done for the Rayleigh textures. It was found that for \( K=10 \) the goodness of fit hypothesis for the marginal distribution of the generated signal was successfully verified and the relative errors of the correlation coefficients did not exceed 10%. The results of the simulation experiments (all for \( K=20 \)) are presented on the fig.1-6. The figures represent the generated Rayleigh textures in the form of black and white images. In 3 cases the estimated correlation function are presented too.

References

NOVEL SCHUR TYPE ALGORITHM FOR NEAR TO TOEPLITZ LINEAR SYSTEMS WITH MULTICHANNEL ENTRIES

A. Liavas & S. Theodoridis, Dept. of Computer Eng., University of Patras, Patras 26500, Greece

An efficient Schur type algorithm for block $\rho$-Toeplitz system solution is developed. The algorithm avoids matrix operations since it exploits the near to Toeplitz property of the coefficient matrix on a scalar level. As a consequence it can be implemented on a linear array processor consisted of scalar operators locally interconnected.

1. INTRODUCTION

Multichannel system identification, multirate filtering and seismic signal enhancement are some typical cases which result in the solution of a well structured block linear system of equations (block Toeplitz or block near to Toeplitz).

The notion of $\rho$-Toeplitz was introduced in [1], with $\rho$ serving as a measure of how near to Toeplitz the matrix is.

All the algorithms for block $\rho$-Toeplitz system solution derived so far are straightforward generalizations of the single channel case [1, 2]. Matrix variables of order $k$ and block vectors with entries $(k \times k)$ matrices appear in the place of scalar and vector variables respectively, complicating the resulting architectures associated with the Schur type algorithms, which offer enhanced parallelism compared to their Levinson type counterparts. Recently motivated by [3], we proposed a stairwise Levinson type algorithm for the solution of a block $\rho$-Toeplitz system which exploits the near to Toeplitz property of the coefficient matrix on a scalar level avoiding order $k$ matrix operations [4]. In this paper starting from the stairwise Levinson type algorithm we will develop a Schur type algorithm which can be mapped in a straightforward way on a linear array processor, involving scalar operations and simplified localized interprocessor communications.

2. THE ALGORITHM

Suppose that we want to efficiently solve

$$R_K = -d$$

(1)

where $R$ is a $(p \times p)$ block near-to-Toeplitz matrix with entries $(k \times k)$ matrices and $d$ is a $(p \times 1)$ block vector with entries $(k \times 1)$ vectors.

We define $R_m$ ($1 \leq m \leq p$) the block matrix that results from $R$ after deleting its last $(p-m)$ block rows and columns and $S_m$ ($1 \leq m \leq p-1$) the block matrix that results from $R_{m+1}$ after deleting its first block row and column. Thus

$$R_{m+1} = \begin{pmatrix} R_m & \ast \\ \ast & \ast \end{pmatrix} = \begin{pmatrix} \ast & \ast \\ \ast & S_m \end{pmatrix}$$

(2)

where * denotes block elements of no current interest.

Since $R$ is a block near to Toeplitz matrix then [2]:

$$S_{p-1} - R_{p-1} = \sum_{i=2}^{p} (\pm \mu_i) \mu_i^T$$

(3)
where \( n_i \) and \( n'_i \) are \(((p - 1) \times 1)\) block vectors with entries \((k \times 1)\) vectors. \( \rho \) is assumed to be small with respect to \( p \). If there is no representation in (3) with smaller number of terms, \( \rho \) is called the displacement rank. It can easily be seen that the low rank property of the difference of the two submatrices \( S_{p-1}, \hat{R}_{p-1} \) is carried out for all principal submatrices of \( R \). Thus if one defines

\[
H = \begin{pmatrix} n_1 & \cdots & n_p \end{pmatrix} \quad H' = \begin{pmatrix} n'_1 \\ \vdots \\ n'_p \end{pmatrix}
\]

then

\[
S_m - R_m = H_m \Sigma H'_m \quad \text{for} \quad m = 1, \ldots, p-1
\]

(5)

\( H, H' \) can be thought as \(((p - 1) \times 1)\) block vectors with entries \((k \times \rho)\) matrices. \( H_m (H'_m) \) results from \( H (H') \) after deletion of its last \((p-m)\) block rows (columns). \( \Sigma \) is a \((\rho \times \rho)\) signature matrix with \( \pm 1 \) on its main diagonal and zeros elsewhere.

In order to derive the stairwise Levinson type algorithm we define [4]:

a) \( R_{mk+i}^j \): \((mk + i) \times (mk + i)\) submatrix of the block matrix \( R \), whose top leftmost element is the \((j, j)\) element of \( R \).

\[
R_{mk+i}^j = \begin{pmatrix} r(j, j) & \cdots & r(j, mk+j+i-1) \\ \vdots & \ddots & \vdots \\ r(mk+j+i-1, j) & \cdots & r(mk+j+i-1, mk+j+i-1) \end{pmatrix}
\]

(6)

b) \( \xi^{(r, y)}_{mk+i} \): \(1 \times (mk + i)\) vector. It is part of the \( r \)-th row of \( R \) with leftmost element \( r(r, y) \).

\[
\xi^{(r, y)}_{mk+i} = \begin{pmatrix} r(r, y) & \cdots & r(r, mk+y+i-1) \end{pmatrix}
\]

(7)

c) \( \xi^{(x,y)}_{mk+i} \): \((mk + i) \times 1\) vector. It is part of the \( y \)-th column of \( R \) with topmost element \( r(x, y) \).

\[
\xi^{(x,y)}_{mk+i} = \begin{pmatrix} r(x, y) \\ \vdots \\ r(x, mk+y+i-1) \end{pmatrix} (8)
\]

d) \( S_{mk+i} \): \((mk + i) \times (mk + i)\) principal submatrix of the block matrix \( S_{p-1} \). At this point we should note that since \( S_{p-1} \) results from \( R \) after deleting its first block row and column — on a scalar level, after deleting its first \( k \) rows and columns — then:

\[
S_{mk+i} \equiv R_{mk+i}^{k+i}
\]

(9)

It can easily be seen from (2-5) that for \( m = 0, \ldots, p-2 \) and \( i = 1, \ldots, k \):

\[
S_{mk+i} \equiv R_{mk+i}^{k+i} = R_{mk+i}^{i} + H_{mk+i} H'_{mk+i}
\]

(10)

where \( H_{mk+i} (H'_{mk+i}) \) results from \( H (H') \) after deleting its last \((p - m - 1)k - i\) rows (columns).

e) \( d_{mk+i} \): \((mk + i) \times 1\) vector that results from \( d \) after deleting its last \((p - m)k - i\) elements.

One block order update is achieved in \( k \) single steps by the recursive solution of

\[
R_{mk+i}^{i} \xi_{mk+i} = -d_{mk+i} \quad i = 1, \ldots, k
\]

(11)

The three basic internal variables of the stairwise Levinson type algorithm of [4] are given in Table 1. It is well known that Levinson type algorithms are not well suited for parallel implementation. Their drawback stems from the way the reflection coefficients of order \( mk + i + 1 \) are computed [4]. They require the computation of a \((mk + i)\)-dimensional inner product or matrix-vector product which in a parallel environment can be performed in at least \( O(\log_2 (mk + i)) \) time units.

On the other hand, Schur type algorithms bypass
the above computational bottleneck by avoiding the
direct computation of inner products. These, are in
turn provided by a set of lattice type operations which

This can be achieved by defining a set of auxiliary
error variables, whose values for certain combinations
of their indices coincide with the required inner prod-

Define for \( mk + i \leq n \leq pk \) the error variable:

It is readily observed that \( \beta^{F}_{mk+i} = e^{F}_{mk+i}(mk + i) \).

It turns out that for each time instant \( n \), variables
\( e^{F}_{mk+i}(n) \) are propagated in order via lattice type op-

Furthermore, these operations can be per-
formed concurrently for all the time instants \( n \) involved
and this is the point where enhanced parallelism is
achieved compared to Levinson type algorithms.

In Table 2 we give the definitions of some of the
internal variables of the stairwise Schur type algeb-

(due to space limitation), while their order updates are
given in Table 3. (Similar results state for the rest
internal variables).

The algorithm can be performed on a linear array
processor composed by \( O(pk) \) locally intercon-
ected computing elements of the multiply-accumulate type
in \( O((\rho + k)pk) \) time units. Interprocessor com-
munications involve only scalar interchanges (\( \rho \)-dimensional
internal variables are passed element by element) mak-
ing the resulting structure simple.

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal variables of the Levinson type</strong></td>
</tr>
<tr>
<td><strong>Stairwise Algorithm</strong></td>
</tr>
<tr>
<td>[ 1 \leq j \leq k \quad \text{and} \quad 1 \leq \mu \leq k ]</td>
</tr>
<tr>
<td>[ a_{mk+i}^{\mu} = \begin{cases} (R_{mk+i}^{\mu})^{-1} L_{mk+i}^{(\mu,\mu-1)} &amp; \mu \neq 1 \ (R_{mk+i}^{\mu})^{-1} L_{mk+i}^{(k+1,k)} &amp; \mu = 1 \end{cases} ] (1.1)</td>
</tr>
<tr>
<td>[ b_{mk+i}^{\mu} = \begin{cases} (R_{mk+i}^{\mu-1})^{-1} L_{mk+i}^{(j-i, mk+j)} &amp; 0 \leq i &lt; j \ (R_{mk+i}^{\mu})^{-1} L_{mk+i}^{(k+j-i, (m+1)k+j)} &amp; j \leq i &lt; k \end{cases} ] (1.2)</td>
</tr>
<tr>
<td>[ F_{mk+i} = -(R_{mk+i}^{1})^{-1} H_{mk+i} ] (1.3)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Table 2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Internal variables of the Schur type</strong></td>
</tr>
<tr>
<td><strong>Stairwise Algorithm</strong></td>
</tr>
<tr>
<td>[ e_{mk+i}^{\mu}(n) = L_{mk+i-1}^{(n,1)} e_{mk+i-1} + d_{n} ] (2.1)</td>
</tr>
<tr>
<td>[ e_{mk+i}^{F}(n) = L_{mk+i-1}^{(n,1)} F_{mk+i-1} + b_{n}^{t} ] (2.2)</td>
</tr>
<tr>
<td>[ e_{mk+i}^{a_{\mu}}(n) = \begin{cases} L_{mk+i}^{(n,\mu)} e_{mk+i-1}^{\mu} + r(n,\mu-1) &amp; \mu \neq 1 \ L_{mk+i}^{(n,1)} e_{mk+i-1}^{1} + r(n+k, k) &amp; \mu = 1 \end{cases} ] (2.3)</td>
</tr>
<tr>
<td>[ e_{mk+i}^{b(j)}(n) = \begin{cases} L_{mk+i-1}^{(n,j-i+1)} e_{mk+i-1}^{j} + r(n, mk+j) &amp; j \geq i \ L_{mk+i}^{(n,k+j-i+1)} e_{mk+i-1}^{j} + r(n, (m+1)k+j) &amp; j &lt; i \end{cases} ] (2.4)</td>
</tr>
<tr>
<td>[ \beta_{mk+i}^{C} = e_{mk+i}^{C}(mk+i) ] (2.5)</td>
</tr>
<tr>
<td>[ \beta_{mk+i}^{F} = e_{mk+i}^{F}(mk+i) ] (2.6)</td>
</tr>
<tr>
<td>[ \beta_{mk+i}^{a}(n) = e_{mk+i}^{a}(mk + \mu + i - 1) ] (2.7)</td>
</tr>
</tbody>
</table>
Table 3
Order updates of the variables of Table 2

\[
\begin{align*}
&\text{for } n = mk + i + 1 \text{ to } pk \text{ in parallel do } \\
&\quad e_{mk+i+1}^e(n) = e_{mk+i}^e(n) + b_{mk+i}^{(i)}c_{mk+i}^e \\
&\quad e_{mk+i+1}^F(n) = e_{mk+i}^F(n) + c_{mk+i}^{(i)}b_{mk+i}^F \\
&\quad e_{mk+i+1}^{a(t)}(n) = e_{mk+i}^{a(t)}(n) + c_{mk+i}^{(i)}a_{mk+i}^{(t)} \\
&\quad e_{mk+i+1}^{a(1)}(n) = e_{mk+i}^{a(1)}(n) + c_{mk+i}^{(i)}a_{mk+i}^{(1)} \\
&\quad e_{mk+i+1}^{b(i)}(n) = e_{mk+i}^{b(i)}(n) + c_{mk+i}^{(i)}b_{mk+i}^{(i)} \\
&\quad (e_{mk+i}^e(n) e_{mk+i}^{a(1)}(n)) K_{mk+i}, \\
&\end{align*}
\]

end for \ n

\[
\begin{align*}
&\text{for } j = 1 \text{ to } k \ (j \neq i) \\
&\quad \mu = j \circ i \\
&\quad \text{for } n = mk + \mu + i \text{ to } pk \text{ in parallel do } \\
&\quad \quad e_{mk+i+1}^{a(\mu)}(n) = e_{mk+i}^{a(\mu)}(n) + \\
&\quad \quad \quad + c_{mk+i}^{(i)}a_{mk+i}^{(\mu)} \\
&\quad \quad e_{mk+i+1}^{b(j)}(n) = e_{mk+i}^{b(j)}(n) + \\
&\quad \quad \quad + c_{mk+i}^{(i)}b_{mk+i}^{(j)} \\
&\quad \end{align*}
\]

end for \ n

end for \ j

REFERENCES:


[3] G. Gentiis, N. Kalouptsidis: ‘Multichannel IS filtering and system identification and Block adap-

Using Hidden Variable Fractal Interpolation to Model One-Dimensional Signals*

Greg Vines and Monson H. Hayes, III

Georgia Tech Lorraine
2-3 Rue Marconi, 57070 Metz, France

Abstract

Hidden variable fractal interpolation has been proposed as a flexible way to model one-dimensional signals through the use of iterated function systems. While methods have been proposed for signal reconstruction and for determining map parameters for the hidden variable model, no algorithm exists for determining the necessary hidden variables. In this paper we propose and evaluate several methods for determining these hidden variables, including an optimal approach based on a least-squares solution to the problem. The various methods are tested on several data samples and results are provided showing an improvement in the SNR over modeling with the fractal interpolation method alone.

1 Introduction

Iterated Function Systems (IFSs) have recently received attention in the literature as a new technique for signal modeling [1, 2]. This interest stems from the fact that an IFS is simple in form and yet capable of producing complicated functions, many of which are fractal in nature [3]. IFSs have, in fact, been used to create images and other waveforms that closely resemble those found naturally [4]. In this paper we investigate the use of the Hidden Variable Fractal Interpolation (HVFI) technique to model one-dimensional signals.

Because IFSs are typically implemented with affine maps, the resulting data is self-affine. This property may not always be desirable. The use of the HVFI method allows a self-affine model to be created in a higher dimensional space, while the projection of this data is not constrained to be self-affine [3, 5]. The problem in using the HVFI method is how to determine the best set of data in the additional dimension.

2 Background

The fractal interpolation method of modeling data is based on selecting points from the data samples and creating IFS maps to interpolate between these points. The maps may then be used to recreate the original data. The task is therefore two-fold: first to determine the appropriate interpolation points, and second to determine the best map parameters. The determination of the map parameters has typically been accomplished through a minimization of some error criterion. The selection of interpolation points has turned out to be a tougher problem. Present methods rely on iterative searches with no guarantee that an optimal set of maps will be chosen.

HVFI is a method where a data sequence is placed into a higher dimensional space and modeled there. We will work exclusively with one-dimensional sequences in this paper, which will be interpreted as a two-dimensional graph, residing in \( x \times y \)-space, and generate \( z \) data for these sequences to place them into three dimensions where they are modeled with an IFS. The original data can then be recovered by taking a projection of the reconstruction of the higher dimensional data. Through the selection of the third dimension’s data, the HVFI model offers a great deal of flexibility.

2.1 Hidden Variable Fractal Interpolation

The HVFI method is based on using an IFS to interpolate between selected sample points in \( R^3 \). We are given the original data sequence, \( x[n] \), which consists of \( N \) data samples for \( N_1 \leq n \leq N_2 \), and have to determine the additional dimension’s data, \( z[n] \), prior to modeling. A variety of methods will be examined to determine this \( z[n] \) data. With a given \( z[n] \) and \( x[n] \), the \( j \)th map out of \( P \) maps is set up in the following manner

\[
\begin{bmatrix}
    z[n] \\
    x[n]
\end{bmatrix} = \begin{bmatrix}
    a_j & 0 & 0 \\
    c_j & d_j & h_j
\end{bmatrix} \begin{bmatrix}
    n \\
    x[n]
\end{bmatrix} + \begin{bmatrix}
    e_j \\
    g_j
\end{bmatrix}, \quad (1)
\]
where the 0 entries are necessary to insure the resulting function is single valued. The \( j \)th map covers \( M_j \) of the data samples for \( M_j \leq n \leq M_{j+1} \). The set of \( P \) maps are constructed to cover the entire data sequence from \( N_1 \) to \( N_2 \). Figure 1 illustrates how a typical map might be set up to generate points from \( M_1 \) to \( M_3 \). For notational simplicity the \( j \) subscript will be omitted in references to the individual maps from now on.

3 Map Parameters

In previous implementation of the fractal interpolation method, the interpolation points were used to eliminate some of the variables by forcing the endpoints of the maps to be exactly equal to the original data \([6]\). Experimental results have shown that a more accurate model is obtained if the interpolation points are not forced to be exactly correct \([7]\) and we will take advantage of this increased accuracy. The map parameters are determined through a least squares minimization of the error between the data, \( x[n] \) and \( z[n] \), and the mapped data, \( w_k[n] \) and \( w_l[n] \),

\[
\xi = \sum_{n=N_1}^{N_2} (w_k[n] - x[n])^2 + (w_l[n] - z[n])^2, \quad (2)
\]

where \( w_k[n] \) and \( w_l[n] \) are the map functions for the \( x[n] \) and \( z[n] \) data respectively. Using least squares to minimize this error gives the following sets of equations:

\[
\begin{bmatrix}
S \\
ST
\end{bmatrix}
\begin{bmatrix}
c \\
d \\
h \\
f
\end{bmatrix}
= \sum_{i=0}^{M-1} S \cdot z[M_1 + i], \quad (3)
\]

and

\[
\begin{bmatrix}
S \\
ST
\end{bmatrix}
\begin{bmatrix}
k \\
m \\
g \\
f
\end{bmatrix}
= \sum_{i=0}^{M-1} S \cdot z[M_1 + i], \quad (4)
\]

where

\[
S = \begin{bmatrix} n & z[n] & z[n] & 1 \end{bmatrix}^T,
\]

\[
n = N_1 + \left( \frac{N-1}{M-1} \right) i,
\]

and \( T \) indicates the transpose.

Once the map parameters have been calculated, the map must be checked to see if it is a contraction mapping. This can be done by examination of the sub-matrix

\[
A = \begin{bmatrix} d & h \\ l & m \end{bmatrix}.
\]

Essentially this matrix is performing a rotation and scaling of the \( x[n] \) and \( z[n] \) data during the mapping. If the determinant of \( A \) has a magnitude less than one, then it is a contraction mapping.

The remaining tasks in using the HVFI method are the determination of the \( x[n] \) data and selection of the interpolation points. Several methods were developed to determine the \( x[n] \) data based on a least squares minimization of the error for each map.

4 Hidden Data Determination

4.1 An Optimal Approach

The optimal \( x[n] \) data is obtained in a least-squares sense through the error equation (2) for all of the maps. Because the \( x[n] \) data is used by all of the maps together, it is not possible to work with each map individually. For each of the \( P \) maps there is an error equation of the form (2). Examination of (2) with regard to \( z \) shows that there are three distinct ranges depending on the value of \( i \) or \( n \), where \( n \) is defined as in (5).

1. \( M_1 \leq n \leq M_2 \) and \( i \neq \frac{(M_1 - N_1)(M + 1)}{M - N} \)
2. \( M_1 \leq n \leq M_2 \) and \( i = \frac{(M_1 - N_1)(M + 1)}{M - N} \)
3. \( n \) outside of \{ \( M_1 \cdot M_2 \) \}

The range \( M_1 \leq n \leq M_2 \) is in terms of \( i \) given by

\[
\left( \frac{(M_1 - N_1)(M + 1)}{M - N} \right) \leq i \leq \left( \frac{(M_2 - N_1)(M + 1)}{M - N} \right).
\]

Taking the partial derivative of (2) with respect to \( z \) for each of these three cases yields three equations:

\[
m \cdot z[n] - z[M_1 + i] = -k \cdot n - l \cdot z[n] - g, \quad (8)
\]

\[
(h^2 + m^2 - m) \cdot z[n] + (1 - m) \cdot z[M_1 + i] = (1 - n)k \cdot n + l \cdot z[n] + g + h(z[M_1 + i] - c \cdot n - d \cdot z[n]) - f, \quad (9)
\]

\[
-(h^2 + m^2) \cdot z[n] + m \cdot z[M_1 + i] = h(c \cdot n + d \cdot z[n] + f - z[M_1 + i]) + m(k \cdot n + l \cdot z[n] + g). \quad (10)
\]

For continuous functions, the IFS maps are constructed to be 'just touching', which means that adjacent maps share end points. For the discrete case
this is not necessary and would result in an overdetermined system of equations. Therefore the maps are set up with adjacent, instead of overlapping, end points. With adjacent end points, the \( P \) maps provide \( N \) equations for the \( N \) unknowns. The nice feature of this system of equations is the sparseness of the matrix; there are only two non-zero coefficients in each row - one at \( n \) and the other at \( M_i + i \), except in the second case above, where \( n = M_i + i \). The system of equations was solved using an iterative method, both the Gauss-Seidel and the Conjugate-Gradient methods were implemented with similar results.

The algorithm to determine the map parameters and the \( z[n] \) data proceeds as follows.

1. Initialize \( z[n] \)
2. Calculate the map parameters using the method from section 3
3. Calculate \( z[n] \) data
4. Calculate error and compare to previous; if the change is small then stop, otherwise go to step 2

Obviously the last step offers some flexibility in that the iterations may be allowed to continue if the error is decreasing, or stopped after a certain number of iterations, and in addition, a check should be made for divergence. The initial value chosen for \( z[n] \) will have some affect on the convergence of the algorithm. There is no guarantee of convergence for a given initial value. The only value which should not be used is \( z[n] \) itself, which will lead to a singular set of equations when the map parameters are calculated.

4.2 Two Quick Iteration Methods

Rather than solve the system of equations for all cases, if instead \( z[n] \) and \( z[M_i + i] \) are treated as different variables then a simpler solution is derived. When the partial of (2) with respect to \( z[n] \) and \( z[M_i + i] \) is set to zero, the following two equations are obtained:

\[
(k^2 + m^2)z[n] + (-m)z[M_i + i] = \\
h(c \cdot n + d \cdot x[n] + f - z[M_i + i]) \\
+ m(k \cdot n + l \cdot x[n] + g) \tag{11}
\]

and

\[
(m)z[n] + (-1)z[M_i + i] = k \cdot n + l \cdot z[n] + g. \tag{12}
\]

These equations can be solved for either \( z[n] \) or \( z[M_i + i] \) resulting in a single equation to calculate \( z[n] \) directly from \( z[n] \). The two equations obtained are:

\[
z[M_i + i] = \frac{m}{h}(x[M_i + i] - c \cdot n - d \cdot x[n] - f) \\
+ k \cdot n + l \cdot z[n] + g. \tag{13}
\]

and

\[
z[n] = \frac{x[M_i + i] - c \cdot n - d \cdot z[n] - f}{h}. \tag{14}
\]

While these two equations are not optimal, they can generate the necessary \( z[n] \) much more quickly than solving the above system of equations.

The above iterative algorithm is again used to determine the best set of maps and corresponding \( z[n] \) data.

4.3 Independent Hidden Variables

With the aforementioned methods one problem which occurs is the interdependence between \( z[n] \) and \( z[M_i] \) during the reconstruction process. Errors tend to propagate and in an interest of stopping this, the link from \( z[n] \) to \( z[M_i] \) was broken. This was implemented by setting the \( l \) variable in all of the maps equal to zero. With this modification equation (4) becomes

\[
\sum_{l=0}^{M-1} \left[ \begin{array}{c} k \\ m \\ g \end{array} \right] = \sum_{l=0}^{M-1} \delta \cdot z[M_i + i],
\]

where

\[
\delta = \left[ \begin{array}{ccc} n & z[n] & 1 \end{array} \right]^T.
\]

This method was implemented with each of the methods for determining \( z[n] \) and the results are given in the following section.

Another variation on this idea is to generate the \( z[n] \) data as suggested previously, create an IFS to model the \( z[n] \) data, use this IFS to recover \( z[n] \), and finally use \( z[n] \) when the map parameters are determined for \( x[n] \). This method offers the advantage that \( z[n] \), which is used in determining the \( z[n] \) map parameters, can be reconstructed without error. This method did not show any improvements over the others and will not be discussed further.

4.4 Non-Iterative Method

All of the aforementioned methods rely on iteratively calculating \( z[n] \) and the map parameters in order to find a solution to the nonlinear system of equations. The computational requirements can be greatly reduced by avoiding these iterations. The system was tested after one iteration of the Independent Hidden Variable method as described in section 4.3 and the results are provided in the next section. The single iteration was set up by first solving for the map parameters, then determining the \( z[n] \) data and finally with this new \( z[n] \) data, recalculating the map parameters.

5 Results

Because the task of determining the interpolation points is still an unsolved problem, we used an exhaustive
search for a fixed number of maps to allow comparison between the various methods proposed for determining $z[n]$.

Table 1 shows results for the different methods discussed including results from a regular IFS, without the $z[n]$ data, for comparison. The tests were conducted with a variety of test files and numbers of maps. Each of the test data sets is a file from an image file and was normalized prior to modeling to facilitate comparisons. Because the exhaustive search rapidly becomes computationally infeasible for more than a few maps, calculations were not made for more than 4 maps.

<table>
<thead>
<tr>
<th>File</th>
<th>Method</th>
<th>2 Maps</th>
<th>3 Maps</th>
<th>4 Maps</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Regular IFS</td>
<td>20.0</td>
<td>23.2</td>
<td>24.5</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>20.2</td>
<td>23.7</td>
<td>25.2</td>
</tr>
<tr>
<td></td>
<td>Quick: $z[n]$</td>
<td>20.3</td>
<td>23.3</td>
<td>23.1</td>
</tr>
<tr>
<td></td>
<td>Quick: $z[M_1 + i]$</td>
<td>20.3</td>
<td>22.9</td>
<td>23.1</td>
</tr>
<tr>
<td></td>
<td>Independent $z[n]$</td>
<td>20.3</td>
<td>23.7</td>
<td>25.2</td>
</tr>
<tr>
<td></td>
<td>Non Iterative</td>
<td>20.3</td>
<td>23.7</td>
<td>25.2</td>
</tr>
<tr>
<td>2</td>
<td>Regular IFS</td>
<td>22.6</td>
<td>25.6</td>
<td>26.8</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>22.7</td>
<td>25.4</td>
<td>26.1</td>
</tr>
<tr>
<td></td>
<td>Quick: $z[n]$</td>
<td>22.8</td>
<td>25.2</td>
<td>27.5</td>
</tr>
<tr>
<td></td>
<td>Quick: $z[M_1 + i]$</td>
<td>18.0</td>
<td>24.9</td>
<td>27.4</td>
</tr>
<tr>
<td></td>
<td>Independent $z[n]$</td>
<td>22.9</td>
<td>26.3</td>
<td>28.2</td>
</tr>
<tr>
<td></td>
<td>Non Iterative</td>
<td>22.4</td>
<td>26.3</td>
<td>28.2</td>
</tr>
<tr>
<td>3</td>
<td>Regular IFS</td>
<td>25.4</td>
<td>31.0</td>
<td>33.3</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>25.4</td>
<td>31.3</td>
<td>33.4</td>
</tr>
<tr>
<td></td>
<td>Quick: $z[n]$</td>
<td>25.3</td>
<td>30.9</td>
<td>33.1</td>
</tr>
<tr>
<td></td>
<td>Quick: $z[M_1 + i]$</td>
<td>25.2</td>
<td>31.1</td>
<td>33.6</td>
</tr>
<tr>
<td></td>
<td>Independent $z[n]$</td>
<td>25.4</td>
<td>31.3</td>
<td>33.6</td>
</tr>
<tr>
<td></td>
<td>Non Iterative</td>
<td>25.4</td>
<td>31.2</td>
<td>33.6</td>
</tr>
<tr>
<td>4</td>
<td>Regular IFS</td>
<td>26.4</td>
<td>30.4</td>
<td>33.0</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>26.4</td>
<td>30.4</td>
<td>33.0</td>
</tr>
<tr>
<td></td>
<td>Quick: $z[n]$</td>
<td>26.2</td>
<td>30.4</td>
<td>33.1</td>
</tr>
<tr>
<td></td>
<td>Quick: $z[M_1 + i]$</td>
<td>24.8</td>
<td>30.1</td>
<td>32.8</td>
</tr>
<tr>
<td></td>
<td>Independent $z[n]$</td>
<td>26.2</td>
<td>30.4</td>
<td>33.3</td>
</tr>
<tr>
<td></td>
<td>Non Iterative</td>
<td>26.3</td>
<td>30.4</td>
<td>33.3</td>
</tr>
<tr>
<td>5</td>
<td>Regular IFS</td>
<td>11.9</td>
<td>15.6</td>
<td>16.9</td>
</tr>
<tr>
<td></td>
<td>Optimal</td>
<td>14.0</td>
<td>17.0</td>
<td>19.8</td>
</tr>
<tr>
<td></td>
<td>Quick: $z[n]$</td>
<td>14.0</td>
<td>17.3</td>
<td>19.6</td>
</tr>
<tr>
<td></td>
<td>Quick: $z[M_1 + i]$</td>
<td>13.6</td>
<td>16.8</td>
<td>18.9</td>
</tr>
<tr>
<td></td>
<td>Independent $z[n]$</td>
<td>14.3</td>
<td>17.3</td>
<td>19.9</td>
</tr>
<tr>
<td></td>
<td>Non Iterative</td>
<td>14.2</td>
<td>17.0</td>
<td>19.9</td>
</tr>
</tbody>
</table>

Table 1: HVIFS Model Performance

6 Conclusion

Each of the methods provided similar results, which suggests using the simplest method to reduce the computational requirements. It turned out that the selection for the initial $z[n]$ made a difference and in general the best results were obtained with a constant value of 0.01.

The simple method for calculating $z[n]$ with independent hidden variables and initial $z[n] = 0.01$ gave the best results of all the methods presented. Of all the iterative approaches, this method also requires the least computations for solving a particular map. The non-iterative version of the independent hidden variables method also provided very good results with significantly fewer computations required. The difference in the results of these two methods is so small that the non-iterative method is probably the best approach overall.

References


All-poles modelling using a linear combination of cumulants

Josep Vidal and Jose A. R. Fonollosa

Universitat Politècnica de Catalunya
E.T.S.E.Telecomunicació, Apdo. 30002, 08080 Barcelona, Spain
E-mail: pepe@tsc.upc.es Fax:34-3-4016447 Tel:34-3-4016457

We address the problem of estimating the parameters of a pure AR causal model excited with non-Gaussian i.i.d. samples of an unobservable process. Output samples may be corrupted with colored Gaussian noise. Other types of noise are allowed provided that they have a set of m-th order statistics whose value is zero, if the same set of statistics of the signal be different of zero. It is shown that each sample of the impulse response of the AR system that generates the process may be expressed as a linear combination of cumulant slices of any order. The resulting algorithm is shown to be well-behaved and to provide consistent estimates while reducing the complexity significantly with respect to other approaches.

I. Introduction

Last years have witnessed a great development of higher-order based techniques in system identification problems. Most of the efforts made in ARMA modeling in the past were based on the properties of the autocorrelation function as an optimum way to deal with Gaussian processes and linearities. However, in many applications were non-Gaussian processes or nonlinearities are present, second order approaches fail to find proper estimations. This is the case also if the system to identify is not minimum phase. Cumulants and their Fourier transform (polyspectra) do contain information about deviations of normality, nonlinearity and phase [4], and as such their use becomes very interesting. Another important property of higher-than-two order statistics (HOS) is that they give unbiased results even when the observed signal is corrupted with additive colored Gaussian noise. Therefore, the interest in their use has risen in other domains, as blind equalization, speech recognition, array processing, non-linear filtering, time-frequency spectral analysis, etc.

In this paper we will deal with the all-poles spectrum estimation of a non-Gaussian process corrupted with noise using HOS. For a complete review on the topic see [3]. The approach followed here makes use of the principles found in [7] and [8], that is combination of different cumulant slices in a linear algebra problem to render a consistent algorithm that guaranties uniqueness. Cumulants of different orders can be combined, and other cumulant slices added to improve the performance. The final algorithm requires much less computation than the HOS Yule-Walker equations while still giving low variance estimates.

II. AR impulse response determination

Consider a zero-mean and ergodic process \( y(n) \), its m-th order cumulant function is defined as the coefficients of the Taylor series expansion of its characteristic function [4]. More precisely the third and fourth order cumulants of the process are defined as:

\[
C_3, y(i,j) = E\{y(n)y(n+i)y(n+j)\}
\]

We will assume that \( y(n) \) is in \( \mathbb{R} \) and that:

\[
C_4, y(i,j,k) = E\{y(n)y(n+i)y(n+j)y(n+k)\} - \\
E\{y(n)y(n+i)\}E\{y(n)y(n+j)\} - \\
E\{y(n)y(n+i)\}E\{y(n)y(n+k)\} - \\
E\{y(n)y(n+j)\}E\{y(n)y(n+k)\}
\]

By fixing m-2 lags of this function, a 1-dimensional sequence is obtained, usually called a 1-D cumulant slice. Assume that \( y(n) \) is an AR(p) causal process:

\[
y(n) = \sum_{i=1}^{p} a(i)y(n-i) + c(n)
\]

and fitting the following hypotheses:

H1: The innovation process \( e(n) \) is stationary, i.i.d., non-Gaussian, with finite m-th order cumulants \( \gamma_{m,e} \), and

\[
\gamma_{m,e} = 0.
\]

H2: \( H(2) \) is exponentially stable, that is, all poles are inside the unit circle.

H3: The process \( z(n) \) is independent of \( e(n) \), zero-mean, Gaussian and of unknown power spectrum. \( z(n) \) is allowed to be non-Gaussian if \( \gamma_{m,e} = C_{m,z}(0,...,0) = 0 \) for some \( m \geq 3 \) provided that \( \gamma_{m,e} \neq 0 \) for the same \( m \).

The property that makes HOS useful is that cumulants of any order are zero for a Gaussian process, so the cumulants of \( x(n) \) equal the cumulants of \( y(n) \) provided that \( z(n) \) be Gaussian.

The impulse response \( h(n) \) of the linear time-invariant system that generates \( y(n) \) satisfies the following recursion:

\[
\sum_{i=0}^{p} a(i)h(n-i) = \delta(n)
\]

Using this recursion and the Brillinger-Rosenblatt summation formula [5] that relates the m-th order cumulants of \( x(n) \) with the impulse response (for \( m \geq 2 \)):

\[
C_m(x(i_1, i_2, ..., i_m)) = \gamma_m \sum_{n=-\infty}^{\infty} \prod_{k=0}^{m-1} h(n+i_k) \quad i_0 = 0
\]
Linearly combining equation (3) with the AR parameters and using (2) we get:
\[
\sum_{k=0}^{m-1} a(k) C_{m,n}(i_1, i_2, \ldots, i_{m-1}) = \gamma_m \prod_{k=0}^{m-1} h(i_k)
\] (4)

Repeating the operation (m-1) times for the (m-1) lag coefficients, \(\gamma_m\) can be obtained [9]:
\[
\sum_{l_1=0}^{P} \ldots \sum_{l_{m-1}=0}^{P} a(l_1) \ldots a(l_{m-1}) C_{m,n}(l_1, \ldots, l_{m-1}) = \gamma_m
\] (5)

In [1] it was shown that the impulse response \(h(n)\) can be obtained from (4) taking the lags \(l_1=l_2=\ldots=l_{m-1}=0\):
\[
\sum_{l=0}^{P} a(l) C_{m,n}(l, 0, \ldots, 0) = \gamma_m h(i) \quad h(0) = 1
\] (6)

However this is not the only linear combination of cumulants samples that yields \(h(n)\). In fact, it is easy to see from (4) that any linear combination of the type:
\[
W(i) = \sum_{l_1=0}^{P} \sum_{l_2=0}^{P} \ldots \sum_{l_{m-1}=0}^{P} a(l_1) w(l_2) \ldots w(l_{m-1}) C_{m,n}(l_1, l_2, \ldots, l_{m-1})
\] (7)

renders \(h(n)\), that is \(W(i) = \mu h(i)\), provided that:
\[
\mu = \gamma_m \prod_{l_1=0}^{P} w(l_1) h(i) = 0
\] (8)

Relations (7) and (8) are also satisfied by any linear combination of slices of order equal or greater than two. We will try to go further showing that the weighted sum of cumulant slices (w-slice):
\[
C_w(i) = w_2 C_{2,n}(i) + \sum_{j=M}^{N} w_3(j) C_{3,n}(i,j) + \sum_{j=M}^{N} \sum_{l=0}^{N} w_4(j,l) C_{4,n}(i,j,l) + \ldots
\] (9)

may yield the impulse response \(h(n)\) without previous knowledge of \(a(i)\), if the weights are appropriately chosen. This is proven in the following proposition, letting constants \(M\) and \(N\) unspecified for the moment:

**Proposition 1:** If \(C_w(i)\) is causal, then \(C_w(i) = C_w(0) h(i)\).

*Proof:* Let us linearly combine the w-slice of (9) with the AR parameters:
\[
\sum_{k=0}^{P} a(k) C_{w}(n-k) = w_2 \sum_{k=0}^{P} a(k) C_{2,n}(n-k) + \sum_{j=M}^{N} w_3(j) \sum_{k=0}^{P} a(k) C_{3,n}(n-k,j) + \sum_{j=M}^{N} \sum_{l=0}^{N} w_4(j,l) \sum_{k=0}^{P} a(k) C_{4,n}(n-k,j,l) + \ldots
\] (10)

combining (2), (4) and (10) the following equality is obtained:
\[
\sum_{k=0}^{P} a(k) C_{w}(n-k) = \gamma_m w_2 + \gamma_2 \sum_{j=0}^{N} w_3(j) h(j) + \gamma_4 \sum_{j,l>0} w_4(j,l) h(j) h(l) + \ldots = \gamma_m h(0) \delta(n)
\] (11)

Note that taking \(n=0\) in (11) the Yule-Walker set of equations for the w-slice \(C_w(i)\) are obtained. If the weights \(w_2, w_3(j), w_4(j,l), \ldots\) are chosen in such a way that:
\[
C_w(i) = 0 \quad i < 0
\] (12)

by similarity between (11) and (2) we get that:
\[
C_w(i) = C_w(0) h(i)
\]

This is an important theoretical result but does not help much in finding a proper set of weights because the conditions stated in (12) suppose an infinite set of linear equations to be solved. Fortunately, we can reduce the number of equations considering the following proposition:

**Proposition 2:** If \(C_w(i) = 0\) for the set \(A:\{i=1..P\}\), being \(P\) an upper bound of \(p\), then \(C_w(i) = C_w(0) h(i)\) for the same set \(A\).

*Proof:* The proof is very simple and it follows from (11) taking \(0 \leq n \leq P\) and by similarity with (2).

It is worth noting that second order statistics can also be used as constructing the w-slice if the signal is known to be non corrupted by noise.

### III. AR w-slice algorithm

We have just found a hint on how to estimate the impulse response through a limited size linear algebra problem. More specifically, proposition 2 gives the set of equations to be considered. In practice, estimation of the AR parameters needs only \(p+1\) samples of \(h(n)\), so the resulting set can be condensed in the matrix equation:
\[
S_a w = 1
\] (13)

where \(S_a\) is the anticausal w-slice matrix:
\[
S_a = \begin{pmatrix}
C_{2,n}(-P) & C_{3,n}(-P_j) & \cdots & C_{4,n}(P_j,k) & \cdots \\
\vdots & \vdots & \ddots & \vdots & \vdots \\
C_{2,n}(0) & C_{3,n}(0,j) & \cdots & C_{4,n}(0,j,k) & \cdots \\
\end{pmatrix}
\]

\(w\) is the weights vector:
\[
w = (w_2 \quad w_3(j) \quad w_4(j,k) \quad w_5(j,k,l) \quad \ldots)^T
\]

and \(1\) the anticausal impulse response:
\[
1 = (0, \ldots, 0, 1)^T
\]

Although in general the rows-rank of \(S_a\) is unknown and may depend on the actual parameters of the process, there is always at least one exact solution to (13). Considering (7) and (8) it may be seen that if the weights \(w(i)\) equal the actual AR parameters, \(h(n)\) is obtained. This is a proof of existence of solution of the vector \(w\) but not of uniqueness. In fact we are not concerned with uniqueness since proposition 1 shows that any solution yielding (12) is a valid one. The use of the Moore-Penrose inverse matrix \(S_a^d\) in this case ensures that, among the subspace of exact solutions, the minimum-norm one is obtained if the superscript and subscript \(N\) and \(M\) in (9) and (10) are specified in such a way that the solution \(w(i)= a(i)\) be
included. It means that if an upper bound P of p is known, the range of values are to be N=0 and M-P. If SVD is used in the computation of $S_a^p$, the solution will always be well conditioned.

The final AR w-slice algorithm can be resumed in three steps:

S1: Computation of the minimum norm weights that yields a causal w-slice with $C_w(0)=1$:

$$w_m = S_a^p 1 \quad (14)$$

S2: Estimate the causal part of the impulse response using:

$$\hat{h} = S_c w_m \quad (15)$$

where $S_c$ denotes the causal counterpart of $S_a$:

$$S_c = \begin{pmatrix} C_{2x}(P) & C_{3x}(P,j) & \cdots & C_{4x}(P,j,k) \\ \vdots & \vdots & \ddots & \vdots \\ C_{2x}(0) & C_{3x}(0,j) & \cdots & C_{4x}(0,j,k) \end{pmatrix}$$

and $\hat{h}$ is the estimated impulse response:

$$\hat{h} = (\hat{h}(0), \hat{h}(1), 1)^t$$

S3: Solve (2) by simple backsubstitution, that is:

$$\begin{pmatrix} \hat{h}(0) & 0 & \cdots & 0 \\ \hat{h}(1) & \hat{h}(0) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \hat{h}(P-1) & \hat{h}(P-2) & \cdots & \hat{h}(0) \end{pmatrix} \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_P \end{pmatrix} = \begin{pmatrix} -\hat{h}(1) \\ -\hat{h}(2) \\ \vdots \\ -\hat{h}(P) \end{pmatrix} \quad (16)$$

Once the AR parameters have been estimated, the higher-order moments $\gamma_p$ of $e(n)$ can be computed from (5). It is worth pointing that the values of M and N are parameters of design, and as they are chosen to consider more cumulative lags, a progressive reduction in variance might be expected. However this reduction may not be so because it depends also on the variance of the estimated cumulants being included. Of course, the length of the data record as well as the theoretical minimum variance limit the improvement. The backsubstitution approach in S3 suffers from the effect of the accumulation of the variance of the estimated AR parameters as the order is increased. This can be solved choosing deliberately a big value of $P$ and overdetermining the system in (16). In solving the overdetermination Total Least Squares or Least Squares can be used. Simulations will show the improvement in this case. This issue has little importance in those applications using the impulse response rather than the AR parameters. In these cases step S3 is skipped. Note that although order determination cannot be stated from the rank of $S_a$ this decision can be done on the final vector $a$.

IV. Computational complexity

Once the methods have been described, some comments on their computational cost are in order. The simulations in section V will exhibit the behavior of the method AR w-slice version in section III with respect to the High Order Yule-Walker equations. The main computational cost for each algorithm consists of two parts: 1) the sample mean estimation of the cumulants, which depends only on the number of data points and 2) the computation of the vector of parameters, which depends mainly on the order of the model. The computation of the first part is specially significant if the data record is large, and might exceed the second one. The computational burden of the second part is mainly due to the matrices size.

Solving a 3rd order Yule-Walker set of equations requires SVD on a $p \times p(p+1)$ matrix, if $p+1$ slices are used to ensure identifiability. The AR w-slice algorithm proposed in III, with the use of 3rd order statistics, solves a matrix of size $(p+1) \times 2(p+1)$ if $M=N=p$ in step S1. Computations of step S2 imply $p(2p+1)$ floating point operations. Step S3 needs $p(p+1)/2$ flops if the system is not overdetermined. Although being conceptually different, the minimum norm solution to the AR w-slice equations - used in the Yule-Walker equations solution - are computed by means of the Moore-Penrose inverse matrix via SVD. In all cases the number of flops required for an $m \times m$ matrix (being $m=n$) is $m^2 + 17/3 n^3$ [2]. Table I summarizes the total number of flops required in each case, once the cumulants have been computed. Clearly the computational demands of the AR w-slice proposed algorithm are significantly smaller than those of the other approach.

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>3rd order Yule-Walker</th>
<th>AR w-slice</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matrices size</td>
<td>$p \times n(p+1)$</td>
<td>$(p+1) \times 2(p+1)$</td>
</tr>
<tr>
<td>Floating point operations</td>
<td>$p^3 + 20p^2 + 20p + 2$</td>
<td>$2p^3 + 22p^2 + 5p + 5$</td>
</tr>
</tbody>
</table>

V. Simulations

The goal of these simulations is to illustrate the performance of the Yule-Walker HOS equations versus the AR w-slice algorithm proposed in section III. Both methods use in this case third-order statistics and, for meaningful comparisons, all are adapted to deal with the same non-redundant set of cumulants. All examples involve pure causal AR processes. The driving noise is formed with zero-mean, i.i.d. samples of an exponential process with $\gamma_2 = 1$. In each example 100 independent Monte Carlo runs are performed, and the sample mean of each realization is subtracted in every run.

Example 1. As previously mentioned, the use of backsubstitution on (16) accumulates the variance of the estimates as the order p increases. This problem may be solved by overestimating the order P when solving $h(n)$, and then by constructing (16) as an overdetermined system. The objective of this example is to compare the use of the Least Squares and the Total Least Squares in the solution of the overdetermination. In a general case, LS solution is preferred if the noisy observations are present in the r.h.s. of the equation to be solved. TLS is better if the errors are present both in the system matrix and in the r.h.s. vector. In this case nothing can be said a priori since r.h.s. vector contains noisy observations and the system matrix consist of actual data (zeros in the upper part) and noisy observations ($h(n)$ samples). First, non-overdetermined AR w-slice algorithm is run with $M=N=2p$ and $P=p$, solving (16) by backsubstitution. Then we overdetermine the
algorithm setting $M=N=2p$ and $P=2p$, in such a way that the same set of non-redundant lags are used. Results in table II for an AR (4) process, show that overdetermination improves the results significantly and that insignificant differences are found between TLS and LS solutions. In the sequel, simulations involving the AR w-slice algorithm will be done taking $M=N=2p$ and $P=2p$ and LS solving of equation (16).

<table>
<thead>
<tr>
<th>Table II. Ex. 1, 2000 samples, 100 Monte Carlo runs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pars.</td>
</tr>
<tr>
<td>a(1)</td>
</tr>
<tr>
<td>a(2)</td>
</tr>
<tr>
<td>a(3)</td>
</tr>
<tr>
<td>a(4)</td>
</tr>
</tbody>
</table>

Example 2. In general the performance of all algorithms depends on the bandwidth of the process being tested. This is so because all algorithms use a finite number of slices, whereas narrow band AR processes cumulants spread over a large set. The performance is also influenced by the central frequency of the process spectrum. So, rather than testing specific well-chosen values for the AR parameters we have preferred to carry out a more general simulation. Bandpass processes of the kind $H(\theta)=(1-a(1)\theta^2+2a(2)\theta^2)\Gamma$ are used. The damping factor $r$ takes the value $r=0.85$. Twenty different values of $\theta$, between 0 and $2\pi$ are used and for each $\theta$, 100 Monte Carlo realizations have been run with record lengths of 1000 samples. No noise is present in this case. Variances are shown in figures 1 and 2 along with the deviations of the mean values from the expected ones. As can be seen, the AR w-slice and Yule-Walker equations present a similar behavior in variance. In mean, the AR w-slice present a better performance. We do not present further simulations for the lack of space, but more can be found in [10].

![Figure 1. Standard deviation of a(1) versus θ (central frequency of the band) for both methods.](image1)

VI. Conclusions

We have studied how to combine cumulant slices in the estimation of an AR system from its output samples. It has been shown that, the proposed AR w-slice algorithm exhibits a behavior comparable to the Yule-Walker equations, but a much smaller computational burden. It represents a good trade-off between cost and statistical behavior, with the additional feature of being capable to deal with statistics of different orders within the same framework. Further work points to the analytical computation of the statistical behavior of the method in comparison with other AR approaches as well as its optimization using the approach in [6].

References

TRANSFER FUNCTION OF LINEAR CONTINUOUS TIME-VARYING SYSTEMS VIA NON-COMMUTATIVE ALGEBRA

M. GUGLIELMI
Laboratoire d'Automatique de NANTES U.R.A. C.N.R.S. 823
Ecole Centrale de NANTES
1 rue de La Noé 44072, NANTES cedex 03, FRANCE
Tel: (33) 40 37 16 48
Fax: (33) 40 74 74 06

Keyword: linear continuous system, non-stationary, transfer function, non-commutative polynomials

Abstract: In this paper, we pay attention to continuous systems described by linear differential equations with time-varying coefficients. Some properties of such systems can be obtained from the study of non-commutative polynomials of the formal variable $\lambda \frac{d}{dt}$. In our approach, it is shown that it is possible to introduce the transfer function by the means of "skew" rational fraction. But, in this case, each rational fraction has two factorizations: the left one and the right one whose factors are in relation. From that, we can enlarge the transfer function concept for non-stationary systems. Then, we precise the association rules for serial or parallel systems from the computation rules over rational "skew" fractions.

1- INTRODUCTION:

During the past decade, considerable research has been done about linear stationary systems via the transfer function approach. By means of this representation, it may be solved some control problems like pole placement, reference model... The basic tool used is an algebraic set of operations over commutative polynomials of the formal LAPLACE variable in the continuous case, or of the $z$-variable for discrete systems.

Generalizations along this line to linear time-varying systems is difficult. One way to enlarge the transfer function concept is done through formal power series. For discrete systems, Kamen and al [1] have proposed a solution via an extension of formal power series with coefficients in a ring of time functions. In the continuous case, Emre and al [2] have showed that transfer matrices can also be defined as formal power series and they proposed a general solution to time-varying problems like observers... These approaches lead to important results like, for example, those obtained in realization theory. Unfortunately, this description is not always very easy to use (for example: interconnection of systems, relation with the original differential equation...). Here, we propose another approach for continuous single input-single output (SISO) systems described by $A(p) y = B(p) u$, in which $A(p)$ and $B(p)$ are non-commutative polynomials with coefficients in a subset of $C^{\infty}$. Then, a necessary and sufficient condition to define the rational non-commutative fraction field is that the "skew" polynomial ring must verify the two ORS's conditions [3] (here "skew" means only that the variable does not commute with the coefficients). The problem is that each rational fraction has two different factorizations, the left one described by $X^{-1}$ and the right one by $Y^{-1}X$. In this way, we develop a theory which resembles the time-invariant case. We define the transfer function of a SISO time-varying system by the means of the isomorphism with the rational non-commutative field. The major consequence is that the transfer function has two expressions: the right one and the left one which are in relations.

This paper is organised as follows: after the preliminaries in the second section, we present, in the third one, the elementary operations as the computation of the Greatest Common Left Divisor (G.C.L.D.) or the Least Common Right Multiple (L.C.R.M.) over non-
commutative polynomials. These two polynomials are the basic elements since that,
in general, they are sufficient to introduce all the other operations like Greatest Common
Right Divisor and Least Common Left Multiple. In the final part of this section, we
consider the field of the "skew" rational fractions. Then, in the fourth section, we
enlarge the definition of the transfer function of time-varying systems. From that, we give,
in the next section, some results about classical operations over systems (like serial
and parallel combinations) for which the results are obtained from arithmetic
operations over rational skew fractions. As in the stationary case, all these computations
need the G.C.L.D. and the L.C.R.M.

2- DEFINITIONS:

In the whole paper, we define (E) as the S.I.S.O system described by an ordinary
differential equation (I) with time-varying coefficients:

\[
\begin{align*}
    y^{(n)}(t) + a_1(t)y^{(n-1)}(t) + \ldots + a_n(t)y^{(0)}(t) &= b_0(t)u^{(n)}(t) + b_1(t)u^{(n-1)}(t) + \ldots + b_p(t)u^{(0)}(t) \\
    y^{(1)}(t) &= \frac{dy(t)}{dt}, \quad u^{(1)}(t) = \frac{du(t)}{dt}
\end{align*}
\]

where:

\[
\begin{align*}
    y^{(1)}(t) &= \frac{dy(t)}{dt}, \quad u^{(1)}(t) = \frac{du(t)}{dt}
\end{align*}
\]

The functions \(a_i(t)\) and \(b_j(t)\) belong to an ordinary differential field \(K\).

Recall that \(K\) is an ordinary differential field if [4]

1) it is a field
2) there exists a unique derivation \(\lambda\) denoted \(\lambda[a] = \dot{a}\) which verifies the
usual properties:

- \(\lambda[a + b] = \lambda[a] + \lambda[b] = \dot{a} + \dot{b}\)
- \(\lambda[ab] = \dot{a}b + a\dot{b}\)

We suppose, here, that all the coefficients \(a_i\) and \(b_j\) are in a subset of \(C^\infty\)
functions of time without zero divisors.

We define \(\lambda\) as the operator \(\frac{d}{dt}\). Then, all the properties of \((I)\) may be obtained from
those of the polynomial equation:

\[
\begin{align*}
    \lambda^n + a_1(t)\lambda^{n-1} + \ldots + a_n(t) y(t) &= \quad \text{(I)} \\
    \lambda^n + b_1(t)\lambda^{n-1} + \ldots + b_p(t) u(t) &= \quad \text{(II)}
\end{align*}
\]

Or in the formal form:

\[
\lambda^n[y(t)] = [P(\lambda)] [u(t)]
\]

3- NON-COMMUTATIVE POLYNOMIALS:

Let us define \(\Pi(\lambda)\) as the general polynomial set:

\[
\Pi(\lambda) = [P(\lambda)] = a_0(t)\lambda^n + a_1(t)\lambda^{n-1} + \ldots + a_n(t),
\forall a_i(t)\in K\text{ for } i \in [0,n]
\]

where \(K\) is a differential field.

When \(a_0(t) = 1\), \(P(\lambda)\) is monic.

For the multiplication, we define the non-commutative operation:

\[
\lambda a = a\lambda + \dot{a} \quad \forall a \in K
\]

with \(\dot{a} = \frac{da}{dt}\)

It is easy to show that \(P(\lambda)\) with
the two internal laws (addition and multiplication) is a non-commutative ring [5]

3-1: Division and Greatest Common
Right Divisor (G.C.R.D.):

It is possible to define the
right division:

Definition:

For every pair \((P_1(\lambda), P_2(\lambda))\in \Pi(\lambda) \ast \Pi(\lambda)\)
with \(d^\circ P_1(\lambda) > d^\circ P_2(\lambda)\), it exists a unique
polynomial pair \((Q(\lambda), R(\lambda))\) which satisfies:

\[
\begin{align*}
    P_1(\lambda) &= Q(\lambda) P_2(\lambda) + R(\lambda)
\end{align*}
\]

with \(d^\circ R(\lambda) < d^\circ P_2(\lambda)\)

and \(d^\circ Q(\lambda) = d^\circ P_1(\lambda) - d^\circ P_2(\lambda)\)

Thus we have the EUCLID's algorithm:

\[
\begin{align*}
    P_1(\lambda) &= Q_1(\lambda) P_2(\lambda) + P_3(\lambda) \\
    P_2(\lambda) &= Q_2(\lambda) P_3(\lambda) + P_4(\lambda) \\
    \ldots &= \ldots \\
    P_{n-1}(\lambda) &= Q_{n-1}(\lambda) P_n(\lambda)
\end{align*}
\]

The G.C.R.D. of \((P_1(\lambda), P_2(\lambda))\) is the well
defined monic polynomial: \(P_n(\lambda)\)

3-2: Least Common Right Multiple
(L.C.R.M.):

We can define the L.C.R.M. of \([P_1(\lambda), P_2(\lambda)]\) as the lowest degree
monic polynomial \(M(\lambda)\) which admits both
\(P_1(\lambda)\) and \(P_2(\lambda)\) as right divisors:
\[ M(\lambda) = Q_1(\lambda) P_1(\lambda) = Q_2(\lambda) P_2(\lambda) \]

In the general case, ORE [5] showed that the existence of EULID's algorithm implies the existence of L.C.R.M.

III-3: Explicit formula of the L.C.R.M.:

From all the quotients obtained in the EULID's algorithm, the L.C.R.M. is calculated by the following formula:

\[ M(\lambda) = [P_1(\lambda), P_2(\lambda)] = a P_{n-1}(\lambda) P_n(\lambda)^{-1} P_{n-2}(\lambda) P_n(\lambda)^{-1} ... P_2(\lambda) P_3(\lambda)^{-1} P_1(\lambda) \]

The constant (i.e. not dependent of \( \lambda \)) \( a \) is chosen so that \( M(\lambda) \) is monic.

Remarks:

1 - The product of two polynomials \( P_1(\lambda) P_2(\lambda) \) is not usually right divisible by \( P_1(\lambda) \).

2 - In the general case, left-hand division differs from right-hand division in that it cannot always be performed. Here, the definition of the differential operator allows to assert the existence of the right-hand division of two polynomials as the Least Common Left Polynomial [5].

3-4: Rational fraction:

With commutative polynomials, it always be considered the rational fraction field. This is not possible when the polynomials are "skew" (that means only that, for these polynomials, the coefficients do not commute with the variable). Nevertheless, a ring has a rational left field and a rational right one if the two ORE's conditions are satisfied [3]:

\[ \forall P_1(\lambda), P_2(\lambda) \in \Pi(\lambda) \otimes \Pi(\lambda), \]

\[ \exists \hat{P}_1(\lambda) \in \Pi(\lambda) \text{ and } \hat{P}_2(\lambda) \in \Pi(\lambda) \]

such that \( \hat{P}_2(\lambda) P_1(\lambda) = \hat{P}_1(\lambda) P_2(\lambda) \)

(left condition)

\[ \forall P_1(\lambda), P_2(\lambda) \in \Pi(\lambda) \otimes \Pi(\lambda), \]

\[ \exists \check{P}_1(\lambda) \in \Pi(\lambda) \text{ and } \check{P}_2(\lambda) \in \Pi(\lambda) \]

such that \( \check{P}_1(\lambda) P_2(\lambda) = P_1(\lambda) \check{P}_2(\lambda) \)

In this case, we may consider the field:

\[ \text{Frac} = \{ P(\lambda), Q(\lambda) \in \Pi(\lambda) \otimes \Pi^*(\lambda) \}
\]

\[ = \{ Q_1(\lambda)^{-1} P_1(\lambda), (P_1(\lambda), Q_1(\lambda)) \in \Pi(\lambda) \otimes \Pi^*(\lambda) \}
\]

with \( \Pi^*(\lambda) = \Pi(\lambda) - \{0\} \)

It is obvious to show that, with the two internal laws (addition and multiplication), it is a "skew" field. For details see Dixmier [6].

Notice that each rational fraction has two representations: the left one and the right one. Notice also that the two ORE's conditions are not necessary; see Krob [3].

Remark: for a polynomial \( P(\lambda) \in \text{Frac} \), we may define the unique inverse \( P(\lambda)^{-1} \) by:

\[ P(\lambda)^{-1} P(\lambda) = P(\lambda) P(\lambda)^{-1} = 1 \]

It is obvious that:

\[ [P(\lambda) P_1(\lambda)]^{-1} = P_1(\lambda)^{-1} P(\lambda)^{-1} \]

4- TRANSFER FUNCTION:

In this section, we present the definition of the transfer function based on the rational fraction field. One way to enlarge the transfer function concept for time-varying systems is to generalize the formal power series. In our approach, the transfer function can be defined from the rational fraction field. One problem is that each transfer function has two representations which are in relation.

4-1: Definition:

If \( (\Sigma) \) is a system defined by a differential equation \( \dot{y}(t) = P(\lambda) y(t) + P(\lambda) u(t) \), we can describe \( (\Sigma) \) by the formal relation:

\[ [G(\lambda)](y(t)) = [P(\lambda)](u(t)) \]

As in the stationary case, the left transfer function (L.T.F.) of the system \( (\Sigma) \) may be defined by the rational fraction:

\[ H(\lambda) = G(\lambda)^{-1} P(\lambda) \]
But as $H(\lambda)$ has another representation in the rational field $\mathbb{Frac}$, we say that the right transfer function (R.T.F) of $(\Sigma)$ is the other representation of $H(\lambda)$.

$$H(\lambda) = P_1(\lambda)Q_1^{-1}(\lambda)$$

**Remark:** The left representation is directly issued from the differential equations.

**R.T.F.:**

$$P(\lambda)Q_1(\lambda) = Q(\lambda)P_1(\lambda)$$

That means that, from the L.T.F. (R.T.F.) the other one is easily computed with the remaining factors of the L.C.L.M. (L.C.R.M.).

**Example:**

Let \( \dot{y} + \frac{1}{t}y = u \)

$$Q(\lambda) = 1 + \frac{1}{t} \text{ and } P(\lambda) = 1$$

The L.C.L.M of $Q(\lambda)$ and $P(\lambda)$ is $\lambda^2 = (\lambda + \frac{1}{t})(\lambda - \frac{1}{t})$

so $Q_1(\lambda) = 1$ and $P_1(\lambda) = (\lambda - \frac{1}{t})$

**4-3: Minimal transfer function:**

As in the stationary case, we can define the minimal transfer function.

First, we introduce the equivalence relation of two transfer function.

**Theorem 1:** Two transfer functions $H_1(\lambda)$ and $H_2(\lambda)$ are equivalent if and only if in their L.T.F. $Q_1^{-1}(\lambda)P_1(\lambda)$ and $Q_2^{-1}(\lambda)P_2(\lambda)$, it exists a polynomial $D(\lambda)$ such as:

$$P_1(\lambda) = D(\lambda)P_2(\lambda) \text{ and } Q_1(\lambda) = D(\lambda)Q_2(\lambda)$$

(if $d^n(P_1(\lambda)) > d^n(P_2(\lambda))$ and $d^n(Q_1(\lambda)) > d^n(Q_2(\lambda))$)

For demonstration see [7]

**Example:**

Let \( \ddot{y} = u + \frac{1}{t}u \)

The L.T.F is $H(\lambda) = Q^{-1}(\lambda)P(\lambda) = \lambda^2 = (\lambda + \frac{1}{t})(\lambda - \frac{1}{t})$.

That corresponds to the differential equation:

$$\dot{y} + \frac{1}{t}y = u$$

(The first differential equation is the derivative of the last one)

**Theorem 2:** Two transfer functions $H_1(\lambda)$ and $H_2(\lambda)$ are equivalent if and only if in their R.T.F. $P_1(\lambda)Q_1^{-1}(\lambda)$ and $P_2(\lambda)Q_2^{-1}(\lambda)$, it exists a polynomial $D(\lambda)$ such as:

$$P_1(\lambda) = P_2(\lambda)D(\lambda) \text{ and } Q_1(\lambda) = Q_2(\lambda)D(\lambda)$$

(if $d^n(P_1(\lambda)) > d^n(P_2(\lambda))$ and $d^n(Q_1(\lambda)) > d^n(Q_2(\lambda))$)

From that, we can define the minimal transfer function:

**Definition:** The transfer function of a linear time-varying system is minimal if, for the L.T.F. (R.T.F.) representation, the numerator and the denominator are left-co-prime (right co-prime).

5- ALGEBRAIC RULES OF COMBINATION OF NON-STATIONARY SYSTEMS:

From the preliminary results, we consider, now, the elementary operations on SISO non-stationary systems. For the two associations (serial and parallel), it is possible to compute the transfer function of the compound system [7].

**5-1: Transfer function for series combination:**

For two systems $\Sigma_1$ and $\Sigma_2$ described by their L.T.F $Q_1(\lambda)^{-1}P_1(\lambda)$ and $Q_2(\lambda)^{-1}P_2(\lambda)$, it is possible to find the left transfer function of $\Sigma$ of the series combination of $\Sigma_1$ and $\Sigma_2$:

We have: $\frac{y_1}{u} = Q_1(\lambda)^{-1}P_1(\lambda)$ and $\frac{y_2}{u} = Q_2(\lambda)^{-1}P_2(\lambda)$

Thus $\frac{y_2}{y_1} = \frac{Q_2(\lambda)^{-1}P_2(\lambda)}{Q_1(\lambda)^{-1}P_1(\lambda)}$

If $M(\lambda) = \widehat{P_2(\lambda)}P_2(\lambda) = \widehat{Q_1(\lambda)}Q_1(\lambda)$

and $\frac{y_2}{y_1} = Q_2(\lambda)^{-1}P_2(\lambda)Q_1(\lambda)^{-1}P_1(\lambda)$

$$= \left[ \widehat{P_2(\lambda)} \cdot \widehat{Q_2(\lambda)} \right]^{-1} \widehat{Q_1(\lambda)}P_1(\lambda)$$
Thus:

\[
\hat{y}_2 = (p_2(\lambda)q_2(\lambda))^{-1} \hat{q}_1(\lambda)p_1(\lambda)
\]

Example:

If \((\Sigma_1) \Rightarrow (\lambda + \frac{1}{t})^{-1}\lambda\) and \((\Sigma_2) \Rightarrow (\lambda - \frac{1}{t})^{-1}\lambda\)

we have: \(Q_1(\lambda) = \lambda + \frac{1}{t}\), \(P_1(\lambda) = \lambda\)
\(Q_2(\lambda) = \lambda - \frac{1}{t}\), \(P_2(\lambda) = \lambda\)

So the L.C.R.M. of \((1, \lambda + \frac{1}{t})\) is \(\lambda + \frac{1}{t}\)
\(\hat{q}_1(\lambda) = 1\), \(\hat{p}_2(\lambda) = \lambda + \frac{1}{t}\)

\((\Sigma) \Rightarrow [ (\lambda + \frac{1}{t}) (\lambda - \frac{1}{t})]^{-1}\lambda = \lambda^{-2}\lambda\)

and the L.T.F. of the \((\Sigma)\) is equal to \(\lambda^{-1}\)

5-2: Transfer function for parallel combination:

The problem is easier to solve. For two systems \(\Sigma_1\) and \(\Sigma_2\) described by their L.T.F. \(q_1(\lambda)^{-1}p_1(\lambda)\) and \(q_2(\lambda)^{-1}p_2(\lambda)\), it is possible to find the left transfer function of \(\Sigma\) of the parallel-combination of \(\Sigma_1\) and \(\Sigma_2\):

\[
\frac{y_1}{u} = Q_1(\lambda)^{-1}p_1(\lambda) \quad \text{and} \quad \frac{y_2}{u} = Q_2(\lambda)^{-1}p_2(\lambda)
\]

If \(M(\lambda) = \hat{q}_1(\lambda)Q_1(\lambda) = \hat{q}_2(\lambda)Q_2(\lambda)\)
Thus:

\[
\frac{y}{u} = M(\lambda)[\hat{q}_1(\lambda)p_1(\lambda) + \hat{p}_2(\lambda)P_2(\lambda)]
\]

Example:

If \((\Sigma_1) = (\lambda + \frac{1}{t})^{-1}\lambda\) and \((\Sigma_2) = (\lambda - \frac{1}{t})^{-1}\lambda\)

\[
M(\lambda) = \text{L.C.R.M.} \{Q_1(\lambda), Q_2(\lambda)\} = \lambda^2 + \frac{1}{t} \lambda - \frac{1}{t^2}
\]

Thus:

and \((\Sigma)\) has the L.T.F:

\[
\frac{y}{u} = [\lambda^2 + \frac{1}{t} \lambda - \frac{1}{t^2}][2(\lambda + \frac{1}{t})]
\]

6- CONCLUSION:

By using theory of non-commutative polynomials, we have been able to define the transfer function of a linear system with time varying coefficients in the continuous case. The major difference with the stationary case is that it appears that the transfer function has two factorizations. The relation between the factors of each one is given. The left representation is closed to the differential equation. For the minimal transfer function, we achieve a necessary and sufficient condition. Besides, we have presented the algebraic rules for compound systems.

We think that these results can be obtained without difficulty to discrete non-stationary systems and can be enlarged to multivariable systems.

ACKNOWLEDGEMENTS:

I should like to thank Jan JEZEC for his critical and incisive comments which have done much to help improve the quality of this paper.

BIBLIOGRAPHY:


Note interne Institut BLAISE PASCAL n° 90-2 Janvier 1990


A TIME-VARYING PRONY MODEL

F. MOLINARO, F. CASTANIE (Member EURASIP)

GAPSE/ENSEEIHT, 2 rue Camichel, 31071 Toulouse, France

Abstract. The communication proposes a new Prony model with time-varying coefficients for modelling of nonstationary signals. This method leads to an extension of several techniques of stationary spectral estimation to the nonstationary case. The determination of the time-varying parameters requires five steps: estimation of the time-varying AR parameters, estimation of the right poles of the linear time-varying system, modelling of these poles, computation of new poles, and least-square estimation of the amplitudes.

1. INTRODUCTION

Parametric spectral estimation methods are well known in signal processing. They are, however limited, due to the fact that they necessarily assume a stationary signal. Many actual signals are nonstationary processes: speech, seismic or engine vibration... For these kinds of signals, in the last 20 years, some authors [1], [2], [3] have developed time-dependent models. Such models are an extension of the time-invariant ARMA model to the nonstationary case. ARMA time-varying coefficient are approximated by a weighted combination of basis functions [4].

This paper proposes an extension of the stationary Prony model [5],[6], to the nonstationary case. This time-varying Prony model is adapted for time-varying multicomponent signals.

In what follows the time-varying Prony model is introduced and defined. In Part 3 the different steps for parameter estimation are detailed. The model is applied to a simulation case in Part 4.

2. THE MODEL

The order \( p \) time-varying Prony model under consideration is defined by:

\[
x(n) = \sum_{k=1}^{p} A_k \omega_k n^k + \varepsilon(n) \quad n = 0, 1, \ldots, N - 1
\]

with the particular following for \( Z_k(n) \):

\[
Z_k(n) = e^{\alpha_k(n) - j\omega_k(n)}
\]

with \( \alpha_k(n) = \sum_{j=0}^{N_k} \alpha_{kj} r^j \) and \( \omega_k(n) = \sum_{j=0}^{N_k} \omega_{kj} r^j \)

\[
A_k = A_k e^{i\theta_k} \text{ is constant}
\]

\( A_k \): constant amplitude.

\( \theta_k \): constant phase.

\( \alpha_k(n) \): polynomial time-varying damping factor.

\( f_k(n) \): polynomial time-varying frequency.

It should be noted that complex amplitudes are not time function in our model, the nonstationary part being limited to the poles.

2.1 Definition of time dependent right poles

Kamen [7] has defined the left shift operator \( \hat{s} \) by:

\[
z^i z^{-i} = z^i
\]

\[
z^i x(n) = x(n + i)
\]

\( \alpha(n), z^i x(n) = \alpha(n), x(n + i) \)

\[ P_k(n) \text{ are time-dependent poles if they have satisfied the following equation:} \]

\[
\left[ 1 + \sum_{k=1}^{p} \alpha_k(n) z^{-k} \right] \hat{s}(1 - z^{-1} P_k(n)) = 0
\]

(1)

where \( \hat{s} \) denotes skew multiplication [7]. Let \( P_p(n) \) be a right pole of the system, there exists \( \beta_{p-1,k}(n) \) such that:

\[
\hat{s}(1 - z^{-1} P_p(n)) = \left( 1 + \sum_{k=1}^{p} \beta_{p-1,k}(n) z^{-k} \right) \left( 1 - z^{-1} P_p(n) \right)
\]

(2)

(1) and (2) gives:

\[
\alpha_1(n) = \beta_{p-1,1}(n) - P_p(n)
\]

\[
\alpha_k(n) = \beta_{p-1,k}(n) - \beta_{p-1,k-1} P_p(n - k + 1) \quad k = 2, \ldots, p - 1
\]

\[
\alpha_p(n) = -\beta_{p-1,p-1}(n) P_p(n - p + 1)
\]

Expressing this system in the matrix form:

\[
\begin{bmatrix}
1 & 0 & \ldots & 0 & \beta_{p-1,1}(n) \\
-\beta_{p-1,1}(n) & 1 & \ldots & 0 & \beta_{p-1,2}(n) \\
0 & -\beta_{p-1,2}(n) & \ldots & 0 & \beta_{p-1,3}(n) \\
\vdots & \vdots & \ddots & \ddots & \ddots \\
0 & 0 & \ldots & -\beta_{p-1,p-1}(n) & \beta_{p-1,p}(n) \\
0 & 0 & \ldots & 0 & \beta_{p-1,p+1}(n)
\end{bmatrix}
\begin{bmatrix}
\alpha_1(n) \\
\alpha_2(n) \\
\vdots \\
\alpha_p(n)
\end{bmatrix} = \begin{bmatrix}
\beta_{p-1,1}(n) \\
\beta_{p-1,2}(n) \\
\vdots \\
\beta_{p-1,p+1}(n)
\end{bmatrix}
\]

(3)

\( P_p(n) \) is recursively computed given its initial \( p \) values and the time varying coefficients \( \alpha_k(n) \).
With this formulation, p right poles $P_1(n), P_2(n), \ldots, P_p(n)$ can be determined. An advantage of this time-dependent pole factorization is that the value of each pole at any given instant is obtained independently of the other poles.

2.2 Justification of the model

The two following theorems justify this model.

**Theorem 1:** [7] If the Vandermonde determinant $V(0)$ is nonzero, then for any initial values $x(0), x(1), \ldots, x(p-1)$ there exist constants $B_1, B_2, \ldots, B_p$ such that the solution of the time-varying equation:

$$x(n) = - \sum_{k=1}^{p} a_k(n)b_k x(n-k)$$

can be written in the form:

$$x(n) = \sum_{k=1}^{p} B_k \phi_k(n)$$

$$\phi_k(n) = P_k(n-1)P_k(n-2)\ldots P_k(0)$$

$P_1(n), P_2(n), \ldots, P_p(n)$ are p right poles

$V(0)$ is defined by:

$$V(0) = \begin{bmatrix}
   1 & 1 & 1 \\
   P_1(0) & P_2(0) & P_3(0) \\
   P_1(1)P_1(0) & P_2(1)P_2(0) & P_3(1)P_3(0) \\
   \vdots & \vdots & \vdots \\
   P_1(p-2)P_1(0) & P_2(p-2)P_2(0) & P_3(p-2)P_3(0)
\end{bmatrix}$$

**Theorem 2:** If the right poles are modeled in the particular form:

$$P_k(n) = e^{\alpha_k(n)+i\omega_k(n)}$$

with $\alpha_k(n) = \sum_{j=0}^{s_k} \alpha_{kj}n^j$ and $\omega_k(n) = \sum_{j=0}^{s_k} \omega_{kj}n^j$

Then it comes:

$$x(n) = \sum_{k=1}^{p} B_k Z_k(n)$$

**Proof:** Namely by denoting $S_j = \sum_{k=0}^{s_j} jk^j$, we obtain:

$$S_j = n \left( \sum_{k=0}^{s_j} k^j \right)$$

$$S_j = \frac{1}{j+1} \left[ n^{j+1} - n - C_j^1S_j - C_j^2S_{j-2} - \ldots - C_j^jS_1 \right]$$

with $S_{s_j} = 1$, every $s_{ij}$ can be computed.

Then:

$$\phi_k(n) = \sum_{j=0}^{s_k} \alpha_{kj}n^j, \quad \omega_k(n) = \sum_{j=0}^{s_k} \omega_{kj}n^j$$

With:

$$\alpha_{kj} = \sum_{i=0}^{N_A} \alpha_{ik}S_{ij}$$

$$\omega_{kj} = \sum_{i=0}^{N_A} \omega_{ik}S_{ij}$$

$\alpha_{kj}$ is a weighted combination of $\alpha_{ik}$ ($k = 0, \ldots, N_A$)

$\omega_{kj}$ is a weighted combination of $\omega_{ik}$ ($k = 0, \ldots, N_A$).

3. COMPUTATION OF THE PARAMETERS

According to the above properties, we can derive the following algorithm in five steps:

- Estimation of the time-varying AR parameters $\alpha_k(n)$
- Estimation of the right poles $P_k(n)$
- Modelling of the right poles
- Calculation of $Z_k(n)$
- Estimation of $B_k$

3.1 Estimation of the time-varying AR parameters

The time-dependent AR modeling [4] of $x(n)$ is represented by the following equation:

$$x(n) = - \sum_{k=1}^{p} \alpha_k(n)x(n-k) + e(n)$$

Parameters $\alpha_k(n)$ are a function of $n$. They are approximated by a weighted combination of a number $\varphi$ of functions $f_j(n)$.

$$\alpha_k(n) = \sum_{j=0}^{\varphi} \alpha_{kj}f_j(n)$$

One may use different sets of functions: Legendre, prolate, Fourier [4].

3.2 Estimation of the time varying right poles

Resolution of the matrix system of Eq.(3) gives:
\[ P_p(n) = -\alpha_1(n) - \sum_{j=2}^{n-2} \frac{\alpha_{p-j}(n)}{\prod_{i=1}^{j-1} P_p(n-i)} \]

Computation of the right poles is initialized with a stationary Prony modeling on a fixed window at the beginning of the signal [6].

### 3.3 Modellisation of the poles

The pole model is given by:

\[ P_k(n) = e^{-\alpha_k(n)^2 - j\omega_k(n)^2} \]

with

\[ \alpha_k(n) = \sum_{j=0}^{N_k} \alpha_{kj} n^j \]
\[ \omega_k(n) = \sum_{j=0}^{N_w} \omega_{kj} n^j \]

For each pole we estimate the module with a least-square polynomial estimation, and the phase with the specific estimation described in [8].

### 3.4 Computation of \( Z_k(n) \)

With (see proof of theorem 2):

\[ \alpha_{kj} = \sum_{i=0}^{N_k} \alpha_{kj} S_{ij} \]
\[ \omega_{kj} = \sum_{i=0}^{N_w} \omega_{kj} S_{ij} \]

The \( Z_k(n) \) can be determined easily.

\[ Z_k^2(n) = e^{\sum_{j=0}^{N_k} \alpha_{kj} n^j} \cdot e^{\sum_{j=0}^{N_w} \omega_{kj} n^j} \]

### 3.5 Estimation of \( B_k \) amplitudes

\( B_k \) amplitudes are estimated with the help of a conventional Least-Square procedure

\[ B = [B_1 ... B_p]^T = (V^HV)^{-1}V^HX \]

\( V \) is a Van Der Monde matrix, whose term is given by:

\[ V = [v_{mn}] = [Z_{mn}(n)] \quad n = 0, ..., N - 1 \quad m = 1, ..., p \]

\( V^H \) denotes its conjugated transposed matrix.

\( X = [x(0) ... x(N - 1)]^T \) is the signal vector.

### 4. APPLICATION OF A SIMULATED SIGNAL

To validate this model, a simulation signal is chosen. It is composed of two chirp signals and can be expressed by:

\[ x(n) = s_1(n) + s_2(n) + w(n) \]
\[ = \cos(\omega_{11}n + \omega_{12}n^2) + \cos(\omega_{21}n + \omega_{22}n^2) + w(n) \]

\( w(n) \) being a white noise. Different SNR levels are also chosen. In order to illustrate the properties of our method, various cases of time-frequency trajectories are given: three signals (see Fig. 1, 2, 3) with a change of linear modulation of \( s_2(n) \), i.e. parameter \( \omega_{22} \).

Fig. 1 shows that signal 1 is composed of two chirps with separate frequencies.

![Fig. 1 - Signal 1 frequency modulations](image)

\( \omega_{11} = 0.255, \omega_{12} = 0.0005, \omega_{21} = 0.1, \omega_{22} = 0.0003 \)

In signal 2 frequency linear modulation of \( s_2(n) \) is such that the trajectories become very close to one another (see Fig. 2).

![Fig. 2 - Signal 2 frequency modulations](image)

\( \omega_{11} = 0.255, \omega_{12} = 0.0005, \omega_{21} = 0.1, \omega_{22} = 0.001 \)

In case 3, the two trajectories intersect (see Fig. 3).

![Fig. 3 - Signal 3 frequency modulations](image)

\( \omega_{11} = 0.255, \omega_{12} = 0.0005, \omega_{21} = 0.1, \omega_{22} = 0.0013 \)
Parameter estimation error for each signal is given in Tab.1,2,3 respectively. Error is defined by:

\[ E = \frac{\omega_{ul} - \omega_{uj}}{\omega_{uj}} \]

<table>
<thead>
<tr>
<th>SNR in db</th>
<th>Error in % ( \omega_{11} )</th>
<th>Error in % ( \omega_{12} )</th>
<th>Error in % ( \omega_{21} )</th>
<th>Error in % ( \omega_{22} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>6.4 ( 10^{-1} )</td>
<td>9.1 ( 10^{-1} )</td>
<td>4.7 ( 10^{-1} )</td>
<td>1.6</td>
</tr>
<tr>
<td>20</td>
<td>8.4 ( 10^{-1} )</td>
<td>3.5</td>
<td>3.6 ( 10^{-1} )</td>
<td>2.8</td>
</tr>
<tr>
<td>10</td>
<td>7.6</td>
<td>16</td>
<td>49</td>
<td>63</td>
</tr>
</tbody>
</table>

**Tab. 1 - Error signal 1 = \( \frac{\omega_{ul} - \omega_{uj}}{\omega_{uj}} \)**

<table>
<thead>
<tr>
<th>SNR in db</th>
<th>Error in % ( \omega_{11} )</th>
<th>Error in % ( \omega_{12} )</th>
<th>Error in % ( \omega_{21} )</th>
<th>Error in % ( \omega_{22} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>7.7 ( 10^{-1} )</td>
<td>8.6 ( 10^{-1} )</td>
<td>3.4</td>
<td>4.1 ( 10^{-2} )</td>
</tr>
<tr>
<td>20</td>
<td>3.5</td>
<td>23</td>
<td>9.4</td>
<td>8.1</td>
</tr>
<tr>
<td>10</td>
<td>23</td>
<td>47</td>
<td>14.7</td>
<td>6.1</td>
</tr>
</tbody>
</table>

**Tab. 2 - Error signal 2 = \( \frac{\omega_{ul} - \omega_{uj}}{\omega_{uj}} \)**

<table>
<thead>
<tr>
<th>SNR in db</th>
<th>Error in % ( \omega_{11} )</th>
<th>Error in % ( \omega_{12} )</th>
<th>Error in % ( \omega_{21} )</th>
<th>Error in % ( \omega_{22} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>8.7</td>
<td>43</td>
<td>15.9</td>
<td>4.9</td>
</tr>
<tr>
<td>20</td>
<td>8.4 ( 10^{-2} )</td>
<td>4.7</td>
<td>6.4</td>
<td>10.6</td>
</tr>
<tr>
<td>10</td>
<td>2</td>
<td>7.1</td>
<td>29.1</td>
<td>12.9</td>
</tr>
</tbody>
</table>

**Tab. 3 - Error signal 3 = \( \frac{\omega_{ul} - \omega_{uj}}{\omega_{uj}} \)**

Estimation of frequency parameters is better for separate frequencies (see Tab.1,2,3). The intersection of the two chirps increases estimation error (see Tab.3). For a high SNR, i.e more than 10 db, this model gives good frequency estimation.

5. CONCLUSION

The present paper has introduced a Prony model with time-varying poles. This model is well suited to frequency modulated nonstationary multicomponent signals.

We have shown that it is possible to introduce a new model, based upon a linear combination of time-varying exponentials with constant amplitudes. With the restriction of the time-varying poles to a well specified class (namely damping factors and instantaneous frequencies being time polynomials), we have proposed a method and the corresponding algorithm to estimate the complete new parameter vector.

Special attention has been paid to the retrieval of frequency modulation laws in simulated signals.

The method has proved capable of giving accurate measurements of time-frequency trajectories. The properties of the corresponding amplitude estimators will be dealt with in a forthcoming submission.

REFERENCES


AAK MODEL REDUCTION FOR TIME-VARYING SYSTEMS

A.-J. van der Veen and P. Dewilde

Department of Electrical Engineering, Delft University of Technology
2628 CD Delft, The Netherlands

The paper summarizes a procedure to obtain the optimal low degree "generalized Hankel norm approximant" of a bounded upper operator which is described by a high-order (time-varying) system. In the classical Hardy space theory this is known as the "model reduction problem" and it has resulted in a solution which goes back to the work of Adamyan, Arov and Krein on the Schur-Takagi problem. Our approach extends the frontiers of that theory to cover general, not only Toeplitz, upper operators as well. The procedure is computational and utilizes time-varying state space representations of the operators.

1. INTRODUCTION

The singular value decomposition (SVD) has been used by Adamyan, Arov and Krein (AAK)[1] to obtain approximations in the context of complex function theory, in relation with the approximation of infinite-size Hankel matrices. The AAK paper is written in an analytic setting of a model reduction problem but the same problem can be put into operator theory, in particular of upper triangular Toeplitz operators (or Toeplitz matrices). In the present paper, the aim is to extend the model reduction theory to the time-varying context, by considering (bounded) upper operators

\[ T = \begin{bmatrix}
  \cdots & \cdots & \cdots & \cdots \\
  0 & T_{01} & T_{02} & \cdots \\
  T_{11} & T_{12} & \cdots & \cdots \\
 & & & \ddots
\end{bmatrix} \]

which are no longer taken to be Toeplitz (the central (0,0) element is distinguished by a box). T can be regarded as the transfer operator of a causal time-varying system which acts on a row sequences, in which case the i-th row of T is the impulse response of T to an impulse applied at time i. The notion of a (physical) system is however not essential: the approximation theory applies to a general class of (bounded) upper operators which have a finite-dimensional computational model. The derivation of such models is the subject of a recently developed realization theory for such operators [2]. The approximation theory in this paper draws heavily onto [2], but in this short paper we omit most of those results and notation, in particular on the spaces to which each of the operators belong. A complete account is presented in the full paper [3], the outline of which runs more or less in parallel with [4]. For a given T, a time-varying state space realization (alias a computational model for T) has the form

\[ x_{k+1} = x_k A_k + u_k B_k \\
\]

\[ y_k = x_k C_k + u_k D_k \]

where T is called a locally finite operator if the individual \( \{ A_k, B_k, C_k, D_k \} \) all have finite (but possibly time-varying) dimensions. We can assemble the matrices \( \{ A_k \}, \{ B_k \} \) etc. as operators on spaces of sequences of appropriate dimensions, by defining \( A = \text{diag}(A_k), B = \text{diag}(B_k), C = \text{diag}(C_k) \) and \( D = \text{diag}(D_k) \). Together these operators define a realization T of T:

\[ x Z T = x A + u B \\
y = x C + u D \]

(2)

The diagonal operators act on sequences \( u = [\cdots, u_k, u_{k+1}, \cdots], x = [\cdots, x_k, x_{k+1}, \cdots] \), and the shift operator \( Z \) on these sequences is defined by \( x Z T = [\cdots, x_k, x_{k+1}, \cdots] \). The realization in (2) is equivalent to (1), but more convenient to handle in equations because the time-index has been suppressed. Shifted diagonal operators are \( A^{(f)} = Z T A Z = \text{diag}(A_{n+1}) \) and \( A^{(f)} = Z A Z = \text{diag}(A_{n+1}) \). An important aspect of these sequences is that the dimensions of their components can vary in time. Suppose that \( x_k \in B_k \), with \( B_k = C_{n_k} \) an Euclidean space of dimension \( N_k \), then we define \( B \) to be the space of sequences \( x \) with entries in \( B_k \), and hence \( x \in B \) and \( A : B \to B^{(f)} \). Even input- and output sequences can have varying dimensions. We will typically use \( A \) for input sequences and \( N \) for output sequences, and hence \( T : A \to N \). If \( I - A Z \) has a bounded inverse that is again upper (this requires the spectral radius of \( AZ \)

\[ \text{Research was supported in part by the commission of the EC under the ESPRIT BRA program 6632 (NANA2).} \]
to be less than one), then the realization $T$ is such that
\[ T = D + BZU - AZH^{-1}C. \]

We will assume throughout the paper that this is the case. Hankel operators are obtained as follows. Denote a certain time instant ($k$ say) as ‘current time’, apply all possible inputs in the ‘past’ with respect to this instant, and measure the corresponding outputs in ‘the future’, from the current time instant on. The operator mapping this input sequence to the restricted output sequence corresponds to an upper-right part of the matrix representation of $T$. Denote this part (or rather its mirrored one-sided infinite equivalent acting on the reversed input sequence) by $H_k$, then for example
\[ H_k = \begin{bmatrix} T_{12} & T_{13} & T_{14} & \cdots \\ T_{22} & T_{23} & T_{24} & \cdots \\ T_{32} & T_{33} & T_{34} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}. \]

In analogy with the time-invariant case, we call $H_k$ the (time-varying) Hankel operator at time $k$, and $H_T = \text{diag}(H_k)$ the time-varying Hankel operator. For the time-invariant case, all $H_k$ are the same and the entries of each of them are constant along anti-diagonals. Neither of this is true in the time-varying setting, but nonetheless the main features carry over (this regards the rank of $H_k$ and a certain shift-invariance property). The Hankel norm of an operator $T$ is $\|T\|_H = \|H_T\| = \sup\|H_k\|$, the supremum over the norms of the individual Hankel operators. In terms of the Hankel norm, the main approximation result reads as follows.

**Theorem 1.** \((31)\) Let $T$ be a bounded, strictly upper, locally finite operator and let $\Gamma$ be a Hermitean diagonal operator. Let $H_k$ be the Hankel matrix of $\Gamma^{-1}T$ at time instant $k$. Suppose that the singular values of each $H_k$ decompose into two sets $\sigma_k$ and $\sigma_{k,2}$ with $\sigma_k$ containing all ($N_k$ say) singular values larger than one and $\sigma_{k,2}$ containing the singular values smaller than one, and suppose furthermore that the infimum over $k$ of $\sigma_k$ is itself larger than one, and the supremum over $k$ of $\sigma_{k,2}$ is itself smaller than one. Let $N$ be the sequence \(\{N_k\}\).

Then there exists a strictly upper locally finite operator $T_0$ of system order at most $N$ such that
\[ \|\Gamma^{-1}(T - T_0)\|_H \leq 1. \]

A proof of this theorem appears in the full paper [3]. In this summarizing paper we will focus on the computational aspects involved in obtaining one such Hankel-norm approximant, and present recursive formulas by which a state space realization of $T_0$ can be obtained.

### 2. SUMMARY OF THE PROCEDURE

The problem that we shall solve in this section is the model reduction problem for a strictly upper operator described by a strictly stable “higher order model”. Let $\Gamma$ be a diagonal and hermitian operator. We shall use $\Gamma$ as a measure for the local accuracy of the reduced order model. It will also parametrize the solutions. We will look for a contractive operator $E$ such that $E = (T^* - T^*)\Gamma^{-1}$ where $T^*$ is an operator which is not necessarily upper triangular, but whose strictly causal part will assumed to be bounded and have state space dimensions of low order — as low as possible for a given $\Gamma$. Once we have such a contractive $E$, it is immediately verified that it satisfies
\[ \|\Gamma^{-1}(T - T^*)\|_H = \|E\| \leq 1. \]

Let $T_0$ be the strictly causal part of $T^*$. Then
\[ \|\Gamma^{-1}(T - T_0)\|_H \leq \|\Gamma^{-1}(T - T^*)\| \leq 1, \]
and $T_0$ is a Hankel-norm approximant when $T^*$ is an operator-norm approximant. To find $T_0$ we start by determining a (minimal) factorization of $T$ in the form $T = \Delta^\dagger U$ where $\Delta$ and $U$ are upper operators which have finite state space dimensions of the same size as that of $T$, and $U$ is inner: $UU^* = U^*U = I$. A $J$-unitary operator $\Theta$ is an operator with block decomposition
\[ \Theta = \begin{bmatrix} \Theta_{11} & \Theta_{12} \\ \Theta_{21} & \Theta_{22} \end{bmatrix}, \]
and signature matrices
\[ J_1 = \begin{bmatrix} I_{M_1} & -I_{M_1} \\ \end{bmatrix}, \quad J_2 = \begin{bmatrix} I_{M_2} & -I_{M_2} \end{bmatrix}, \]

such that $J_1\Theta J_2 = J_2 \Theta J_1 = J_1$. We look for a locally finite $J$-unitary $\Theta$-operator with upper block operators chosen such that
\[ \left[ U^* - T\Gamma^{-1} \right] \Theta = [A' - B'] \]
consists of two upper operators. A solution to this problem exists if certain conditions on a Lyapunov equation associated to $\Gamma^{-1}T$ are satisfied (see below). $\Theta$ will again be locally finite. Then, because $\Theta$ is $J$-unitary, we have that $\Theta_{22}\Theta_{22} = I + \Theta_{12}\Theta_{12}$, and because $\Theta$ is locally finite, $\Theta_{22}$ will exist (but not necessarily upper) and $\Sigma_{12} = -\Theta_{12}\Theta_{22}$ will be contractive. From (4) we have
\[ B' = -U^*\Theta_{12} + T^*\Gamma^{-1}\Theta_{22}. \]
Define the approximating operator $T'$ as
\[ T' = B'\Theta_{22}^\dagger \Gamma, \]
then $E := (T^* - T')I = -U^* \Sigma_{12}$. Because $\Sigma_{12}$ is contractive and $U$ unitary, we infer that $\|E\| \leq 1$, so that $T'^* = B^* \Theta_{22} \Gamma^* I$ is indeed an approximant with an admissible modeling error. In view of the target theorem 1, we have to show that the strictly causal part of $T'$ has the stated number of states and to verify the relation with the Hankel singular values of $\Gamma^{-1}T$. This will done in the remaining part of this section.

Factorization of $T$

For a factorization $T = \Delta^* U$, we start from a realization of $T$ in output canonical form, i.e., such that $AA^* + CC^* = I$. For each time instant $k$, augment the state transition matrices $[A_k, C_k]$ of $T$ with as many extra rows as needed to yield a unitary (hence square) matrix $U_k$:

$$U_k = \begin{bmatrix} A_k & C_k \\ B_k^* & D_k^* \end{bmatrix}. \tag{7}$$

Assemble the individual matrices $\{A_k, B_k, C_k, D_k\}$ in diagonal operators $\{A, B, C, D\}$, then the corresponding operator $U$ is a state space realization for $U$: $U = D_k + B_k Z (I - AZ)^{-1} C$. $U$ is well-defined and upper, and it is unitary because it has a unitary realization. It is straightforward to verify that $\Delta = UT^*$ is indeed upper. Note that the number of rows added to $[A_k, C_k]$ is time-varying, so that $U$ (and hence also $\Delta$) has a time-varying number of inputs.

Construction of $\Theta$

The next step is to construct a locally finite and block-upper $J$-unitary $\Theta$ that satisfies equation (5). Let $B$ be the space in which the state sequences of the realization $\Theta$ of $\Theta$ live. $\Theta$ will be $J$-unitary in the sense of (3) if $\Theta$ satisfies

$$\Theta^* \begin{bmatrix} J_B & J_1 \\ J_1 & J_2 \end{bmatrix} \Theta = \begin{bmatrix} J_B \Theta_0 \Theta_1 \Theta_2 \end{bmatrix} \tag{8}$$

where $B = B, \Theta = \Theta_0 \Theta_1 \Theta_2$ is a certain decomposition of $B$, and

$$J_B = \begin{bmatrix} I_B & 0 \\ 0 & -I_B \end{bmatrix} \tag{9}$$

is a corresponding signature matrix. (In this context, we call $J_B$ the state signature sequence.) Let $\{A, B, C, 0\}$ be the realization for $T$ used in the previous section (it is in output canonical form), and let $\{A, B, C, D\}$ be the realization for the inner factor $U$ of $T$. We submit that $\Theta$ satisfying (4) has a realization $\Theta$ of the form

$$\Theta = \begin{bmatrix} X \\ I \end{bmatrix} \begin{bmatrix} A & C_1 & C_2 \\ B_1 & D_{11} & D_{12} \\ \Gamma^{-1} B & D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} (X^{-1})^* \\ I \end{bmatrix} \tag{10}$$

which is a square matrix at each time instant $k$, and where $X$ and $C_i, D_i$ are yet to be determined. Note that the state sequence space $B$ is the same for $\Theta$ and $T$. $X$ is a boundedly invertible diagonal state transformation operator which is such that $\Theta$ is $J$-unitary as in (8). Writing $\Lambda = X^T J_B X$, the signature of $\Lambda$ will determine $J_B$, and it is straightforward to derive that $\Lambda = I - M$, where $M$ is given by

$$M^{-1} = A^* M A + B^* \Gamma^{-1} \Gamma^{-1} B. \tag{11}$$

$M$ is the solution of one of the Lyapunov equations associated to $\Gamma^{-1}T$, and can be determined recursively from the given realization of $T$ via $M_{k+1} = A_k^* M_k A_k + B_k^* \Gamma^{-1} \Gamma^{-1} B_k$. It is closely related to the Hankel operator of $\Gamma^{-1}T$. In particular, the non-zero singular values of the Hankel operator of $(\Gamma^{-1} T)$ at time $k$ are equal to the eigenvalues of the matrix $M_k$. Using $\Lambda = I - M$, it follows that a singular value of the $k$-th Hankel operator of $\Gamma^{-1}T$ that is larger than 1 corresponds to a negative eigenvalue of $A_k$. Hence $J_B$, the signature of $\Lambda$, is determined by the Hankel singular values of $\Gamma^{-1}T$. With $X$ known, the remaining unknown columns in (10) can be determined straightforwardly and independently for each time instant $k$ as the $J$-unitary complement of the columns that are already known, such that the resulting $\Theta$ is square and $J$-unitary at each point in time.

At this point we have covered the first part of theorem 1: we have constructed a $J$-unitary $\Theta$ and from it an operator $T_0$ which is a Hankel-norm approximant of $T$. It remains to verify the complexity assertion, which stated that the dimension of the state space of $T_0$ is at most equal to $N$: the number of Hankel singular values of $\Gamma^{-1}T$ that are larger than one, or the number of negative entries in the state signature $J_B$ of $\Theta$. Assume that the strictly causal part $T_0$ of $T'$ exists as a bounded operator. In view of its definition in (6), it can be shown that (under the conditions of theorem 1) the input state space of $T_0$ is contained in the input state space of $\Theta_2$, and that the dimension of this space is at most equal to $N = \#(J_B)$, the number of negative entries in the state signature sequence of $\Theta$. This proves the second part of theorem 1.

3. STATE REALIZATION OF $T_0$

The preceding section has given a general expression for a Hankel-norm approximant $T_0$. It is possible to give a state realization for $T_0$ in terms of the state realizations of $T, U$ and $\Theta$, which have been derived in the preceding sections. Necessary for the construction is to obtain a realization for $B'$ and for the upper part of $\Theta_2$. The former is straightforward from its definition (4); the latter is more difficult because $\Theta_2$ is not causal, and requires the construction of a (unitary) realization $\Sigma$ that is related to $\Theta$ in the following way. Paritize the state $x$ of $\Theta$ according to the
signature \( J^2 \) into \( x = [x_1 \ x_2] \), and partition \( \Theta \) likewise:

\[
\Theta = \begin{bmatrix}
    x_1^{(i)} & x_2^{(i)} & \mathbf{a}_2 & \mathbf{b}_2 \\
    x_1 & x_2 & \mathbf{a}_1 & \mathbf{b}_1 \\
    a_1 & b_1 & a_2 & b_2
\end{bmatrix}, \quad (12)
\]

then the corresponding \( \Sigma \) is defined by the relation

\[
[x_1 \ x_2 \ a_1 \ b_1] \Theta = [x_1^{(i)} \ x_2^{(i)} \ a_2 \ b_2]
\]

(inputs of \( \Sigma \) have positive signature) and has partitioning

\[
\Sigma = \begin{bmatrix}
    x_1^{(i)} & x_2^{(i)} & \mathbf{a}_2 & \mathbf{b}_2 \\
    x_1 & x_2 & \mathbf{a}_1 & \mathbf{b}_1 \\
    a_1 & b_1 & a_2 & b_2
\end{bmatrix}, \quad (13)
\]

From this definition, the entries in \( \Sigma \) can be determined as

\[
\begin{aligned}
    F_{11} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    G_{11} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    H_{11} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    H_{10} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2
\end{aligned}
\]

\[
\begin{aligned}
    F_{12} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    G_{12} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    H_{12} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    H_{11} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2
\end{aligned}
\]

\[
\begin{aligned}
    F_{21} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    G_{21} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    H_{21} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    H_{22} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2
\end{aligned}
\]

\[
\begin{aligned}
    F_{22} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    G_{22} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    H_{22} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \\
    H_{21} & = \mathbf{a}_1 \quad \mathbf{a}_2 \quad \mathbf{a}_2 \quad \mathbf{a}_2
\end{aligned}
\]

Note that the matrix \( \Sigma \) only depends on the entries of \( \Theta \), so that it can be computed independently for each time instant. \( \Sigma \) is the state realization of the operator \( \Sigma \) which is the scattering operator that can be associated to \( \Theta \); from equation (13) it is seen that the realization is not causal: the state \( x \) is computed instead of used.

Let \( (A,B,C,0) \) be a state realization for \( T \) as in theorem 1 with \( M \) defined by the recursion in (11), let \( (A,B_s,C,D_s) \) be a realization for the inner factor \( U \) of \( T \), the interpolating \( \Theta \) as in (10), and \( \Sigma \) corresponding to \( \Theta \) as above. The following additional recursions are needed to generate a state realization of \( T_a \):

\[
S^{(i)} = F_{21} + F_{22}(I - SF_{12})^{-1}SF_{11}
\]

\[
R = F_{12} + F_{11}R^{(i)}(I - F_{21}R^{(i)})^{-1}F_{22}
\]

\[
C_s = [G_{22} + G_{21}R^{(i)}(I - F_{21}R^{(i)})^{-1}F_{22}] (I - SR)^{-1}.
\]

In terms of these quantities, a state realization \( (A_s,B_s,C_s,0) \) of \( T_a \) is given by

\[
A_s = F_{22}(I - SF_{12})^{-1}
\]

\[
B_s = H_{22} + F_{22}(I - SF_{12})^{-1}SH_{12}
\]

\[
C_s = C_s [D_{11}D_{21} + C_3(J - M)L] + A_s Y^{(i)}A^* M C
\]

\[
Y = A_s Y^{(i)} A^* + C_3 C_2^*.
\]

in which a fourth recursively determined quantity \( Y \) is introduced. A check on the dimensions of \( A_s \) reveals that the state realization for \( T_a \) has indeed a state space dimension given by \( N = \#(J_{bs}) \): the number of Hankel singular values of \( T \) that are larger than 1. The realization is given in terms of four recursions: two for \( M \) and \( S \) that run forward in time, the other two for \( R \) and \( Y \) that run backward in time and that depend on \( S \). One implication of this is that an optimal approximant of \( T \) can only be derived if the model of \( T \) is known for all time, which is of course not very surprising. It is possible to obtain results in finite time for special cases, such as (1) finite \((n \times n)\) matrices, in which case \( M, S, R, Y \) have zero dimensions outside an interval of size \( n \), and (2) systems which are time-invariant outside a finite interval of size \( n \), in which case the recursions become (eigenvalue) equations outside that interval, and can be solved analytically. This brings another remarkable feature of this algorithm into light: in the time-invariant case, an eigenvalue decomposition is needed to split \( \Theta \) into upper and lower parts, i.e., to retrieve the stable and unstable poles of the approximant. In the current setting, the notion of a "time-varying pole" is ill-defined, because e.g., the state dimension can vary in time. However, the above recursions avoids this notion altogether, while at the same time \( \Theta \) is never explicitly formed.

REFERENCES


OPTIMUM INTERPOLATION FOR TIME-LIMITED AND BAND-LIMITED SIGNALS

VICTOR NEAGOE, Member EURASIP

Department of Electronics and Telecommunications, Polytechnic Institute of Bucharest, Bucharest 16, Romania 77206*

This paper proposes an optimum method of uniform sampling reconstruction for signals assumed to be both band-limited and also time-limited. The method is based on the estimation of the Legendre nonuniformly taken samples of the Fourier transform in order to be consistent with the given sequence of uniform time samples. An analytical formula of the maximum reconstruction error and the corresponding condition of convergence for interpolation are deduced. The computer simulations prove the important advantage of the proposed method over the classical Shannon interpolation.

1. INTRODUCTION

The Shannon sampling theorem and its variants [1, 3] are well known as performing the reconstruction of a band-limited signal, from the knowledge of its corresponding infinite series of uniformly taken samples. However, in practice we have only time-limited signals, so that the signal reconstruction must be performed from a finite sequence of samples, which leads to the truncation of Shannon series and implicitly to the occurrence of corresponding truncation errors [1, 4].

Recently, Neagoe [2] has presented an optimum nonuniform sampling theorem for time-limited signals, by minimizing the maximum instantaneous interpolation error. The sampling moments are taken according to the Chebyshev polynomial roots and then the Discrete Cosine Transform (DCT) is applied on the vector of nonuniformly taken samples. However, although it is an optimum model for time-limited signals, firstly, it does not consider the constraint of band-limitation as a previous assumption for optimization and secondly, it seems a cumbersome task to build a practical device for nonuniform (asynchronous) signal sampling in the time domain.

Neagoe [3] also proposes a nonuniform sampling theorem for time-limited signals, to the aim of high accuracy preserving the integral signal characteristics (such as energy and Fourier Transform). It consists of the cascade of nonuniform sampling in the time domain according to Legendre polynomial roots followed by the Discrete Legendre Transform (DLT) applied on the vector of nonuniformly taken samples. We retain from the above method the idea of Legendre nonuniform sampling but in the frequency domain instead of the time domain and we further apply this idea in the now proposed model.

This paper presents a new method for signal interpolation giving several improvements over the previous two models of the author, namely it takes into consideration both the time-limiting and also the band-limiting conditions as a priori constraints for optimization. Moreover, it considers the usual uniform sampling in the time domain, which is very simple and practical, but the method estimates the nonuniformly taken samples of the Fourier Transform in the frequency domain, giving us a high accuracy interpolation formula, according to the Gauss quadrature rule.

2. THE NEW INTERPOLATION METHOD

Interpolation Theorem: Assume we have a real-valued signal, g(t), considered to be both time-limited and also band-limited \([g(0,N) \rightarrow \mathbb{R}, 0 \leq N, \|g\|_{\infty} < \infty}\), \(G(f) = 0\) for \(|f| > H\), where \(G(f)\) is the Fourier transform of \(g(t)\), so that \(g(t) = \int_{-1}^{1} G(f)e^{2\pi ift} dt\), \(i^{2} = -1\) and also assume that the vector of uniformly taken signal samples \(\mathbf{g} = (g_j = g(j))_{j=0}^{N} = (g_0, g_1, \ldots, g_N)^T\) is given.

Then:

(a) The optimum interpolation formula minimizing the maximum instantaneous error is:

\[
\hat{g}_n(t) = \sum_{k=0}^{N} \hat{g}_j(t)g_k(t),
\]

where \(\hat{g}_j(t)\) is a real-valued function, \(\psi(t)_{j=0,N} \rightarrow \mathbb{R}\), given by

\[
\psi(t)_{j=0,N} = \sum_{k=1}^{N} r_{j,k} \psi_k(t), \quad (j=0,1,\ldots,N),
\]

where \(r_{j,k} = e^{2\pi i jk}, \quad (j=1,2,\ldots,N),\quad (k=1,2,\ldots,1/(2N))\)

is the ratio between the present sampling period and the Shannon sampling period, \(\{r_{j,k}\}_{j=0}^{N}\)

is the set of ordered roots of the \((n+1)\)th degree Legendre polynomial \((-1 \leq r_{j,k} \leq 1)\), and \(\psi_k \in \mathbb{C}\) is the general element of the matrix \(\hat{A} = (\psi_k)_{j=0,N}, k \leq j \leq N\), being the row index, and \(j\) the column index, so that \(\sum\)


*
Author's mailing address: V. Neagoe, P.O. Box 16-37, Bucharest 16, Romania 77500, Phone: 40-0-714769.
we have reduced an analytical formula for the inversion of the matrix \( \mathbf{C} \), i.e.,

\[
(\mathbf{C}^{-1})_{ij} = \frac{-1}{2} \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \frac{\sin \left( \frac{\pi (i-k) j}{n} \right) \sin \left( \frac{\pi (j-l) i}{n} \right)}{\sin \left( \frac{\pi i}{n} \right) \sin \left( \frac{\pi j}{n} \right)}
\]

with the result of the previous steps.

Theorem 2: Let \( \mathbf{C} \) be the given matrix. Then, the following formula holds:

\[
(\mathbf{C}^{-1})_{ij} = \frac{1}{2} \sum_{k=0}^{\infty} \sum_{l=0}^{\infty} \frac{\sin \left( \frac{\pi (i-k) j}{n} \right) \sin \left( \frac{\pi (j-l) i}{n} \right)}{\sin \left( \frac{\pi i}{n} \right) \sin \left( \frac{\pi j}{n} \right)}
\]

3. INTERPOLATION PERFORMANCE AND CONCLUDING REMARKS

Table I shows the numerical results of the interpolation error bound for the proposed method, as compared to other methods. We may show that, unlike the Shannon interpolation, the error bound for the proposed method is significantly lower. The Shannon bound is also included for comparison.

\[
\begin{align*}
\phi(n) & = \frac{1}{2^n} \sum_{k=0}^{2^n-1} \left( \psi \right)_{n+k}
\end{align*}
\]

where \( \psi \) is the positive integer, \( n \) is the integer part of \( \frac{2^n}{2} \), and \( \phi(n) \) is the effective integration of \( \psi \).
\[ \mathcal{E} = \frac{\mathcal{E}_{\text{Shannon}}}{\sqrt{\mathcal{E}_1}} = \frac{\mathcal{E}_{\text{Chebyshev}}}{\sqrt{\mathcal{E}_1}} = \frac{\mathcal{E}_{\text{Neagoe2}}}{\sqrt{\mathcal{E}}} \]

\[ \mathcal{E}_{\text{Shannon}} \leq C_1(r) \cdot \frac{f}{n} \]  

\[ \mathcal{E}_{\text{Chebyshev}} \leq C_2(r) \cdot \frac{f}{n+1} \cdot \left( \frac{r \pi e}{4} \right)^{n+1} \]  

\[ \mathcal{E}_{\text{Neagoe2}} \leq C_3(r) \cdot \frac{f}{n+1} \cdot \left( \frac{r \pi e}{4} \right)^{3(n+1)} \]  

We can remark the strong exponential convergence towards zero of the error bound for the presented method (the exponent being \(2n+2\)) by comparison with the exponential convergence for the Neagoe previous nonuniform sampling method (with exponent \((n+1)\)), and with the weak \(n^{-1}\) convergence of the error bound for Shannon interpolation.

We have also evaluated by computer simulation the experimental interpolation performances of the new method. We have considered the signal

\[ g(t) = A_0 + \sum_{i=1}^{5} A_i \cos(2\pi f_i t), \]  

whose parameters are given in Table II. The signal is sampled with a sampling period equal to 1 \((r=2\pi w=0.45)\), using \(N=11=9\) samples. The experimental interpolation results are given in Table II. For the considered example, we deduce that the interpolation error becomes at least 110 times smaller by applying our proposed method, as compared to Shannon formula.

REFERENCES


### Table II

**Computer simulation performances obtained for the new interpolation method by comparison with Shannon formula**

\(n=8\) \((n+1=9\) uniformly taken samples\)

\( g(t) = \mathbb{R}, \ g(f) = 0 \) for \(|f| > W\), where

\[ g(t) = \int_{-W}^{W} g(f) e^{2\pi ft} df \]

\[ g(t) = A_0 + \sum_{i=0}^{5} A_i \cos(2\pi f_i t) \]

\( f_0 = 0 \)

\( f_1 = 0.05 \)

\( f_2 = 0.115 \)

\( f_3 = 0.14 \)

\( f_4 = 0.175 \)

\( f_5 = 0.225 \)

\( A_0 = 0.5 \)

\( A_1 = 1 \)

\( A_2 = 1/(2.3) \)

\( A_3 = 1/(2.8) \)

\( A_4 = 1/(3.5) \)

\( A_5 = 1/(4.5) \)

\[ \mathcal{E}_j = \left| \hat{g}(j + 0.5) - g(j + 0.5) \right| \]

<table>
<thead>
<tr>
<th>( j )</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>( 10^4 \cdot \mathcal{E}_j )</td>
<td>0.37</td>
<td>0.34</td>
<td>0.17</td>
<td>0.28</td>
<td>0.22</td>
<td>0.17</td>
<td>0.95</td>
<td>4.16</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Normalized interpolation error</th>
<th>The New Method</th>
<th>Shannon Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>3541</td>
</tr>
<tr>
<td>TYPE OF INTERPOLATION ERROR BOUND</td>
<td>NUMBER OF TAKEN SAMPLES</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>INTERPOLATION ERROR BOUND FOR SHANNON UNIFORM SAMPLING RECONSTRUCTION</td>
<td>BROWN BOUND</td>
<td></td>
</tr>
<tr>
<td>$E_{SH} = \max_{t \in (\frac{Q}{2} - \frac{\pi}{2}; \frac{Q}{2} + \frac{\pi}{2})} \frac{</td>
<td>g(t) - \hat{g}_n(t)</td>
<td>}{\sqrt{E}}$</td>
</tr>
<tr>
<td>(for a signal $g(t)$ uniformly sampled in $N=n+1$ points: $j=0,1,...,n$; $n$ is even; sampling interval $= 1$; $r=1/(2\pi)$; $N$ = signal bandwidth; $G(f)$ = Fourier ($\hat{g}(f)$)); where $G(f)=0$, for $</td>
<td>f</td>
<td>&gt; W$; $E_1 = \frac{1}{\sqrt{W}} \int</td>
</tr>
<tr>
<td>PIPE BOUND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{PI} = \max_{t \in (\frac{Q}{2} - \frac{\pi}{2}; \frac{Q}{2} + \frac{\pi}{2})} \frac{</td>
<td>g(t) - \hat{g}_n(t)</td>
<td>}{\sqrt{E}}$</td>
</tr>
<tr>
<td>YAO-THOMAS BOUND</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{YT} = \max_{t \in (\frac{Q}{2} - \frac{\pi}{2}; \frac{Q}{2} + \frac{\pi}{2})} \frac{</td>
<td>g(t) - \hat{g}_n(t)</td>
<td>}{\sqrt{E}}$</td>
</tr>
<tr>
<td>INTERPOLATION ERROR BOUND FOR NEAGOŞ NONUNIFORM TIME SAMPLING RECONSTRUCTION</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{N1} = \max_{t \in (0,n)} \frac{</td>
<td>g(t) - \hat{g}_n(t)</td>
<td>}{\sqrt{E}}$</td>
</tr>
<tr>
<td>(for a signal $g(t)$ uniformly sampled in $N=n+1$ points over the interval $[0,n]$, according to Chebyshev polynomial roots; $r_W, G(f)$, and $E_1$ have the significances given above).</td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTERPOLATION ERROR BOUND FOR THE PRESENTED OPTIMUM UNIFORM SAMPLING RECONSTRUCTION METHOD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{N2} = \max_{t \in (\frac{Q}{2} - \frac{\pi}{2}; \frac{Q}{2} + \frac{\pi}{2})} \frac{</td>
<td>g(t) - \hat{g}_n(t)</td>
<td>}{\sqrt{E}}$</td>
</tr>
<tr>
<td>(for a signal $g(t)$, uniformly sampled in $N=n+1$ points: $j=0,1,...,n$; $n$ is even; sampling interval $= 1$; $r, W, G(f)$ have the significances given above, while $E = \int_0^\pi g^2(t)dt$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER OF $N=n+1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N=5$</td>
<td>$N=9$</td>
<td>$N=17$</td>
</tr>
<tr>
<td>BROWN BOUND</td>
<td>0.625904</td>
<td>0.250261</td>
</tr>
<tr>
<td>PIPE BOUND</td>
<td>0.534243</td>
<td>0.213697</td>
</tr>
<tr>
<td>YAO-THOMAS BOUND</td>
<td>0.325943</td>
<td>0.131817</td>
</tr>
<tr>
<td>INTERPOLATION ERROR BOUND FOR THE PRESENTED OPTIMUM UNIFORM SAMPLING RECONSTRUCTION METHOD</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$E_{N2} = \max_{t \in (\frac{Q}{2} - \frac{\pi}{2}; \frac{Q}{2} + \frac{\pi}{2})} \frac{</td>
<td>g(t) - \hat{g}_n(t)</td>
<td>}{\sqrt{E}}$</td>
</tr>
<tr>
<td>(for a signal $g(t)$, uniformly sampled in $N=n+1$ points: $j=0,1,...,n$; $n$ is even; sampling interval $= 1$; $r, W, G(f)$ have the significances given above, while $E = \int_0^\pi g^2(t)dt$)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>NUMBER OF $N=n+1$</td>
<td></td>
<td></td>
</tr>
<tr>
<td>$N=5$</td>
<td>$N=9$</td>
<td>$N=17$</td>
</tr>
<tr>
<td>0.019037</td>
<td>0.009795</td>
<td>0.003942</td>
</tr>
<tr>
<td>0.009080</td>
<td>0.006079</td>
<td>0.002630</td>
</tr>
</tbody>
</table>
NONUNIFORM DISCRETE FOURIER TRANSFORM AND ITS APPLICATIONS IN SIGNAL PROCESSING

SANJIT K. MITRA, SONALI CHAKRABARTI, and EDUARDO ABREU

Department of Electrical and Computer Engineering
University of California, Santa Barbara, CA 93106

The nonuniform discrete Fourier transform (NDFT) of a sequence of length N is defined as samples of its z-transform at N arbitrarily located distinct points on the z-plane. The NDFT reduces to the conventional DFT when the sampling points are located on the unit circle at equally spaced angles. The properties of the NDFT and its applications in signal processing such as spectral analysis and FIR filter design are outlined. Extension of the concept of the NDFT to two dimensions is discussed.

1. INTRODUCTION

In many applications, the problem of computing samples of the z-transform X(z) of a sequence \( \{x(n)\} \) of length N is of interest. One popularly used approach is to compute the discrete Fourier transform (DFT) of the sequence, which corresponds to evaluating its z-transform at N equally spaced points on the unit circle. However, in certain applications such as spectral analysis, it is preferable to sample the z-transform X(z) at N points that are nonuniformly spaced. There have been various efforts to achieve such nonuniform sampling [1],[2].

A better approach, considered in this paper, is to generalize the definition of the DFT. The nonuniform discrete Fourier transform (NDFT) of a sequence \( \{x(n)\} \) of length N is defined as a sequence of length N that corresponds to samples of the z-transform X(z) at N distinct points located arbitrarily on the z-plane. The NDFT reduces to the DFT when the N points are chosen at equally spaced angles on the unit circle. The freedom offered by the NDFT in choosing the points on the z-plane can be exploited in many signal processing applications.

2. DEFINITION

The nonuniform discrete Fourier transform (NDFT) of a sequence \( \{x(n)\} \) of length N is defined as

\[
X(z_k) = \sum_{n=0}^{N-1} x(n)z_k^{-n}, \quad k = 0, 1, \ldots, N-1, \quad (1)
\]

where \( z_0, z_1, \ldots, z_{N-1} \) are distinct points located arbitrarily on the z-plane. This can be expressed in a matrix form as

\[
X = D \cdot x
\]

where

\[
X = \begin{bmatrix}
X(z_0) \\
X(z_1) \\
\vdots \\
X(z_{N-1})
\end{bmatrix}, \quad x = \begin{bmatrix}
x(0) \\
x(1) \\
\vdots \\
x(N-1)
\end{bmatrix}, \quad (2)
\]

and

\[
D = \begin{bmatrix}
1 & z_0^{-1} & z_0^{-2} & \cdots & z_0^{-N+1} \\
1 & z_1^{-1} & z_1^{-2} & \cdots & z_1^{-N+1} \\
1 & z_2^{-1} & z_2^{-2} & \cdots & z_2^{-N+1} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & z_{N-1}^{-1} & z_{N-1}^{-2} & \cdots & z_{N-1}^{-N+1}
\end{bmatrix}, \quad (3a)
\]

Note that the NDFT matrix D is a Vandermonde matrix and therefore is non-singular provided the N sampling points are distinct. Thus the inverse NDFT is simply

\[
x = D^{-1}X. \quad (4)
\]

The DFT has been widely used in analyzing the frequency content of signals. Typically, a continuous-time signal is sampled, a finite-duration window is applied to it to obtain a sequence of length N, and then an N-point DFT is used to compute samples of the spectrum of the windowed sequence. Often, misleading results are obtained using this procedure. For example, if the signal contains two closely spaced sinusoids, the resulting DFT may fail to resolve the corresponding peaks in the spectrum. For fixed length transforms, and some knowledge of the signal, the resolution can be improved by using the NDFT to compute the spectral samples. A simple example to illustrate this idea is shown in Figure 1.
The discrete-time Fourier transform of the windowed sequence is shown in Figure 1(a). Figure 1(b) shows the spectrum obtained by using a 64-point DFT. The unsatisfactory resolution obtained was improved by using a 64-point NDFT, where the spectrum was sampled more densely in the critical region.

![Graphs showing the comparison between DFT and NDFT](image)

**Figure 1.** (a) Original spectrum, (b) 64-point DFT and NDFT

## 3. PROPERTIES OF THE NDFT

Some relevant properties of the NDFT are outlined in this section.

### 3.1. Basic Properties

Let \( X(z_k) \) and \( Y(z_k) \) denote the NDFT of the sequences \( x(n) \) and \( y(n) \) respectively. For notational convenience we denote an NDFT pair as

\[
(\mathbf{x}, \mathbf{X}) \leftarrow \text{NDFT} \rightarrow (\mathbf{z}, \mathbf{Z})
\]

**Property 1.** Linearity:

\[
\alpha x(n) + \beta y(n) \leftarrow \text{NDFT} \rightarrow \alpha X(z_k) + \beta Y(z_k)
\]

Properties 2-5 hold if the sampling points \( z_0, z_1, \ldots, z_{N-1} \) are chosen in complex conjugate pairs.

**Property 2.**

\[
x^*(n) \leftarrow \text{NDFT} \rightarrow X^*(z_k^*)
\]

**Property 3.**

\[
\text{Re}\{x(n)\} \leftarrow \text{NDFT} \rightarrow X_e(z_k) = \frac{1}{2} \left[ X(z_k) + X^*(z_k^*) \right]
\]

**Property 4.**

\[
\text{Im}\{x(n)\} \leftarrow \text{NDFT} \rightarrow X_o(z_k) = \frac{1}{2} \left[ X(z_k) - X^*(z_k^*) \right]
\]

Property 5 holds if \( x(n) \) is real.

**Property 5.** Symmetry Properties.

\[
X(z_k) = X^*(z_k^*)
\]

\[
\text{Re}\{X(z_k)\} = \text{Re}\{X(z_k^*)\}
\]

\[
\text{Im}\{X(z_k)\} = -\text{Im}\{X(z_k^*)\}
\]

\[
|X(z_k)| = |X(z_k^*)|
\]

\[
\text{arg}\{X(z_k)\} = -\text{arg}\{X(z_k^*)\}
\]

### 3.2. Linear Convolution Using the NDFT

Let \( \{y(n)\} \) be the linear convolution of the sequences \( \{x(n)\} \) and \( \{h(n)\} \). Using vector notation this convolution can be expressed as

\[
y = x * h \tag{5}
\]

where the vectors \( x \) and \( h \) are of length \( N \) and \( y \) is of length \( 2N-1 \). Let \( X \) and \( H \) denote the \( N \)-point NDFT of \( x \) and \( h \) respectively, i.e.

\[
X = D \mathbf{x}, \quad H = D \mathbf{h}, \tag{6}
\]

and let

\[
x_c = D^{-1}[X \cdot H] \tag{7}
\]

where the dot "\( \cdot \)" denotes the componentwise product of the vectors \( X_1 \) and \( X_2 \). Evaluating the \( z \)-transforms of both sides of Eq. (5) at \( z = z_0, z_1, \ldots, z_{N-1} \) we get

\[
X \cdot H = [D \ C] y = D [I \ D^{-1}C] y. \tag{8}
\]

where \( I \) is the \( N \times N \) identity matrix and

\[
C = \begin{bmatrix}
z_0^{-N} & z_0^{-N-1} & \cdots & z_0^{-2N+2} \\
z_1^{-N} & z_1^{-N-1} & \cdots & z_1^{-2N+2} \\
\vdots & \vdots & \ddots & \vdots \\
z_{N-1}^{-N} & z_{N-1}^{-N-1} & \cdots & z_{N-1}^{-2N+2}
\end{bmatrix}. \tag{9}
\]

Substituting Eq. (8) in Eq. (7), we arrive at

\[
x_c = [I \ D^{-1}C] y. \tag{10}
\]

From Eq. (10) we conclude that if \( x_c \) is made to have length \( 2N-1 \) by augmenting the sequences \( x \) and \( h \) with \( N-1 \) zeroes at the end, and then using an NDFT...
matrix of size $(2N-1)\times(2N-1)$, $x_\nu$ will correspond to the linear convolution of $x$ and $h$. Thus linear convolution can be obtained using the NDFT and the zero padding technique, similar to the DFT.

4. FIR FILTER DESIGN USING THE NDFT

In our proposed FIR filter design method, the desired frequency response is first evaluated at $N$ nonuniformly spaced points around the unit circle. The filter coefficients are then obtained by an $N$-point inverse NDFT of these frequency samples. This procedure is mathematically equivalent to computing the coefficients of an interpolating polynomial of order $N$ that has the desired values at the specified frequency locations [3]. This approach is thus a generalization of the classical frequency sampling approach but yields improved results.

As an example we considered the design of linear phase lowpass filters with nearly equiripple frequency responses. The desired response was obtained by the method described in [4]. In this approach, separate functions derived from Chebychev polynomials are used to approximate the passband and stopband of the desired equiripple frequency response. In our proposed design method, since the frequency samples are nonuniformly spaced, they were chosen to be in the passband and stopband only. It was found that the best locations for the frequency samples were at the extrema of the desired equiripple response. Analytic expressions were found, specifying these locations precisely from the given filter specifications. In the NDFT formulation in Eq. (2), the Vandermonde matrix $D$ was constructed using the frequency sample locations and $X$ contained the sample values. The impulse response $x$ was then obtained by solving this linear system of equations. In many cases, the resulting interpolated frequency responses were found to be very close to optimal responses obtained by using the Parks-McClellan algorithm for the same filter specifications. Moreover, the CPU design time needed was considerably less when compared to the Parks-McClellan algorithm.

4.1. A Design Example

A lowpass filter was designed with the following specifications: passband edge $\omega_p$ at $0.3\pi$, stopband edge $\omega_s$ at $0.36\pi$, passband ripple $\delta_p = 0.01$, and stopband ripple $\delta_s = 0.005$. Table 1 shows a comparison of the two filters, each of length 71, designed using the proposed approach and the Parks-McClellan algorithm respectively.

<table>
<thead>
<tr>
<th>Table 1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Design method</td>
</tr>
<tr>
<td>----------</td>
</tr>
<tr>
<td>NDFT</td>
</tr>
<tr>
<td>Parks-McClellan</td>
</tr>
</tbody>
</table>

5. THE 2-D NDFT

The nonuniform discrete Fourier transform of a 2-D sequence can be defined in a manner similar to that of a 1-D sequence. Thus, the 2-D NDFT of a sequence $x(m,n)$ of size $M\times N$ is defined as

$$
\hat{X}(z_{1k}, z_{2k}) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) z_{1k}^m z_{2k}^n, \quad k = 0, 1, \ldots, MN - 1
$$

where $(z_{1k}, z_{2k})$ represent distinct frequency locations, which can be chosen arbitrarily but in such a way that the inverse transform exists.

For the sake of clarity, this is illustrated by means of a simple example. Consider the case when $M = N = 2$. Then Eq. (10) can be expressed in a matrix form as

$$
\hat{X} = DX
$$

where

$$
\hat{X} = \begin{bmatrix}
\hat{X}(z_{10}, z_{20}) \\
\hat{X}(z_{11}, z_{21}) \\
\hat{X}(z_{12}, z_{22}) \\
\hat{X}(z_{13}, z_{23})
\end{bmatrix}, \quad X = \begin{bmatrix}
x(0,0) \\
x(0,1) \\
x(1,0) \\
x(1,1)
\end{bmatrix}
$$

$$
D = \begin{bmatrix}
1 & z_{20}^{-1} & z_{10}^{-1} & z_{10}^{-1} & z_{20}^{-1} \\
1 & z_{11}^{-1} & z_{11}^{-1} & z_{21}^{-1} & z_{12}^{-1} & z_{22}^{-1} \\
1 & z_{12}^{-1} & z_{22}^{-1} & z_{12}^{-1} & z_{22}^{-1} & z_{13}^{-1} & z_{23}^{-1} \end{bmatrix}
$$

In general, $D$ is a $MN \times MN$ 2-D Vandermonde matrix. In the special case when the frequencies are chosen on a nonuniform rectangular grid, i.e.,

$$
z_{1,m} = z_{1,m+pM}, \quad m = 0, 1, \ldots, M-1, p = 1, 2, \ldots, N-1,$$

$$
z_{2,n} = z_{2,n+qN}, \quad n = 0, 1, \ldots, N-1, q = 1, 2, \ldots, M-1,$$

Eq. (11) can alternatively be expressed in terms of two 1-D Vandermonde matrices, $D_1$ and $D_2$, of sizes $M \times M$ and $N \times N$ respectively, as

$$
\hat{X} = D_1 X D_2^{-1},
$$

where

$$
\hat{X} = \begin{bmatrix}
\hat{X}(z_{10}, z_{20}) \\
\hat{X}(z_{11}, z_{21}) \\
\hat{X}(z_{10}, z_{20}) \\
\hat{X}(z_{11}, z_{21})
\end{bmatrix}.
\[ X = \begin{bmatrix} x(0,0) & x(0,1) \\ x(1,0) & x(1,1) \end{bmatrix}, \]

\[ D_1 = \begin{bmatrix} 1 & z_{10}^{-1} \\ 1 & z_{11}^{-1} \end{bmatrix}, \quad D_2 = \begin{bmatrix} 1 & z_{20}^{-1} \\ 1 & z_{21}^{-1} \end{bmatrix}. \]

As can be seen, the 2-D DFT is a special case of the 2-D NDFT when the points are located on a uniform rectangular grid. In general the 2-D NDFT has \( MN \) degrees of freedom, which is far greater than the \( M+N \) degrees of freedom available when using a rectangular grid. This additional freedom can be harnessed in applications such as 2-D FIR filter design.

The 2-D NDFT can be used for the design of 2-D FIR filters by using a generalization of frequency sampling design technique. This method holds great potential for designing filters with unusual passband shapes and for achieving non-rectangular sampling geometries. Although several efforts for the design of 2-D FIR filters using nonuniform frequency sampling have been reported [5],[6], a general design procedure for 2-D nonseparable filters has not yet been found.

As an example we considered the design of 2-D half-band filters using this approach, e.g. diamond filters and 90° fan filters. Figure 2 shows the magnitude response and the contour plot of a 9\times9 diamond filter designed using the proposed approach. Due to the octagonal symmetry and the half-band nature of the frequency response, only 6 of the 81 impulse response coefficients are independent. These were found by taking 6 frequency samples of unit value, located nonuniformly as shown in Fig. 2(b). The maximum ripple obtained is only 0.003. A 9\times9 diamond filter with the same transition bandwidth, designed using the frequency transformation method, has a maximum ripple of 0.029.

6. CONCLUDING REMARKS

The concept of the nonuniform discrete Fourier transform, which corresponds to sampling the z-transform at arbitrarily chosen points on the z-plane, was presented in this paper. The NDFT was defined and its properties were studied. The advantage offered by the NDFT in choosing the points on the z-plane was illustrated by an example in spectral analysis. In particular, this idea was applied to nonuniform frequency sampling design of FIR filters. In many cases the resulting filters were practically equal to optimal equiripple filters, and the CPU design time was considerably reduced. A filter design example was presented to illustrate this technique. The NDFT was also extended to 2-D. An example of diamond filter design was presented to demonstrate the potential of the 2-D NDFT in 2-D nonseparable FIR filter design. Finally, the concept of the NDFT can easily be extended to higher dimensions.

Acknowledgements

This work was supported in part by a University of California MICRO grant with matching support from Granger-Teletra, Inc.

References


UNIQUENESS AND STABILITY OF DISCRETE ZERO-CROSSINGS AND MAXIMA WAVELET REPRESENTATIONS

Zeev Berman and John S. Baras

Electrical Engineering Department and Systems Research Center
University of Maryland, College Park, MD 20742, USA
e-mail: berman@src.umd.edu

A general signal description, called an inherently bounded Adaptive Quasi Linear Representation (AQLR), motivated by two important examples, namely the wavelet maxima representation, and the wavelet zero-crossings representation, is introduced. It is shown, that the dyadic wavelet maxima (zero-crossings) representation is, in general, nonunique. Using the idea of the inherently bounded AQLR, a global BIBO stability is shown. For a special case, where perturbations are limited to the continuous part of the representation, a Lipschitz condition is satisfied.

1 Introduction

S. Mallat in [4] and, together with Zhong, in [5], introduced zero-crossings and extrema of the wavelet transform as a multiscale edge representation. Two important advantages of these methods are low algorithmic complexity and flexibility in choosing the basic filter. Moreover, [4] and [5] propose reconstruction procedures and show accurate numerical reconstruction results from zero-crossings and maxima representations. In [4, 5], as in many other works in this area, the basic algorithms were developed using continuous variables. The continuous approach gives an excellent background to motivate and justify the use of either local extrema or zero-crossings as important signal features. Unfortunately, in the continuous framework, analytic tools to investigate the information content of the representation are not yet available. The knowledge about properties of the representations is mainly based on empirical reconstruction results. From the theoretical point of view, there are still important open problems, e.g. stability, uniqueness, and structure of a reconstruction set (a family of signals having the same representation).

Our objective is to analyze these theoretical questions using a model of an actual implementation. The main assumption is that the data is discrete and finite. The discrete multiscale maxima and zero-crossings representations are defined in a general set-up of a linear filter bank, however, the main goal is to consider a particular case where the filter bank describes the wavelet transform. Since reconstruction sets of both maxima and zero-crossings representations have a similar structure, a general form called the Adaptive Quasi Linear Representation (AQLR) is introduced. Moreover, many generalizations of the basic maxima and zero-crossings representations fit into the framework of the AQLR.

We first present conditions for uniqueness, then apply these conditions to the wavelet transform-based representation, and then obtain a conclusive result. It turns out, that neither the wavelet maxima representation nor the wavelet zero-crossings representation is, in general, unique.

The next subject is stability of the representation. This issue is of great importance because there are many known examples of unstable zero-crossings representations. In order to improve the stability properties, Mallat has included additional sums in the standard zero-crossings representation and, together with Zhong, they have introduced the wavelet maxima representation, as a stable alternative to the zero-crossings representation. Using the idea of the inherently bounded AQLR, global BIBO (bounded input, bounded output) stability is shown. For a special case, where perturbations are limited to the continuous part of the representation, a Lipschitz condition is satisfied.

\footnote{Research supported in part by NSF grant NSF-CDR-86-00108 under the Engineering Research Centers Program.}
2 Multiscale Maxima Representation

Consider $\mathcal{L}$, a linear space of real, finite sequences:

$$\mathcal{L} \triangleq \{ f: f = \{f(n)\}_{n=0}^{N-1}, f(n) \in \mathbb{R} \}.$$  

Let $X$ and $Y$ denote operators on $\mathcal{L}$ which provide the sets of local maxima and minima, respectively, of a sequence $f \in \mathcal{L}$. The formal definitions are:

$$Xf = \{ k : f(k+1) \leq f(k) \text{ and } f(k-1) \leq f(k) \}$$

$$Yf = \{ k : f(k+1) \geq f(k) \text{ and } f(k-1) \geq f(k) \}$$

In this work, in order to avoid boundary problems, an $N$-periodic extension of finite sequences is assumed.

Let $W_1, W_2, \ldots, W_J, S_J$ be linear operators on $\mathcal{L}$. The multiscale local extrema representation, $R_m f$, is defined as:

$$R_m f \triangleq \left\{ \{ XW_j f, YW_j f, \{W_j f(k)\}_{k=1}^{N-1}, S_j f \} \right\}_{j=1}^{J}.$$ 

where $EW_j f = XW_j f \cup YW_j f$.

Following [6], $R_m f$, will be called the multiscale maxima representation as well. In the particular case, when $W_1, W_2, \ldots, W_J, S_J$ correspond to a wavelet transform, $R_m f$ will be called the wavelet maxima representation.

The determination of the extrema point sets is highly nonlinear. However, for the given extrema sets, $XW_j f$ and $YW_j f$, the remaining data, called the sampling information, are obtained by a linear operation of sampling an image of a linear operator at fixed points.

Let $T_m f$ denote the linear operator associated with the sampling information. Then, $R_m f$ is written in an alternative way as:

$$R_m f = \left\{ \{ XW_j f, YW_j f \} \right\}_{j=1}^{J} T_m f.$$  

(1)

For a given representation $R_f$, a reconstruction set $\Gamma(R_f)$ is defined as a set of all sequences satisfying this representation, i.e.

$$\Gamma(R_f) \triangleq \{ \gamma \in \mathcal{L} : R\gamma = R_f \}.$$  

(2)

It is clear that in order to satisfy a given maxima representation, a sequence $h \in \mathcal{L}$, in addition to obeying the sampling information $T_m h = T_m f$, needs to meet the requirement that $W_j h$ has local extrema at the points of $XW_j f$ and $YW_j f$. Loosely speaking, we have to assure that $W_j h$ is increasing after a minimum and before a maximum and it is decreasing otherwise. Straightforward analysis yields:

Theorem 1 $R_m f$ is a given multiscale maxima representation. $h \in \Gamma(R_m f)$ if and only if

$$T_m h = T_m f$$  

(3)

$$t(k) \cdot (W_j h(k+1) - W_j h(k)) > 0$$  

(4)

The last inequality should be satisfied for $j = 1, 2, \ldots, J$ and for almost all $k$ (if two consecutive $k$’s belong to $EW_j f$ then the first is omitted here). $t(k)$ is called the type of $k$ and can be either $+1$ or $-1$.

The maxima representations can be cast into the form $Rf = \{Vf, Tf\}$, where $Vf$ is a set of points and $T$ is a linear operator which may depend on $Vf$. However, the key feature of the maxima representation is the fact that the set $Vf$ yields additional constraints, in the form of linear inequalities, which do not appear directly in $Rf$. Stimulated by this observation, we define the following general family of signal representations.

Definition 1 $Rf = \{Vf, Tf\}$ is called an Adaptive Quasi Linear Representation (AQLR) if there exists a linear operator $A$ and a sequence $a$ such that:

$$a \in \Gamma(Rf) \Rightarrow Ax = Tf \text{ and } Aa > a.$$  

(5)

$A, a$ may depend on $Vf$, but they must be independent of $Tf$.

The reasoning behind the name “Adaptive Quasi Linear Representation” (AQLR) is as follows. This representation is adaptive since $T, A, a$ depend on the sequence $f$ (via the set $Vf$). It is quasi linear because it is based on a set of linear equalities and inequalities.

Clearly, the following is true.

Proposition 1 Any multiscale maxima representation is an AQLR.

The next definition is a generalization of an essential boundedness property of the wavelet maxima representation.

Definition 2 An AQLR is called inherently bounded if there exists a real $K > 0$ such that

$$a \in \Gamma(Rf) \Rightarrow \|a\| \leq K\|Tf\|.$$  

The coefficient $K$ can depend on the parameters of the representation e.g. $N, J, W_1, \ldots, W_J, S_J$ but it must be independent of $Vf$ and $Tf$.

Proposition 2 The wavelet maxima representation is an inherently bounded AQLR.

It turns out that the wavelet maxima representation, as defined here, provides bounds for $\|W_j h\|$, $\|S_J h\|$. These bounds, due to Parseval’s equality, yield a bound for the original sequence $h$. For details, the reader is referred to [3].
3 The Multiscale Zero-Crossings

In defining the multiscale zero-crossings representation, we essentially follow [4], but minor changes are necessary due to our basic assumption that only a discrete signal version is available. Let \( Z \) be an operator which provides the set of zero-crossings of a given sequence \( f \in \mathcal{L} \), i.e.

\[
Z f = \{ k : f(k - 1) \cdot f(k) \leq 0 \}.
\]  

(6)

Mallat in [4] has stabilized the zero-crossings representation by including the values of the wavelet transform integral calculated between consecutive zero-crossing points. Therefore, the multiscale zero-crossings representation, \( R_s f \), is defined as:

\[
R_s f = \{ \{ ZW_j f, U_j^{(s)} f \}_{j=1}^{n(k)} \},
\]  

(7)

where

\[
U_j^{(s)} f(k) = \sum_{l=k}^{n(k)-1} W_j f(l).
\]

(8)

\( k \) and \( n(k) \) are two consecutive zero-crossings of \( W_j f \).

As in the maxima representation case, for fixed sets \( ZW_j f \), the remaining data \( U_j^{(s)} f \) and \( S_j f \) are obtained by a linear operator, denoted by \( T_j \). The zero-crossings representation can also be written as:

\[
R_s f = \{ \{ ZW_j f \}_{j=1}^{n(k)}, T_j f \}.
\]  

(9)

We have the following characterization of the reconstruction set.

**Theorem 2** Let \( R_s f \) be a given multiscale zero-crossings representation, \( h \in \Gamma(R_s f) \) if and only if:

\[
T_j h = T_j f,
\]  

(10)

\( t(k) \) is the type of \( k \) and can be either +1 or -1. The last inequality should be satisfied for almost all \( k \) (if \( W_j f(k) = 0 \) then \( k \) is omitted).

As an immediate consequence of Theorem 2 we have:

**Proposition 3** Any multiscale zero-crossings representation is an AQLR.

Moreover:

**Theorem 3** The wavelet zero-crossings representation is an inherently bounded AQLR.

4 Nonuniqueness

A representation \( Rf = \{ Vf, Tf \} \) is said to be unique, if the reconstruction set \( \Gamma(Rf) \) consists of exactly one element. We have the following uniqueness characterization for AQLR's.

**Lemma 1** Let \( Rf = \{ Vf, Tf \} \) be an AQLR. Then \( Rf \) is unique if and only if the kernel of the operator \( T \) is trivial, i.e. \( NT = \{ 0 \} \).

The proof is clear from topological considerations. Nevertheless, an elementary but constructive proof is given in [3].

This claim has some significant consequences. Using the above lemma, an algorithm which tests for uniqueness can be developed. Perhaps the most important consequence of Lemma 1 is the fact that uniqueness of the representation \( Rf \) is equivalent to uniqueness of the underlying irregular sampling \( T f \). In other words, in the unique case, all the information about the signal is already contained in \( T f \). Additional constraints \( Af > a \) are redundant. On the other hand, from the signal compression, understanding and interpretation point of view, it seems to be desirable that a little information would be specified explicitly by \( T f \) and as much as possible information about a signal should be described implicitly by \( Af > a \). Therefore, in our opinion, the most important and interesting features of AQLR’s appear in the nonunique case.

Using the previous lemma, we are able to show that:

**Theorem 4** A discrete dyadic wavelet maxima (zero-crossings) representation based on a discrete low pass filter \( H(\omega) \) is given. If \( H(\pi) = 0 \), \( J \geq 3 \), and \( N \) is a multiple of \( 2^J \) then there exists a sequence \( f \) which has a nonunique maxima (zero-crossings) representation.

As a universal example of nonuniqueness the following sequence is proposed.

\[
f(n) = \cos(2\pi \frac{n}{2^J}) \quad n = 0, 1, \ldots, N - 1.
\]  

(11)

Observe that the same sequence is proposed for all dyadic wavelet transforms and for both the maxima representation and the zero-crossings representation.

The example of the nonunique maxima representation is described in [1, 2]. Now, let us consider the zero-crossings representation based on the wavelet transform defined in [4]. Let \( J = 5 \) and \( N = 256 \). Consider two sequences:

\[
f(k) = \cos \left( \frac{2\pi k}{32} \right)
\]  

\[
f_1(k) = \cos \left( \frac{2\pi k}{32} \right) + 0.1 \cdot \sin \left( \frac{2\pi k}{16} \right).
\]

It can be shown that they have the same zero-crossings representation. Figure 1 describes these sequences, while Figure 2 gives their first level wavelet transforms (other levels wavelet transforms are very similar).
Let $\Gamma_p$ be the corresponding reconstruction set. In general, the distance, $d$, between two reconstruction sets, $\Gamma$ and $\Gamma_p$, is defined as:

$$d(\Gamma, \Gamma_p) \triangleq \sup \{ \| \gamma - \gamma_p \| : \gamma \in \Gamma, \gamma_p \in \Gamma_p \}.$$ 

Observe, that for an inherently bounded AQLR, $d(\Gamma, \Gamma_p)$ is always finite. The measure of the perturbation in the reconstruction set is the difference between $d(\Gamma, \Gamma_p)$ and the size of $\Gamma$ which is defined as follows:

$$s(\Gamma) \triangleq d(\Gamma, \Gamma) = \sup \{ \| \gamma - \gamma_p \| : \gamma, \gamma_p \in \Gamma \}.$$ 

$s(\Gamma)$ and $d(\Gamma, \Gamma_p)$ describe the largest possible Euclidean norm of a reconstruction error, from the original representation and from a perturbed one, respectively.

One remark is in order. In general, for an arbitrary $\Delta(Tf)$, the associated reconstruction set may be empty and then $d(\Gamma, \Gamma_p)$ would not be defined. In the sequel, it is assumed that this problem is treated by a reconstruction algorithm and hence $\Delta(Tf)$ yields a nonempty $\Gamma_p$. In this case, the following Lipschitz condition is satisfied.

**Theorem 5** For all inherently bounded AQLR, there exists $K > 0$ such that:

$$d(\Gamma, \Gamma_p) \leq K \cdot \| \Delta(Tf) \| + s(\Gamma).$$ 

**Conclusions**

The described theoretical results about uniqueness and stability are new. In our opinion, the most significant contribution of this work is to create a framework to define and analyze a wide family of representations. Important examples are generalizations of a basic maxima representation obtained by using only a subset of local extreme points. Their properties are the subject of the ongoing research.

**References**


A Simple and Reliable Phase Unwrapping Algorithm

Eduard KRAJNIK
Faculty of Electrical Engineering, Czech Technical University, 166 27 Prague 6, Czechoslovakia

Abstract

A great majority of phase unwrapping algorithms are based on the observation that the values of the unwrapped phase $\Phi$ differ at each frequency $\omega_k$ from the principal value of the phase $\text{ARG}(X(e^{j\omega}))$ by an integer multiple of $2\pi$: $\Phi(\omega_k) = \text{ARG}(X(e^{j\omega_k})) + 2\pi n_k$ and determine the correct value of $n_k$. We propose a novel method in which the unwrapped phase values are computed recursively: $\Phi(\omega_{k+1}) = \Phi(\omega_k) + \Delta\Phi_k$. We first ensure that the frequencies $\omega_k$ and $\omega_{k+1}$ are such that $|\Delta\Phi_k| < \pi$ (a sufficient condition for this is given) and the phase increment $\Delta\Phi_k$ can be then unequivocally expressed by the principal value of the phase of the quotient of $\Phi(\omega_{k+1})$ and $\Phi(\omega_k)$. Thus, neither the detection of discontinuities nor the addition of multiples of $2\pi$ are needed. If the fast Fourier transform is used to evaluate $\Phi(\omega_k)$ a sufficient value of its size $N$ is shown to be any integer not less than $2 \max_{\omega \in [0,2\pi]} |\Phi(\omega)|$.

Introduction

Signal processing in the cepstral domain has proven to be useful in many deconvolution problems (seismology, speech, radar, sonar and others). The processing requires the computation of the cepstrum which is usually performed using the discrete Fourier transform and the complex logarithm function. In this process we encounter the problem of correctly choosing the phase values so that they will represent samples of the continuous phase function - the problem which is commonly called phase unwrapping. Though "exact" methods for phase unwrapping have been published recently [2, 3], the most popular method still seems to be Tribollet's adaptive integration algorithm [4], which gives an estimation to the unwrapped phase. The accuracy of the method depends critically on the FFT size $N$ and the author claims that "it is not possible to estimate a priori how large the FFT size $N$ should be in order to accurately unwrap the phase".

We propose a modification in the algorithm which not only enables a priori estimate of the FFT size $N$ but also increases the accuracy up to the precision of FFT.

The algorithm

Let $x_n$ be a sequence of a finite length $L$ and $X(e^{j\omega})$ its Fourier transform; let $X(e^{j\omega}) \neq 0$ for all $\omega \in [0,2\pi]$. Then there exists a continuous function (unwrapped phase) $\Phi(\omega)$ such that

$$X(e^{j\omega}) = |X(e^{j\omega})| \exp(j\Phi(\omega))$$

Let $\omega_k = 2\pi k/N$, $k = 0, 1, \ldots, N-1$, be the frequencies where the values $\Phi_k = \Phi(\omega_k)$ are required. If we are able to choose $N$ such that the phase $\Phi$ never changes by more than $\pi$ between any two adjacent frequencies $\omega_k, \omega_{k+1}$, we can unequivocally compute the phase increment $\Delta\Phi_k$ between $\omega_k$ and $\omega_{k+1}$:

$$\Delta\Phi_k = \text{ARG}\left(\frac{X(e^{j\omega_{k+1}})}{X(e^{j\omega_k})}\right)$$

where $\text{ARG}$ denotes the principal value of the phase. The unwrapped phase samples are then computed recursively:

$$\Phi_{k+1} = \Phi_k + \Delta\Phi_k$$

Unlike Tribollet's algorithm, the phase values are not approximated and their accuracy depends only on the
accuracy of the values $X(e^{in\theta})$; the cumulative rounding error in the sequence $\Phi_k$ can easily be eliminated by the comparison of $\Phi_k$ and the nearest value of \( A_RG(X(e^{in\theta})) + 2\pi m_k \) for a few values of $k$. Also, neither the detection of discontinuities nor the addition of multiples of $2\pi$ are needed. Of course, these advantages have to be paid for; but the price is reasonable: we have to ensure that the phase $\Phi$ never changes by more than $\pi$ between any two adjacent frequencies $\omega_k$, $\omega_{k+1}$. This, in principle, can be achieved by a suitable choice of $N$. A sufficient value can be derived by using the Mean Value Theorem:

$$
\Phi(\omega_{k+1}) - \Phi(\omega_k) = \frac{d}{d\sigma} \Phi(\tau)(\omega_{k+1} - \omega_k)
$$

for some $\tau \in (\omega_k, \omega_{k+1})$. Hence

$$
|\Phi(\omega_{k+1}) - \Phi(\omega_k)| \leq M |\omega_{k+1} - \omega_k| = 2\pi M/N
$$

where

$$
M \geq \max_{\omega_k \in [0,2\pi]} |\Phi'(\omega)|
$$

The condition $|\Phi(\omega_{k+1}) - \Phi(\omega_k)| < \pi$ thus will be satisfied for any $N > 2M$ and the problem is reduced to finding an upper bound $M$ of the phase derivative $\Phi'$, preferably $\max_{\omega_k \in [0,2\pi]} |\Phi'(\omega)|$. The numerical computation of a good approximation to the global maximum of a function is difficult in general; it is impossible to find the maximum with a prescribed tolerance in a finite number of function evaluations, unless we have some a priori information about the function. Efficient algorithms do exist, however, for unimodal functions or $C^2$ functions provided an upper bound of the second derivative is known [4].

Let us now investigate the maximum of the phase derivative $\Phi'$. We express the phase derivative in a similar way as in [1]. Setting $y_k = nx_k$ and denoting $Y(e^{i\omega})$ the Fourier transform of $y_k$, we have

$$
\Phi'(\omega) = \frac{X_R(e^{i\omega})Y_R(e^{i\omega}) + X_I(e^{i\omega})Y_I(e^{i\omega})}{X_R^2(e^{i\omega}) + X_I^2(e^{i\omega})}
$$

where

$$
X(e^{i\omega}) = X_R(e^{i\omega}) + jX_I(e^{i\omega}) \quad \text{and} \quad Y(e^{i\omega}) = Y_R(e^{i\omega}) + jY_I(e^{i\omega})
$$

After a rearrangement we get

$$
\Phi'(\omega) = \sum_{k=0}^{L} \left| \sum_{n=1}^{L} k x_n x_k \cos(k - l) \omega \right| + \sum_{k=0}^{L} \left| \sum_{n=1}^{L} k x_n x_k \cos(k - l) \omega \right| = \frac{\sum_{k=0}^{L} a_k \cos k \omega}{\sum_{k=0}^{L} b_k \cos k \omega} = \frac{A(\omega)}{B(\omega)}
$$

where

$$
a_k = \sum_{n=1}^{L} n x_n^2, \quad b_k = \sum_{n=1}^{L} x_n^2
$$

$$
a_k = \sum_{n=1}^{L} (2n + k) x_n x_{n+k} \quad \text{for} \ k = 1, \ldots, L
$$

$$
b_k = 2 \sum_{n=1}^{L-k} x_n x_{n+k} \quad \text{for} \ k = 1, \ldots, L
$$

Since $\Phi'$ is the quotient of two trigonometric polynomials $A$, $B$, it is not a unimodal function and though it is a $C^2$ function, a reasonable upper bound of its second derivative cannot be obtained. We therefore find

$$
M_B = \min_{\omega \in [0,2\pi]} |B(\omega)| \quad \text{and} \quad M_A = \max_{\omega \in [0,2\pi]} |A(\omega)|
$$

using the algorithm globmin given in the Appendix. We note that globmin can be used to find both a minimum and a maximum of a function since max $f(t) = -\min(-f(t))$. The upper bounds of the second derivatives of $A$ and $B$ required in globmin are taken $\max k^2 |a_k|$ and $\sum k^2 |b_k|$ respectively. Then we select $N > 2M_A/M_B$.

If $X$ has a zero extremally close to the unit circle resulting $N$ may be unacceptably large. In such a case, we face a problem whose numerical solution will always be dubious and certain compromises are necessary. A detailed discussion of this case will be given in [5].

The presented method gives a sufficient condition for simplification of phase unwrapping. The accuracy of the result is limited only by the accuracy of FFT.

**Appendix**

For the reader's convenience we present a minimization algorithm globmin with guaranteed convergence [4]. Globmin returns the global minimum of the external function func(x:real): real defined on [a, b] and the point x where the function attains its minimum; c is an initial guess at x (any value in [a, b] will do), m is an upper bound of the second derivative of func on [a, b], the constants macheps, c, t are the relative machine precision and tolerances respectively; it is assumed that func(x) is computed with an absolute error less than e and the minimum is returned with an absolute error less than t + 2e.
FUNCTION globmin(a, b, c, m: real; 
var x: real): real;

LABEL 1;

CONST macheps = 1e-15;
e = 1e-12;
t = 1e-7;

VAR y0, y1, y2, y3, yb, y, h: real;
a0, a2, a3, d0, d1, d2, r, s: real;
z0, z1, z2, m2, q, p, q: real;

BEGIN
a0 := b;
y0 := func(b);
yb := y0;
y := y2;
a2 := a;

IF y < y THEN y := y0 ELSE x := a;

IF (m > 0) AND (a < b) THEN BEGIN
m2 := 0.5*(1+16*macheps)*m;
IF (c < a) OR (c > b) THEN
  c := 0.5*(a+b);
y1 := func(c);
d0 := a2-c;
h := 9/11;

IF y1 < y THEN BEGIN x := c;
y := y1
END;

REPEAT
  d1 := a2-a0;
d2 := c-a0;
z2 := b-a2;
z0 := y2-y1;
z1 := y2-y0;
  r := d1*d1+z0*d0+z0*z1;
  q := 2*(d0+z1-d1*z0);
p := r;
  q := q;
  IF y >= y2 THEN BEGIN
    IF q*(r*(y-b)+2*z2*q-(y-y))+t) <
      z2*m2*r*(z2+q-r) THEN
      BEGIN
        a3 := a2+r/q;
y3 := func(a3);
        IF y < y THEN BEGIN x := a3;
y := y3
      END
    END;

END

END;

END;
ELSE a3 := r;
  IF a3 < r THEN a3 := r;
  IF a3 >= b THEN BEGIN a3 := b;
y3 := yb
END
ELSE y3 := func(a3);
  IF y3 < y THEN BEGIN x := a3;
y := y3
END;

END;
d0 := a3-a2;
  IF a3 > r THEN BEGIN
    p := 2*(y2-y3)/(m*d0);
    IF (abs(p) < (1+9*macheps)*d0) AND
        (0.5*m2*(d0+0.01*p)> (y2-y)+(y3-y)+2*t) THEN BEGIN
      a3 := 0.5*(a2+a3);
h := 0.9*h;
goto 1
END
END;

END;
a0 := c;
c := a2;
a2 := a3;
y0 := y1;
y1 := y2;
y2 := y3
UNTIL a3 >= b;

END;
globmin := y;
END;

References


2-D FFT Algorithms with Parallel Bit Reversion on Hypercube Computers

G. Angelopoulos  I. Pitas

Department of Electrical Engineering, University of Thessaloniki, Thessaloniki 540 06, GREECE

This paper discusses the parallel computation of the 2-D Discrete Fourier Transform by using three different Fast Fourier Transform (FFT) algorithms. The examined algorithms are: Row-Column FFT, Vector Radix FFT and Polynomial Transform FFT. Hypercube computers with non-shared memory are considered. The proposed implementations allow both numeric computations and the bit-reversion operation to be performed in parallel.

1. INTRODUCTION

High performance parallel computers is an interesting alternative to the serial computation of 2-D FFTs. The RC FFT is usually proposed for parallel 2-D DFT calculation [4,5,6,7]. In [5,6] transposition of the data after the row FFTs is proposed. A RC FFT on a Transputer-based Perfect Shuffle machine is presented in [4]. Another parallel multidimensional FFT on d-dimensional Perfect Shuffle permutation network is presented in [8]. A parallel version of the VR FFT in a Cube network of PEs is presented in [6]. Parallel implementations of PFFTts have not been reported, at least to the authors’ knowledge. In this paper we propose suitable implementations of the above mentioned 2-D FFTs for parallel systems with non-shared memory and hypercube topology. The proposed implementations allow both numeric computations and the bit-reversion operation, which is required in these FFTs, to be performed in parallel.

The 2-D Discrete Fourier Transform of an $N_1 \times N_2$ -point digital signal $x(n_1, n_2)$, $0 \leq n_1 < N_1$, $0 \leq n_2 < N_2$, is defined by the equation:

$$X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} x(n_1, n_2) W_{N_1}^{n_1 k_1} W_{N_2}^{n_2 k_2} \quad (1)$$

$$W_{N_i} = \exp(-j2\pi/N_i), \quad i = 1, 2.$$  

In most practical cases $N_1 = N_2 = N = 2^p$.

The 2-D DFT of a $N \times N$ -point sequence defined in equation (1) can be computed in two successive steps which consist the Row-Column (RC) FFT [1]:

$$X'(n_1, k_2) = \sum_{n_2=0}^{N-1} x(n_1, n_2) W_{N}^{n_2 k_2} \quad (2)$$

$$X(k_1, k_2) = \sum_{n_1=0}^{N_1-1} X'(n_1, k_2) W_{N_1}^{-n_1 k_1} \quad (3)$$

Alternatively, the 2-D DFT can be calculated by working both along columns and rows simultaneously. The decimation in time version of the Vector Radix $2 \times 2$ (VR) FFT is derived by expressing the $N \times N$ -point DFT in terms of four $N/2 \times N/2$ -point DFTs [1]. This procedure can be repeated until we reach transforms of size $2 \times 2$. The resulting VR FFT algorithm requires 25% less complex multiplications than the RC FFT.

The DFT of a 2-D $N \times N$ -point sequence defined in equation (1) can also be calculated by using the following steps:

$$X(n_z) = \sum_{n_3=0}^{N-1} x(n_1, n_2) W^{-n_3 n_z} \quad (4)$$

$$\tilde{X}(2k_1+1)k_2(z) = \sum_{n_1=0}^{N-1} X(n_1(z)) z^{2n_1 k_1} \mod (z^N + 1) \quad (5)$$

$$X[(2k_2 + 1)k_1, k_2] = \tilde{X}(2k_1+1)k_2(z) \mod (z - W^{2k_2 + 1}) \quad (6)$$

where $W = \exp(-j\pi/N)$. These relations represent the Polynomial Transform FFT (PTFFT) [2]. Relation (4) describes a multiplication of $x(n_1, n_2)$ by $W^{-n_3}$, followed by a polynomial transform of length $N$ with root $z^2$. Equation (6) is equivalent to a multiplication by $W^{n_2}$ and an 1-D DFT. A final unscrambling step is required, because the indices $(2k_2 + 1)k_1$ in (6) are scrambled. After this brief introduction to the 2-D FFT algorithms we proceed to their parallel implementation on a hypercube computer.
2. 2-D FFTS ON HYPERCUBE COMPUTERS

A hypercube computer of dimension p consists of \( P = 2^p \) PEIs located at the corners of the p-dimensional cube [3]. The edges of the hypercube represent bidirectional communication links. Each PE has \( p \) links. Any two PEIs are at most \( p \) links apart. Each PE is labeled by a number from 0 to \( P - 1 \) so that two PEIs are connected if and only if the \( p \)-bit binary representations of their labels differ only in one bit. Furthermore, each of the \( p \) dimensions is characterized by a unique bit position in the binary representation of the PE labels. Figure 1 shows a hypercube computer with \( P = 16 \) PEIs. Usually, a parallel machine is connected to a host computer via PE 0 for input/output operations. In the case of the 2-D FFTs, the distribution of the data to the PEIs can be performed by embedding a pseudo binary tree on the hypercube [6]. The same procedure holds for the gathering of the results into the PE 0.

RC FFT. Let us consider a hypercube with \( P \) PEIs and the binary representation of their labels \((l_0, l_1, \ldots, l_p)\). A 2-D signal of size \( N \times N \), where \( N \geq 2P \), can be divided into \( 2P \) stripes of size \( N/(2P) \) rows. These stripes are labeled with \((p + 1)\) bit labels and are distributed to the local memories of the PEIs. Each PE stores on its local memory the two stripes labeled by \( i \) and \( P + i \). According to the distribution of the signal described above, the computations of the FFTs of the rows can be performed in parallel without any interprocessor communication. Column DFT computation requires communications among the PEIs, because each column is distributed on \( P \) PEIs. The proposed algorithm has two phases. In the first phase, the first \( p = \log_2 P \) stages of the 1-D FFTs are computed. After finishing each of these stages, data exchange is required that is done in pairs of PEIs. Two PEIs whose labels differ only in their \( l_0 \) bit of their binary label representation \((l_0, l_1, \ldots, l_p)\) exchange data after the first stage. These two PEIs are one link apart. The PE which has its \( l_0 \) bit equal to zero exchanges its second stripe with the first stripe of the other member of the pair. After these communications, which are done in parallel in all PEIs, the calculation of the butterflies of the second stage is performed in parallel. This process is continued for the rest of the stages of the first phase. In the second phase, the last \( \log_2(N/P) \) stages of the radix-2 1-D FFT algorithm are performed without communications. This version of the parallel RC FFT algorithm requires the transfers of \((N^2/2) \log_2 P\) data items that are performed in \( p = \log_2 P \) steps, one in each dimension of the hypercube.

DFT output scrambling along the horizontal direction can be performed in parallel at each PE. The unscrambling along columns can be performed in parallel in two steps. In the first step, each PE performs the bit-reversed ordering of each column segment residing on this PE. In the second step, each PE performs a bit-reversed ordering of its label \((l_0, l_1, \ldots, l_p)\). Thus, the logical labels of the PEIs change and each PE with new label \( i = (l_0, \ldots, l_p-1, l_p) \) holds the output data rows \( k_i \) for which \( i = k_i \mod P \).

VR FFT. Let us consider a hypercube with \( P \) PEIs, where \( P \) is a power of 4, and a 2-D signal of size \( N \times N \), where \( N \geq 2P^{1/2} \). Such a signal can be divided into \( 4P \) square blocks of size \((N/2P^{1/2}) \times (N/2P^{1/2})\) that cover the entire 2-D signal in a row-wise manner, as can be seen in Figure 2. These blocks are labeled with \((p + 2)\) bit labels. The \( p \)-bit labels \((l_0, l_1, \ldots, l_p)\) of the PEIs can be split into labels of the form \((r_p, r_{p-1}, \ldots, r_2, c_0, c_1, c_2)\). For simplicity, these labels can be written as \((r_p, r_{p-1}, \ldots, r_2, c_1, c_0)\), where \( q = p/2 \). The distribution of the data into the PEIs is done by storing the four square blocks \((x_{r_0}, \ldots, r_2, x_{c_0}, \ldots, c_2)\) on the PE \((r_{p-1}, r_{p-2}, c_1, c_0)\), where \( x \) denotes the "don’t care" bit value. Figure 2 shows the partitioning of an \( N \times N \) signal into \( 4P \) = 16 square blocks and the associated labeling.

The proposed algorithm has two phases. In the first phase, the first \( q = p/2 = \log_2 P \) stages of the VR 2 \times 2 FFT algorithm are performed. After finishing the computations in each of these stages, an exchange of data among PEIs is required. The exchange of data is done among groups of PEIs. Each group consists of four PEIs. In the first stage of this phase, four PEIs are clustered in the same group if their labels differ only in their \( r_{q-1} \) and \( c_{q-1} \) bits. There are \( 4/P \) such groups in a hypercube having \( P \) PEIs. The four PEIs of a group are interconnected in a 2-D sub-hypercube (square), as shown in Figure 3. They are distinguished by their \( r_{q-1} \) and \( c_{q-1} \) bits which form a binary number \( r_{q-1}c_{q-1} \) for each PE of a group. The range of label \((r_{q-1}, c_{q-1})\) is from 0 to 3. By labeling the four outputs of the butterflies with the labels \( a, b, c, \) and \( d \), the state of the data after the first phase of the algorithm is shown in Figure 3a. The goal of the communication is to gather the butterfly outputs having the same label to a single PE. Each of the four PEIs in a group retains one quarter of its data and sends the three quarters of its data to the three other members of its group. This is achieved in two steps. In each step, exchange of data is performed through the links of a specific dimension of the 2-D hypercube that the four PEIs construct. In the first step, exchange of data is performed between processors which have the same \( r_{q-1} \) bit label and different \( c_{q-1} \) bit label. These processors are characterized by a label of the form \( r_{q-1}x \). In the second step, exchange of data is performed between processors which have the same \( c_{q-1} \) bit label and differ
ferent $r_{p,q}$ bit label, that is a label of the form $x_{p,q}$.  These two steps are shown in Figures 3b and 3c respectively. In the entire hypercube, the clustering of the PEs into groups of four PEs in this stage is represented by the labels $(x_{p+2} ... r_0 x_{p+1} ... c_0)$. The PEs that exchange data in these two steps are characterized by the labels $(r_{p+1} r_{p+2} ... r_0 x_{p+1} ... c_0)$ and $(x_{p+2} ... r_0 x_{p+1} ... c_0)$ respectively. In this way, each of the four PEs in a group receives $(3/4)N^2/P$ data from the three other members of its group. However, $(1/4)N^2/P$ data pass through a PE in order to achieve data transfers between non adjacent (diagonally lying) PEs in a group. Thus, the total number of transferred data in the entire hypercube is $N^2$. After these communications, which are done in parallel in all PEs, the calculation of the butterflies of the next stage is performed in parallel. This process is continued for the rest of the stages of the first phase of the VR FFT algorithm with the rest of the bits of the PE labels. The clustering of the PEs into groups of four requires that $P$ is a power of 4. In the second phase, the last log$_2(N/P^{1/2})$ stages of the VR FFT algorithm are computed on the PEs without communications. The total amount of data transfers is $N^2 \log_2 P = (N^2/2) \log_2 P$ data items and they are performed in $p = \log_2 P$ steps, one in each dimension of the hypercube.

In the VR 2 x 2 decimation in time 2-D FFT algorithm, the output data must be unscrambled in bit-reversed order in both dimensions $k_1$ and $k_2$ of the 2-D signal, after the main (numerical) processing. The bit-reversion operation can be performed in parallel on a hypercube computer. Each PE performs the bit-reversed ordering of the data that are locally stored.

The four square blocks that each PE holds constructs a 2-D array of size $(N/P^{1/2}) \times (N/P^{1/2})$. All PEs working in parallel and independently perform the bit-reversion operation on both dimensions of their arrays. Subsequently, each PE labeled $(r_{p+1} r_{p+2} ... r_0 x_{p+1} ... c_0)$ performs a bit-reversed ordering on each split part of its label and the new label becomes $(r_{p+1} ... r_{p+2} ... r_0 x_{p+1} ... c_0)$. Thus, each PE, whose new row sub-label is $i = (r_{p+1} ... r_{p+2} ... r_0 x_{p+1} ... c_0)$, holds the elements of the output data rows $k_1$, for which $i = k_1 \mod P^{1/2}$. Similarly, each PE, whose new column sub-label is $j = (c_0 c_1 ... c_{p+1} ... c_0)$, holds the elements of the output data columns $k_2$, for which $j = k_2 \mod P^{1/2}$. Finally, the gathering of the results into the PE 0 is performed by embedding a pseudo binary tree on the hypercube [9].

PT FFT. A stripe partitioning of a 2-D signal into $2P$ stripes and the distribution of them into the PEs is assumed, as in the RC FFT. All multiplications by $W^{n \sigma}$ and $W^{n \eta}$ involved in the PTFFT can be performed in parallel without any communication. For the computation of the polynomial transform, communication among the PEs must be performed. The proposed algorithm has two phases. In the first phase, the first log$_2 P$ stages of the polynomial transform are computed. The communications required in these stages are the same to the ones needed in the RC FFT algorithm. Similarly, the total amount of the transferred data is $(N^2/2) \log_2 P$ data items and the data transfers are performed in $p = \log_2 P$ steps, one in each dimension of the hypercube. In the second phase, the last log$_2(N/P)$ stages of the polynomial transform are performed without communications. The FFTs of rows can be performed in parallel without any communication. The gathering of the results into the PE 0 can be performed by embedding a pseudo binary tree on the hypercube [9].

3. DISCUSSION

All three algorithms described in the previous section have the same communication load. Thus the deciding factor for the choice among the three algorithms is their computational complexity. The VR FFT algorithm has been implemented on a parallel machine, because of its efficiency over the RC FFT and its simplicity over the PTFFFT algorithm. It was implemented on the Telnet T-Node 16-32 machine in hypercube configuration by using HELIOS C programming language. This machine consists up to 32 PEs (transputers T800) and has reconfigurable architecture. The total time (processing plus communications) for $P = 1$, $P = 16$ PEs and the achieved speedups for various values of $N$ are shown in Table 1. The PE communications have been performed by using the read-write functions, that provide relatively slow interprocessor communication with large startup overheads. The total time includes the time required for the computation of the complex exponentials. As expected, the speedup increases with the signal size $N \times N$, because in this case the computational load is much larger than the communication load. Furthermore, in the case of large signal sizes, the effect of startup time in the communication time is reduced, because large data blocks are transferred in each communication operation.

Acknowledgements This work has been partially supported by ESPRIT PC project No4012 and by a project of the Greek Ministry of Technology (PENED 89).

References


### TABLE 1

<table>
<thead>
<tr>
<th>N</th>
<th>P = 1</th>
<th>P = 16</th>
<th>Speed Up</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>100</td>
<td>70</td>
<td>1.43</td>
</tr>
<tr>
<td>64</td>
<td>470</td>
<td>110</td>
<td>4.27</td>
</tr>
<tr>
<td>128</td>
<td>2210</td>
<td>240</td>
<td>9.21</td>
</tr>
<tr>
<td>256</td>
<td>9810</td>
<td>860</td>
<td>11.41</td>
</tr>
<tr>
<td>512</td>
<td>43970</td>
<td>3510</td>
<td>12.53</td>
</tr>
</tbody>
</table>

#### Figure 1: Hypercube with 16 processors.

#### Figure 2: Partitioning of a 2-D signal in 16 square blocks. The numbers in the boxes denote square block labels, \( r_1 r_2 c_1 c_2 \) is their binary representation.

#### Figure 3: Data exchange in VR FFT among four PEs; (a) Initial state; (b) First exchange; (c) Second exchange.
Time-warped Polynomials for Signal Representation and Coding

Wilfried Philips

Laboratory for Electronics, St.-Pietersnieuwstraat 41, B9000 Gent, Belgium
E-Mail: philips@lem.ruag.ac.be Tel: 32-91-64.3835 Fax: 32-91-64.35.94

This paper introduces a new class of discrete, orthogonal transforms, whose base functions are time-warped polynomials. The general properties of the base functions are described. The class generalizes at least three well-known transforms. A data compression application on Electrocardiograms based on one of the new transforms is presented. At a given signal quality, the new transform allows a compression more than twice that of the DCT.

1 Introduction

Orthogonal transforms are widely used in signal processing. Although infinitely many of them exist, only a few of them are used in practice [1]. These include the Discrete Cosine Transform (DCT) [1], the Walsh-Hadamard Transform, etc. The transform matrices of the Karhunen-Loève Transform (KLT) and the Singular Value Decomposition (SVD) are not fixed, but they depend on the signal's statistics and the signal's data respectively [1]. These transforms can only be described by enumerating all of their base functions, or by storing the data from which they were derived. The large overhead involved in describing the transform limits the practical usefulness of these adaptive transforms.

This paper describes a class of transforms that are completely determined by specifying a single function \( f(x) \). The base functions of these transforms are time-warped polynomials, i.e., functions of the form \( P_n(f(x)) \), where \( P_n(f) \) is a polynomial in \( f \). By parametrizing the warping function \( f(x) \), transforms depending on only a few numbers can easily be devised. By changing the parameters that determine \( f(x) \), the transform can be adapted to a given signal. Since the overhead is relatively small, it is even possible to vary the transform parameters in time.

The new class of base functions is introduced in section 2. It includes the DCT and two different polynomial transforms. Section 3 describes the main properties of the time-warped base functions and gives some guidelines for choosing the right transform for a given application.

In section 4, a data compression method for Electrocardiograms (ECGs) based on a specific time-warped transform is compared to a DCT compression scheme. It is shown that for the same quality, the Squared Error (RMSSE), the compression factor of the time-warped transform is more than twice that of the DCT.

2 The New Transform

Let \( S = \{ a_i = -1 + 2i/L : i = 0 \ldots L \} \) be a set of sample points on the real line. The inner product of two functions \( g(x) \) and \( h(x) \) on \( S \) is defined as \( g \cdot h = \sum_{a_i} g(a_i)h(a_i) \). The norm of \( g(x) \) is \( ||g(x)|| = \sqrt{g \cdot g} \). A function \( \phi_n(x) \) on \( S \) is a complete orthogonal base on \( S \) if the \( \{ \phi_n(x) \} \) are mutually orthogonal. Hence, if they satisfy \( \phi_m \cdot \phi_n = \delta_{m,n} \) for all \( m, n \) for which \( 0 \leq m, n \leq L \).

Let \( s(x) \) be a function on \( S \) and \( s_N(x) = \sum_{n=0}^N A_n \phi_n(x) \). Due to the completeness of \( B, s_N(x) = s(x), \) for all \( x \in S \). On the other hand, for \( N < L, s_N(x) \) is that combination of the base functions \( \phi_n(x) \), \( n = 0 \ldots N \) that minimizes the RMSSE \( \sqrt{\sum |s(x) - s_N(x)|^2} \) between \( s(x) \) and \( s_N(x) \) on \( S \).

If the set of base functions \( \phi_n(x) \), \( n = 0 \ldots L \) is suitably chosen, the approximation \( s_N(x) \) will be close to \( s(x) \) on \( S \), even if \( N \) is much smaller than \( L \). In that case, the coefficients \( A_n \), \( n = 0 \ldots N \) are a sufficient, though approximate, representation of the signal \( s(x) \). This is exploited in data compression applications.

In adaptive transform coding, the orthogonal base used for representing the signal \( s(x) \) depends on \( s(x) \) itself. If the base is matched to a given signal, fewer coefficients are needed to reduce the RMSSE for this signal to a given level. On the other hand, the overhead data needed to describe which transform was used by the coder, can significantly reduce the net compression factor. This overhead is small for transforms that depend on a few parameters only. Such transforms could be devised in several ways. As far as the author knows, the subject has not yet been investigated in detail in the signal processing literature. The time-warped polynomial bases that are introduced below, can be represented by a few parameters only. They can be computed at high speed and generalize some well-known transforms. This makes them very promising for adaptive coding.

Let the function \( f(x) \) be defined on \( S \). It is easily shown that the powers \( f^n(x), n = 0 \ldots L \) of \( f(x) \), are linearly independent on \( S \), if \( f^i(x) \neq f^j(x) \) when \( i \neq j \). In the following, \( f(x) \) is a strictly increasing, continuous function with \( f(-1) = 1 \) and \( f(1) = L \). For such an \( f(x) \), the independence property holds trivially. In practice, \( f(x) \) is also slowly varying.

In general, the non-orthogonal functions \( f^n(x) \) are not very suitable for signal representation. As described above, a set \( B \) of orthogonal base functions \( \phi_n(x) \) should be used instead. These functions \( \phi_n(x) = P_n(f(x)) \) are obtained by Gramm-Schmidt Orthogonalization (GSO) of the functions \( f^n(x) \) in the order \( n = 0, 1, \ldots L \).

Since \( P_n(f) \) is a polynomial of precise degree \( n \) in \( f \), the orthogonal base functions are called time-warped polynomials. It can be shown that the orthogonal functions \( P_n(f(x)) \) satisfy a three-term recursion, which is similar to the recursion for orthogonal polynomials [3, p. 42].
Since the recursion is much faster than GSO, it should be used to generate the orthogonal base. A fast transform generation is essential in adaptive transform coding, where the orthogonal base must be generated at both the coder (transmitter) and the decoder (receiver). In such applications, the transform might even vary in time.

The orthogonal base $B$ is completely determined by the warping function $f(x)$. It can be shown that $B$ is also obtained for any warping function $f'(x) = af(x) + b$, where $a \neq 0$ and $b$ are constants. In practice, a function that can be completely specified by a few parameters only will be chosen for $f(x)$. The orthogonal base $B$ is then completely determined by the value of these parameters.

For specific choices of $f(x)$, two kinds of Discrete Legendre Transforms (DLT1 and DLT2) and the Discrete Cosine (DCT) are obtained:

- DLT1 [4] $f(x) = x$.
- DLT2 [5] $f(x_i) = x_i$, where $x_i, i = 0, \ldots, L$ are the zeros of the (ordinary) Legendre polynomial $P_{L+1}(x)$ of degree $L$, where it is assumed that $x_i > x_j$ if $i > j$.
- DCT [1] $f(x_i) = \cos((\pi + 1/2 \pi) x_i / N)$, where $N = L + 1$.

The last two transforms are usually defined in different ways, but the definitions given here are in agreement with the usual ones.

3 General Properties

The theory of orthogonal polynomials [3, p. 44] predicts that $P_n(f(x))$ has exactly $n$ zeros in the real interval $[-1, 1]$, if the conditions of the previous sections are fulfilled. Therefore, the base functions of all degrees oscillate in $[-1, 1]$.

In general, the following approximation is valid [6]:

$$P_n(f(x)) \approx \sqrt{2/(\pi L)} \sqrt{\frac{d}{dx} / |\sin \theta|} \cos(n \theta + \Gamma(\theta)), \quad (1)$$

where $\theta = \arccos(f(x))$ and $\Gamma(\theta)$ depends only upon $f$ and $\theta$.

The previous equation is valid if $n$ is small compared to the number of sample points $L + 1$, because then the discrete orthogonality relation approximates a continuous one. On the other hand $n$ should not be too small, because (1) is really an asymptotic relation for large $n$. The precise range of $n$-values for which equation (1) holds, depends on $f(x)$ and is difficult to determine in general. Experiments using discrete Legendre Polynomials [7] suggest that equation (1) accurately predicts the behavior of $P_n(f(x))$ for $n$-values as low as 5 and as high as $L/2$. Also, eq. (1) holds exactly for the DCT.

Equation (1) shows that the base functions are approximately amplitude and frequency modulated sine waves. The amplitude $A(x)$ and the angular frequency $\omega(x)$ are given by

$$A(x) = \sqrt{2/(\pi L)} \sqrt{\frac{d}{dx} / |\sin \theta|} = \sqrt{2/(\pi L)} \sqrt{g(x)} \quad (2)$$

$$\omega(x) = \frac{d}{dx} \left[ n + \frac{d}{dx} \right] = \frac{d}{dx} |\sin \theta| = g(x) \left[ n + \frac{d}{dx} \right] \quad (3)$$

Both $A^2(x)$ and $\omega(x)$ are proportional to $g(x) = \frac{d}{dx} / |\sin \theta|$ if $n \gg |\frac{d}{dx}|$.

4 ECG Data Compression

![Figure 1: Some base functions and their corresponding $g(x)$-functions.](image)

Figure 1 shows the base functions of degree 20, for different choices of $g(x)$. These examples suggest that $g(x)$ accurately predicts the local base function frequency, even for non-classical transforms.

The previous results can be used to qualitatively determine the best choice of $f(x)$, for any particular application (see next section).

### 4 ECG Data Compression

The top curves of figures 2 and 3 show original ECG signals. These signals were sampled at 500 Hz and at a resolution of 10 bits per sample. By convention, the ECG waves are labelled P, Q, R, S and T (see fig. 2). The ECG signal is highly non-stationary and quasi-periodic.

In the transform coding methods described in this article, the ECG signal is first segmented into intervals. In the following, $S$ represents one such interval. The signal $s(x)$ on $S$ is approximated by the weighted sum $s_N(x)$, as described in section 2. For best results, each interval should correspond to one ECG period [7]. Such an interval choice allows the coefficients $A_n$ to be coded efficiently, since they will only change slowly from one interval to another. The coefficients $A_n$ are quantized and coded using a variable word-length coding scheme. The same uniform quantizer is used for all coefficients. More detailed descriptions of the coefficient quantiser and encoder can be found in [7].

Different transforms, i.e., different sets of base functions $\phi_n(x)$, can be compared by computing the RMSE $\Delta_\phi$ or the Peak Error (PE) for specific values of $N$. Obviously, transforms producing a lower RMSE or a lower PE are to be preferred for data compression purposes. In this section, the DCT is compared to the DLT and to a third transform, that is called the Linear Spline-Warped Transform (LSWT). Its warping function $f(x)$ is a linear spline and is shown in fig. 4. Since the spline has unity slope in $[a, b]$, $f(x)$ is completely determined by the breakpoints $a$ and $b$ and the slopes $\alpha$ and $\beta$ in the intervals $[-1, a]$ and $[b, 1]$. In the following, $a$ and $\beta$ are chosen equal for simplicity. The breakpoints of $f(x)$ coincide with the end and the beginning of succeeding QRS complexes. Therefore, by choosing $\alpha > 1$, the amplitude and frequency of the base functions can be increased in the vicinity of the high-frequency QRS complexes.
The DCT is widely used in practice because its RMSE closely approximates the minimal one, for stationary signal statistics [1]. However, the ECG is a highly non-stationary signal and therefore it is to be expected that the DCT is not close to optimal, for the compression of ECG data. In the remainder of this article, it will be shown that the DLT1 is better suited for representing the non-stationary ECG signal.

For the DLT1, eq. (1) reduces to Laplace's approximation [7, 8], the amplitude becomes \( A(x) = \sqrt{\beta} \left( \frac{\pi L \sin \theta}{\sin \theta} \right) \) and the angular frequency \( \omega(x) = (n + 1/2) / \sin \theta \). Both \( A(x) \) and \( \omega(x) \) increase near the boundary of \( S \), i.e., for \( |x| = |\cos \theta| \) approaching 1. The base functions of the DCT, on the other hand, have a constant amplitude and frequency on the whole interval \( S \).

The non-stationary behaviour of the discrete Legendre polynomials qualitatively matches that of the ECG-signal in a single R-R interval. This is because the high-frequency, high-amplitude QRS-complexes are located on the boundaries of the R-R intervals, while the other waves, which generally have lower frequency and amplitude, are located away from these boundaries (see figures 2 and 3). Therefore, for the DLT1, \( S \) is (approximately) an R-R interval. Numerical results [7] show that for the DLT1, both the RMSE and the PE increase sharply if \( S \) deviates substantially from an R-R interval. For the DCT, the PE and RMSE are virtually independent of the interval alignment, because of the stationary characteristics of its base functions.

Figures 2 and 3 show some DCT and LSWT approximations. The (optimal) interval boundaries used, are indicated in the figures, while table 1 lists numerical values of the PE and the RMSE for these and other approximations. For the LSWT, \( a \) and \( b \) were chosen as described above. For a given ECG, the value of \( \alpha = \beta \) that minimized \( e^2 \) was used. This optimal value of \( \alpha \) is computed by a gradient-descent type algorithm and depends on both the order \( N \) and on the ECG being coded. In the table, \( CR \) is the compression ratio, i.e., the ratio of the number of bits in the original signal and the number of coefficients needed to store the quantized coefficients and some overhead data, such as interval lengths.

The LSWT clearly results in the highest compression and in the lowest RMSE. For instance, at a CR of 22.1, the RMSE for the LSWT is 16\( \mu \)V for the ECG of fig. 2. At a comparable RMSE (17\( \mu \)V) the CR for the DCT is only 8.1. The values of the PE are less consistent, since it is generally more randomly distributed. However, the PEs for the DLT1 and the LSWT are significantly lower than for the DCT.
Results for the ECG of figure 2 ($\Delta = 1/16$ $\mu$V, $a = -0.83$, $b = 0.83$, $\alpha = \beta = 2.7$)

<table>
<thead>
<tr>
<th>CR</th>
<th>RMSE</th>
<th>PE</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>DCT</td>
<td>DLT1</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>30</td>
<td>16.0</td>
<td>21.1</td>
</tr>
<tr>
<td>50</td>
<td>12.2</td>
<td>15.4</td>
</tr>
<tr>
<td>75</td>
<td>8.1</td>
<td>12.5</td>
</tr>
<tr>
<td>100</td>
<td>8.0</td>
<td>11.7</td>
</tr>
</tbody>
</table>

Results for the ECG of figure 3 ($\Delta = 1/64$ $\mu$V, $a = -0.75$, $b = 0.65$, $\alpha = \beta = 3.3$)

<table>
<thead>
<tr>
<th>CR</th>
<th>RMSE</th>
<th>PE</th>
</tr>
</thead>
<tbody>
<tr>
<td>N</td>
<td>DCT</td>
<td>DLT1</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>30</td>
<td>17.0</td>
<td>19.9</td>
</tr>
<tr>
<td>50</td>
<td>11.6</td>
<td>13.8</td>
</tr>
<tr>
<td>75</td>
<td>8.5</td>
<td>10.2</td>
</tr>
<tr>
<td>100</td>
<td>6.7</td>
<td>8.3</td>
</tr>
</tbody>
</table>

Table 1: Compression (CR), peak error (PE) and RMS error (RMSE) for the DLT1, the DCT, and the LSWT in function of the number of coefficients $N+1$. The quantization step $\Delta$ is the same for all transforms. The parameters of the LSWT warping function are listed.

Also, at very high compressions (e.g. for $N = 30$ in table 1) the PE of the LSWT is clearly better than that of the DLT1. For the ECG of fig. 2, the LSWT performs slightly worse as far as PE is concerned. However, a higher compression largely compensates the small difference (10%) in PE.

Numerical results for other ECG signals confirm that the LSWT leads to both a higher compression and a lower RMSE, for a given quantization step $\Delta$. Therefore, it is the method of choice for ECG data compression. The LSWT can be further improved by allowing different values for $a$ and $b$ and by optimizing the positions $a$ and $b$ of the spline break points. This has not yet been examined. A slight disadvantage is that some computations are needed to determine the optimal spline parameters. These computations are only needed at the coder however. Furthermore, the spline parameters can be optimized for a given class of ECG signals (instead of for a single signal). This will tend to lower the mean compression for a given RMSE, but the results will never be worse than for the DLT1, which is still far superior than the DCT.

5 Conclusion

The paper has introduced a new set of transforms, that includes well-known transforms such as the DCT. The transform base functions can be generated quickly and are fully determined by a single (parametrized) function. The main properties of the base functions have been described. The usefulness of the base functions has been demonstrated in an ECG data compression application.

Acknowledgement

The author is grateful to the Belgian National Fund for Scientific Research, for supporting this work financially.

References


Design of Signal-Matched Wavelets - Theory and Results

T. Greiner, E. Eberle, M. Pandit

University of Kaiserslautern, PO-Box 3049, W-6750 Kaiserslautern, Germany

Abstract: We propose a new method to design wavelets. These wavelets contain signal characteristics and are designed by determining the impulse responses of the wavelet digital filters by a linearized optimization process. The achieved wavelet filters fulfill the wavelet conditions and yield better results in texture analysis. The 'Eigenfilters' of Ade and a Karhunen-Loève related maximum variance criterion are the basis of the matching. The new filters are tested with the Brodatz textures.

1. Introduction

The wavelet transform is a relatively new tool in signal processing, which finds application in the analysis of non-stationary processes and the multiresolution decomposition of signals. In the latter application, which is the current context, orthogonal wavelets are employed for viewing images at various resolutions, thereby achieving a set of independent information levels.

2. Characteristics of the wavelet transform

We use the nomenclature of Mallat [8] and consider the wavelets \( \psi_{jk}(x) \) as the set of functions which are obtained by translation \( 'k ' \) and dilation \( 'j ' \) of the basic wavelet function \( \psi(x) \). For dyadic wavelets with a translation and dilation grid of a power of the basis two we get

\[
\psi_{jk}(x) = (\sqrt{2^j} \cdot \psi_{j/2}((x-2^j'k'))_{j,k} \tag{1}
\]

with \( \psi_{j/2}(x) = 2^j \cdot \psi(2^j x) \)

The function \( \psi(x) \) is so chosen, that the wavelet functions \( \psi_{jk}(x) \) are orthogonal. Well known types of orthogonal wavelet functions are: the classical Haar functions and the wavelets (based on the work of Lemarie [7]) suggested by Mallat [8] and those proposed by Daubechies [3]. Furthermore a scaling function \( \Phi(x) \) is introduced, which is related to \( \psi(x) \) in the frequency domain by a filter \( g(n) \)

\[
\Phi(\omega) = \sqrt{\frac{2}{\pi}} \cdot g\left(\frac{\omega}{2}\right) \cdot \Phi\left(\frac{\omega}{2}\right) \tag{2}
\]

whereby the tilde represents the Fourier transform.

The functions \( \psi_{jk}(x) \) and \( \Phi_{jk}(x) \) form a pair of orthonormal bases and are used to resolve a function \( f(x) \) into a 'bandpass' and a 'lowpass' component, respectively. In the context of digital signal processing the functions \( \psi_{jk}(x) \) and \( \Phi_{jk}(x) \) are used with the filter \( g(n) \) and a filter \( h(n) \). These filters form a Quadrature-Mirror-Filter (QMF) filter pair [11], whereby \( h(n) \) is a lowpass filter and \( g(n) \) is a highpass filter. The impulse responses themselves are determined on the wavelet and scaling functions using the following equations

\[
h(n) = \langle \phi_{2^{-1} \cdot (u)}, \phi((u-n)) \rangle
\]

\[
g(n) = \langle \psi_{2^{-1} \cdot (u)}, \phi((u-n)) \rangle
\]

\(< > \) denotes the scalar product.

Multiresolution decomposition consists in convolving the signal with the scaling filter \( h(n) \) and the wavelet filter \( g(n) \), respectively. This is done iteratively by downsampling both filtered signal components and filtering again the lowpass signal with \( h(n) \) and \( g(n) \). The signal components achieved by filtering with \( g(n) \) are called 'Detail'-signals. Hence, a procedure is given, which leads successively to a multiresolution representation of the analyzed signal.

The filters \( h(n) \) and \( g(n) \) must fulfill several conditions. These are the 'Finite Energy' condition, the 'Mean' condition, the 'Orthogonality' condition and the 'Regularity' condition. Daubechies [3] formulated these conditions for discrete filters with finite support.

The 'Orthogonality' condition is mathematically formulated by

\[
\sum_n g(n-2k) \cdot g(n-2 \cdot 1) = \delta_{k1} \tag{4a}
\]

and the 'Regularity' condition demands

\[
\sum_n n^k \cdot g(n) = 0 \quad k=0, 1, \ldots, N-1 \tag{4b}
\]

resulting, that \( H(z) \) has \( N \) zeros at \( z = (-1) \). This constraint implies the convergence of the iterated filter...
H(z) to a continuous limit function \( H_s(z) \) [9]

\[
H_s(z) = \prod_{p=0}^{\infty} H(z^{2^p})
\]

(3)

i.e. the corresponding scaling function. Since this condition is only necessary, but not sufficient, the convergence cannot be guaranteed. The strategy is to put as many zeros as possible to (-1). It depends on the application, whether a continuous limit function is not achievable, the filters designed can be useful for multiresolution decomposition.

2.1 Two-dimensional wavelets

In this paper we investigate the wavelet decomposition for the purpose of texture analysis. Hence 2-d filters \( H(\omega_1, \omega_2) \) are necessary. This is obtained by a separable 2-d filter \( H(\omega_1, \omega_2) \), i.e. the filter can be expressed by two 1-d filters \( H(\omega_1) \) and \( H(\omega_2) \). Additionally, a orientation selectivity is achieved. The results are a low-pass-lowpass filter \( H(\omega_1)H(\omega_2) \), two mixed filters \( H(\omega_1)G(\omega_2), G(\omega_1)H(\omega_2), \) and a high-pass-highpass filter \( G(\omega_1)G(\omega_2) \) building a pyramid structure with 4 components on each level.

3. Signal-matched analysing wavelet filters

All known orthogonal wavelet functions are designed on the basis of mathematical criteria, i.e. solving a dilation equation. Mallat solves this equation for the continuous case and uses a discrete approximation, while Daubechies proposes a solution for the digital filter case directly. Whereas the procedures of Mallat and Deubaechies do not take the signal to be analysed into consideration, the procedure given by Tewfik in [10] is signal oriented. Here, the wavelet is designed by minimizing the approximation error up to a given scale in the multiresolution decomposition and reconstruction process. Because of the complicated optimization process Tewfik restricts himself by using the upper bound of the error as the optimization criterion.

Each resolution level decomposes the image into a highpass and a lowpass part by filtering. Thus image characteristics should influence the filter pairs \( h(n_1), g(n_1) \) (horizontal filters) and \( h(n_2), g(n_2) \) (vertical filters). Hence, first we design the filters \( h(n) \) and \( g(n) \) instead of the wavelet function directly. Subsequently the wavelet function can be calculated from the corresponding filter pair as long as the filters converge.

In this paper we propose two approaches for designing wavelet filters for texture analysis which are matched to the textures to be analysed. The filters are designed by minimizing the mean square error between their impulse response and the impulse response of a given signal matched filter at the same time considering the wavelet conditions or by following a Karhunen-Loeve-Transform related maximum variance criterion.

4. Signal matched filters for texture analysis

In texture analysis, one basic filter design method is the 'Eigenfilter' method proposed by Ade [1].

This approach leads to a decomposition of the image into a series of uncorrelated images, each of which contains progressively less information. The impulse responses \( f_{\text{eigen}}(n_1,n_2) \) are the normalized eigenvectors \( y_i \) of the estimated covariance matrix \( C_{\text{eig}} \) of the image signal \( s(n_1,n_2) \), ordered according to declining eigenvalues arranged as 2-d filters.

The design of 'Eigenfilters' can be traced back to the discrete Karhunen-Loeve-Transform, which is a signal matched transform reducing the number of given components by a maximum variance extraction in the first few transformed components \( s'(n_1,n_2) \).

This approach is derived by multiplying the covariance matrix \( C_{\text{eig}} \) by a column vector \( y \), which can be used as a digital filter

\[
\mathbf{y}^T \mathbf{C}_{\text{eig}} \mathbf{y} = \sigma^2_{\text{eig}} - \max .
\]

(6a)

with the normalization constraint \( \mathbf{y}^T \mathbf{y} = 1 \)

(6b)

Again the vectors \( y \) are the normalized eigenvectors of \( C_{\text{eig}} \).

Aiming at separable solutions, the covariance matrix is estimated for the lines and rows of the image.

5. Optimization of wavelet filters under nonlinear constraints

The optimization process is formulated in the spatial domain. Because of the high nonlinearity of the optimizing process we use an iterative Lagrange-Newton approach [5] with a linearized equation system. A similar method has been used for designing Linear-Phase QMF-filters achieving the 'Perfect-Reconstruction' property [6].

Two approaches are possible. We start with an approximation of the impulse response of one 'Eigenfilter' \( f_{\text{eigen}}(n) \) by an optimized scaling filter \( h_0(n) \) or a wavelet filter \( g_0(n) \). Here two design steps are required; first the 'Eigenfilters' have to be designed and secondly, their impulse responses have to be approximated.

In contrast to this two step solution a scaling filter \( h_0(n) \) or a wavelet filter \( g_0(n) \) can be designed directly by maximizing the variance of the filtered image and fulfilling the wavelet conditions.

For both approaches the used wavelet constraints lead to highly nonlinear equations. We have to minimize
\[
L(\mathbf{\psi}_{\text{wavelet}}; \lambda) = Q(\mathbf{\psi}_{\text{wavelet}}) - \sum_i \lambda_i c_i(\mathbf{\psi}_{\text{wavelet}})
\]
(7a)

with

\[
Q(\mathbf{\psi}_{\text{wavelet}}) = \sum_n (\mathbf{\psi}_{\text{wavelet}}(n) - \mathbf{\psi}_{\text{eigen}}(n))^3
\]
(7b)

or \[
Q(\mathbf{\psi}_{\text{wavelet}}) = \mathbf{\psi}_{\text{wavelet}}^T \Sigma_{\text{Cov}} \mathbf{\psi}_{\text{wavelet}}
\]
(7c)

\[
\lambda_i - \text{Lagrange Multipliers}
\]

\[
c_i - \text{wavelet filter conditions}
\]

\[
\mathbf{\psi}_{\text{wavelet}} - \text{filter to be determined}
\]

\[
\Sigma_{\text{Cov}} - \text{Covariance matrix}
\]

The optimization problem can be solved iteratively, using a Taylor approximation of the nonlinear equations on each iteration step and truncating after the linear term.

For the optimization it is necessary to set

\[
\nabla L(\mathbf{\psi}; \lambda) = 0
\]
(8)

The Taylor approximation results in

\[
\nabla L(\mathbf{\psi}^{(k)}; \lambda^{(k)}; \Delta \lambda^{(k)}) = \nabla L(\mathbf{\psi}^{(k)}; \lambda^{(k)}) + \nabla^2 L(\mathbf{\psi}^{(k)}; \lambda^{(k)}) \Delta \lambda^{(k)}
\]
(9)

The iteration is formulated with

\[
\begin{bmatrix}
\Delta \psi^{(k)} \\
\Delta \lambda^{(k+1)}
\end{bmatrix} =
\begin{bmatrix}
\psi^{(k)} & -\Delta \lambda^{(k)} \\
-\psi^{(k)} & \lambda^{(k)}
\end{bmatrix}^{-1}
\begin{bmatrix}
\nabla L^{(k)} \\
\nabla^2 L^{(k)}
\end{bmatrix}
\]
(10a)

\[
\mathbf{\psi}^{(k+1)} = \mathbf{\psi}^{(k)} + \Delta \mathbf{\psi}^{(k)}
\]
(10b)

\[
\Lambda - \text{Hesse-Matrix of the Lagrange equation}
\]

\[
\nabla - \text{Jacoby-Matrix of the wavelet conditions}
\]

as long as \(\Delta \psi\) and \(\Delta \lambda\) exceed a specified level. The starting value are the impulse response of the selected 'Eigenfilter' for the first optimization approach or the corresponding normalized eigenvector for the second approach.

6. Results

The design proposed was carried out, filters which contain signal characteristics and additionally fulfill the orthogonal wavelet conditions were obtained. The filters were matched to the Brodatz textures No. 5 ("Expanded Mica") and No. 9 ("Grass lawn"). Both images were normalized getting the same mean and variance values. The designed filters are 6-tap filters with \(H(z)\) having 2 zeroes at \(z=\pm 1\).
Fig. 3. Iterated scaling function (Example 2)

resolution steps. The result of the first filter proposed show distinct differences in the two textures. Because of the separable solutions this is true both for filtering with $H(\omega_1)G(\omega_2)$ and $G(\omega_1)H(\omega_2)$. The second filter example reveals the greatest differences. Even though one does not obtain a continuous scaling function, the filter can be employed for texture analysis. The result of the third example exhibits the strongest resemblance with the classical case. By virtue of the maximum-variance criterion, more than 99% of the energy of the first texture is contained in the lowpass component. The analysis of other textures yields similar results.

<table>
<thead>
<tr>
<th>Pyramid level</th>
<th>Filter</th>
<th>Mallat</th>
<th>Daubechies</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st level</td>
<td>HH</td>
<td>99.02</td>
<td>97.68</td>
<td>97.11</td>
<td>92.76</td>
<td>99.32</td>
</tr>
<tr>
<td></td>
<td>HG</td>
<td>1.91</td>
<td>1.83</td>
<td>1.83</td>
<td>4.40</td>
<td>0.44</td>
</tr>
<tr>
<td></td>
<td>GH</td>
<td>2.01</td>
<td>0.24</td>
<td>0.94</td>
<td>2.22</td>
<td>0.20</td>
</tr>
<tr>
<td></td>
<td>GG</td>
<td>0.06</td>
<td>0.05</td>
<td>0.12</td>
<td>0.62</td>
<td>0.04</td>
</tr>
<tr>
<td>2nd level</td>
<td>HH</td>
<td>93.21</td>
<td>92.81</td>
<td>90.71</td>
<td>84.28</td>
<td>95.00</td>
</tr>
<tr>
<td></td>
<td>HG</td>
<td>3.70</td>
<td>5.04</td>
<td>5.10</td>
<td>8.36</td>
<td>3.18</td>
</tr>
<tr>
<td></td>
<td>GH</td>
<td>2.84</td>
<td>1.63</td>
<td>3.25</td>
<td>4.88</td>
<td>1.35</td>
</tr>
<tr>
<td></td>
<td>GG</td>
<td>0.25</td>
<td>0.52</td>
<td>0.94</td>
<td>2.48</td>
<td>0.47</td>
</tr>
<tr>
<td>3rd level</td>
<td>HH</td>
<td>87.11</td>
<td>80.30</td>
<td>78.20</td>
<td>70.07</td>
<td>85.91</td>
</tr>
<tr>
<td></td>
<td>HG</td>
<td>6.80</td>
<td>12.54</td>
<td>11.30</td>
<td>12.59</td>
<td>9.14</td>
</tr>
<tr>
<td></td>
<td>GH</td>
<td>4.78</td>
<td>5.42</td>
<td>6.83</td>
<td>11.48</td>
<td>3.50</td>
</tr>
<tr>
<td></td>
<td>GG</td>
<td>1.32</td>
<td>1.34</td>
<td>3.67</td>
<td>5.86</td>
<td>2.45</td>
</tr>
</tbody>
</table>

Table 1. Energy contents of the different approaches for Brodatz Texture No. 5 achieved with filters matched in comparison with the classical wavelet filters.

Table 2. Energy contents of the different approaches for Brodatz Texture No. 9 achieved with filters matched to Brodatz Texture No. 5.

<table>
<thead>
<tr>
<th>Pyramid level</th>
<th>Filter</th>
<th>Mallat</th>
<th>Daubechies</th>
<th>Example 1</th>
<th>Example 2</th>
<th>Example 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>1st level</td>
<td>HH</td>
<td>95.89</td>
<td>94.74</td>
<td>92.27</td>
<td>81.38</td>
<td>97.52</td>
</tr>
<tr>
<td></td>
<td>HG</td>
<td>2.11</td>
<td>4.00</td>
<td>3.88</td>
<td>8.37</td>
<td>1.21</td>
</tr>
<tr>
<td></td>
<td>GH</td>
<td>2.92</td>
<td>1.08</td>
<td>3.29</td>
<td>7.99</td>
<td>1.65</td>
</tr>
<tr>
<td></td>
<td>GG</td>
<td>0.07</td>
<td>0.10</td>
<td>0.56</td>
<td>2.26</td>
<td>0.12</td>
</tr>
<tr>
<td>2nd level</td>
<td>HH</td>
<td>87.90</td>
<td>84.27</td>
<td>81.10</td>
<td>66.60</td>
<td>85.76</td>
</tr>
<tr>
<td></td>
<td>HG</td>
<td>5.49</td>
<td>8.30</td>
<td>8.88</td>
<td>13.59</td>
<td>6.49</td>
</tr>
<tr>
<td></td>
<td>GH</td>
<td>6.05</td>
<td>5.64</td>
<td>7.93</td>
<td>13.66</td>
<td>5.33</td>
</tr>
<tr>
<td></td>
<td>GG</td>
<td>0.56</td>
<td>1.79</td>
<td>2.09</td>
<td>6.24</td>
<td>1.42</td>
</tr>
<tr>
<td>3rd level</td>
<td>HH</td>
<td>76.18</td>
<td>66.32</td>
<td>65.46</td>
<td>44.97</td>
<td>69.10</td>
</tr>
<tr>
<td></td>
<td>HG</td>
<td>9.96</td>
<td>13.20</td>
<td>13.85</td>
<td>24.91</td>
<td>11.28</td>
</tr>
<tr>
<td></td>
<td>GH</td>
<td>10.41</td>
<td>13.67</td>
<td>14.41</td>
<td>19.21</td>
<td>12.22</td>
</tr>
<tr>
<td></td>
<td>GG</td>
<td>3.43</td>
<td>6.81</td>
<td>6.34</td>
<td>10.91</td>
<td>6.80</td>
</tr>
</tbody>
</table>

7. References

A NEW FAST ALGEBRAIC CONVOLUTION ALGORITHM

Torsten MINKWITZ & Reiner CREUTZBURG

University of Karlsruhe, Faculty of Informatics, Institute of Algorithms and Cognitive Systems, P. O. Box 6980, D-7500 Karlsruhe 1, Germany

phone: (+49)-721-6084325; fax: (+49)-721-606803; e-mail:minkwitz@ira.uka.de; creutzbu@ira.uka.de

This paper introduces a new fast algorithm to do convolutions, which is a recurring problem in signal processing. The concepts used are fully algebraic. Therefore the algorithm does not produce roundoff errors. For modestly large signal lengths it is faster than any other previously described method. The calculations in our new method are done in the field of rational numbers and in finite field extensions of Q. The algorithm performs cyclic convolutions of lengths $N = 2^n$. This can be interpreted as a multiplication of polynomials modulo the polynomial $X^N - 1 \in \mathbb{Q}[X]$. Since $X^N - 1 = (X^{N/2} - 1)(X^{N/2} + 1)$ for $N$ even, the solution to be sought is a fast algorithm to do convolutions modulo the irreducible polynomial $X^{N/2} + 1 \in \mathbb{Q}[X]$. Precisely that has been achieved by our work. The algorithm introduced here calculates the cyclic convolution of two signals at a cost of $O(N \log(N))$ operations in $\mathbb{Q}$. The algorithm was successfully implemented and the measured execution times are given.

1 Introduction

The Chinese remainder theorem and existing FFT algorithms provide for mechanisms to develop divide and conquer strategies for algorithms in algebraic computing. Solving a problem of size $N = p \cdot q$ is reduced to solving the same problem $p$ times for size $q$. If this reduction and its inverse can be done rapidly, then a faster algorithm can be achieved. Convolutions are particularly well suited for such an approach, because discrete Fourier transforms have the convolution property [1-8,10-15].

Theorem 1 If $A, B$ are vectors of length $N$ over a field containing all $N$-th roots of unity, then if $\ast$ denotes the cyclic convolution and $\odot$ denotes pointwise multiplication:

\[ A \ast B = DFT^{-1}(DFT(A) \odot DFT(B)). \]

A convolution of size $N/2$ can always be computed by a cyclic convolution of size $N$. The higher order coefficients are set to zero and no actual overflow occurs. The FFT is a fast method of computing the DFT. It is based on the fact that the roots of unity form a cyclic group $\mathbb{C}_N$. Instead of taking the single group elements (roots of unity) of this large group, first the normal subgroup $\mathbb{C}_q (N = p \cdot q)$ is taken and the DFT is computed for the smaller group $\mathbb{C}_N/\mathbb{C}_q$. The cosets are now used instead of the single roots of unity. Thereafter the same algorithm is applied recursively for the cosets. Substituting the DFTs with FFT's in (Theorem 1) yields an efficient method to do convolutions. In real applications the coefficients will often be elements of the ring $\mathbb{Z}$ of integers. However, the maximum size DFT that can be done in $\mathbb{Z}$ is 2, since higher roots of unity are missing. Therefore the integers are embedded in the field $\mathbb{C}$ of complex numbers, which includes all roots of unity, or in a suitable number ring, which contains just the necessary roots of unity. The next section discusses some drawbacks of these approaches next to describing some alternative methods that avoid just those difficulties, but don't achieve a comparable efficiency for large problem sizes.

If the vectors are interpreted as polynomials, their convolution is a polynomial multiplication. This interpretation will be used in the later sections to introduce a new algorithm, which does not suffer from any of the disadvantages that the others have. It uses the concept of finite field extensions to use a FFT type divide and conquer strategy.

2 Previous Algorithms

A number of different methods have been designed to do convolutions [1-8,10-15]. If the coefficients are integers, two have been suggested in the introduction, which will be covered in this section. Besides the simple method which is taught in school for polynomial multiplications, there is also more efficient algorithms that don't require the concept of roots of unity. As an example the algorithm of Karatsuba is described here. Another algorithm that uses the convolution property of FFTs is the method described by Schönhage, which does convolutions over the ring of integers by mapping the vectors into large numbers and then doing fast multiplications of those large numbers.
2.1 Convolution in the field of complex numbers \( \mathbb{C} \)

As was mentioned in the introduction, the convolution property of the FFT yields a fast algorithm since the necessary roots of unity exist in \( \mathbb{C} \). The calculations are done with floating point arithmetic. The problem comes from the fact that there is no exact mapping of all roots of unity into floating point numbers on a machine [8]. Also the use of floating point arithmetic results in roundoff errors. The convolution can therefore be only of limited precision. Another disadvantage of this method is that complex arithmetic with floating point numbers is usually slower than simple integer arithmetic. Therefore the speedup achieved by this algorithm is somewhat diminished.

2.2 Convolutions using NTTs

Number theoretic transforms (NTT) were introduced as a generalization of the DFT over residue class rings of integers.

The NTT (Fermat number transform) uses the transform length \( N = 2^d \cdot 1 \), \( a = 2 \) as a primitive \( N \)-th root of unity modulo \( m \) and the Fermat number modulus

\[
m = F_d = 2^{2^d} + 1 = 2^b + 1, \quad (d \geq 0, \quad b = 2^d).
\]

Since the transform length is always a power of two, the well-known radix-2 FFT-algorithm can be applied to the NTT. Note that since \( a = 2 \), the multiplications with roots of unity are only shifts on a binary machine.

2.3 Karatsuba’s Algorithm

This method [10] uses a simple recursion scheme. The vectors are interpreted as polynomials. The formula for a polynomial multiplication of length \( N \) is:

\[
A \ast B = \sum_{i=0}^{2N-2} (\sum_{j+k=i} a_j \cdot b_k)X^i.
\]

If \( N \) is even, then the vectors \( A \) and \( B \) consist of upper halves \( A_1 \) and \( B_1 \), (coefficients \( a_i, b_i, N/2 \leq i < N \)) and lower halves \( A_0 \) and \( B_0 \), (coefficients \( a_i, b_i, 0 \leq i < N/2 \)). Now the multiplication formula is

\[
A \ast B = (A_1 \ast B_1)X^{N/2} + (A_0 \ast B_0)X^{N/2} + A_1 \ast B_0 + A_0 \ast B_1.
\]

This last formula shows how one multiplication of length \( N \) is reduced to only 3 multiplications of length \( N/2 \) instead of \( 4 \) in the previous formula. If \( N/2 \) is also even, the scheme can be applied recursively. Therefore for \( N \) a power of 2, instead of \( O(N^2) \), this method computes length \( N \) polynomial multiplications with \( O(N\log_2N) \approx O(N^{1.65}) \) elementary multiplications. For small \( N \), this method is faster than the others described here.

2.4 Schönhage’s Algorithm

Instead of explicitly doing a convolution, here the coefficients are mapped into a large integer and the convolution is done by multiplying those large integers. If the maximum coefficient is \( l \) bits long and the convolution is of length \( N \), then the numbers to be multiplied are:

\[
A' = \sum_{i=0}^{N-1} a_i \cdot 2^{(l + \log(N) + 1)},
\]

\[
B' = \sum_{i=0}^{N-1} b_i \cdot 2^{(l + \log(N) + 1)}.
\]

These numbers are multiplied by the efficient Schönhage–Strassen algorithm and the result of the convolution can be read from the result of the multiplication of numbers. The \( i \)-th coefficient is in the \( 2l + \log(N) + 1 \) bits beginning at the \( (i(2l + \log(N) + 1)) \)-th bit. For a fixed maximum size of resulting coefficients, this method has a cost of \( O(N \cdot \log(N) \cdot \log \log(N)) \) operations. The disadvantage it has in comparison with the algorithm introduced in the following section is the cumbersome arithmetic used in the Schönhage–Strassen algorithm [14, 15].

3 Convolution with finite field extensions

Instead of looking at direct convolutions, the rest of this paper will deal with cyclic convolutions of lengths \( N = 2^n \), \( n \) an integer, which are polynomial multiplications modulo the polynomial \( X^n - 1 \), [3, 4, 12]. These can be used to do direct convolutions of length \( N/2 \). The splitting of \( X^n - 1 \) yields (see also fig. 1)

\[
X^n - 1 = (x - 1) \prod_{i=0}^{n-1} (X^2 + 1).
\]

If a fast method is known to do convolutions modulo \( X^M + 1 \) for \( M \) a power of two, the Chinese remainder theorem suggests a fast algorithm to do cyclic convolutions.

\[
\begin{align*}
X^n - 1 & \\
\frac{X^n}{2} - 1 & \\
\frac{X^n}{4} + 1 & \\
\frac{X^n}{8} - 1 & \\
\ldots & \\
X - 1 &
\end{align*}
\]

Fig. 1. Structure for Chinese remaindering
3.1 Extending the field of rational numbers \( \mathbb{Q} \)

The field \( \mathbb{Q} \) of rational numbers does not include the necessary roots of unity to do cyclic convolutions of lengths greater than 2, [12]. Therefore \( X^M + 1 \) is irreducible over \( \mathbb{Q} \) for \( M \) a power of two and \( M > 2 \).

**Definition** 1  a) \( n \in \mathbb{Z} \), then \( [n]^* \) denotes the mapping to the next smaller power of two of \( n \).

b) \( n \in \mathbb{Z} \), then \( [n]^* \) denotes the mapping to the next greater power of two of \( n \).

In order to improve the situation, calculations will be done in \( \mathbb{Q}(\omega^{[\sqrt{M}]^*}) \), where \( \omega \) is a \((2M)^{th}\) root of unity. \( X^M + 1 \) is reducible over this field. It now splits into

\[
(*) \quad X^M + 1 = \prod_{i=1}^{[\sqrt{M}]^*} (x^{[\sqrt{M}]^*} - \omega^{[\sqrt{M}]^*} \cdot (i-1)).
\]

The Chinese remainder theorem shows that doing multiplications modulo \( X^M + 1 \) can be performed by calculating the remainders modulo the factors of \( X^M + 1 \), doing separate multiplications modulo these, and, by interpolation, calculating the product. Since the original coefficients were rationals, the remainders modulo the factors in (*) are all conjugate. Therefore only one remainder needs to be considered (v. fig. 2). This remainder has \( [\sqrt{M}]^* \) coefficients and can be acquired without any real calculations. To multiply in this modulo, a double length cyclic convolution is performed to do a full multiplication. The remainder is calculated thereafter. The cyclic convolution is a polynomial multiplication modulo \( X^{2[\sqrt{M}]^*} - 1 \). Since the \( 2[\sqrt{M}]^* \)-th roots of unity exist in the selected field extension, the DFT exists and with the convolution property and an FFT-type algorithm, the cyclic convolution can be calculated efficiently.

With the help of the FFT, the transformed coefficients can be multiplied separately. The coefficients have dimension \([\sqrt{M}]^*\) over \( \mathbb{Q} \). So they are polynomials in \( \psi = \omega^{[\sqrt{M}]^*} \) of degree \([\sqrt{M}]^* - 1\). Since \( \psi^{[\sqrt{M}]^*} \equiv -1 \), the same algorithm as before applies for the multiplication of the coefficients, as it did for polynomials modulo \( X^M + 1 \), however, this time of square root length.

3.2 Investigation of time expense

The cyclic convolution is a multiplication of polynomials modulo \( X^N - 1 \). That problem is reduced to multiplying once modulo \( X^{N/2} - 1 \) and once modulo \( X^{N/2} + 1 \) by the Chinese remainder theorem. Just scanning the input takes \( O(N) \) operations to do, so the algorithm will at least have that cost. Therefore reducing the problem to half results in a runtime which is at most half of that of the original length. Since the above reduction and its inverse can be performed in \( O(N) \) steps, the investigations can be limited to the expense of multiplying modulo \( X^M + 1 \). Twice the operations for that problem plus the \( O(N) \) from the reduction are an upper bound for the total cost.

The reduction modulo \( X^{[\sqrt{M}]^*} - \omega^{[\sqrt{M}]^*} \) takes \( O(M) \) steps, since it is just a copy of the input (v. example 1.). The FFT is known to have a cost of \( O(n \cdot \log(n)) \) for input of length \( n \). Therefore the double length FFTs need \( O(\sqrt{M} \cdot \log(M)) \) operations in the field extension. Since the coefficients are of length \( \sqrt{M} \), every one of these operations requires \( O(\sqrt{M}) \) steps (only additions, subtractions and shifts occur). So the total expense so far is

\[
O(\sqrt{M} \cdot \sqrt{M} \cdot \log(M)) = O(M \cdot \log(M)).
\]

What is left is the complexity of multiplying \( 2 \cdot \sqrt{M} \) coeffieients of length \( \sqrt{M} \). Since the same algorithm applies, the complexity is again

\[
O(2 \cdot \sqrt{M} \cdot \sqrt{M} \cdot \log(\sqrt{M})) = O(M \cdot \log(M)).
\]

So every recursion brings about this complexity. It stops after about \( \log(\log(M)) \) recursion steps. Therefore, the total complexity is:

\[
O(M \cdot \log(M) \cdot \log(\log(M))).
\]

4 Results

All convolution algorithms were tested on a Sun-3 computer. The programs are implemented in machine code. Every input coefficient is coded in a 32 bit integer. The resulting coefficients are 64 bit integers. The times are measured in milliseconds.
<table>
<thead>
<tr>
<th>$N$</th>
<th>School Method</th>
<th>Karatsuba</th>
<th>New Algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.099</td>
<td>0.366</td>
<td>3.833</td>
</tr>
<tr>
<td>8</td>
<td>0.349</td>
<td>0.833</td>
<td>9.333</td>
</tr>
<tr>
<td>16</td>
<td>1.117</td>
<td>2.183</td>
<td>21.990</td>
</tr>
<tr>
<td>32</td>
<td>3.583</td>
<td>6.199</td>
<td>52.331</td>
</tr>
<tr>
<td>64</td>
<td>11.666</td>
<td>17.165</td>
<td>119.995</td>
</tr>
<tr>
<td>128</td>
<td>34.832</td>
<td>45.331</td>
<td>269.989</td>
</tr>
<tr>
<td>256</td>
<td>104.906</td>
<td>108.529</td>
<td>601.642</td>
</tr>
<tr>
<td>512</td>
<td>323.320</td>
<td>269.999</td>
<td>1349.946</td>
</tr>
<tr>
<td>1024</td>
<td>966.628</td>
<td>614.975</td>
<td>2040.852</td>
</tr>
<tr>
<td>2048</td>
<td>2940.852</td>
<td>1516.600</td>
<td>6390.782</td>
</tr>
<tr>
<td>4096</td>
<td>8816.314</td>
<td>3349.856</td>
<td>13782.792</td>
</tr>
<tr>
<td>8192</td>
<td>26488.942</td>
<td>7899.884</td>
<td>29532.132</td>
</tr>
<tr>
<td>16384</td>
<td>79868.812</td>
<td>17182.646</td>
<td>63447.492</td>
</tr>
<tr>
<td>32768</td>
<td>230767.675</td>
<td>37098.516</td>
<td>133227.924</td>
</tr>
</tbody>
</table>

These times were measured for programs written in the C programming language. C lacks 64 bit integers as basic data types. Therefore, the values stem from running the code for 32 bit input and output. The code for the FFTs uses the floating point coprocessor M68831 of Motorola. A floating point number is also coded in 32 bits. The FFT method in the last column assumes the input to be real. The times are measured in milliseconds.

5 Summary

The results of the implementation show that the new algorithm is suitable for a good range of signal lengths. The advantage of using integers instead of floating point numbers makes it far better than the FFT for all practical problem sizes. However it must be mentioned that 32 bits are not enough to represent the resulting signals for very large lengths. One either has to use machine code to add 64 bit arithmetic to the C programming language or one has to accept roundoff errors by using fixed point arithmetic. The 64 bit machine code implementation, which produced the values of the first table is really a C program with only the main loops being coded in machine language. The structure of the algorithm suggests an easy way to implement it on a parallel hardware and possible direct hardware implementations in VLSI. The method does not require a fast floating-point unit and is therefore efficiently usable on a wide range of different machines.

6 References

SIMPLE, OPTIMAL REGULARITY ESTIMATES FOR WAVELETS

Olivier Rioul
CNET Centre Paris B, CRPE
38-40 rue du Général Leclerc, 92131 Issy-Les-Moulineaux
France.

The new criterion of regularity is of increasing interest in applications involving wavelet decomposition schemes. In this paper, regularity is fully characterized on filter taps, resulting in easily implementable, optimal regularity estimates which can be used for any filter.

1. INTRODUCTION

Perhaps the biggest potential of wavelet theory has been claimed for signal compression schemes [1,6] in which the signal is decomposed into several resolution levels using a "discrete wavelet transform (DWT)" [3,8]. In fact, the DWT was soon recognized to be equivalent to an octave-band tree filter bank which was proposed for some time in subband coding of images [9]. In this particular context, the novelty of wavelet theory comes down to the choice of the filters present in a two-band filter bank: "Wavelet" filters are regular.

In order to provide an intuitive feel for what regularity represents, consider the following iterated interpolations with low-pass filter impulse response $G(z)$ which are obviously present in DWT's [8].

$$Y(z) = X(z^2)G(z).$$

Iterating (1) $j$ times yields

$$Y^j(z) = X(z^{2^j})G^j(z)$$

where

$$G^j(z) = G(z)G(z^2)G(z^4)\ldots G(z^{2^j-1}).$$

The sequence $g_k^j$ corresponding to (3) is the equivalent impulse response at $j$th stage of the reconstruction. Now, for special choices of $G(z)$, the temporal shape of the $g_k^j$'s, plotted against $n2^{-j}$ (i.e., with the same temporal extent), rapidly converges to a "regular" limit function $\varphi(t)$ as $j \to \infty$ (see Fig. 1). However, for "bad" choices of $g_n$, $\varphi(t)$ may be highly irregular; the iterated scheme may even diverge, even though $G(z)$ is a "good" half-band low-pass filter [8]. Note that filters are assumed FIR here, $\varphi(t)$ is compactly supported. A first definition of the regularity order of $\varphi(t)$ is the number of times it is continuously differentiable; this is clearly a smoothness requirement on the temporal waveforms of the $g_k^j$'s.

The band-pass impulse responses present in a DWT are obtained with the same iterated interpolation procedure as (2), which is initialized with the high-pass filter impulse response $h_n$ [8]. The resulting limit function $\psi(t)$ is the continuous-time wavelet prototype [3,8]. Here we restrict ourselves to the convergence of the $g_k^j$'s toward $\varphi(t)$, because $\varphi(t)$ and $\psi(t)$ share the same regularity properties [3,6].

Several intuitive arguments have been raised which hint that this property should be useful in image coding applications [1,8]. First, requiring that the signal is analyzed by smooth "basis functions" $g_k^j$ and $h_k^j$ ensures that no artificial discontinuity—not due to the signal itself—appears in the transform coefficients, which are inner products of the signal with these basis functions. That is, regularity would lead to a "better" representation of the signal by the transform coefficients. Second, any quantization error made in a coefficient at some resolution level results, at reconstruction, in an error signal that is proportional to the basis function corresponding to this resolution level. It is therefore natural to require that this perturbation be smooth, rather than discontinuous: A discontinuous perturbation is likely to strike the eye more than a smooth one for the same m.s.e. distortion level.

However, understanding the role of regularity in a DWT-based compression scheme requires precise evaluation of it. One difficulty is that it is a mathematical notion which is expressed on $\varphi(t)$ rather than on
filters tape \( g_n \). Therefore, the characterization of regularity on any set of coefficients \( g_n \) is a difficult problem, which was first addressed in the wavelet context by Daubechies [3]. A number of regularity order estimates, most of them based on the spectrum \( |G(e^{i\omega})|^2 \), have been investigated [2,3,4]. Unfortunately, these estimates turn out to be suboptimal in general and sometimes computationally expensive.

This paper presents a complete characterization of regularity on the filter tape \( g_n \) in simple terms, restricting to the one-dimensional case. This method is original in that all regularity properties of \( \varphi(t) \) are translated into equivalent\(^1\) properties of the discrete-time sequences \( g_n \).

2. CONTINUITY

It can be shown [6] that as long as the resulting limit function is regular, the type of convergence of the \( g_n \) is "uniform," which is a strong type of convergence. Uniform convergence of the \( g_n \)'s is in fact equivalent to the existence and continuity of \( \varphi(t) \) [6]. Continuity (or uniform convergence) [6] is equivalent to the following intuitive conditions.

\[
G(1) = 2, \quad G(-1) = 0, \quad \lim_{j \to \infty} \max_n |g_n^j - g_n| = 0. \tag{6}
\]

Condition (4) is simply a normalization requirement on \( G(z) \), while (5) is crucial for convergence and regularity, as explained in section 3.1. The basic requirement (6) is that the difference between two successive values of \( g_n \) tend to zero uniformly in \( n \). Hence, no jumps or discontinuities should appear anywhere in the iterated sequences \( g_n \) as \( j \) increases, and the limit function is continuous.

However, even when \( \varphi(t) \) is required to be continuous, it may not appear to be smooth at all, as shown in Fig. 2. It is therefore natural to require more, namely that \( \varphi(t) \) possess \( N > 0 \) continuous derivatives. This is done next.

3. DERIVATIVES

The limit function \( \varphi(t) \) has regularity order \( N \) if its \( N \)th derivative, \( d^N \varphi(t)/dt^N \), is continuous. To characterize this on \( g_n \), consider the first-order finite difference sequence \( \delta g_n^j \), defined as the difference of the slopes of the "discrete curve" \( g_n^j \) plotted against \( n2^{-j} \).

\[
\delta g_n^j = \frac{g_n^{j+1} - g_n^{j-1}}{2^{-j}}. \tag{7}
\]

The corresponding z-transform is \( \Delta G^j(z) = [2(1 - z^{-1})G(z)]^j \). Applying \( N \) times the operator \( \delta \) yields the finite difference of \( g_n \) of order \( N \), \( \delta^N g_n \), given by \( \Delta^N G^j(z) = [2N(1 - z^{-1})G(z)]^j \).

Since the role of the derivative of \( \varphi(t) \) of order \( N \) is played in the discrete-time domain by \( \delta^N g_n \), it can be shown [6] that regularity order \( N \) is simply characterized by uniform convergence of \( \delta^N g_n \).

3.1. The role of zeroes at \( z = -1 \) in \( G(z) \).

In fact, the \( N \)th derivative of \( \varphi(t) \) can be obtained from the same iterated interpolation procedure as (2), where \( N \) zeroes in \( G(z) \) have been removed [6]. As a side result of this and (5), \( G(z) \) should have at least \( N + 1 \) zeroes at \( z = -1 \) to achieve regularity order \( N \). Note that adding one zero at \( z = -1 \) in \( G(z) \) will increase its regularity order by one since removing one amounts to "differentiate." Therefore, zeroes at \( z = -1 \) have a favorable effect for regularity. This was used by Daubechies in [3] to design regular, orthonormal "wavelet" filters, by imposing as many zeroes at \( z = -1 \) as possible in \( G(z) \) for a given filter length. Note that imposing such zeroes in \( G(z) \) amounts to requiring that the frequency response \( G(e^{i\omega}) \) is "flat" about half the sampling frequency \( \omega = \pi \).

However, the effect of zeroes at \( z = -1 \) may be killed by the other zeroes present in \( G(z) \). The rest of this paper aims at quantifying the "destructive effect" of zeroes in \( G(z) \) that are not located at \( z = -1 \) in order to quantify regularity accurately.
4. HÖLDER AND SOBOLEV REGULARITY

We first extend regularity orders to arbitrary, real-valued numbers. A popular extension [2,3] uses a spectral approach to regularity which regards it as a spectral localization. This definition is typically based on Sobolev spaces [2,6]. However, it masks the effect of regularity on the temporal waveform of \( \varphi(t) \) and does not use phase information of \( G(e^{j\omega}) \). This may be inappropriate: Fig. 2 shows an example of \( \varphi(t) \) for which the best Sobolev exponent \( r \) is negative, although it can shown that \( \varphi(t) \) is in fact continuous.

These limitations are overcome in the following definition of Hölder regularity. The function \( \varphi(t) \) is regular of order \( \alpha, \) \( 0 < \alpha \leq 1, \) if

\[
|\varphi(t + h) - \varphi(t)| < ch^\alpha.
\]

(8)

This controls the way infinitesimal slopes of \( \varphi(t) \), \( |\varphi(t + h) - \varphi(t)|/|h| \), grow as \( h \) becomes indefinitely small. For higher regularity orders \( r = N + \alpha, \) \( 0 < \alpha \leq 1, \) the same definition is used on the \( N \)th derivative of \( \varphi(t) \). This definition is more compatible with continuity and differentiability because it can be shown that [6] if \( \varphi(t) \) is a limit function of \( \varphi_n \), then \( \varphi(t) \) possesses \( N \) continuous derivatives if and only if it has some Hölder regularity order \( r \) greater than \( N \). We have seen that this property is not shared by Sobolev regularity. In the following we therefore concentrate on Hölder regularity.

There is a slight irritation in that \( \varphi(t) \) possesses \( N \) continuous derivatives only when its Hölder regularity is \( r = N + \epsilon, \) where \( \epsilon > 0 \) is arbitrarily small. To simplify our presentation, we drop the \( \epsilon \) in the sequel and regard regularity orders within an arbitrarily small constant.

4.1. Hölder regularity order \( 0 < \alpha \leq 1 \)

To characterize Hölder regularity \( \alpha, \) \( 0 < \alpha \leq 1, \) on \( g_n \), we can do an analogy with (8), replacing \( \varphi(t) \) by \( g_n \) with \( t = n2^{-j} \) and \( h = 2^{-j}. \) This gives

\[
|g_{n+1} - g_n| < c2^{-j\alpha}.
\]

(9)

This property, along with (4), (5), is indeed equivalent to (8) [5]. This gives an intuitive interpretation of Hölder regularity: The slopes of \( g_n \) plotted against \( n2^{-j} \), \( |g_{n+1} - g_n|/2^{-j} \), grow less than \( 2^{1-\alpha} \) as \( j \rightarrow \infty. \)

For example, bounded slopes mean that regularity order is 1, i.e., \( \varphi(t) \) is almost continuously differentiable. And less regularity allows slopes to increase indefinitely. This explains why \( \varphi(t) \), although continuous, may sometimes be quite "nasty" as in Fig. 2.

4.2. Arbitrary regularity orders

Since derivatives of \( \varphi(t) \) correspond to finite differences of \( g_n \), a natural discrete-time characterization [6] of regularity order \( r = N + \alpha, \) \( 0 < \alpha \leq 1, \) is (4), (5), and (9) written for \( \delta^N g_n^j, \) i.e.,

\[
|\delta^N g_{n+1}^j - \delta^N g_n^j| < c2^{-j\alpha}.
\]

(10)

A remarkable fact is that (10) can be extended to negative values of \( \alpha \) [6]. That is, even if (10) "fails," i.e., gives a negative regularity order for \( \delta^N g_n^j, \) it can be used to prove that \( g_n \) has some (positive) regularity if \( N > \alpha. \) It is therefore worthwhile to consider negative regularity orders. In particular, assume that \( G(z) \) has exactly \( N \) zeroes at \( z = -1. \) The maximum number of \( \alpha \leq 0 \) for which (10) holds is then the exact amount of regularity lost due to the destructive effect—discussed in section 3.1—of the zeroes in \( G(z) \) that are not located at \( z = -1 \) [7].

4.3. Regularity and rate of convergence

In practical systems involving a discrete implementation of the DWT, the number of iterations \( j \) is limited. It is therefore questionable to study the limit function as \( j \rightarrow \infty. \) However, the rate of convergence of \( g_n \) to \( \varphi(t) \) is faster as regularity is high (the difference tends to 0 as \( 2^{-j\alpha} \) [6]). The convergence is even faster for higher regularity orders (see Fig. 1).

5. OPTIMAL REGULARITY ESTIMATES

A regularity estimate \( r \) is here said to be optimal if \( \varphi(t) \) is at least regular of order \( r - \epsilon \) and is not regular of order \( r + \epsilon, \) where \( \epsilon > 0 \) is arbitrarily small.

A simple algorithm [6], which was independently derived using the "Littlewood-Paley theory" by Cohen and Daubechies [2], gives the optimal Sobolev regularity order. However this is not optimal for Hölder regularity in general: Hölder regularity is always greater than Sobolev regularity by at most 1/2 [6]. This gives suboptimal Sobolev lower and upper bounds for Hölder regularity. Sobolev regularity depends on the modulus of the spectrum while filters that differ only by their phase have Hölder regularity orders that differ by at most 1/2. In the following we provide sharp lower and upper bound estimates based on characterization (10).

5.1. Lower bound

Since (10) must be satisfied for infinitely many \( j \)'s and with an unknown constant \( c, \) this is impossible to check in practice. Fortunately, this task can be reduced to a finite-time computer search [6,7]:

Algorithm 1 (Lower bound on Hölder regularity). Let \( N > 0 \) be the exact number of zeroes at \( z = -1 \) in low-pass interpolation FIR filter \( G(z), \) normalized such that
\( G(1) = 2. \) If \( G(z) \) only has zeroes at \( z = -1, \) stop. The Hölder regularity order is \( N. \) To estimate the amount of regularity lost due to the other zeroes, compute \( F(z), \) defined as

\[
G(z) = 2^{-N}(1 + z^{-1})^N F(z).
\] (11)

Let \( j \) be any positive integer. Compute the (positive) number

\[
\beta_j = \frac{1}{j} \log_2 \max_{0 < n < 2^j} \sum_k |f_{n-2^j k}|,
\] (12)

where \( f_{n} \) is given by

\[
F(z) = F(z) F(z^2) \ldots F(z^{2^{j-1}}).
\] (13)

The Hölder regularity order of \( G(z) \) is at least \( N - \beta_j. \)

A matrix formulation can be shown \([8]\) to be equivalent to a Hölder regularity estimate which was derived by Daubechies and Lagarias \([2,4]\) using a very different approach. While the method they describe in \([4]\) is only manageable for very short filters \( G(z), \) Algorithm 1 gives nearly optimal results (as \( j \) increases) for any filter: In fact, \( N - \beta_j \) tends (at most as \( 1/j \)) to the optimal Hölder regularity order as \( j \to \infty \) \([6]\). In practice, the exact (optimal) regularity order \( r \) is generally obtained to two decimal places after \( j = 20 \) iterations. This algorithm can be easily implemented by recursive calls to the same small subroutine \([7]\).

### 5.2. Upper bound

One possible drawback of Algorithm 1 is its exponentially increasing numerical complexity \([7]\). Now, assume that one retains only the values \( n = 0 \) and \( 2^j - 1 \) in the computation of the maximum in \((12)\): This results of course in a much faster algorithm. The obtained estimate clearly gives an upper bound of Hölder regularity as \( j \to \infty \) since \( \beta_j \) is under-estimated. We give here the matrix formulation of this algorithm, which simplifies to the computation of a spectral radius of one matrix \([8]\):

**Algorithm 2 (Sharp Hölder regularity upper bound).** Let \( G(z), F(z), N > 0 \) be as in Algorithm 1 and let \( K > 1 \) be the length of \( F(z). \) Form the matrix

\[
F = (F_{i,j}),
\]

\( 0 \leq i,j \leq K - 2, \) defined by

\[
F_{i,j} = f_{n-i+j+1}
\] (14)

and compute its spectral radius \( \rho. \) The Hölder regularity order of \( G(z) \) is bounded by \( N - \text{max}(|f_0|, |f_{K-1}|, \rho). \)

The resulting estimates are very close to the optimal Hölder regularity order, as seen in Fig. 3.

---

### 6. Conclusion

The method presented here, which characterizes regularity on discrete-time sequences, was found to be powerful: We have provided regularity estimates that are, in contrast with earlier ones \([2,3,4]\), easily implementable, optimal, and of general applicability. Local regularity \([4]\) can also be studied as alternatives to global regularity using this method. \([7]\).

---

### REFERENCES


CERTAIN ASPECTS OF FIXED POINT FFT AND WFTA ROUNDDING ERRORS COMPARISON

Ewa ŁUKASIK
Institute of Computing Sciences, Technical University of Poznań, PL-60-965 Poznań, ul. Piotrowo 3a, Poland

Ryszard STASIŃSKI
Institute of Electronics and Telecommunications, Technical University of Poznań, PL-60-965 Poznań, ul. Piotrowo 3a, Poland

After the revision of to date results of fixed point FFT and WFTA precision comparison some new conclusions are presented based on the experimental investigation. Several aspects of the problem are sketched out. The stress is laid on statistical parameters of the rounding errors. The useful measure is introduced: difference of accurate bits of FFT and WFTA.

1. INTRODUCTION
Numerical errors introduced by limited word length became a factor of special interest for algorithms used in signal processing. Their value coincides with the structure and computational speed of algorithms, as well as with the hardware design. Fixed point realizations still find wide acceptance in practice, mainly when there is a need of fast realizations. Algorithms discussed in this paper concern computing the Discrete Fourier Transform given by the formula:

\[ X(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{j2\pi nk/N} \]

There is a large literature treating of the Fast Fourier Transform (FFT) rounding errors. The analysis of the errors of Winograd Fourier Transform Algorithm (WFTA, which is the fastest one, has been the subject of the works reported in [1] and then in [4,5,6]. In this paper we present a comparison of errors introduced by thoroughly prepared (optimized) fixed point WFTA and FFT algorithms for real data in two's complement arithmetics. The errors of special interest are:
- random errors resulting from word length reduction after multiplication and/or addition/subtraction and shift, assuming that input signals are not deterministic,
- errors when input signals are deterministic, three various methods of quantization.

2. THE CHOICE OF ALGORITHMS
Complex radix-2 FFTs, in both versions - decimation in time (DIT) and decimation in frequency (DIF) are the most widely used algorithms for computing Discrete Fourier Transform. On the other hand split-radix FFT [3] is the one with minimal computational complexity among all FFTs and minor rounding errors, esp. in its real valued DIF version. The structure of both algorithms is modular and easy to implement. The main problem concerning their fixed point realizations accuracy is only multiplication by the coefficient equal to 1, which can be easily skipped over, and the place of scaling. These algorithms were chosen as the representatives for the comparison with WFTA from the point of view of rounding errors magnitude.

The Winograd Fourier Transform Algorithms, being in general more complicated than FFT, have a very important advantage of the least number of multiplications among all the DFT algorithms. This feature is still very important because generally it makes them faster then other DFTs especially taking into account the fact, that in all kinds of implementations the cost of executing multiplication is bigger than that of addition. Although the structure of WFTAs causes that rounding errors are unavoidably bigger [7] but there are more possibilities to alter their structure in order to diminish them [5]. The multiplications in WFTA are placed in the middle of the algorithm. In 'classical' WFTAs [2] all the multiplications are the scalar ones. In the algorithms examined in this paper some of the multiplications are scalar and some of them are polynomial products (PP) modulo \( z^2+1 \). This was one of the measures undertaken in order to construct WFTA-like algorithms with diminished values of rounding errors in fixed point realizations. Basic modules for these algorithms were as follows:
- \( N=2^t \)- point module, the simplest, requiring the use of PP mod \( z^2+1 \) starting from \( t=4 \), causing the smallest rounding errors;
- N=15-point module, constructed with the 3-point and 5-point small DFTs: 5-point one is based on the 4-point circular convolution, in which FP mod z^4 +1 has to be computed; in 3-point DFT all the multiplications are scalar;
- N=17-point module, being the most complicated and the most erroneous one: it is built of a 16-point circular convolution, where FP mod z^16 has to be developed in five stages to get FP mod z^{1+1} inside; a rather big expansion of the algorithm is observed (effect of algorithm lengthening and growing its dimension in internal stages), resulting in greater rounding errors.

The above modules may form two subclasses of algorithms, having the following sizes:
1. N=2^3·15 - point DFTs, eg N=30, 60, 120, 240,
2. N=2^5·255 - point DFT, eg N=510, 1020, 2040, 4080 ...

These sizes are compatible to-radix-2 and split radix FFT algorithms sizes i.e. N=2^k and they are consistent with those specified in [7] as the least erroneous. Smaller computational complexity is their advantage over FFT.

In fixed point realizations all three mentioned above algorithms of computing DFT have some characteristic features:
- all data and coefficients are the fractional numbers,
- scalings by the factor 2 is implemented at each stage of additions to ensure keeping the intermediate results as fractions (that results in 1/N·X(k) instead of original definition (I)),
- overscaling (dividing a sum of r components by the number 2^r) never occurs.

3. FFT AND WFTA ROUNDING ERRORS COMPARISON

3.1. Random rounding errors

The important factor of fixed point rounding error comparison is the quantization method. For the analysis presented in this paper three of them were used:
- rounding (R),
- truncation (T),
- random rounding (RR), where the value 0.5 is randomly rounded or truncated.

The most suitable aspect of FFT and WFTA rounding errors comparison is mean square error. It may be computed analytically using the method presented in [6] or by simulations for a large number of experiments. The second method was chosen to perform the comparison. For three DFT algorithms presented above the same experiments were carried out. For the Gaussian noise as the input signal the given algorithm was used twice: first in its floating point version, the results X(k) being treated as the exact ones and then in simulated fixed-point version, that operated on 16-bit data words, giving the erroneous result [X(k)]. The squared difference:

\[ \Delta X(k) = \left( \frac{1}{N} \cdot X(k) - \frac{1}{N} \cdot X(k) \right)^2 \]

\( k \)- number of DFT sample, \( k = 0, 1, \ldots, N-1 \)
\( i \)- number of experiment, \( i = 0, 1, \ldots, N-1 \)

was the error measure. For big number of experiments (N=1024) we get the information about the statistical features of each \( X(k)_i \) error by computing the mean value of all \( N \) values of \( \Delta X(k) \). The results are presented in rising order on graphs for the representatives of mentioned above sizes of FFT and WFTA, i.e. N=128 and 120, N=1024 and 1020 points (Fig.1). Additionally we introduce two criteria for algorithms comparison:
- maximal mean-square error of all the results (P_m),
- average mean-square error of all the results (P).

The most implementation-oriented measure of the accuracy is the difference in number of accurate bits of FFTs and WFTAs for both criteria and three quantization methods, presented in table 1.

<table>
<thead>
<tr>
<th>N</th>
<th>error</th>
<th>Arithmetics/Algorithms</th>
<th>R</th>
<th>RR</th>
<th>T</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>P</td>
<td>WFTA/FFT</td>
<td>-0.27</td>
<td>0.08</td>
<td>0.09</td>
</tr>
<tr>
<td>128</td>
<td>P</td>
<td>WFTA/FFTs</td>
<td>-0.26</td>
<td>0.08</td>
<td>0.36</td>
</tr>
<tr>
<td>P_m</td>
<td></td>
<td>WFTA/FFT</td>
<td>-0.40</td>
<td>0.19</td>
<td>0.14</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WFTA/FFTs-r</td>
<td>-0.51</td>
<td>0.18</td>
<td>0.24</td>
</tr>
<tr>
<td>255</td>
<td>P</td>
<td>WFTA/FFT</td>
<td>0.72</td>
<td>1.11</td>
<td>1.35</td>
</tr>
<tr>
<td>256</td>
<td>P</td>
<td>WFTA/FFTs-r</td>
<td>0.88</td>
<td>1.09</td>
<td>1.64</td>
</tr>
<tr>
<td>P_m</td>
<td></td>
<td>WFTA/FFT</td>
<td>0.07</td>
<td>1.20</td>
<td>1.37</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WFTA/FFTs-r</td>
<td>-0.03</td>
<td>1.16</td>
<td>1.41</td>
</tr>
<tr>
<td>1024</td>
<td>P</td>
<td>WFTA/FFT</td>
<td>0.98</td>
<td>1.39</td>
<td>1.62</td>
</tr>
<tr>
<td>1024</td>
<td>P</td>
<td>WFTA/FFTs-r</td>
<td>1.16</td>
<td>1.37</td>
<td>1.91</td>
</tr>
<tr>
<td>P_m</td>
<td></td>
<td>WFTA/FFT</td>
<td>0.24</td>
<td>1.64</td>
<td>1.53</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WFTA/FFTs-r</td>
<td>0.08</td>
<td>1.63</td>
<td>1.62</td>
</tr>
</tbody>
</table>

R-rounding, RR-random rounding, T-truncation, s-r - split-radix.

The main conclusions of this comparison are:
- optimization of WFTA algorithms leads to the best results for rounding; - in general the differences do not exceed 2 bits,
- experimental results for WFTA are concordant with the model presented in [7], whereas the model for FFTs is too pessimistic. It does not include the effect of noise attenuation caused by multiplication by fractional coefficients in every step of the algorithm. In WFTA such a noise reduction performs only once.
Fig.1. Mean-square errors ($\varepsilon^2$) of $N$ output samples of three various DFT algorithms and three quantization methods in rising order: a) $N=120/128$, b) $N=1020/1024$
Fig. 2. Error mean value ($\mu$) and standard deviation ($\sigma$) for three DFT algorithms, N=255/256, rounding.

Fig. 2. presents the example of mean value (offset) and standard deviation of FFTs and WFTA (N=255, rounding). Since the offset is the significant error component (for truncation even more) it may be compensated at the output of the algorithm. However standard deviation, being rather stable for all output samples, remains higher for WFTA than for both FFTs for the same reasons as mentioned above.

3.2. Other aspects of FFT and WFTA accuracy comparison

A question may be raised, concerning the worst-case input function causing the biggest rounding error at the DFT output. Some useful functions for algorithms rounding error comparison are sime, cosine or impulse functions. However they do not involve in computation all the algorithms graph paths and nodes, being the rounding error sources, i.e. some of them are omitted and final error does not take maximal value. Only for random input function this may not occur. Indeed the highest rounding errors we have got for realizations of Gaussian noise as input functions. A sequence of 510 random N-sample realizations was examined for all N to catch those giving maximal error values. In Table 2 the results are listed. Evidently errors of WFTA are bigger than of any of FFTs. These results confirm previous conclusions.

4. CONCLUSIONS

All efforts leading to the minimization of rounding errors in WFTA did not give results compensating the structural features determining higher values of rounding errors of WFTA than for FFT.

These structural features are:
- lengthening and growing internal dimensions of the algorithms;
- little influence of coefficients on the noise reduction.

They are involved in all kinds of errors. The results are slightly better than general ones given in [1].

<table>
<thead>
<tr>
<th>$N$</th>
<th>error</th>
<th>WFTA</th>
<th>FFT</th>
<th>FFTs-r</th>
</tr>
</thead>
<tbody>
<tr>
<td>120</td>
<td>$\rho$</td>
<td>1.04</td>
<td>1.14</td>
<td>1.06</td>
</tr>
<tr>
<td></td>
<td>$\rho_{\max}$</td>
<td>2.68</td>
<td>2.80</td>
<td>3.04</td>
</tr>
<tr>
<td>255</td>
<td>$\rho$</td>
<td>2.09</td>
<td>1.18</td>
<td>1.11</td>
</tr>
<tr>
<td></td>
<td>$\rho_{\max}$</td>
<td>6.67</td>
<td>3.28</td>
<td>3.22</td>
</tr>
<tr>
<td>1024</td>
<td>$\rho$</td>
<td>2.52</td>
<td>1.21</td>
<td>1.14</td>
</tr>
<tr>
<td></td>
<td>$\rho_{\max}$</td>
<td>9.46</td>
<td>4.10</td>
<td>4.43</td>
</tr>
</tbody>
</table>

$\rho = \sqrt{\rho} / Q$, $\rho_{\max} = \sqrt{\rho_{\max}} / Q = 2^{-(b-1)}$

REFERENCES

A NOVEL FAST FIR FILTERING ALGORITHM AND ITS IMPLEMENTATION*

Sau-Gea Chen & Ren-Jer Tsai

Department of Electronics Engineering, National Chiao Tung University
Hsinchu, Taiwan, ROC

A class of novel convolution algorithms are proposed which reduce the number of multiplications by more than 50%, and in some cases decrease the number of additions by more than 50% per output point, compared with the direct convolution. As such, their roundoff noises are greatly reduced. Under the same direct-form L1-norm scaling scheme, the problem of increased dynamic range is irrelevant. The algorithm is well applied to IIR, and 2-D convolutions with more computation reductions both in multiplication and addition counts. When implemented, the algorithms maintain architectural regularity.

1. INTRODUCTION

There are numerous fast convolution algorithms in the literature [1,2,3,4], which can be categorized as two different classes. The first one is the well-known transform-domain FFT approach. Though asymptotically the FFT-based algorithms are optimal, they require dedicated hardware implementation for bit-reversed data routing, in addition that every single output sample may not be available at the instant.

The second one is done directly in the temporal domain. Some of the popular algorithms include the iterated FIR filtering [2,5] that builds long convolution iteratively by using short Winograd convolution algorithms. Asymptotically, the complexity for the filtering operations with \( N \) by \( N \) filter kernel costs \( N^2 \times 5.885 \) multiplications per output point which is inferior to FFT approach for long convolution length. Another time-domain fast convolution is done by multirate processing [3,4] of decimating output signals. the original \( N \) multiplications required for a single output is reduced to \( 3N^2/4 \) multiplications due to the decomposition of the convolution as three shorter convolutions each with size of \( N/2 \) that produce two output points at the same time. The algorithm can also be iteratively decomposed to its fullest, but accordingly there are disadvantages of using multi-clocking system.

Here a novel temporal-domain filtering algorithm is proposed which reduces number of multiplications from \( N \) to \( N/2 \) at the expense of increased additions from \( N-1 \) to \( 3N/2-1 \). The saving of \( N/2 \) multiplication operations dwarfs the penalty of \( N/2 \) extra addition operations. Moreover, the new algorithm preserves the regular FIR filtering structures. Compared with the multirate convolution algorithm, the algorithm reduces \( N^2/4 \) multiplications, while costs \( 3N/4 \) more additions. More reductions in multiplication and even addition counts are obtained when the 1-D algorithm is extended to 2-D convolutions. For instance, in the new 2-D algorithm for a 2-D quadrilateral symmetric convolution, the numbers of multiplications and additions required are only respectively \( 1/8 \) and \( 3/8 \) that of by direct convolution.

In what follows, we shall first elaborate on the new 1-D convolution algorithm. Then, extension of the algorithm to various convolutions are detailed. Finally, example realizations for the algorithms and their implementation considerations are discussed.

2. THE ALGORITHM

Given the input sequence \( x(n) \) and filter coefficients \( h(n) \), the convolution output sequence \( y(n) \) can be reformulated as three parts:

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

\[
= \sum_{k=0}^{N-1} [x(n-2k) + h(2k + 1)][x(n-2k-1) + h(2k)]
\]

\[
- \sum_{k=0}^{N-2} h(2k)h(2k + 1) - \sum_{k=0}^{N-2} x(n-2k)x(n-2k-1)
\]

The first part has complexity of \( N/2 \) multiplication and \( 3N^2/2-1 \) addition operations per output point. The second one can be precalculated and fixed during the rest of entire convolution operation, thus contributes only one addition operation.

* This work was supported in part by National Science Council under Contract NSC-81-0404-E009-130.
The term
\[ \sum_{k=0}^{N/2-1} x(n-2k)x(n-2k-1) \]
can be computed by defining a recursive sum-of-product equation as
\[ p(n) = p(n-2) + x(n)x(n-1) - x(n-N)x(n-N-1) \]
where \( x(k) = 0 \) for \( k < 0 \), \( p(k) = 0 \) for \( k < 0 \)
thus contributes only one multiplication and two additions. The total computational complexity is thus \( N^2/2+1 \) multiplications and \( 3N^2/2+3 \) additions.

3. 2-D ALGORITHMS

3.1 Nonseparable 2-D Convolution Algorithm

A nonseparable 2-D convolution operation can be considered as a stacking of 1-D convolutions, and the operation can be decomposed as
\[
y(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x(i-k,j-l)h_k(j) - p(i,j) - C_n
\]
where
\[
z(i,j,k,l) = x(i-k,j-l) + h_k(2l+1),
\]
\[
z(i,j,k,l) = x(i-k,j-l) + h_k(2l+1)
\]
\[
p(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x(i-k,j-l)x(i-k,j-l-1)
\]
\[
p(i,j) = p(i-1,j) + p_{aw}(i,j) + c_n(i,j)
\]
\[
p_{aw}(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x(i-k,j-l)x(i-k,j-l-1)
\]
\[
p_{aw}(i,j) = p_{aw}(i-j-2) + x(i,j)x(i,j-1) - x(i,j-M)x(i,j-M-1)
\]
\[
p_{aw}(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} x(i-k,j-l)x(i-k,j-l-1)
\]
\[
p_{aw}(i,j) = p_{aw}(i-N,j)
\]
\[
C_n = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} h_k(2l)h_k(2l+1) = \text{constant}
\]
If signals are processed rowwise, then for an \( MN \times M \times N \) 2-D filter kernel, the number of multiplications and additions required for a single output point will asymptotically be \( MN^2/2+1 \) and \( 3MN^2/2+5 \) respectively.

3.2 2-D Separable Convolution Algorithm

A 2-D separable convolution can be executed as a concatenation of two 1-D algorithms, i.e. by the nested algorithm as follows,
\[
y(n,m) = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} x(n-j,m-k)h_k(j)
\]
\[
y(n,m) = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} x(n-j,m-k)h_k(j)
\]
\[
y(n,m) = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} x(n-j,m-k)h_k(j)
\]
\[
y(n,m) = \sum_{j=0}^{M-1} \sum_{k=0}^{N-1} x(n-j,m-k)h_k(j)
\]
which amounts to \( (N^2/2+1)(M^2/2+1) \) multiplications and \( (3MN^2+5N^2)/2+3 \) additions. Compared with the direct convolution, there is a close to 4-fold reduction in multiplications, at the expense of 1.5-fold increase in additions.

3.3 2-D Quadrirsymmetry Convolution Algorithm

Frequently, we require that a 2-D filter be symmetric so as to have linear-phase response. The algorithm gives a much reduced computation for quadrirsymmetry is as follows,
given \( h(k,l)=h(M-k,l)=h(k,N-l)=h(M-k,N-l) \)
\[
y(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} h(k,l)z(i-k,j-l) + x(i-k,j-l+1)
\]
\[
y(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} h(k,l)x(i-k,j-l+1)
\]
\[
y(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} h(k,l)z(i-k,j-l) + x(i-k,j-l+1)
\]
\[
y(i,j) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} h(k,l)x(i-k,j-l+1)
\]
By taking advantage of the symmetry, the equation can be further simplified as
\[
y(i,j) = \sum_{k=0}^{N/2-1} y_s(i,j) + \sum_{k=0}^{N/2-1} y_s(i,2k-M+1,j)
\]
\[
y(i,j) = \sum_{k=0}^{N/2-1} y_s(i,j) + \sum_{k=0}^{N/2-1} y_s(i,2k-M+1,j)
\]
The second term of RHS had already been computed for the earlier indices \( i \). The first term of RHS can be similarly decomposed as before,
\[
\sum_{k=0}^{N/2-1} y_s(i,j) = \sum_{k=0}^{N/2-1} \sum_{l=0}^{N-1} x(i,j,k,l)x_s(i,j,k,l) - C_n - p(i,j)
\]
\[
x_s(i,j,k,l) = x_s(i,j,k,l) + h(k,2l+1)
\]
\[
x_s(i,j,k,l) = x_s(i,j,k,l) + h(k,2l+1)
\]
\[
x_s(i,j,k,l) = x(i-k,j-l-2) + x(i-k,j-N+2l+1)
\]
\[
x_s(i,j,k,l) = x(i-k,j-l-2) + x(i-k,j-N+2l+2)
\]
\[
p(i,j) = \sum_{k=0}^{M/2-1} \sum_{l=0}^{N/2-1} x(i,j,k,l)x_s(i,j,k,l)
\]
\[
p(i,j) = \sum_{k=0}^{M/2-1} \sum_{l=0}^{N/2-1} x(i,j,k,l)x_s(i,j,k,l)
\]
\[ p_{aw}(i,j) = \sum_{l=0}^{N/4-1} x_2(i,j,0,l)x_4(i,j,0,l) \]
\[ p_{aw}(i,j) = \sum_{l=0}^{N/4-1} x_3(i,j,M/2,l)x_4(i,j,M/2,l) \]
\[ = p_{aw}(i-M/2,j) \]

The algorithm costs \( MN/8 + N/4 \) multiplications and \( 3MN/8 + 3N/4 + M/2 + 2 \) additions. Close to eight-fold reduction in multiplications and 2.7-fold reduction in additions are obtained, compared with the direct convolution.

4. IMPLEMENTATION

In hardware, the algorithms can be realized by most of the conventional FIR filter structures. One of many possible realizations for the 1-D convolution algorithm is shown in Figure 1. Figures 2 and 3 depict possible structures for a 2-D \( 4 \times 4 \) quadrissymmetry filter. All the structures maintain high degree of regularity.

4.1 Word Length

For the 1-D realizations, assumed (B+1)-bit fixed-point representation and stationary white uncorrelated rounding noises [6] for the multiplications, then the variance of roundoff noise of this structure is equal to

\[ \sigma^2 = (N/2 + 1) \times 2^{-28} / 12 \]

which is half that of direct-form realization. To remedy the more severe overflow problem than the direct-form realization, the same \( L_1 \)-norm scaling scheme as that of direct form is sufficient to eliminate possible overflows, i.e. choose a scaling constant that confines \( x(n) \) as follows,

\[ \max(|x(n)|) < \frac{1}{\sum |h(l)|} \]

then it is enough that the adders are implemented with (B+1) bits as the direct form. The overflow bits in the addition operations are don't-care. In the multiplier implementations, (B+2)-bit operand size have to be allowed for the possible carry bit from the addition before multiplication and multiplication results are rounded to (B+1) bits as usual. Similar argument is also applied to 2-D convolver design.

5. CONCLUSION

Several efficient convolution algorithms are proposed in this work, which achieve more than 50% reductions in multiplication counts, and in some cases also the reduction of addition counts. The approach is well applied to 1-D or 2-D IIR filtering operations. The algorithms can be easily adapted to systolic-array-like realizations. But care must be taken in deciding the scanning schemes for 2-D input signals, which deserved further study. The algorithms are under further investigation when they are extended to other cases of filter symmetries [7]. Likewise, the algorithms can be combined with multirate approaches for possible more efficient filter realizations.

REFERENCES

Figure 1. 1-D filter structure based on the new convolution algorithm

\[ y(n) = \sum_{k=0}^{N-1} h(k)h(2k+1) \]

\[ x(n), h(n) \]

Figure 2. Filter structure for a 2-D 4x4 quadrisymmetric convolution based on the new 2-D algorithm

\[ y(i,j) = h(0,0)h(0,1) + h(0,0)h(1,0) + h(1,1)h(0,0) + h(1,1)h(1,0) \]

\[ x(i,j), x(i,j-1), x(i,j-2), h(i,j), h(i,j-1), h(i,j-2) \]

Figure 3. Another realization for 4x4 quadrisymmetric filter

\[ Y_k \]
A study of the filter roundoff noise in subband coding systems

RENAUD Jean-Pierre (**), LECH Henri (**)

(*) Lernout and Hauspie Speechproducts
Rosendaalstraat 14
8900 Ieper (Belgium)

(**) Faculté Polytechnique de Mons
Boulevard Dolez 31
7000 Mons (Belgium)

Tel : 32 65 374322 Fax : 32 65 374300

ABSTRACT

The influence of filter roundoff noise in subband coding systems is examined in this paper. First of all, we present a methodology which allows to determine the energy propagation of signals in tree-structured systems. Next, the output level of filter roundoff noise is determined as a function of the number of subbands, for the fixed-point arithmetic (FXP) on one hand, and for the floating-point arithmetic (FLP) on the other hand. Results of simulation programs implemented on a PC are presented in order to verify the predictions. Finally, practical case of listening tests where roundoff noise influence is obvious, is presented, with some possible solutions to eliminate it.

1. INTRODUCTION

Subband coding systems are nowadays widely used for sound [1,2,3] and speech processing [4]. They indeed allow a significant bit-rate reduction without altering the quality of the processed signal. These systems are based on digital signal processors (DSP) which work with a fixed-point (FXP) or floating-point (FLP) arithmetic.

Our aim is to determine the noise performances of the subband coding systems according to the arithmetic of the DSP used, and the complexity of the whole system. This problem is important as experiments show that an higher complexity yields better bit-rate reduction. Moreover, DSP producers are continuously improving the processing capacity of their chips, in terms of accuracy and speed. This allows then the realization of more complex algorithms.

Section 2 presents the architecture of the subband coding systems studied and gives the methodology which allows the determination of the signal energy propagation in these multirate systems. In the next section, this methodology is used to compute the roundoff noise as a function of the different parameters of the subband system, as the processor arithmetic, the number of subbands, etc. Then, results of simulations made on a PC in order to verify theoretical predictions are presented and commented. Section 4 describes a practical example where roundoff noise is put in evidence by listening tests.

Conclusions are given in the last section.

2. ARCHITECTURE OF SUBBAND CODING SYSTEMS AND DETERMINATION OF ENERGY PROPAGATION

2.1 Architecture of subband coding systems

A subband coding system can schematically be divided into three main phases: the analysis, the subband processing and the synthesis.

The analysis consists in the splitting of the input signal in a certain number of signals each of which represents a determined part of the original signal spectrum, and which are called "subbands". This operation is done, in principle, by a band-pass filters bank which isolates the different spectrum parts, followed by a decimation process which preserves the original bit-rate.

At this step, the subband processing allows to reduce the bit-rate needed to store or transmit the signal. This phase is not taken into account in this paper.

Finally, the synthesis reconstructs the signal from the subbands. It is the dual operation of the analysis and consists in an interpolation process followed by a band-pass filters bank.

The analysis and the synthesis are obviously elaborated so as to preserve the original signal if there is no subband processing. This implies that "aliasing" and "imaging" effects generated respectively by the decimation and the interpolation processes, must be eliminated. This is supposed to be achieved by employing "quadrature mirror filters" (QMF) [5].

In this paper, we only consider systems which architecture is based on the "dichotomous tree structure". In these systems, the input signal spectrum is split, at the first stage, in two half-band signals. At next stage, each of these last signals are "spectrally" divided in two again, and so on. If N stages are implemented, 2^N subbands are created. This structure is the most often used in subband coders [2,3].

It also allows a very symmetric implementation of the analysis and synthesis which is, practically, obtained by the combination of elementary "direct" cells of figures 1 and 2.
In practice, as the processed signals are of finite duration, we do not compute the global energy. We prefer a mean energy estimator (on $n$ samples) called "variance" $\sigma_x^2$, which is given by the following formula:

$$\sigma_x^2 = \frac{1}{n} \sum_{i=1}^{n} x_i^2$$

It can be shown that the variance can be freely used to compare (mean) energy of signals.

2.2.2 Propagation of the energy in the cells

The energy contribution of the signals issued from the filters cells is straightforward if these filters are considered as ideal. However, in the cases of the decimation and the interpolation, the determination of the input-output energy relation is not obvious. Relations between transmittances of the input and output signals [see (10)] lead to the following results for the interpolation:

- $d$: input signal at $f_E/2$ ($\Phi = 2\pi f / f_E$)
- $y$: output signal at $f_E$ ($\Phi = 2\pi f / f_E$)
- $K = 1/2$
- $E_y = E_d/2$

For the decimation, the relations are the following:

- $x$: input signal at $f_E$ ($\Phi = 2\pi f / f_E$)
- $d$: output signal at $f_E/2$ ($\Phi = 2\pi f / f_E$)
- $K = 2$
- $E_d = E_x + 2 \text{COV}_x$

where:

COV$_x$: correlation between the low frequency half-band spectrum and the high frequency half-band spectrum.

Practical experiments made on different types of music signals have shown that COV$_x$ is always negligible. It is then easy to compute the energy relations between all the signals of each direct and polyphase cells.

3. ROUND OFF NOISE IN SUBBAND SYSTEMS

In this section, we are going to determine the characteristics of the roundoff noise in the subband systems and its influence on the signal to noise ratio.

3.1 Generality on the roundoff noise

The implementation of the subband analysis and synthesis implies the utilization of DSP. These processors use the fixed-point arithmetic (FXP) or the floating-point arithmetic (FPL). Whatever the arithmetic chosen, the finite length of represented word leads to errors, called "roundoff errors". For that reason, each multiplication operator constitutes an error source in both FXP and FPL. However, addition operator only creates errors in FLP.
Real operators can be modeled [8] as ideal operators followed by a roundoff error source of variance $x^2_{FP}$ in the FXP case, $x^2_{LP}$ in the FLP case ($x^2$ is the variance of the signal being rounded). $x^2_{FP}$ and $x^2_{LP}$ are only depending on the length of representation (word for FXP, mantissa for FLP).

The influence of the roundoff noise on the signal to noise ratio at the output of the system is going to be investigated. This signal to noise ratio (SNR) is defined as follows:

$$\text{SNR} = 10 \log \frac{E_y}{E_o}$$

$E_y$ = energy of the output signal

$E_o$ = energy of the error signal at the system output.

3.2 Roundoff noise of systems implemented with the direct cells

The roundoff errors sources of the two direct cells are located at the output of each filter. Note that each FIR-OMF (direct or polyphase) filter introduces only one source because we suppose that the filter is implemented following the direct canonical structure [7] and that the DOP used is able to perform successive "multiply and add" with high accuracy (those operations are made with longer word/mantissa length of representation).

By hypothesis [8], all the roundoff sources are not correlated with the signal.

It can thus be shown that the SNR for FXP and FLP arithmetics in the case of $N$ stages subband system, is given by the following relations:

$$\text{SNR}_{FP} = 10 \log \frac{E_y^2}{x^2_{FP} 4(2^N - 1)} \quad (\text{dB}) \quad (2)$$

$$\text{SNR}_{LP} = -10 \log \frac{E_y^2}{x^2_{LP} 2(2^N - 1)} \quad (\text{dB}) \quad (3)$$

3.3 Roundoff noise of systems implemented with the polyphase cells

For the polyphase cells, noise sources are located at the filters output. More sources exist in the FLP case and are situated at the output of each addition operator.

Following the same consideration as in section 3.2, it is possible to demonstrate:

$$\text{SNR}_{FP} = 10 \log \frac{E_y^2}{x^2_{FP} 2(2^N - 1)} \quad (\text{dB}) \quad (4)$$

$$\text{SNR}_{LP} = -10 \log \frac{E_y^2}{x^2_{LP} 4(2^N - 1)} \quad (\text{dB}) \quad (5)$$

By comparing relations (2) and (4), on one hand, and relations (3) and (5), on the other hand, we conclude that the roundoff noise performances of the subband system depends of the implementation chosen.

3.4 Simulations

In order to verify the SNR presented in section 3.2 and 3.3, we have developed a simulation program on a PC. This program is able to compute noise variances by comparing results obtained on one hand, by an "ideal system" where arithmetic is realized with FLP "double" accuracy, and on the other hand, by a "real system" where arithmetic is realized with FLP "float" accuracy ($x^2_{FP} = -151.93$ dB) or FLP "int" accuracy ($x^2_{FP} = -10.79$ dB).

In the tables 1 and 2, we compare the results obtained using the relations (2) to (5) and by simulations ($x^2 = 70.57$ dB)

<table>
<thead>
<tr>
<th>N</th>
<th>DIRECT</th>
<th>POLYPHASE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory</td>
<td>Experiments</td>
</tr>
<tr>
<td>1</td>
<td>75.35</td>
<td>75.36</td>
</tr>
<tr>
<td>2</td>
<td>70.57</td>
<td>70.53</td>
</tr>
<tr>
<td>3</td>
<td>66.89</td>
<td>66.90</td>
</tr>
<tr>
<td>4</td>
<td>63.58</td>
<td>63.56</td>
</tr>
<tr>
<td>5</td>
<td>60.43</td>
<td>60.46</td>
</tr>
</tbody>
</table>

Table 1 (FPX)

<table>
<thead>
<tr>
<th>N</th>
<th>DIRECT</th>
<th>POLYPHASE</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Theory</td>
<td>Experiments</td>
</tr>
<tr>
<td>1</td>
<td>148.94</td>
<td>148.90</td>
</tr>
<tr>
<td>2</td>
<td>145.93</td>
<td>145.44</td>
</tr>
<tr>
<td>3</td>
<td>144.17</td>
<td>144.21</td>
</tr>
<tr>
<td>4</td>
<td>142.92</td>
<td>142.94</td>
</tr>
<tr>
<td>5</td>
<td>141.95</td>
<td>141.91</td>
</tr>
</tbody>
</table>

Table 2 (FLP)

These last tables show that experimental results are in perfect concordance with the theoretical predeterminations, excepted for the polyphase structure in FLP. A more detailed study shows that this difference comes from the analysis cells and is an effect of the correlation between signals processed by the addition operators.

4. LISTENING TESTS AND FIXED-POINT ARITHMETIC

The methodology presented in section 3 shows how to compute the output SNR when no subband processing occurs. This SNR can thus be considered as an upper limit of the SNR obtainable in subband coders.
As the SNR decreases as the number \( N \) of stages of the dichotomous tree structure increases, it may exist a limit above which noise in audio subband coders becomes audible.

We have put this problem in evidence for FXP arithmetic and polyphase implementation with a particular piece of music (N° 65 of the SGAM-EBU: R. STRAUSS).

The experiments have been led with a restitution station constituted by digital amplifier and headphones. The amplifier is calibrated in order that when we impose a certain gain "G", the acoustic level \( L \) of restitution in the headphone is related to the numeric variance \( \sigma^2 \) of the piece of music by the following relation:

\[
L(dB) = G(dB) + \sigma^2(dB)
\]

Listening tests have shown that, when \( G \) is equal to 31 dB, a shrill noise becomes audible when 5 stages are implemented. This noise is not audible when 4 stages or less are used.

Obviously, the noise origin comes from roundoff errors. Moreover, the shrill nature of the noise is explained with the help of the perceptual property of the ear known as the auditory threshold [9]. Indeed, only a small part of the noise spectrum of a 5 stages system (from 2500 Hz to 5000 Hz) is above the auditory threshold (cf. figure 5).

![Auditory threshold diagram](image)

Filter roundoff noise spectrum

Figure 5

To avoid these problems, many solutions can be found. We may use bigger length of representation, or use the FLP arithmetic. It is even possible to create subbands with different "spectral width" in order that the spectrum noise matches the auditory threshold.

5. CONCLUSIONS

A methodology is presented for the computation of the output filter roundoff noise of subband coding system based on "dichotomous tree structure". The computation of the output roundoff noise is made as a function of the number of subbands for the FXP and FLP representation, for the direct and polyphase implementation of the system. This allows to compare the influence of each feature. Moreover, in the case of fixed-point representation, listening tests show that the roundoff noise can be considered as an upper limit of coding noise performances of the subband systems.

REFERENCES

DESIGN OF FINITE WORDLENGTH 2-D IIR FILTERS USING SIMULATED ANNEALING

Jan Radecki, Janusz Konrad and Eric Dubois
Institut National de la Recherche Scientifique
INRS-Telecommunications
16 Place du Commerce, Verdun
Quebec, Canada, H3E 1H6

Abstract
This paper proposes a new approach to the design of two-dimensional (2-D) infinite impulse response (IIR) filters with finite precision coefficients. An objective function is proposed which combines magnitude, phase, step response and stability errors. This function being multidimensional and, in general, non-convex is minimized using simulated annealing. Development of this method constitutes the first step in a feasibility study of the application of 2-D IIR filters to the processing of video signals. Initial results on the design of low-pass filters are very encouraging and compare favourably with similar finite impulse response (FIR) designs.

1. INTRODUCTION
In this paper we propose a new approach to the design of finite wordlength 2-D IIR filters. This approach rests on a multiple-term objective function minimized using simulated annealing and is influenced by a potential application of the method to the design of filters for video processing.

As has been demonstrated in [1], 2-D non-separable digital filtering is a very attractive alternative to 1-D filtering (analog or digital) in certain video processing tasks. For example, in an NTSC encoder/decoder a 2-D digital filter is capable of precise band-limitation of luminance and chrominance components, thus substantially reducing component crosstalk in comparison with 1-D filters [1]. To date only FIR filters have been used for such applications. In this contribution we investigate a possibility of designing 2-D IIR filters instead. Such filters have a potential for substantial savings in filter complexity and thus in implementation costs.

The approach we propose is an extension of our earlier work on the design of finite precision FIR [2] and IIR [3,4] filters. It is based on simultaneous minimization of four error terms. Similarly to various other methods [5,6,7], the first three terms deal with magnitude response, group delay (or phase) and stability. The principle term describing selective properties of the filter is specified through constraints on the magnitude response in the frequency domain. Since IIR filters are in general characterized by a nonlinear phase response and since the human visual system (HVS) is sensitive to phase errors, the second error term ensures that this nonlinearity is as small as possible. The third error term enforces filter stability with a given minimum stability margin, which is particularly important in fixed-point hardware implementations. Unlike in other methods [5,6,7], however, our objective function contains the fourth error term which controls spatial properties of the filter by measuring step response overshoots. Such overshoots appear as ringing at sharp intensity transitions in an image, and are very objectionable to the HVS. The objective function resulting from the above constraints is multidimensional and, in general, non-convex. Moreover, due to the nature of calculations involved in evaluating step response overshoots, the fourth error term cannot be expressed analytically. Thus, to avoid local minima and to deal with step response overshoots we use simulated annealing to carry out minimization. This is in contrast with optimization methods used in other design methods of 2-D IIR filters.

2. STATEMENT OF THE PROBLEM
Recently Wan and Fahmy [8] have given necessary and sufficient conditions for quadrantal symmetry of a transfer function. Their result states that the denominator of a transfer function possessing quadrantal symmetry must be separable. Since it is usually required that magnitude response of a filter used in image (video) processing be of quadrantal symmetry (no preference in frequency orientations), the above result suggests that optimal separable-denominator filters can be designed.

Thus, we consider a 2-D IIR filter with separable-denominator transfer function defined as follows:

\[ H(z_1, z_2) = \frac{\sum_{m=0}^{M} \sum_{n=0}^{N} a(m, n) \cdot z_1^{-m} \cdot z_2^{-n}}{\prod_{k=1}^{L} \left[ \sum_{m=0}^{2} b(m, k) \cdot z_1^{-m} \cdot \prod_{l=1}^{K} \left[ \sum_{n=0}^{2} c(n, l) \cdot z_2^{-n} \right] \right]} \]  \hspace{0.5cm} (1)

The cascaded form above is characterized by a relatively low coefficient sensitivity to errors, and together with denominator separability it permits simple calculation of the stability margin. Our design method, however, is
general enough to handle arbitrary form of the transfer function \( H(x_1, x_2) \), also with a non-separable denominator. The major concern is the cost of computing the stability error.

Assuming that \( b(0, k) = c(0, l) = 1 \) for \( h = 1, \ldots, K, l = 1, \ldots, L \) we define the vector of unknowns \( a \) containing all \((M + 1)(N + 1) + 2(K + L)\) coefficients. Requiring that coefficients be represented by \( b \) bits and that the maximum absolute value of a coefficient be \( A \), it follows that the quantization step \( q \) and the coefficient domain \( S \) should be defined as
\[
q = \frac{A}{2^b - 1}, \quad S = \{ a : a = -A + qi, i = 1, \ldots, 2^b \}.
\]

Thus, the vector \( a \) belongs to the finite set \( S^{(M+1)(N+1)+2(K+L)} \).

Let \( \omega = (\omega_1, \omega_2) \in \mathbb{R}^2 \) be a 2-D frequency, \( H(e^{j\omega_1}, e^{j\omega_2}) = H(\omega) \cdot e^{j\phi(\omega)} \) be the frequency response of filter (1) and
\[
\tau_1(\omega) = -d\phi(\omega)/d\omega_1, \quad \tau_2(\omega) = -d\phi(\omega)/d\omega_2
\]
be its partial group delays with respect to \( \omega_1 \) and \( \omega_2 \), respectively. Based on the above discussion, our design method for 2-D IIR filters rests on the following minimization:
\[
\min_{\{a\}} \mathcal{E}(\mathbf{a}), \quad \mathcal{E}(\mathbf{a}) = \mathcal{E}_m + \lambda_1 \cdot \mathcal{E}_1 + \lambda_1 \cdot \mathcal{E}_2 + \lambda_1 \cdot \mathcal{E}_3, \quad (2)
\]
where \( \lambda_1 \)'s are weights and \( \mathcal{E}_1, \mathcal{E}_2 \) are error terms defined as follows:

1. magnitude response error \( \mathcal{E}_m \):
\[
\mathcal{E}_m = \max_{\omega \in \Omega} |\gamma_m(\omega) \cdot |H(\omega) - D(\omega)|/\gamma_{\max}(\omega),
\]
where \( D(\omega) \) is the desired magnitude response, \( \gamma_m(\omega) \) is a frequency-dependent weighting function, and \( \Omega \) is a discrete set of frequencies \( \omega \) over which magnitude and group delay errors are evaluated.

2. group delay error \( \mathcal{E}_1 \) (instead of minimizing phase nonlinearity the maximum spread of the group delay is minimized):
\[
\mathcal{E}_1 = \max_{\omega \in \Omega} \left\{ \max_{\omega \in \Omega} |\tau_1(\omega) - \tau_1(\omega)| - \min_{\omega \in \Omega} |\tau_1(\omega) - \tau_1(\omega)|, \right. \\
\left. \max_{\omega \in \Omega} |\tau_2(\omega) - \tau_2(\omega)| - \min_{\omega \in \Omega} |\tau_2(\omega) - \tau_2(\omega)| \right\},
\]
where \( \gamma_1(\omega) \) is a frequency-dependent weighting function used to deemphasize the importance of transition and stop bands.

3. stability error \( \mathcal{E}_2 \):
\[
\mathcal{E}_2 = \left\{ \begin{array}{ll} q/(1 - \delta) - |z_k|_{\max} & \text{for } |z_k|_{\max} < 1 - \delta, \\
\zeta & \text{for } |z_k|_{\max} \geq 1 - \delta,
\end{array} \right.
\]
where \( |z_k|_{\max} \) is the magnitude of the pole closest to the unit circle, \( \delta \) is a required stability margin, \( \zeta \) is a large positive value penalizing introduction of poles outside the circle with radius \( 1 - \delta \).

4. overshoot error \( \mathcal{E}_3 \) (maximum overshoot of the unit step response):
\[
\mathcal{E}_3 = \max_{n_1, n_2 \in \mathbb{Z}^2} |\eta(n_1) - \eta(n_2)|,
\]
where \( \eta \) is filter response to the 2-D corner unit step (first quadrant and positive axes are 1, otherwise 0) and \( N \) is a discrete set of 2-D spatial locations \( n \in \mathbb{Z}^2 \) (set of integers) over which the error is evaluated.

Since the error \( \mathcal{E}_3 \) is related to the transition bandwidth of the magnitude response of the filter, the overshoot constraint is in a partial conflict with the magnitude constraint, and as such must be used with caution. Step response overshoots are also related to group delay variations (different delays for different frequencies). Our intent is to partially trade-off overshoots in the step response for magnitude ripples. Such a trade-off is expected to be beneficial to image quality since a magnitude response ripple becomes dispersed over the whole image and thus is less objectionable than a well localized spatial overshoot. To achieve this goal the weight \( \lambda_0 \) must be carefully selected.

3. SOLUTION

The cost function \( \mathcal{E}(\mathbf{a}) \) under minimization is non-quadratic (no linear system can be solved), non-differentiable (no gradient-based technique can be used) and multi-variable. In fact, it is likely to have multiple minima, thus requiring global optimization. A method which we propose to use here is simulated annealing [9], which under certain conditions is able to localize the global minimum of an arbitrary cost function by executing a random walk in the space of solutions.

In the implementation of simulated annealing we have used the Metropolis algorithm [10] as follows. If \( a_1 \) is the coefficient vector at time \( t \) and \( a_{1t} \) is the vector after a change of one coefficient, then \( \Delta \mathcal{E}_i = \mathcal{E}(a_{1t}) - \mathcal{E}(a_1) \) is a cost function change due to a change of one coefficient. At every time instant \( t \) temperature \( T \) is chosen \((T \to 0 \text{ with } t \to \infty)\) and the following decision is made:

1. \( \Delta \mathcal{E}_i \leq 0 \Rightarrow \text{accept } a_{1t} \) unconditionally: \( a_{t+1} = a_{1t} \),
2. \( \Delta \mathcal{E}_i > 0 \Rightarrow \text{accept } a_{1t} \) with probability \( p = e^{-\Delta \mathcal{E}_i/T} \).

The process is repeated until \( T \) is sufficiently low, or as long as a certain criterion imposed on \( \mathcal{E}(\mathbf{a}) \) or on the acceptance ratio has not been met.

4. RESULTS

We have tested the above method by designing 2-D lowpass IIR filters of various orders and coefficient precisions, as well as different pass-band shapes such as diamond, hexagonal, circularly symmetric. Below we present some results obtained for chrominance filters used in modulation and demodulation of the NTSC composite signal. Those filters are used to band-limit the chrominance I and Q signals in order to reduce component cross-talk [1], and thus are called I- and Q-filters, respectively. We have designed those filters with 8-bit
coefficient precision, and we have compared them with corresponding 8-bit FIR filters proposed in [1].

To deemphasize the importance of exact magnitude value in the transition band and of the group delay constancy in the transition and stop bands, the weights \( r_m \) and \( r_r \) have been appropriately shaped. The overshoot error \( E_o \) has been implemented thus far by finding the tallest peak (deepest valley) closest to the step response origin and then by searching its neighbourhood, but staying on the same side of the step response rise, for minimum (maximum) accessible by a monotonic descent (ascent).

Table 1 shows principle parameters of the designed filters. All filters have been designed with coefficient precision of 8 bits. The IIR filters have been designed using the method proposed here, while the FIR filters have been obtained by the method proposed in [2]. The IIR filters have 33 coefficients altogether (\( M = N = 4 \), \( K = L = 2 \)), while the FIR filters have 165 coefficients (15 by 11) out of which 48 are independent. In our experience, higher order (8-bit lowpass I and Q) FIR filters do not provide significant improvement due to additional coefficients being quantized to zero, which is not the case with IIR filters. Fig.1 shows the desired magnitude response \( D(\omega) \) of the I-filter, while Fig.2 shows magnitude responses of filters designed to approximate \( D(\omega) \). Note somewhat larger departure of the IIR filter’s magnitude response from the desired response, which is confirmed by larger pass-band ripples and smaller stop-band attenuation (Table 1). Fig.3 shows the partial group delay \( \tau_1 \) of the IIR filter. Not the almost constant value of \( \tau_1 \) in the pass-band (center of the plot). In fact, the maximum group delay spread in the pass-band is well below 1 sample (Table 1). Fig.4 shows responses of both filters to the 2-D corner unit step (Section 2). Qualitatively, it seems that the IIR filter has smaller overshoots. The value of \( E_o \) from Table 1 confirms this observation.

<table>
<thead>
<tr>
<th>I-filter</th>
<th>Q-filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>IIR</td>
<td>FIR</td>
</tr>
<tr>
<td>ripple (pass) [dB]</td>
<td>0.73</td>
</tr>
<tr>
<td>atten. (stop) [dB]</td>
<td>19.5</td>
</tr>
<tr>
<td>( \tau_{max} - \tau_{min} ) (pass)</td>
<td>0.30</td>
</tr>
<tr>
<td>( \tau_{2max} - \tau_{2min} ) (pass)</td>
<td>0.52</td>
</tr>
<tr>
<td>1 - (</td>
<td>z_k</td>
</tr>
<tr>
<td>( E_o )</td>
<td>0.104</td>
</tr>
</tbody>
</table>

Table 1 Comparison of parameters of 8-bit FIR and IIR filters for I and Q specifications

We have tested the designed filters on real video images, especially scrutinizing areas of sharp luminance transitions. The images produced by IIR and FIR filters were almost indistinguishable. A tiny difference in image “softness” was due to slightly different magnitude responses of the filters, which can be compensated for. The IIR filters, however, have performed better in terms of step response overshoots. Spurious “ghosts” were slightly less visible, which indeed is a desirable property of a video filter.

In conclusion, our study indicates that 2D IIR filters may become a viable alternative to FIR filters in video processing, thus reducing computational costs involved.

REFERENCES

Fig. 1  Desired magnitude response $D(\omega)$ of IIR filter

Fig. 3  Group delay $\tau(\omega)$ of IIR IIR filter

(a) 

(b) 

Fig. 2  Magnitude response of IIR (a) and FIR (b) filter

(a) 

(b) 

Fig. 4  Step response of IIR (a) and FIR (b) filter
A simple design method for FIR filters with integer coefficients using a personal computer

Shogo NAKAMURA, Yasumasa MORISHITA, Yoshihiko HORIO
Akiyoshi KAWAHASHI, Yukio KADOWAKI *

Department of Electronic Engineering, Tokyo Denki University, 2-2 Kanda
Nishiki-cho, Chiyoda-ku, Tokyo, 101, Japan

*LSI R & D Center Electronics Devices Division Ricoh Co., LTD,
13-1 Himemuro-cho, Ikeda, Osaka, 563, Japan

FIR filters with linear phases are useful in various fields. However, when we desire the sharper filter characteristics, its hardware and computational complexities inevitably increase so that an available frequency band is limited. Therefore, obtaining a fast filtering architecture of the FIR filters becomes a valuable task.

This paper mainly describes a simple design method of a lower order FIR filter with integer coefficients by using a conventional personal computer.

1. Introduction

There were several efforts to get FIR filters with integer coefficients.[1],[2] Lim proposed a design method using mixed integer linear programming in [1] and provided his nice program. However, since the program is large and takes a lot of computational times, it is oriented to a big computer such as a supercomputer. Sometimes, a simple design approach is desirable in many applications of the real fields. Therefore, as another approach, the method to achieve the linear phase FIR filter with good properties by realizing a hierarchical building block structure by using a lot of lower order FIR subfilters has been proposed.[3],[4] This paper mainly describes a simple design method of the above lower order FIR subfilter with integer coefficients by using a conventional simple personal computer.

Although the proposed approach is not optimal, it is considered that the results obtained are sufficiently good for practical applications. It is also shown by two different examples that this approach is useful.

2. Outline of the proposed FIR filter

Let us explain the properties of the proposed filter of the hierarchical building block structure. This basically has the structure of tapped-cascaded subfilters with linear phases. The linear phase FIR filter of a characteristic which is better than that of the subfilter, can be obtained by changing tapping coefficients appropriately. When the above mentioned filter is regarded again as a subfilter, the filter design can be achieved by using the building block structure such as shown in Fig.1.

Since the processings of a lot of identical subfilters and the tapping coefficients can be executed in parallel, this kind of filter is suitable for a high speed processing and an LSI implementation. For the high speed processing, the arithmetic operations of the subfilter and the tapping parts must be simple. In addition, the use of the multiplier is not desirable from the point of view of LSI implementation because it needs large chip areas in the LSI. So, the adder-based architecture was adopted.
3. A simple design approach

The proposed design is based on a linear equation set which comes from frequency specifications of a desired filter. First, the frequency sampling points which give the desired value of 1 in the pass-band and the value of 0 in the stop-band, are estimated by using the transition band width of a given filter specifications. Thus the frequency sampling points are selected with the same interval of the transition band width from the pass band and the stop band edges such as shown in Fig.2.

\[ \omega_{\text{dif}} = \omega_{\text{stop}} - \omega_{\text{pass}} \quad (1) \]

In Fig.2, \( N_p(1) \) means the nearest frequency sampling point of the pass-band edge and \( N_s(1) \) also means the nearest frequency sampling point of the stop-band edge. \( N_p(M_p) \) and \( N_s(M_s) \) represent the closest sampling points at \( \omega = 0 \) and \( \omega = 0.5 \), respectively. If \( n = M_p + M_s \) sampling points are chosen, they are substituted in the following frequency transfer function so that the \( n \) linear equations are obtained.

\[ H(e^{j\omega}) = h_0 + 2 \sum_{i=1}^{n} h_i \cos(i\omega) \quad (2) \]

\[ k = 1, 2, 3, \ldots, n \]

To solve this simultaneous equation, one more equation is needed. Another frequency sampling point \( \omega_1 \) is taken at the center of the transition band of the filter. Next, the conditional number of this simultaneous equation is calculated. The conditional number \( \text{Cond}(A) \) is defined by the following relation:

\[ \text{Cond}(A) = \|A\| \cdot \|A^{-1}\| \quad (3) \]

where \( A^{-1} \) is the inverse matrix of \( A \), \( A \) is the coefficients matrix of the simultaneous equation such as:

\[ A = \begin{bmatrix}
1 & 2\cos(\omega_1) & 2\cos(2\omega_1) & \ldots & 2\cos(n\omega_1) \\
1 & 2\cos(\omega_2) & 2\cos(2\omega_2) & \ldots & 2\cos(n\omega_2) \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
1 & 2\cos(\omega_n) & 2\cos(2\omega_n) & \ldots & 2\cos(n\omega_n)
\end{bmatrix} \]

\[ \|A\| = \sum_{j=1}^{n} \|a_{ij}\| \quad (4) \]

and \( \|A\| \) is a norm of \( A \) which is defined such as:

\[ \|A\| = \sum_{j=1}^{n} \|a_{ij}\| \quad (5) \]

where \( a_{ij} \) is an element of matrix \( A \).
Fig. 2 An example of initial frequency sampling points.

Now, the original frequency sampling points except for \( \omega_{m} \) are modified by changing the interval \( \omega_{\text{dif}} \) between the frequency sampling points by the eq.(6).

\[
\begin{align*}
N_{p}(i+1) &= N_{p}(i) + \text{DIF}(i) \cdot \text{FACT} \cdot \omega_{\text{dif}} \\
N_{s}(j+1) &= N_{s}(j) + \text{DIF}(j) \cdot \text{FACT} \cdot \omega_{\text{dif}} \\
i &= 1, 2, ..., N_{p}(M_{p} - 1), \\
j &= 1, 2, ..., N_{s}(M_{s} - 1)
\end{align*}
\] (6)

where \( \text{DIF}(i) \) or \( \text{DIF}(j) \) was obtained experimentally such as shown in Fig.3 from the result of the equi-ripple FIR filter design due to the Remez Exchange Algorithm [5] and FACT is a constant number to vary from 0.9 to 1. The conditional number of eq.(2) is repeatedly calculated by using a new frequency sampling points depending on the different values of FACT. When the minimum conditional number is obtained, the corresponding frequency sampling points are fixed in order to solve the coefficients \( \{ h_{i} \} \) of eq.(2).

Solving eq.(2) by using the above coefficients, the coefficients \( \{ h_{i} \} \) of the subfilter can be obtained.

Finally, the coefficients obtained are rounded to several bits. Since this procedure is very simple, the use of a conventional simple personal computer is allowed for the design of the FIR filter.

4. A design example

Figures 4 and 5 show examples designed with different band edges. The relation between the conditional number and the deviations from the desired values can be found from two figures.

![Fig.4 (a) The result of a small conditional number.](image)

![Fig.4 (b) The result of a large conditional number.](image)
Thus, it is observed that the small conditional number offers less quantization effect. The specifications of the filters in Fig. 4 is such as:

Filter length : 11,  max. deviation : 0.1  
Pass-band edge : 0.05  
Stop-band edge : 0.15

The other filter in Fig. 5 is described below such as:

Filter length : 11,  max. deviation : 0.1  
Pass-band edge : 0.15  
Stop-band edge : 0.25

Each frequency-magnitude property is plotted by using the filter coefficients quantized to 7 bits which are also represented in the figures.

6. Conclusion

The authors proposed an approach to simplify the design of the FIR filter. The advantage of the proposed method is that since the basic architecture is a tapped cascaded subfilter structure, the direct design problem of the FIR filter is replaced by design of the simple subfilter, and the design of the subfilter with integer coefficients can be simply performed by using a conventional personal computer.

7. Acknowledgment

The authors wish to express their gratitude to the Research Institute for Technology, Tokyo Denki University and Ricoh Co. LTD. for their financial support.

8. References

SYNTHESIS METHODS FOR WAVE DIGITAL FILTERS EXHIBITING
ARBITRARY AMPLITUDE CHARACTERISTICS

M. Yaseen * and T. Henk
Dept. of Telecommunications and Telematics,
Technical Univ. of Budapest, HUNGARY

Design conceptions are delivered for several wave digital filter (WDF)
structures satisfying arbitrary amplitude specifications given at all
frequencies, i.e., the transition band can also be specified beside the
passband and stopband. The approximation problem is solved using the
interpolation methods combined with the Remez-exchange algorithm.
The different conceptions are compared from the wave realization cost,
sensitivity, dynamic range and nonlinearity points of view.

1. INTRODUCTION

WDF's exhibit excellent properties, including coefficient sensitivity, roundoff noise
overflow level, stability and nonlinear effects under finite-arithmetic conditions [1,2]. The
robustness of WDF's is insured by their passive [3,4] reference structures and specially the
losslessness of the filter two-port [1]. The losslessness within passivity is insured if the
approximated transmission is unity and zero in the passband and stopband respectively.
Otherwise there is no direct relationship between losslessness and low coefficient
sensitivity. This is why these properties are usually not investigated in the transition band
and also in the passband and stopband if the transmission is not unity and zero respectively,
but are in the focus of this paper.

In many applications, e.g., data transmission, sampling, etc., the amplitude characteristic
A(ω) is specified at all frequencies, i.e., also in the transition band while the
transmission is also a given function in the passband. To solve this filter design problem
optimally, design conceptions are delivered for several WDF structures satisfying arbitrary
amplitude specifications given at all frequencies in form of lower and upper
amplitude limit functions:

\[ A_L(\omega) \leq A(\omega) \leq A_U(\omega) \]  \hspace{1cm} (1)

Approximation methods are developed for the considered WDF structures by applying the
interpolation method combined with the Remez-exchange algorithms. The design
conceptions are compared from the wave digital realization cost, sensitivity, dynamic range
and nonlinearity point of view to find an optimal solution.

2. DESIGN CONCEPTIONS

The first design conception is relying on a structure composed of only one reactant
filter. This one filter can be restricted to possess odd degree lattice structure if the
relationship,

\[ A_1(\omega) \equiv A(0) \]  \hspace{1cm} (2)

holds for the approximated amplitude characteristic. In such a case, bireciprocal
filter structure can be employed if further

\[ A^2(\omega)+A^2(\omega,F-\omega)=A^2(0) \]  \hspace{1cm} (3)

holds, where F is the sampling rate. In case of

\[ A(\omega) > A(0) \text{ in some band,} \]  \hspace{1cm} (4)

the constructed filter must be of ladder structure.

The second design conception is relying on a structure composed of cascaded filter and
amplitude equalizer [5]. The third design conception is relying on a structure composed of
two cascaded filters. The first filter in the second or third conception is designed to
satisfy the biggest part of the selectivity as a prototype filter and can be restricted to
possess odd degree lattice structure or maybe bireciprocal one while an amplitude equalizer
or a second filter of lower order insures the final shape of the characteristics. In case of
two cascaded filters, the second filter can be lattice only if

\[ A(\omega) \equiv A(0) \]  \hspace{1cm} (5)

* On leave from Dept. of Electrical Eng., Univ. of Assiut, Assiut, EGYPT.
3. APPROXIMATION AND REALIZATION

The approximation problem is solved by applying interpolation methods combined with the Remez-exchange algorithm similarly to Ref. 6. The lattice and bireciprocal solutions may become of nonminimum phase [7]. If the specification is given in the digital p or \( \omega \) frequency domain, it is transformed to the reference \( \psi \) or \( \omega \) frequency domain by the bilinear transformation [1]:

\[
\psi = \frac{(z-1)/(z+1)}{z = e^{jT}}, \quad \phi = \tan(\omega T/2), \quad T = \text{sampling period}.
\]  

(6)

The approximation procedure is summarized for each conception as follows.

3.1. One Reactant Filter

According to this conception the given specification is approximated by only one filter. The approximation is carried out by constructing the characteristic function \( \Psi(\psi) \):

\[
\Psi(\psi) = h(\psi) / f(\psi) = S_{11}(\psi)/S_{21}(\psi),
\]

(7)

where \( S_{11}(\psi) \) is the reflectance and \( S_{21}(\psi) \) is the transmittance.

In case of lowpass lattice filter, \( h(\psi) \) is an odd polynomial with degree \( n \), which is the degree of the filter and it is an odd number:

\[
(n-1)/2
\]

\[
h(\psi) = \psi \sum_{i=0}^{(n-1)/2} c(i) \psi^{2i},
\]

(8)

with its roots are not restricted to lie on the \( \omega \)-axis. On the other hand, \( f(\psi) \) is an even polynomial of degree \( n-1 \):

\[
(n-1)/2
\]

\[
f(\psi) = \sum_{i=0}^{(n-1)/2} \psi^{2i} = \prod_{i=1}^{(n-1)/2}(\psi^{2} + \beta^{2}(1)),
\]

(9)

so, it is restricted to possess \( \omega \)-axis roots only with transmission zeros \( B(1) \).

The specification for \( \Psi(\psi) \) is obtained from that of the transmittance by:

\[
|\Psi(\omega)|^2 = 1/|S_{21}(\omega)|^2 - 1
\]

(10)

The design procedure is summarized as follows:

1) Calculate the specification of \( \Psi(\omega) \) from that of the transmittance and select a degree for the filter.

2) Select proper values for the transmission zeros \( B(1) \).

3) At a selected set of frequencies determine \( c(1) \) by interpolating \( |\Psi(\omega)| \) in the passband and transition band if specified.

4) By applying the Remez-exchange algorithms, move the interpolation frequencies and recalculate \( c(1) \) for optimal passband and transition band if specified.

5) With the obtained \( c(1) \), apply the Remez-exchange algorithm for new optimal transmission zeros \( B(1) \).

6) Go back to step 3 to re-satisfy the passband and transition band if specified. Continue till the specification is satisfied in the three bands. If the selected degree is not sufficient select new one and go to step 2.

In case of bireciprocal structure, the function \( \Psi(\psi) \) is decomposed:

\[
\Psi(\psi) = \Psi_{1}(\psi) \Psi_{2}(\psi),
\]

(11)

\[
\Psi_{1}(\psi) = \prod_{i=1}^{k/2}(S_{21}^{2}(1)\psi^{2i+1}), \quad \Psi_{2}(\psi) = \prod_{i=1}^{m/2}(\psi^{2} + \beta^{2}(1)) \sum_{i=0}^{m/2} c(i) \psi^{2i},
\]

(12)

such that the first of them is restricted to possess roots on the \( \omega \)-axis only while the second is not restricted on the \( \omega \)-axis. In (12), \( k \) and \( m \) are related to the filter degree \( n \) by:

\[
n = k+m+1,
\]

(13)

furthermore \( B(i) \) and \( c(1) \) are redefined. The design procedure is similar to that of the lattice structure.

In case of

\[
A(\omega) > A(0) \text{ in some band},
\]

(14)

the one filter structure must be of ladder type. In such a case it is approximated by constructing its squared magnitude transmittance:

\[
S_{21}(\psi)S_{21}(-\psi) = \frac{1}{\prod_{i=1}^{n} \psi^{2} + \beta^{2}(1))^{2}} = \frac{N(\psi)}{D(\psi)}, \quad \psi = j\phi
\]

(15)

where \( B(i) \) are the corresponding transmission zeros, furthermore \( B(1) \) and \( c(1) \) are redefined again. The design procedure for \( B(1) \) and \( c(1) \) is similar again to that of the lattice structure. The transmittance of the filter is obtained by the factorization of \( N(\psi) \) and the Hurwitz factorization of \( D(\psi) \).

To realize a lattice filter from its \( \Psi(\psi) \), the following expression is considered:
\[ h(\psi) + f(\psi) = g_1(\psi) g_2(-\psi), \]

where the denominator \( g(\psi) \) of the transmittance \( S_{21}(\psi) \) is related to \( g_1(\psi) \) and \( g_2(\psi) \) by

\[ g(\psi) = g_1(\psi) g_2(\psi). \]

So, by considering the Hurwitz and anti-Hurwitz factors of \( h(\psi) + f(\psi) \), the polynomials \( g_1(\psi) \) and \( g_2(\psi) \) respectively can be determined.

The two all-pass functions of the lattice branches are calculated as:

\[ S_1(\psi) = -g_1(-\psi)/g_1(\psi), \quad S_2(\psi) = g_2(-\psi)/g_2(\psi) \]

and they are realized by factorizing them into cascaded sections of first and second orders [2]. In the bireciprocal case however fourth order sections are also needed [8].

In case of ladder structure, the use of Fujisawa’s theorems [9] results in a lossless ladder two-port terminated with a resistance.

3.2. Prototype Filter Cascaded with Equalizer

According to this conception, explicit formulas [2] can be used to design the prototype filter. The loss characteristic of this filter is subtracted from the original specification to get new specification for the required amplitude equalizer. The equalizer is approximated through the construction of its squared magnitude transmittance:

\[
S_{21}(\psi)S_{21}(-\psi) = \frac{\sum_{i=0}^{m} n(i) \psi^{2i}}{\sum_{i=0}^{m} d(i) \psi^{2i}} = \frac{N(\psi)}{D(\psi)}
\]

\[ = f(\psi)f(-\psi)/g(\psi)g(-\psi), \quad \psi = j\phi \]

where \( g(\psi) \) is a strictly Hurwitzian polynomial, while there is no restriction for \( f(\psi) \). The approximation procedure is summarized as follows:

1) Select the degrees for the filter and the equalizer.
2) Design an odd degree all-pass filter to satisfy the biggest part of selectivity.
3) Get the specification of the equalizer.
4) At a selected set of frequencies interpolate the squared magnitude of the equalizer for having \( n(1) \) and \( d(1) \).
5) Move the interpolation frequencies for optimal amplitude characteristic for new \( n(1) \) and \( d(1) \).
6) Investigate a new lattice filter with changed passband and stopband requirements but with the same degree to find the optimal lattice filter and go to step 3. Continue till the specification is satisfied in the passband, stopband and transition band if specified. If the degrees are not sufficient, select new degrees and go to 2.

7) The transmittance of the equalizer is obtained by any factorization for \( N(\psi) \) and the Hurwitz factorization of \( D(\psi) \).

To realize the equalizer its transmittance is equal to the reflectance of a lossy one-port [4]. The realization is carried out by cascaded sections of first and second orders.

3.3. Two Cascaded Filters

According to this conception, the given specification is approximated such that the first filter with higher order and prototype specification is constructed to satisfy the biggest part of selectivity. This filter can be restricted to possess odd-degree lattice structure or maybe bireciprocal one. Explicit formulas [2] can be used to design this filter. The loss characteristic of this filter is subtracted from the original specification to obtain a new specification for a low order second filter. The second filter can be restricted to be lattice if

\[ A(\omega) = A(0), \]

otherwise it must be of ladder type. The approximation procedure is similar to that of the cascaded filter and equalizer conception.

4. SIMULATION

Considering a model consisting of A/D, digital filter and D/A, the fixed point arithmetic is used with 12 digits for the truncated coefficient wordlength and for A/D and the register wordlength. The magnitude truncation or 2’s complement truncation is applied at the adapter outputs as appropriate [2]. The developed structures are investigated by means of typical examples from scaling, dynamic range, sensitivity and nonlinearity point of view.

5. DESIGN EXAMPLE

A monotonic specification of a data transmission demodulator filter is approximated by 9-th order bireciprocal filter , F=64 kHz. The loss response with the specification can be seen in Fig. 1a. The WD realization is shown in Fig 1b with the following values for the adaptor coefficients:

\[ y_1 = -0.1108907 \quad y_2 = 0.4170397 \quad y_3 = -0.2912512 \]
\[ y_4 = 0.66324614 \]
The dynamic range of this structure is 63.35 dB and its signal to noise ratio in the passband is 62.0 dB.

6. CONCLUSIONS

Several design conceptions are delivered for WDF's exhibiting arbitrarily specified loss characteristics and they are compared by means of typical examples. Comparing them from the number of adaptors point of view the conception of one bireciprocal or two lattices and the conception of bireciprocal + ladder or lattice+ladder are the optimal solutions for \( A(\omega) \leq A(0) \) and for \( A(\omega) > A(0) \) in some band respectively. Comparing them from dynamic range point of view the conception of two filters gives the optimal value for both \( A(\omega) \leq A(0) \) and for \( A(\omega) > A(0) \) in some band respectively. Comparing them from the nonlinearity point of view the conception of two filters gives the optimal behavior. The use of equalizers results in more attenuation in the stopband but at the cost of higher number of adaptors. The two filter solution enables us in case of \( A(\omega) \leq A(0) \) to design a cascaded two lattice structure with total even degree which almost is lower than the degree of one lattice filter structure. The use of two filter conception enables us to restrict the biggest part of the filter structure to be lattice in case of \( A(\omega) > A(0) \) in some band.

REFERENCES

[8] Wegener, W., Wave Digital Filters with Reduced Number of Multipliers and Adders, AEU, Band 33, 1979, Hef 6, pp. 239-243.
EFFICIENT TEMPORAL BAND-LIMITATION BASED ON MOTION ADAPTIVE SPATIAL FILTERING

JONG-HUN KIM, SEONG-DAE KIM

DEPT. OF ELECTRICAL ENGINEERING, VISUAL COMMUNICATIONS LAB., KAIST
373-1, KUSEONG-DONG, YUSEONG-KU, DAEGU 305-701, KOREA

Most of coding applications have been developed without concerning about temporal aliasing which causes visual artefacts such as motion judder, stroke-width modulation, etc. This paper describes a band-limiting temporal filter which is not affected by the aliasing. The proposed filter, Motion Adaptive Spatial Filter, is employed on spatial domain in direction of movement. By performing temporal band-limitation efficiently, it will be applicable to image sequence coding as a prefilter.

1. INTRODUCTION

Unlike a noise removal recursive temporal filter[3], we have different point of view how we can perform a temporal band-limitation efficiently. Recent progress on HDTV which combines spatial and temporal subsampling uses an adaptation of their resolutions according to the movement. If a lowpass temporal filter is applied, more improvement of their quality will be achieved. Moreover, most of coding applications have been studied without the concern of visual artefacts resulting from temporal aliasing. Therefore, it is necessary to investigate a band-limiting temporal filter not affected by the aliasing.

In words, this paper focuses on the elimination of the temporal aliasing components. To devise the filter, de-aliasing temporal filter, we assume that moving objects in image sequence[6] have only constant-velocity rigid translational motion and we restrict it non-interface scanning, since interface scanning can be easily converted to non-interface one by de-interlacing techniques[4]. In the following, for simplicity, lowpass temporal filter will be mainly considered, since other filters can be anticipated by the use of lowpass characteristics.

2. EFFECTS ON TEMPORAL DOMAIN FILTERING

Video signal can be regarded as a three-dimensional one which has horizontal, vertical and temporal components. Let \( f(x,y,t) \) be a three-dimensional continuous video signal. Assuming that its moving components have only constant-velocity rigid translational motion \( v = (v_x, v_y) \), Fourier transform of \( f(x) \) is represented by:

\[
F_3(U_x, U_y, f_t) = F_2(U_x, U_y) \cdot \delta(U_x v_x + U_y v_y + f_t),
\]

where \( F_2(f_x, f_y) \) is a two-dimensional image function, and \( \delta(U_x v_x + U_y v_y + f_t) \) is a tilted plane in 3D frequency space described by the equation \( U_x v_x + U_y v_y + f_t = 0 \), i.e., baseband exists only on the plane. (1) indicates that the greater spatial details or velocity are, the more resultant temporal components will be. Thus, sharp pictures may produce a greater proportion of high temporal frequencies than soft pictures.

(1) is description of a continuous case. Since the used pixels for filtering are 3D sampled data, sampling of \( f(x) \) is expressed by multiplying \( f_3(x,y,t) \) by 3D array of delta functions. Spectrum of the sampled signal is then given by the convolution of Fourier transform \( f_3(x,y,t) \) and delta function. Thus, it is known that spectrum aliasing between the repeated spectras may occur according to increase of the velocity. If the repeated spectras include aliasing components, visual artefacts appear. Especially, moving areas composed of spatial high frequency components may distort psychovisual effects.

To derive procedure of the de-aliasing temporal filtering, we made two assumptions: first, baseband spectrum has no spatial aliasing components, and secondly there exists only purely horizontal motion. Filtered results based on temporal domain samples are shown in Fig. 1(a). They contain spatial high frequency components of adjacent spectras which is temporal aliasing. We previously stated that the presence of spatial high frequencies results in high temporal frequencies. However, Fig. 1(a) presents that the spatial high frequency components produce temporal low fre-

![Fig. 1 Comparisons of temporal bandlimitation.](image)

(a) Temporal domain filtering.
(b) Spatial domain filtering (MASF).
quency ones. In other words, the display causes a disturbance between the spatial high frequency components and the low frequency ones. Therefore, reducing of the effects introduced by temporal aliasing (e.g. motion judder, stroke-width modulation) is desirable.

The attenuation afforded to spatial high frequencies has a great effect in reducing temporal aliasing. Although it is allowable, it may cause severe loss of spatial informations. But on the other hand, higher temporal sampling rate is able to achieve an attenuation of temporal aliasing. If given sampling rate is however low, it requires temporal sampling rate conversion which is known to be ill-posed problem. As a result, it is necessary to develop a sophisticated and general approach.

3. DE-ALIASING TEMPORAL FILTER

Using the relationship between the location of baseband spectrum and the velocity, temporal band-limitation can be described by a spatial filter, i.e., lowpass spatial filtering with its cutoff frequency corresponding to the temporal one provides same effects. From the location of baseband spectrum in (1), a relation between spatial and temporal frequencies is given by,

\[ f_{\text{cutoff}}^x = \frac{1}{|v|} f_{\text{cutoff}}^t. \]  

(2)

Compared with a temporal band-limitation using temporal domain pixels of Fig. 1(a), the MASF has de-aliasing characteristic. Temporally band-limited results which do not include temporal aliasing are shown in Fig. 1(b).

This result, de-aliasing characteristic, can be applicable to any types of band-limitations. For examples, though filter types are given by highpass or bandpass, the MASF will preserve the de-aliasing characteristic. They are illustrated in Fig. 2.

Previously mentioned investigation is one-dimensional case which has only horizontal velocity. Extending the velocity into two-dimension, spatial filtering in terms of the velocity is equal to the filtering along movement trajectory, which results in Motion Adaptive Spatial Filter (MASF). Two-dimensional case is shown in Fig. 3.

Fig. 2 Examples of temporal bandlimitation based on MASF.
(a) Highpass filtering, (b) Bandpass filtering.

With the motivation of the spatial filtering using a velocity, an implementation of a MASF is undertaken. Let \( h(t) \) be lowpass impulse response, then the temporally band-limited form of \( f(t) \) is given by

\[ g(x,t) = \int_{-\infty}^{\infty} h(\tau) f(x,t-\tau) d\tau, \]  

(3)

where a linear phase filter is used concerning the group-delay of response. From an assumption of motion,

\[ f(x,t-t_v) = f(x-v_x t_v, t). \]  

(4)

Thus, (3) is represented by

\[ g(x,t) = \int_{-\infty}^{\infty} h(\tau) f(x-v_x \tau, t) d\tau. \]  

(5)

And its Fourier transform is expressed by

\[ G(f_x,f_y) = H(f_x' - f_x) \cdot F(f_x,f_y). \]  

(6)

\( H(\cdot) \) had initially temporal band-limiting characteristics. Filtering characteristics however vary by a conversion of filtering domain. Since the filtering operation on spatial domain is changed according to the velocity, \( H(\cdot) \) performs a lowpass spatial filtering whose cutoff frequency is adapted by the velocity. Therefore, direct realization of (5) results in de-aliasing temporal filter (MASF).

\( (5) \) is a continuous description of the MASF. Similar results hold in the discrete case : the integral is replaced by summation and \( d\tau \) is represented by combination of \( \Delta \tau \) and \( k \). (5) is then given by

\[ g(x,n) = \sum_{k=-N}^{N} h(k\Delta\tau) \cdot f(x-n\Delta\tau \cdot k, n), \]  

(7)

where velocity and filtering positions are replaced by vector form. And filter length is \( 2N + 1 \), \( \Delta \tau \) is selected to satisfy \( |v(\cdot)| \Delta\tau | \leq |\Delta x| \). If \( \Delta\tau \) is different from the required value, it may cause spatial aliasing.

Let \( \Delta T \) be a frame to frame interval, \( \psi(\cdot)\Delta T \) is then equal to \( \Delta(\cdot) \) which is a displacement between current and previous frames. If \( \Delta(\cdot) \) is defined as a ratio of \( \Delta T \cdot \alpha(x,n) = \Delta T, \)
\[
\frac{D(x,n)}{\alpha(x,n)} = d(x,n), \quad |d(x,n)| = |\Delta x| = 1,
\]

(7) is modified as a following description.

\[
g(x,n) = \sum_{k=-N}^{N} h(k) \cdot f(x - d(x,n) \cdot k,n),
\]

(8)

where \(\Delta r\) is normalized to 1, and \(h(\cdot)\) is determined by (2). (8) is a direct realization of (5). However, due to \(\alpha(\cdot)\) and \(d(\cdot)\), actual data for filtering may be non-grid pixels. Concerning the computation and efficiency, bi-linear filter which is weighted average with nearest neighbor pixel is used.

4. EXPERIMENTAL RESULTS

A zone-plate can be used to study the frequency response of a filter. It has the appearance of concentric rings whose spacing decreases away from the center. And the center of rings is periodically repeated in horizontal or vertical directions. To evaluate a de-aliasing characteristic of the proposed filter, a moving zone-plate without periodicity is generated.

Assuming that velocity function of the moving zone-plate is constant, equation (8) as a temporal filter is evaluated. To verify the de-aliasing characteristic of the MASF, lowpass filtering is compared with direct filtering of Fig. 1(a). Fig. 4 shows that the MASF possesses efficient temporal band-limitation. The MASF results in blurring toward moving direction. The blurring is necessary consequence of temporal band-limitation without severe loss of spatial informations and are not perceived at typical display (30Hz ...).

For an application of real image sequence, "Miss America" is used. Fig. 5 presents filtered result as a function of temporal cutoff frequency, where motion vector based on BMA is used. Fig. 5(d) gives the result with temporal cutoff frequency-15Hz which is half of the temporal sampling frequency. It indicates maximum bound to eliminate the aliasing. Thus, there are no spectral components of adjacent spectra in any Nyquist region. This filtered sequence is applicable to image coding.

Fig. 4 De-aliasing characteristic of MASF.

Each plane indicates Horizontal-Vertical and Vertical-Temporal planes, respectively. The velocity is assumed to be 5 pixel/frame interval, and normalized temporal cutoff frequency is 0.25.

(a) Original planes,

(b) Filtered planes based on temporal domain filtering,

(c) Filtered planes based on spatial domain filtering.
Fig. 5 Temporally band-limited results.
Original frame, (a) (b) (c) (d) indicate filtered frames when normalized temporal cutoff frequency is 0.125, 0.25 and 0.5, respectively.

Clearly, lowpass filtering results in the increase of redundancy for image coding. Thus, data compression efficiency may be improved. Results applied to a motion compensated predictive coder is shown in Fig. 6. As can be seen, the PSNR (Peak Signal to Noise Ratio) for the filtered sequence (Fig. 5(d)) is reduced whereas the entropy is high. Consequently, the MASF can be used as a prefilter for image sequence coding.

5. CONCLUSIONS

In this paper, effects of temporal aliasing and de-aliasing temporal filter are investigated. Unlike a conventional temporal filter, the proposed filter, MASF, achieves efficient temporal band-limitation. Additionally, temporal noises on moving areas are eliminated since they can be regarded to high temporal frequency components. To construct a temporal filter which is not affected by aliasing components, we performed temporal filtering on spatial domain along movement trajectory. The MASF creates blurring on moving areas (e.g. motion blur) resulting from temporal hard-limitation and thereby reduces visual artefacts at sequential display.

The MASF achieves temporal band-limitation with arbitrary cutoff frequency, even half of temporal sampling frequency. The filtered sequence is applicable to an input for image sequence coding. i.e., The MASF will be feasible as a prefilter for coding applications.

ACKNOWLEDGEMENTS

This work was supported by Video Research Center of Dae woo Electronics Co. Ltd. B.W. Lee and G.H. Jang made helpful comments on the manuscript.

REFERENCES

Generalized Theory of Multidimensional M-Band Filter Bank Design

I.A. Shah

A.A.C. Kalker

Philips Research Laboratories,
P.O. Box 80.000, 5600 JA Eindhoven, The Netherlands
Tel: (+31) 40 744174
Fax: (+31) 40 744675
Net: kalker@prl.philips.nl, shah@prl.philips.nl

Abstract

The paper presents a generalized theory of designing M-band $N_1$-dimensional ($N_1$-D) perfect reconstruction filter banks, from $N_2$-dimensional ($N_2$-D) M-band filter banks using transformations, where $N_1$, $N_2$, and $M$ are arbitrary integers. We will state theorems on the preservation of the perfect reconstruction property when transforming from $N_1$-D to $N_2$-D. The theory is applied in the design of bi-orthogonal filters suitable for quincunx subsampling.

1 Introduction

Multidimensional subband coding systems are becoming popular for image representation and coding. This makes designing multidimensional filters for use in filter banks important. Such designs are well-known for 1-D filters, but are more difficult to design for higher dimensions. Various authors have designed 2-D 2-band filter banks starting from a 1-D 2-band filter bank [2] [3] [4]. Recently, the theory for multidimensional QMF designs was presented in [5].

In this paper we present a generalized theory for transforming an M-band $N_1$-D filter bank into an M-band $N_2$-D filter bank for arbitrary integers $N_1$, $N_2$, and $M$. The theorems governing the transformations will be stated. They will be illustrated by the transformation of 1-D bi-orthogonal filters to 2 dimensions.

Section 2 presents theory of the transformations by stating the appropriate theorems. Section 3 will work out the conditions imposed by the theorems for the case of 1-D to 2-D transforms, where the 2-D subsampling matrix is quincunx. Also an example of a 2-D bi-orthogonal filter-bank design is given. In Section 4 extensions are made to the case of M-band designs for $M \geq 2$.

2 General theory of filter transformation

In a previous paper [5] we have shown how a 1-D QMF bank can be transformed into a 2-D QMF bank using a method introduced by McClellan [1]. In this section we recall the McClellan transform and show its generalization.

2.1 The McClellan transform

Given is a zero-phase filter $H(z)$ and a 2-D kernel $K(\xi)$ where $\xi = (z_1, z_2)$. The zero-phase condition on $H(z)$ implies that there exists a filter $H_S(z)$ such that

$$H(z) = H_S\left(\frac{1}{2}(z + z^{-1})\right).$$

The McClellan transform $H^K(\xi)$ of $H(z)$ by $K(\xi)$ is now defined as

$$H^K(\xi) = H_S(K(\xi)).$$

i.e. the 1-D kernel $S(z) = \frac{1}{2}(z + z^{-1})$ is replaced by the 2-D kernel $K(\xi)$. In order to control the shape of the resulting filter it is required that the values of $K(\xi)$ on the unit bi-circle $B = \{(z_1, z_2) : |z_1| = 1, |z_2| = 1\}$ form a subset of the range of $S(z)$, where $z$ ranges over the unit circle. One easily checks that $K(\xi)$ has to be zero-phase (in order to be real valued) and to be bounded between -1 and 1 on $B$.

2.2 Notations and basic facts

In the previous section we have shown that 1-D filters can be transformed into 2-D filters by extraction and insertion of kernels. In our previous paper [5] we have also shown that if we apply this transformation to each filter of a 1-D QMF filter bank the resulting 2-D filter bank is also alias free and perfect reconstructing. Certain restrictions on the 2-D kernel however were needed. More specifically we found that the kernel must live on the non-zero coset of the sampling raster.

Analyzing the proof needed for the QMF filter bank we found that the property which makes things
work is the preservation of the polyphase structure of the analysis/synthesis bank. In order to formalize
the notion of preservation of polyphase structure we have to introduce some notations and definitions.

Let \( \mathcal{L} \) be an \( N \)-dimensional sampling raster
represented by a matrix \( L \). The set of cosets \( C(\mathcal{L}) \) of
\( \mathcal{L} \) has \( \lambda = |\det(L)| \) elements and is endowed with
a canonical additive group structure. Dual to \( C(\mathcal{L}) \) the
set \( W(\mathcal{L}) \) is defined by
\[
W(\mathcal{L}) = \{ w : w^L = 1 \},
\]
where \( 1 \) is the \( M \)-dimensional vector \((1,1,\cdots,1)\) and
where exponentiation by a matrix is defined as usual [6]. The set \( W(\mathcal{L}) \) will be used to characterize
the polyphase components of an \( M \)-D filter \( F(z) \).

**Definition 1** Let \( F(z) \) be a filter. Let \( l \) be a coset
of \( \mathcal{L} \). Then the polyphase component \( P_l(z) \) of \( F(z) \)
is defined as that part of \( F(z) \) which lives on \( l \) (i.e.
has support on \( l \)).

It is clear that \( F(z) \) is equal to the sum of its
polyphase components. Moreover we have a character-
zation of polyphase components in terms of the
set \( W(\mathcal{L}) \).

**Theorem 1** Let \( F(z) \), \( P_1(z) \), \( \cdots \), \( P_l(z) \) be filters
such that
\[
F(z) = \sum_l P_l(z)
\]
and such that
\[
P_l(w \otimes z) = w^{-l}P_l(z),
\]
where \( \otimes \) denotes componentwise multiplication
and where exponentiation by a coset is defined as usual [6]. Then we have
\[
P_l(z) = \hat{P}_l(z),
\]
where \( \hat{P}_l(z) \) defines the polyphase decomposition of \( F(z) \).

Let \( (F,G) \) be an \( M \)-band analysis/synthesis filter bank with
channels \( i = 1, \cdots, M \). By considering the
polyphase matrices of the filter bank one can check
for alias cancellation and also determine the transfer
function \( T(z) \) of the filter bank. For the analysis
bank \( \{ F_i^j(z) \}_{i=1,\cdots,M} \) the polyphase matrix \( F_i(\mathcal{L}) \)
is defined by
\[
F_i(\mathcal{L}) = \{ F_i^j(z) \}_{i=1,\cdots,M}.
\]
and a similar formula holds for the synthesis bank
\( \{ G_i^j(z) \}_{i=1,\cdots,M} \).

Let \( \{ G_i^j(z) \} \) be defined by
\[
G_i^j(z) = G_i^j(z),
\]
then we have the following theorem.

**Theorem 2** The filter bank \( (F,G) \) is alias free and
has transfer function \( T(z) \) if and only if
\[
(G_i^j \circ F_i^j)(z) = T(z)I
\]
where \( \circ \) denotes matrix multiplication and where \( I \)
is the identity matrix.

With this formulation at hand a natural way of
transforming a filter bank is given by extracting a
kernel from or inserting a kernel into the compo-
nents of the polyphase matrices. If such an inser-
tion or extraction is valid (see below), alias cancella-
tion is guaranteed. In that case the transfer function
changes by extraction or insertion of the same kernel.
The problem is the determination of the property of be-
ing a polyphase matrix. The next section discusses
conditions on the kernel to solve this problem.

### 2.3 Validation of Transformations

Before stating a theorem about valid transformation
we need some preliminaries. First of all we need
a formal definition of a valid transformation. Let
\( \mathcal{L}_i \), \( i = 1,2 \), be two sampling rasters of dimension
\( N_i \). Let \( \tau \) be an isomorphism from \( C(\mathcal{L}_1) \) to \( C(\mathcal{L}_2) \).
Finally let \( K(z) \) be a \( N_2 \)-D filter (kernel) with \( N_2 \)-D values. With these notations we are able to give the
definition of a valid transformation.

**Definition 2** Let \( F(z) \) be a \( N_1 \)-D filter with 1-D
values. We say that the transformation
\[
F(z) \rightarrow F(K(z)),
\]
where
\[
F_K(z) = F(K(z))
\]
is a valid transformation with respect to \( \tau \) if for all
\( l \in C(\mathcal{L}_1) \) the following identity holds.
\[
F_l(K(z)) = (F_K)_\tau(\mathcal{L}_1)(z).
\]

The following theorem can now easily be proved.

**Theorem 3** Let \( (F,G) \) be an alias free (with respect
to the sampling raster \( \mathcal{L}_1 \) \( M \)-band analysis/synthesis
bank with transfer function \( T(z) \). Let kernel substitution with \( K(z) \) be valid with respect to
\( \tau \) for every filter in the total filter bank. Then the
filter bank \( (F_K,G_K) \) is also alias free (with respect
to the sampling raster \( \mathcal{L}_2 \) and its transfer function
is given by \( T(K(z)) \)).

For the above theorem to be of any use two problems
have to be solved: given a filter bank which kernels
provide for valid transformations, and secondly,
which of these kernels generate filter banks useful
for applications? We postpone the latter question
to the following section and treat the other question
first. The following theorem holds.
Theorem 4: Given two sampling rasters \( L_i, i = 1, 2 \), an isomorphism \( \tau \) and a kernel filter \( K(z) \) as above, then the following assertions hold.

1. There exist an isomorphism \( \sigma \) from \( W(L_2) \) to \( W(L_1) \) such that for all \( w \in W(L_2) \) and all \( l \in C(L_1) \) the equality
   \[
   \sigma(w)^l = w^{\tau(l)}
   \]
   holds. The morphism \( \sigma \) is uniquely determined by \( \tau \). Moreover for every isomorphism \( \sigma \) a corresponding isomorphism \( \tau \) can be found.

2. Substitution by \( K(z) \) is valid with respect to \( \tau \) for every \( F(z) \) (of the correct dimension) if and only if \( K(z) \) satisfies
   \[
   K(w \otimes z) = \sigma(w) \otimes K(z)
   \]
   for all \( w \in W(L_2) \).

For a specific case this theorem can be reformulated in a less abstract way.

Theorem 5: Let \( K(z) \) be an kernel with 1-D values, i.e. \( N_2 = 1 \). Then substitution by \( K(z) \) is a valid transformation with respect to some \( \tau \) if and only if

1. \( C(L_2) \) is cyclic of order \( \lambda = \lambda_2 \equiv \lambda \).
2. \( K(z) \) is equal to one of its polyphase components \( K_l \) such that \( l \) generates the group \( C(L_2) \).

3 Examples and Design Applications

3.1 The Quincunx Case

In this example we will apply the theorems to transform a 1-D 2-band filter bank consisting of zero-phase filters to a 2-D 2-band filter bank with a quincunx subsampling raster. The transform consists of two phases: first extraction of the kernel \( S(z) = \frac{1}{2}(z + z^{-1}) \), then the insertion of a 2-D kernel \( K(z) \).

In order to check that the extraction of \( S(z) \) is a valid transformation we can apply Theorem 5 where we use subsampling by 2 on both sides of the transformation. The group structure of the coset space is in this case isomorphic to the cyclic space of order 2. Consequently there exists only one generating element, viz. the non-zero coset. Extraction of \( S(z) \) is a valid transformation as \( S(z) \) lives on this non-zero coset. Another way of checking is given by Theorem 4, i.e. we have to verify

\[
S(w_{1d} \otimes z) = \sigma(w_{1d}) S(z),
\]

where \( \sigma(w_{1d}) \defeq w_{1d} \) and where \( w_{1d} \) ranges over \( \{1, -1\} \). This equality is easily checked.

A possible matrix for quincunx subsampling is given by

\[
L_{2d} = \begin{bmatrix}
1 & 1 \\
-1 & 1
\end{bmatrix}
\]

and the corresponding \( w \)'s are given by the set:

\[
W(L_{2d}) = \left\{ \begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \right\}.
\]

The quincunx coset space is a cyclic group of order two. Again by Theorem 5 substitution by \( K(z) \) is valid if \( K(z) \) lives on the non-zero coset of the quincunx sampling raster. As above one can also check validity by applying Theorem 4.

\[
K(w_{2d} \otimes z) = \sigma(w_{2d}) K(z).
\]

The isomorphism \( \sigma \) is unique given that we are dealing with cyclic groups of order two. Its explicit representation is given by

\[
\begin{pmatrix} 1 \\ 1 \end{pmatrix} \rightarrow 1, \begin{pmatrix} -1 \\ -1 \end{pmatrix} \rightarrow -1.
\]

As both phases of the transform are valid transformations the alias cancellation property of the filter bank will be retained. For example, the original 1-D filter bank is perfect reconstructing then the resulting 2-D filter bank will also be perfect reconstructing. Starting from a 1-D bi-orthogonal zero-phase filter bank we can now do an actual 2-D filter-bank design.

3.2 Design Application

The transformation from 1-D to 2- and 3-D QMF filters was discussed in [6]. Here we will use the above technique to transform a set of 1-D bi-orthogonal filters into 2-D “fan” shaped filters, suitable for use in a filter bank with quincunx subsampling.

We start with zero-phase 1-D filters. We perform a valid transformation by extracting the 1-D kernel \( S(z) \). We then insert a valid 2-D kernel \( K(z) \) given by

\[
K = \begin{pmatrix}
0 & -1/4 & 0 \\
1/4 & 0 & 1/4 \\
0 & -1/4 & 0
\end{pmatrix}.
\]

It can be shown that \( K(z) \) is valid by checking that \( K(w_{2d} \otimes z) = \sigma(w_{2d}) K(z) \). Substituting the values for the 2-band quincunx case we get \( K(-z) = -K(z) \). Figures 1 and 2 shows the frequency response of the upper and lower branch of the resulting fan-shape filters.

4 Extensions to M-band Designs

All the theory has been presented in a general way, and is valid for arbitrary dimensions and number of
bands. In this section we concentrate on the application of the theory to the design of 2-D filters with M bands for $M \geq 2$.

The procedure is the same as in the two band case. First extract from each $M_1$-D filter $H(z)$ a valid kernel $S(z)$, that lives on an allowed coset and then insert a valid $N_2$-D kernel. The resulting $N_2$-D filter bank will preserve the alias cancellation and perfect reconstruction properties of the $N_1$-D filter bank. To give a specific example, we consider the transformation of a 1-D 3-band filter bank into a 2-D 3-band filter bank. The extraction of the valid 1-D kernel in this case would mean finding a filter $H_S(z)$ such that

$$H(z) = H_S\left(\frac{1}{2}(z + z^{-2})\right)$$

(22)

and then replacing $\frac{1}{2}(z + z^{-2})$ by a valid 2-D kernel $K(z)$. The practical problem with such a procedure is that finding $H_S(z)$ is more difficult than the 2-band case. It should be noted that the extraction of the 1-D kernel controls the shape of the 2-D filter, which is of importance in practical filter design. An alternative procedure could be to perform a direct 2-D kernel insertion into $H(z)$ and not extract a 1-D kernel at all. If the kernel insertion is valid, this procedure will result in a 2-D filter bank with alias cancellation and perfect reconstruction. However there will be little control over the frequency response of the resulting individual 2-D filters.

5 Summary

In this paper we have presented generalized theory of transforming $M$-band $N_1$-D filter banks into $N_2$-D filter banks, while preserving the $N_1$-D properties. The theory has been used to design 2-band 2-D biorthogonal filters. The design of $M$-band ($M \geq 2$) filter banks is considered.

References


A NEW WLS CHEBYSHEV APPROXIMATION METHOD FOR THE DESIGN OF FIR DIGITAL FILTERS WITH ARBITRARY COMPLEX FREQUENCY RESPONSE

Chong-Yung Chi and Shyu-Li Chiu

Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, ROC.

Abstract: In this paper, motivated by Chi–Kou's weighted least-squares (WLS) Chebyshev approximation method [1] for the design of FIR digital filters with linear phase, we propose a new self-initiated iterative WLS Chebyshev approximation method for the design of FIR digital filters with arbitrary complex frequency response. The proposed method not only inherits all the advantages of Chi–Kou's method but also is computationally more efficient than Chi–Kou's method for the case of linear phase FIR filter design. Two design examples are provided to demonstrate the good performance of the proposed method.

1. Introduction

Recently, Chi and Kou [1] proposed an efficient weighted least-squares (WLS) Chebyshev approximation method for the design of linear phase equiripple FIR digital filters which performs as well as the well-known Parks–McClellan’s algorithm [2] with some extra advantages. For instance, the former does not need the initial guess for a suitable set of extremal frequencies or filter coefficients; it is applicable in the design of any type of linear phase FIR filters with no need of nontrivial modifications required by the latter. However, they are not applicable in the design of FIR digital filters with arbitrary complex frequency response.

Mason and Chit [3] proposed a time-domain least mean square (LMS) approximation method for the design of FIR filters with arbitrary complex frequency response. In this paper, we propose a new WLS Chebyshev approximation method based on Chi–Kou’s approximation method. The proposed approximation method is not only applicable in the case of FIR filter with arbitrary complex frequency response but also computationally more efficient than Chi–Kou’s approximation method for the linear phase case. It can be viewed as a frequency-domain counterpart to Mason–Chit’s LMS method based on similar philosophies. Furthermore, the former has better performance than the latter and is much simpler than some other well-known complex approximation methods such as [4,5].

In Section 2, we present the new WLS Chebyshev approximation method. Then we show two design examples to support the proposed approximation method in Section 3. Finally, we draw some conclusions.

2. The WLS Chebyshev Approximation Method

Assume that the desired complex frequency response $H_d(f)$ is conjugate-symmetric, i.e., $H_d(f) = H_d(-f)$, and that the domain $B_{NT} \subset [0, \frac{\pi}{2}]$ over which $H_d(f)$ is well defined includes $p$ disjoint nontransition bands as follows:

$$B_{NT} = B_1 \cup B_2 \cup \cdots \cup B_p$$

where

$$B_m = \{ f | f_m \leq f \leq f_{m+1} \},$$

and $f_m$ and $f_{m+1}$ denote the specified cutoff frequencies in the $m$th frequency band $B_m$. Then the union of the transition bands, denoted $B_{TS}$, is given by

$$B_{TS} = \{ f | 0 \leq f \leq \frac{\pi}{2}, f \notin B_{NT} \}$$

Assume that the filter to be designed is an $(M-1)$th-order FIR filter with real filter coefficients $h(n)$. The frequency response is then

$$H(f) = \sum_{n=0}^{N-1} h(n) e^{-j2\pi fn}$$

We define the complex approximation error $E(f)$ as

$$E(f) = H_d(f) - H(f) = E_r(f) + jE_i(f)$$

where $E_r(f)$ and $E_i(f)$ are real part and imaginary part of $E(f)$, respectively. Let $W(f)$, $f \in B_{NT}$, be a piecwise-constant function associated with the desired relative approximation error ratio among $p$ nontransition bands, defined as

$$W(f) = \rho_m$$

if $f \in B_m$$\quad (3)$

where $\rho_1 > 0$, $\rho_2 > 0$, $\cdots$, $\rho_p > 0$, $\max(\rho_1, \rho_2, \cdots, \rho_p) = 1$, and the ratio $(1/\rho_1):(1/\rho_2): \cdots : (1/\rho_p)$ denotes the desired relative approximation error ratio among $B_1, B_2, \cdots, B_p$. Our object is to find a set of filter coefficients $h(n)$ by the well-known WLS method such that $H(f)$ is equiripple with $\delta_1, \delta_2, \cdots, \delta_p = (1/\rho_1): (1/\rho_2): \cdots : (1/\rho_p)$ where $\delta_m$ is the maximum error in $B_m$. Next, let us present the WLS estimator on which the new approximation design method is based.

For notational simplicity, let $W_k(k)$, $E(k)$, $E_k(k)$ and $E_k(k)$ also denote $W_k(\pi-k/2N)$, $E(\pi-k/2N)$, $E_k(\pi-k/2N)$, respectively, where $N$ is the total number of uniform samples in the interval $[0, \pi/2]$. Thus, by (1) and (2), we can express $E_k(k)$ and $E_k(k)$ for $k = 0, 1, \cdots, N-1$, in the following

This work was supported by the National Science Council under Grant NSC81-0404-E-007-010.
linear vector form:
\[
\begin{bmatrix}
E_{c} \\
E_{i}
\end{bmatrix} = \begin{bmatrix}
\text{Re} \{H_d\} & D_1 \\
\text{Im} \{H_d\} & D_2
\end{bmatrix} \begin{bmatrix}
h \\
h \end{bmatrix}
\]
(4)
where
\[
h = [h(0), h(1), ..., h(M-1)]',
\]
\[
E_{c} = [E_c(0), E_c(1), ..., E_c(N-1)]',
\]
\[
E_{i} = [E_i(0), E_i(1), ..., E_i(N-1)]',
\]
\[
H_d = [H_d(0), H_d(1), ..., H_d(N-1)]',
\]
\[
\text{Re}(H_d) \quad \text{and} \quad \text{Im}(H_d)
\]
denote real part and imaginary part of \(H_d\), respectively, and \(D_1\) as well as \(D_2\) are \(N\times M\) matrices with the \((i,j)\)th element \([D_1]_{ij} = \cos((i-1)(j-1)\pi/N)\) and \([D_2]_{ij} = -\sin((i-1)(j-1)\pi/N)\), respectively. The sum of weighted error squares is defined as
\[
J(h) = \sum_{k=0}^{N-1} w(k) |E(k)|^2 = E_{c}^t W E_{c} + E_{i}^t W E_{i}
\]
(5)
where
\[
W = \text{diag} [w(0), w(1), ..., w(N-1)]
\]
with \(w(k) \geq 0\) for all \(0 \leq k \leq N-1\). It is well-known that the WLS estimate \(\hat{h}\), which minimizes \(J(h)\), is given by [6]
\[
\hat{h} = \left[D_1^t W D_1 + D_2^t W D_2\right]^{-1} \cdot \left[D_1^t W \text{Re}(H_d) + D_2^t W \text{Im}(H_d)\right]
\]
(6)
A well-known property of WLS estimators is as follows:

(P1) The larger the weight \(w(k)\), the smaller is the associated error \(|E(k)|\).

The new approximation method is an iterative algorithm based on (P1) for finding the optimum \(w(k)\) such that \(|E(k)|\) is equiripple with the desired approximation error ratio among the nontransient bands, or, equivalently, \(|W_e(k)\cdot E(k)|\) for \(k/2N \in B_{NT}\) is equiripple. Before presenting the new design method, we define some notations for ease of latter use:

- Error ripple \(E_{m}^i(k)\):
  \[
  E_{m}^i(k) = |E(k)|, \quad \text{for } k/2N \in B_{m}^i, i = 1, 2, ..., q,
  \]
  where \(q\) is the total number of error ripples in \(B_{m}\)
  and
  \[
  B_{m} = \{ f | t_{m}^{i-1} \leq f \leq t_{m}^i \} \subset B_m
  \]
  where \(t_{m}^{i-1} = f_{m}^{i-1}, f_{m}^i = f_{m}^{i+1}, \text{and } t_{m}^i \in B_m\) \(i = 1, 2, \ldots, q-1\), are the frequencies at each of which \(|E(f)|\) is a local minimum.

- Amplitude \(e_{m}^i\) of error ripple \(E_{m}^i\):
  \[
  e_{m}^i = \max \{ E_{m}^i(k), k/2N \in B_{m}^i \}
  \]
  (8)

- Piecewise-constant function \(R(k)\), \(k/2N \in B_{NT}\):
  \[
  R(k) = W_e(k) \cdot e_{m}^i = e_{m}^i \cdot e_{m}^i \quad \text{if } k/2N \in B_{m}^i
  \]
  (9)

The new design method, which is shown in Fig. 1, begins with the initial weighting function

\[
\begin{align*}
\text{START} \\
\text{SET } w^{(0)}(k) \\
n = 1
\end{align*}
\]

\[
\begin{align*}
\text{COMPUTE FILTER COEFFICIENTS} \\
\text{BY WLS ESTIMATOR}
\end{align*}
\]

\[
\begin{align*}
\text{COMPUTE } E(k), \text{ SEARCH FOR ERROR RIFFLES AND FORM } R(k)
\end{align*}
\]

\[
\begin{align*}
\text{IS THE DESIGNED FILTER EQUIRIPLE?}
\end{align*}
\]

\[
\begin{align*}
\text{UPDATE WEIGHTING FUNCTION } w^{(n)}(k)
\end{align*}
\]

\[
\begin{align*}
\text{STOP}
\end{align*}
\]

Fig. 1. The optimum WLS approximation method.

\[
\begin{align*}
w^{(0)}(k) &= \begin{cases}
W_e(k), & k/2N \in B_{NT} \\
\text{0,} & k/2N \in B_{TS}
\end{cases}
\end{align*}
\]
(10)

Assume that we ended up with the weighting function \(w(k) = w^{(n)}(k)\) at the \(n\)th iteration. For the \(n\)th iteration, the WLS estimate, \(\hat{h}\), is computed by (6), in which \(w(k) = w^{(n-1)}(k)\). Then we check whether \(|W_e(k)\cdot E(k)|\) for \(k/2N \in B_{NT}\) is equiripple by
\[
\left(R_{\max} - R_{min}\right)/R_{max} \leq \sigma
\]
(11)
where \(R_{\max} = \max(R(k), k/2N \in B_{NT}), R_{min} = \min(R(k), k/2N \in B_{NT})\), and \(\sigma\) is a prespecified small positive constant. If \(|W_e(k)\cdot E(k)|\) for \(k/2N \in B_{NT}\) is not equiripple yet, we update the weighting function by

\[
\begin{align*}
w^{(n)}(k) &= \begin{cases}
w^{(n-1)}(k) R(k)/w_{\max}, & k/2N \in B_{NT} \\
w^{(n-1)}(k) R(k)/w_{\min}, & k/2N \in B_{TS}
\end{cases}
\end{align*}
\]
(12)

where
\[
\begin{align*}
w_{\max} &= \max \{w^{(n-1)}(k) \cdot R(k), k/2N \in B_{NT}\}
\end{align*}
\]
(13)

normalizes \(w^{(n)}(k)\) such that \(0 < w^{(n)}(k) \leq 1\) for all \(k/2N \in B_{NT}\).

Although the proposed approximation method was illuminated by the case of real filter coefficients due to \(H_d(f) = H_d(-f)\), it is applicable in the case of complex filter coefficients by replacing the linear
model (4) and the WLS estimate (6) with
\[ E = H_d - D \hat{h} \]
and
\[ \hat{h} = [D^H W D]^{-1} D^H W y \tag{15} \]
respectively, where $D^H$ is the complex conjugate transpose of $N \times M$ matrix $D$ whose elements can be easily determined from the definition of $H(f)$.

The FIR filter to be designed may have to satisfy certain implicit constraints. For instance, type I and type II linear phase FIR filters require $h(n) = h(M - 1 - n)$ which can be easily taken into account in the proposed approximation method as follows. Assume that $M$ is even (type II). Then, $H(f)$ and $H_d(f)$ can be expressed as
\[ H(f) = \exp(-j2\pi fL) \sum_{n=0}^{M/2-1} 2h(n) \cos(2\pi (L-n)) \tag{16} \]
and
\[ H_d(f) = d(f) \exp(-j2\pi fL) \tag{17} \]
where $L=(M-1)/2$ and $d(f)$ is real. The approximation error $E(f)$ can be redefined as
\[ E(f) = d(f) - \sum_{n=0}^{M/2-1} 2h(n) \cos(2\pi (L-n)) \tag{18} \]
which leads to the following real linear vector model:
\[ E = d - D \hat{h} \tag{19} \]
where $d = [d(0), d(1), ..., d(N-1)]^T$, $h = [h(0), h(1), ..., h(M/2-1)]^T$ containing only nonredundant filter coefficients and $D$ is an $N \times (M/2)$ matrix whose elements can be easily determined from (12). The WLS estimate is then
\[ \hat{h} = [D^T W D]^{-1} D^T W d \tag{20} \]

3. Design Examples

Two design examples, which were obtained with $N = 1000$ and convergence parameter $\epsilon = 0.01$, are presented below to support the proposed WLS Chebyshev approximation method.

**Example 1.** Lowpass filter with constant group delay $\tau_d$ (taken from [4,5]):

The desired frequency response $H_d(f)$ is as follows:
\[ H_d(f) = \begin{cases} e^{-j2\pi f \tau_d}, & f \in B_1 = [0, 0.06] \text{ (passband)} \\ 0, & f \in B_2 = [0.12, 0.5] \text{ (stopband)} \end{cases} \tag{21} \]
where $\tau_d=12$. The filter to be designed is a 2-band filter $(p=2)$ of order $M=30$ and
\[ W_{e}(f) = \begin{cases} \rho_1 = 0.1, & f \in B_1 \\ \rho_2 = 1, & f \in B_2 \end{cases} \tag{22} \]
The total iterations spent by the proposed approximation method is 11. The magnitude response and group delay response of the designed filter are shown in Fig. 2(a) and Fig. 2(b), respectively. One can see, from Fig. 2(b), that in the passband, the group delay is nearly constant although it oscillates around 12 with maximum deviation of 1.968. Figure 2(c) shows the absolute approximation error $|E(f)|$ which is equiripple with maximum errors $\delta_1 = 0.0041$ and $\delta_2 = 0.00443$, in the passband and the stopband, respectively. Note that $\delta_1/\delta_2 \approx (1/\rho_1)/(1/\rho_2) = 10$. These results are also comparable with the corresponding results reported in [4] and [5].

**Example 2.** Type II linear phase shaping filter (taken from [3]):

The desired amplitude response $d(f)$ is as follows:
The filter to be designed is a 2-band filter of order \( M = 49 \) and \( W_1(f) = 1 \), \( f \in B_3 = [0, 0.25 \) and \( f \in B_2 = [0.28, 0.5]. \) The total iterations spent by the proposed approximation method is 10. Figures 3(a) and 3(b) show magnitude response and \( |E(f)| \), respectively. From Fig. 3(b), one can see that \( |E(f)| \) is equiripple with maximum error \( \epsilon = \delta = 0.0353. \) For the same example, it was reported in [3] that Mason–Chit’s method spends 412 (>>10) iterations and the corresponding equiripple \( |E(f)| \) has maximum error \( 0.064 \approx 1.8 \delta. \) Hence, this design example supports that the performance of the proposed method is much superior to that of Mason–Chit’s method from both viewpoints of iteration number and maximum error.

4. Conclusions
Let us summarize the characteristics of the proposed WLS Chebyshev approximation method for the design of FIR filters with arbitrary complex frequency response in the following:

(1) Contrast to Mason–Chit’s time–domain LMS method [3] which is also for the design of FIR filters with arbitrary frequency response, the proposed method is a frequency–domain WLS method based on similar philosophies. The latter has better performance than the former although the designed optimum filters by both methods are equiripple.

(2) Contrast to Chi–Kou’s method which is for the design of FIR filters with linear phase, the proposed method adjusts the filter coefficients by simultaneously considering approximation errors in each frequency band and desired error ratio among all frequency bands. Therefore, the proposed method is computationally more efficient than Chi–Kou’s approximation method from the viewpoint of total iterations needed.

(3) The proposed method possesses the same advantages as Chi–Kou’s approximation method as follows. It allows the exact specification of the cutoff frequencies; it does not need any initial guess for a suitable set of extremal frequencies or filter coefficients; it can be directly applied to the design of any type of linear phase FIR filters without need of extra modifications.

(4) The FIR filter to be designed may have to satisfy certain implicit constraints (such as particular symmetries of linear phase FIR filters) which can be easily taken into account in the proposed approximation method by properly reformulating the linear model (4) such that \( h \) contains only nonredundant filter coefficients. However, as long as \( N \) is large enough, the same designed filter will be obtained no matter whether implicit constraints on filter coefficients are considered or not.

Fig. 3. The designed type II linear phase 2-band shaping filter of order \( M = 49 \) for \( B_3 = [0, 0.25 \) and \( B_2 = [0.28, 0.5]. \) (a) Magnitude response and (b) absolute approximation error.

References
MODULATED FILTER BANKS IN MULTIDIMENSIONS WITH ARBITRARY SAMPLING LATTICES

David B. H. Tay and Nick G. Kingsbury

Signal Processing and Communications Laboratory, Department of Engineering, University of Cambridge, Trumpington Street, Cambridge, CB2 1PZ, England.

An analysis of DFT modulated filter banks in multidimensions with arbitrary sampling lattices is presented. The filters are obtained by modulating a prototype filter. An efficient polyphase network implementation will be derived and the issue of aliasing cancellation and perfect reconstruction is addressed. The shape aspects of the frequency subbands are discussed.

1. INTRODUCTION, NOTATIONS AND DEFINITIONS

Over the last decade there has been a great deal of interest in the area of multirate digital filter banks. These systems perform analysis/synthesis of signals by/from subband components and find wide applications in areas such as bandwidth compression, signal scrambling and spectrum analysis. Most of the work concerning multirate digital filter banks has dealt with one dimensional signals and systems but recently there has been an increasing interest in multidimensions [1],[2]. One important class of filter banks is the DFT (Discrete Fourier Transform) modulated filter bank and it has a very efficient polyphase network implementation [3]. This class of filter bank in one dimension has been treated by various authors [4],[5]. A special case in two dimensions where the sampling lattice used was rectangular was presented in [6], but the issue of aliasing cancellation and perfect reconstruction was however not addressed. In this paper we extend and generalize the DFT modulated filter bank to multidimensions with an arbitrary sampling lattice.

We briefly review some definitions and a notation that are useful for compact description of multidimensional multirate systems (see [1] for a more complete treatment). An N-dimensional sequence is denoted by \( z(n) \) where \( n = [n_1 \ldots n_N]^T \) is a vector of integer variables. The \( z \)-transform variable \( z = [z_1 \ldots z_N]^T \) is a vector of complex variables. We define \( z^N = \prod_{i=1}^{N} z_i^N \) and the \( z \)-transform by \( X(z) = \sum_{n\in\Lambda} z(n)z^{-n} \) (\( \Lambda \) is the set of all integer vectors \( n \)). The Fourier transform \( X(\omega) = \sum_{n\in\Lambda} z(n)e^{-j\omega n} \) is obtained by evaluating \( X(z) \) at \( z = [e^{jn_1} \ldots e^{jn_N}]^T \).

A sublattice \( \Lambda_D \) of \( \Lambda \) is defined to be the set of points \( m = Dn \) (\( n \in \Lambda \)). The matrix \( D \) of integer elements is known as the sampling (or periodicity) matrix. A coset of \( \Lambda_D \) is the set of points obtained by shifting \( \Lambda_D \) by an integer vector \( m \) (coset vector). There are exactly \( |\det(D)| \) distinct cosets that are represented by the \( m_0, m_1, \ldots, m_{|\det(D)|-1} \) coset vectors. The subsampling operation is defined as \( y(n) = z(Dn) \) (only points in \( \Lambda_D \) are retained) with effective subsampling rate \( D \). The upsampling is defined as \( y(n) = z(D^{-1}n) \) for \( n \in \Lambda_D \) and \( y(n) = 0 \) elsewhere. If \( d_i \) are the columns of \( D \) we define \( z^D = [z^{d_1} \ldots z^{d_N}]^T \). The z-transform input/output relationships of the subsampling and upsampling operations are respectively:

\[
Y(z) = \frac{1}{D} \sum_{l=0}^{D-1} X(W(k_l) \circ z^{D^{-1}})
\]

(1)

\[
Y(z) = X(z^D)
\]

(2)

where \( k_l \) are the coset vectors of \( D \) (sublattice \( \Lambda_D \)) and \( W(k_l) = \left[ \exp(-j2\pi k_l^T d_i) \right] \) (\( d_i \) are the columns of \( D^{-1} \)). The symbol \( \circ \) denotes the element-wise product i.e. \( x \circ y = [x_1y_1 \ldots x_Ny_N]^T \).

The DFT pair with periodicity matrix \( D \) is defined as:

\[
X(k) = \sum_{l=0}^{D-1} z(m_l) \exp(j2\pi k^T D^{-1} m_l)
\]

(3)

\[
z(n) = \frac{1}{D} \sum_{l=0}^{D-1} X(k_l) \exp(-j2\pi k^T D^{-1} n)
\]

(4)

\( X(k) \) and \( z(n) \) are both periodic with periodicity matrix \( D^T \) and \( D \) respectively (e.g. \( z(n+Dp) = z(n) \) for \( p \in \Lambda \)).

2. MODULATED FILTER BANKS

The DFT modulated filter bank is closely related to the Short-Space Fourier Transform (SSFT) [4] which is fundamental to spectral analysis. There are two representations of the SSFT, namely (i) low-pass and (ii) band-pass. We shall employ the band-pass representation which is given by:

\[
X(m,\omega) = \sum_{n\in\Lambda} h(n) \exp(j\omega n) z(m-n)
\]

The discrete SSFT is obtained by sampling \( X(m,\omega) \) at frequencies \( \omega = 2\pi D^{-T} k_l \) and is critically decimated when \( m \) is subsampled by \( D \). It is obtained directly from a modulated filter bank with sampling matrix \( D \) (there are exactly \( D \) channels). The filter in each channel is obtained by frequency translating (modulating) a prototype filter \( H(z) \) (impulse response \( h(n) \)):
Figure 1: A Typical Channel of Filter Bank - (a) Analysis Section (b) Synthesis Section

\[ H(z) = H(W(k_i) \circ z) \]
\[ h[n] = \exp[j2\pi k_i^T D^{-1} n] h(n) \]

where \( l = 0, \ldots, D - 1 \) is the channel number label. The ideal passband of \( H(z) \) is related to \( D \) (to be discussed later). A typical channel of the filter bank which performs analysis and synthesis of signals is shown in Fig. 1(a) and Fig. 1(b) respectively. Note that in the synthesis section the \( d_l(n) \) \((l = 0, \ldots, D - 1)\) signals are added up in the reconstruction process.

2.1. FREQUENCY DOMAIN ANALYSIS

With the analysis and synthesis sections connected back-to-back the input/output relationship can be obtained using (1) and (2) and it is:

\[ \tilde{X}(z) = 1/D g^T(z) H_{AC} x_M(z) \]

where \( g(z) = [G_0(z) \ldots G_{D-1}(z)]^T \), \( x_M(z) = [X(W(k_0) \circ z) \ldots X(W(k_{D-1}) \circ z)]^T \) and \( H_{AC} \) is the AC (aliasing-component) matrix with its \((l,p)\) element \( [H_{AC}]_{lp} = H_l(W(k_p) \circ z) \). When the analysis filters are modulated versions of a prototype as in (5), we have:

\[ H_{AC} = H (W (k_i + k_p) \circ z) = [H_{AC}]_{lp} \]

i.e. \( H_{AC} = H^T \) - the AC matrix is symmetric. Without loss of generality we can set \( k_0 = [0 \ldots 0]^T \). Then \( W(k_0) = [1 \ldots 1]^T \). The first element of \( x_M(z) \) is then the signal component \( X(z) \) while other elements are aliased versions of \( X(z) \). We can easily verify from (7) that the condition for aliasing cancellation is:

\[ g(z) = D T(z) H_{AC} [1 0 \ldots 0]^T \]

where \( T(z) \) is an arbitrary transfer function. Equation (8) gives the synthesis filters that will achieve aliasing cancellation given the analysis filters.

The polyphase decomposition of \( H(z) \) with respect to \( D \) is \( H(z) = \sum_{p=0}^{D-1} z^{-mp} E_p(z^D) \) where \( E_0(z), \ldots, E_{D-1}(z) \) are the polyphase components of \( H(z) \). It can be verified that \( W^D(k_i) = [1 \ldots 1]^T \). Then: \( H_i(z) = \sum_{p=0}^{D-1} W^{-mp}(k_i) z^{-mp} E_p(z^D) \). We can then express \( h(z) = \left[ h_0(z) \ldots h_{D-1}(z) \right]^T \) as:

\[ h(z) = \Omega \text{diag} \left( z^{-m_0}, \ldots, z^{-m_{D-1}} \right) e(z^D) \]

where \( e(z) = [E_0(z) \ldots E_{D-1}(z)]^T \) and \( \Omega \) is referred to as the Multidimensional Fourier Matrix with elements \( \Omega_{kp} = W^{-mp}(k_i) = \exp[j2\pi k_i^T D^{-1} m] \).

The AC matrix can be expressed in terms of \( h(z) \) as:

\[ H_{AC} = \left[ h(z) | h(W(k_0) \circ z) | \ldots | h(W(k_{D-1}) \circ z) \right] \]

With (9) we arrive at the diagonalization of the AC matrix:

\[ H_{AC} = \Omega \text{diag} \{ z^{-m_0} E_0(z^D), \ldots, z^{-m_{D-1}} E_{D-1}(z^D) \} \Omega^T \]

It can be verified that \( \Omega^T \Omega = \Omega \Omega^T = DI \) so that \( \Omega^{-1} = 1/D \Omega \Omega^T \). By formal inversion of the AC matrix with the aid of its diagonal form, we obtain the solution to (8). After some matrix algebra manipulations we obtain:

\[ G(z) = T(z) \sum_{p=0}^{D-1} W^{-mp}(k_i) z^{mp} E_p^{-1}(z^D) \]

By defining (i) \( R(z) \equiv \prod_{p \neq 0} E_p(z) \);
(ii) \( G(z) = \sum_{p=0}^{D-1} z^{mp} R_p(z^D) \); (iii) \( T(z) \) by the equation:

\[ T(z) = T^*(z) \prod_{p=0}^{D-1} E_p(z^D) = T^*(z) R(z^D) E(z^D) \]

we can express (10) as:

\[ G(z) = T^*(z) G(W(k_i) \circ z) \]

i.e. the synthesis filters can be expressed as modulated versions of a prototype filter \( G(z) \) with a common factor \( T(z) \) which can be interpreted as a post-filter after the reconstruction process. For arbitrary FIR (finite impulse response) analysis filters (with FIR \( E_p(z) \)), the synthesis filters required for aliasing cancellation are in general IIR (infinite impulse response) due to the \( E_p^{-1}(z^D) \) factors in (10). If, however, \( T(z) \) in (11) is FIR, then the synthesis filters will also be FIR as seen from (12). If only aliasing cancellation is required \( T(z) \) can be arbitrary but for perfect reconstruction we require \( T(z) = c \dot{x} q^T \) (\( c \) - constant factor, \( q \) - integer vector). From (11), we see that in general this would require \( T(z) \) to be IIR. The synthesis filters are then IIR unless \( E_p(z) = c_p \dot{x} q^p \) (arbitrary \( c_p, q_p \)). However, this gives trivial FIR analysis and synthesis filters and the filter bank would basically perform only a normal DFT (see Section 2.2 and Fig. 2). From here onwards we shall deal only with the case where both the analysis and synthesis filters are obtained from modulated versions of prototype filters as in (5) (or (6)) and (12) (with \( T(z) = 1 \)).

2.2. TIME DOMAIN ANALYSIS

The output of a typical channel of the analysis filter bank (Fig. 1(a)) is:

\[ y(n) = \sum_{n \in \mathbb{N}} h(n') \exp[j2\pi k_i^T D^{-1} n'] x(Dn - n') \]

The polyphase components of \( h(n) \) and \( x(n) \) are respectively:

\[ c_p(n) = h(Dn + m_p) \quad \text{and} \quad x_p(n) = x(Dn - m_p) \]

for \( p = 0, \ldots, D - 1 \) (note the difference between the definitions of polyphase components of \( h(n) \) and \( x(n) \)). The
polyphase components are obtained by shifting the signal by $m_p$ (or $-m_p$) and subsampling with $D$. By defining the change of variables: $n' = D r + m_p$, we have the following change in summation:

$$
\sum_{n' \in \Lambda} \rightarrow \sum_{r \in \Lambda} \sum_{p=0}^{D-1} (15)
$$

Equation (13) then becomes:

$$
g_r(n) = \sum_{r \in \Lambda} \sum_{p=0}^{D-1} h(D r + m_p) x(D(n-r) - m_p) \exp[j 2\pi k r^T D^{-1}(D r + m_p)]
$$

Using the fact that $\exp[j 2\pi k r^T r] = 1$ with (14), we have:

$$
g_r(n) = \sum_{r \in \Lambda} \sum_{p=0}^{D-1} \exp[j 2\pi k r^T D^{-1} m_p] (\sum_{r \in \Lambda} \exp[j 2\pi k r^T D^{-1} m_p] x(D(n-r))
$$

The summation in $r$ represents a convolution (filtering) of $x_r(n)$ with $h_r(n)$. The summation in $p$ represents a DFT of the filtered polyphase components (cf. eqn. (3)). Hence we obtain the polyphase network implementation as shown in Fig. 2. The processing of the signals is carried out at a reduced sampling rate.

The reconstructed signal from the synthesis filter bank (Fig. 1(b)) is:

$$
\hat{x}(n) = \sum_{i=0}^{D-1} \sum_{n' \in \Lambda} \hat{g}_i(n') g_i(n - D n') (16)
$$

where $g_i(n) = \exp[j 2\pi k_i^T D^{-1} n] g(n)$; and $g(n)$ is the prototype synthesis filter. The polyphase components of $g(n)$ and $\hat{x}(n)$ are respectively:

$$
\hat{r}_i(n) = g_i(D n - m_p) \quad (17)
$$

$$
\hat{x}_p(n) = \hat{z}(D n - m_p) \quad (18)
$$

Using (17) and (16) in (18), we have:

$$
\hat{x}_p(n) = \sum_{i=0}^{D-1} \sum_{n' \in \Lambda} \hat{g}_i(n') g_i(D(n - n') - m_p)
$$

$$
= \sum_{n' \in \Lambda} \hat{r}_p(n - n') (\sum_{i=0}^{D-1} \hat{g}_i(n') \exp[-j 2\pi k_i^T D^{-1} m_p])
$$

Figure 3: Polyphase Network Implementation of Synthesis Filter Bank

The summation in $i$ represents an inverse DFT (IDFT) of the subband signals $\hat{g}_i(n')$ (cf. eqn. (4)). The summation in $n'$ represents a filtering of the output of the DFT operation with $r_p(n)$. We obtain an efficient polyphase network implementation as shown in Fig. 3. Note that up-sampling by $D$, shifting by $m_p$ and summing of the signals corresponds to interleaving the polyphase components $\hat{x}_p(n)$ to obtain $\hat{x}(n)$.

The computation of DFT and IDFT: A fast algorithm that is similar to the Cooley-Tukey algorithm (in 1D) can be formulated for the multidimensional DFT with arbitrary periodicity. Instead of factorizing the integer $N (= \text{length of 1D FFT})$, the factorization of the periodicity matrix $D$ is used to derive the algorithm (see [7] for more details). A different algorithm which is more suitable for highly parallel processing is formulated by using the Smith Normal Form of the periodicity matrix: $D = U \Delta V$. All the matrices have integer elements and $\Delta$ is diagonal while $\|\det U\| = \|\det V\| = 1$. The effect of the matrices $U$ and $V$ is to rearrange (reindex) the input and output data of the DFT respectively. The DFT then becomes separable (due to the diagonal nature of $\Delta$) and can be computed using 1D DFT along the rows and columns (see [8] for more details).

3. FREQUENCY SUBBANDS - SHAPE ASPECTS

From its definition the Fourier transform $X^F(\omega)$ of a discrete signal $x(n)$ is periodic in each of the $\omega$ variables with period $2\pi$. Hence $X^F(\omega)$ needs to be defined only in the unit frequency cell: $U = \{\omega : \omega \in [-\pi, \pi]^N\}$. If a signal $X^F(\omega)$ is subsampled by sampling matrix $D$, then it can be verified from (1) that aliasing is avoided if the signal's passband is confined to $S \equiv \{\omega : \omega = D^{-T} \omega' ; \omega' \in U\}$.

The ideal passband of the prototype filter $H^F(\omega)$ used in the modified filter bank should be $S$. The ideal passband of the modified filter $H^F(\omega)$ (in $H^F(\omega) = H^F(\omega - 2\pi D^{-T} k_i)$ from (5)), is obtained by translating $S$ by the modulation vector $k_i = 2\pi D^{-T} k_i$:

$$
S + k_i \equiv \{\omega : \omega = \omega' + k_i ; \omega' \in S\} \quad (19)
$$

Note that regions that are translated out of $U$ are 'wrapped back' into $U$ due to the $(2\pi)^N$ periodicity of the Fourier
transform. The complete set of cost vectors $k_i$ for $D$ can be obtained from $\{k : k = D^T x, x \in \{0, 1\}^M\}$. The passbands of the complete set of modulated filters $H_i^c(\omega)$, $i = 0, \ldots, D - 1$ cover the whole frequency band $\cup_{i=0}^{D-1} (S + k_i) = U$. The filter bank thus analyzes (decomposes) a full band $(U)$ signal into subbands defined by (19). Subsampling by $D$ maps the ideal subbands defined by (19) into $U$. In practice there will be a degree of aliasing as the filters cannot have the ideal passbands described. The aliasing can however be completely eliminated in the synthesis process as shown in Sec. 2.1. One important aspect of using an arbitrary sampling lattice is that the subband shapes need not be rectangular. This can also introduce a directional aspect which is not present in one-dimensional systems. We will illustrate these aspects in two-dimensions with two examples.

The first example is described by the sampling matrix and cost vectors:

$$D = \begin{bmatrix} 2 & 1 \\ -1 & 1 \end{bmatrix}, \quad \frac{1}{\pi} k_{0,1,2} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

with $|\det(D)| = 3$. The ideal passband $S$ is shown in Fig. 4(a) as the shaded region inside the square which is $U$. The modulation vectors $k_i$ are shown in Fig. 4(b).

The second example is known as hexagonal sampling and is described by the sampling matrix and cost vectors:

$$D = \begin{bmatrix} 1 & 1 \\ 2 & -2 \end{bmatrix}, \quad \frac{1}{\pi} k_{0,1,2,3} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad \begin{bmatrix} 2 \\ -1 \end{bmatrix}$$

with $|\det(D)| = 4$. The ideal passband $S$ and modulation vectors $k_i$ are shown in Fig. 5(a) and Fig. 5(b) respectively. We see that the subband $S$ has more frequency components along the horizontal direction than the vertical direction.

**Design of the prototype filter:** Several general design techniques for multidimensional filters described in [6], [8] (e.g. frequency sampling technique) could be used to design the prototype filter. However, a recently reported technique described in [10] is very suitable for designing filters with a passband described by $S$. A 1D filter $h_1(n)$ with a nominal passband $[-\pi/|\det(D)|, \pi/|\det(D)|]$ is first designed using any of the 1D design techniques. An N-dimensional separable filter is then formed: $h_0(n) = \prod_{i=0}^{N-1} h_i(n_i)$. The result is obtained by sampling $h_0(n)$ with the sampling matrix $D = \text{adj}(D)/|\det(D)|$ D is $h(n) = h_0(Dn)$. 

4. CONCLUSIONS

The DFT modulated filter bank (in multidimensions with arbitrary sampling lattice) can achieve aliasing cancellation with FIR analysis and synthesis filters but in general IIR synthesis filters are required for perfect reconstruction. It has an efficient polyphase network implementation that employs the DFT. The shape of the frequency subbands need not be rectangular and the shape depends on the sampling lattice used.

REFERENCES

OVERSAMPLED DIGITAL LEAPFROG FILTERS

F. Corthay, F. Pellandini
Institute of Microtechnology, University Neuchâtel, Tivoli 28, CH-2003 Neuchâtel-Serrières, Switzerland.

A new digital filter architecture is presented which is based on the leapfrog simulation of passive analog LC filters. This filter type exploits a high oversampling in order to minimize the hardware of the devices. Indeed, due to the oversampling, the filter state variables are coded on a single bit and can thus be interpreted as pulse rates.

Two leapfrog synthesis methods are proposed: they correspond to the ladder and the lattice structures. An analysis method allows to evaluate the effects of the quantization of both coefficients and variables. The integrated circuit implementation of a filter can be realized by the abutment of a small set of basic bit slices.

The proposed filters provide an efficient alternative to classical systems for the processing of pulse frequency modulated signals. The interfaces between this kind of signals and the analog or the digital world are performed by simple readily available building blocks. Due to the high oversampling, the oversampled digital leapfrog filters are limited to applications with band edge frequencies of some kHz.

1. INTRODUCTION

Oversampled Digital LeapFrog Filters (ODLFs) rely on the simulation of an analog passive LC prototype in order to inherit its interesting properties. The analog active LeapFrog (LF) simulation of a reactance in its ladder expansion has been shown to be performed by a chain of integrators [1]. Within ODLFs, the digital implementation of the integrators in the LF structure is carried out by Up-Down Counters (UDCs) and Rate Multipliers (RMs).

The use of these basic operators reveal the inheritance from Quasi-Continuous Digital Filters (QCDFs) [2], [3], [4]. The QCDFs also implement integrator based structures using UDCs and RMs, but their internal signals are defined as pulse rates which are not synchronized to a system clock. This difference leads to a more difficult characterization of the QCDF signals and, above all, serious realization problems due to the asynchronous configuration of the pulses. A novel realization scheme of the RM allowed to refine the signal processing model of this operator and so to extend the insight of the ODLF structure.

With this, ODLFs can be described as a strictly digital filter class. Together with the proposed structure, a complete design method is presented, starting with the LF synthesis, scoping the analysis of transfer function deviation, and concluding with an integrated circuit realization [5].

The study of the ODLFs relies on the model of the workings of its operators. To these operators corresponds a specific coding of the signals. A clear outline of both operators and signals allows a straightforward approach of the complete ODLF investigation.

The proposed filtering method is derived from the fundamental choice to construct the filter hardware at the expense of speed. It offers an insight to the processing of pulse rate coded signals related to various domains such as quartz sensors or sigma-delta (ΣΔ) modulators.

The investigation of ODLFs starts with the presentation of the operators. Then, the synthesis shows the method used to derive an ODLF from its analog prototype. The analysis allows to determine the requirements for the appliance of the filter. An Application Specific Integrated Circuit (ASIC) design method is sketched. Finally, the whole topic is illustrated with the example of a benchmark filter design.

2. ODLF OPERATORS

The digital LF synthesis leads to different possible structures. In the Direct-transform Digital Integration (DDI) synthesis method [6], the integrators of the active LF are realized by accumulators. The integrator gains are set by a digital multiplier. ODLFs are DDI structures whose sampling rate is high enough to allow the coding of the filter variables on a single bit.
Coding the signals on a single bit trims the accumulators down to counters so that a LF accumulator with a positive and a negative input is implemented by an Up-Down Counter (UDC). The UDC implements the LF integrator function:

\[ H(s) = \frac{f_s}{2^{2\text{bits}}} \cdot \frac{1}{s} \]  

(1)

The multipliers setting the integrator gains also have to code their output to a single bit. This is efficiently implemented by Rate Multipliers (RMs). RMs are basically made out of a quantizer, which codes the bit-parallel input to a single bit, and a one bit multiplier which multiplies the quantizer output with a pulse rate coefficient input. The quantization algorithm of the RM makes use of dithering: a dither signal originating from a free running counter is added to the input coming from an UDC and the most significant bit of the sum is kept as the quantizer output. The RM coefficient is restricted to the domain \([0, 1]\).

In review, the cascade of an UDC and a RM is the ODLF equivalent of the integrator of the active LF. Inputs and outputs of this digital replication are coded in the form of pulse rates. Likewise, the filter coefficients too are provided to the RMs as pulse rates. Using unsigned arithmetic, these pulse rates are normalized in the domain \([0, 1]\), where the pulse rate coding the amplitude of 1 is equal to the sampling frequency of the filter.

ODLF synthesis is achieved in the same manner as analog integrator based filter synthesis, only the integrators of the active device are implemented by UDC-RM couples.

3. LF SYNTHESIS

For all-pole filters, the classical LF structure has been introduced by Girling and Good [1]. With this structure, the active filter simulation of a doubly terminated LC ladder two-port is performed by a chain of integrators. Figure 1 displays the schematic appearance of the corresponding fifth order ladder ODLF.

The integrator based simulation of an analog passive filter owning finite zeros is not as straightforward as the ladder LF synthesis. An interesting approach was proposed by Johns & Al. [7]. In this solution, an integrator chain corresponding to a singly loaded ladder determines the filter poles and the output is formed by the weighted sum of the state variables, so setting the zeroes of the filter. However, this structure leads to an ODLF implementation which is not canonic in terms of the number of operators.

For the most common amplitude approximation functions, the simulation of an analog lattice prototype shows to be preferable. The two lattice reactances are simulated by two ladder LF chains and the filter output is derived from the difference of the two chains, as depicted in figure 2. This procedure is similar to the synthesis of Lattice Wave Digital Filters (LWDFs) and a complementary output is also found for ODLFs as it is the case for LWDFs.

Additionally to the UDCs and RMs, the lattice structure requires a bit rate subtractor. In order to keep the output in the form of a bit rate, this operator can be realized as proposed by O'Leary & Al. [8]. Furthermore, a constant negative input \(R_{10}\) is fed to the last integrator of the LF chains in order to keep all state variables positive.

4. QUANTIZATION ERROR ANALYSIS

The analysis of the transfer function deviations is realized with the help of the state-space description of the filter. Two deviation sources are taken into account: the quantization of the coefficients and the quantization of the filter variables. As the oversampling is high and the device is based on integrator structures, the state-space description is made in the continuous time s-domain.
The state-space matrices of the ladder and the lattice filters can be obtained by the inspection of their schematic representation. As the LF links only interconnect neighboring integrators in a chain, the matrix A, whose eigenvalues are the poles of the system, is tridiagonal.

The quantization of the filter coefficients leads to a deterministic distortion of the transfer function. The quantized coefficients are inserted in the state-space matrices in order to estimate the transfer function of the actual filter and to check whether it still meets the specifications. Algorithms for coefficient quantization repeatedly execute this function in order to find an optimal coefficient set. Furthermore, the state-space matrices can be used to examine the sensitivity of the transfer function to coefficient values [9].

As opposed to coefficient quantization, the quantization of the signals inside a filter does not produce a deterministic distortion of the transfer function. Large scale distortions of the signals inside the filter originate from overflow of the UDCs. This effect is avoided by a proper scaling of the UDCs, which is achieved by adding least significant bits to the counters. The range which has to be provided to the counter outputs is estimated from the transfer functions between the input and the state variables. Small scale distortions of the signals emanate from the quantization inside the RM. These noise sources can be inserted in the state-space description of the filter and their cumulative effect is estimated as the maximal deviation of the transfer function.

The number of bits allocated to the UDCs determines the ODLF sampling rate. In effect, adding one bit to the UDCs requires to double the sampling rate in order to leave the transfer function described by equation (1) unchanged.

Analysis is based on reasonable assumptions, but filter simulation still requires a necessary tool for the verification of the proper functioning of the device.

5. INTEGRATED CIRCUIT REALIZATION

As well the UDC as the RM are basically made out of a counter. ASIC realization of these operators is successfully performed by the abutment of bit slices. Without too strenuous constraints, the operators can be designed for being abutted together so a complete LF chain is realized by the abutment of a limited set of basic slices.

The realization of the basic slices using synchronous counters has been developed [10]. More recently, the layout of the slices for asynchronous counters has proved to require even less silicon surface. Using a 1.6 μm technology, the core of the fifth order lattice filter represented in figure 2 requires an approximate area of 0.87 mm² (ca 1.45 mm x 0.6 mm)

6. BENCHMARK FILTER DESIGN EXAMPLE

The CCITT specifications G712 PCM have been extensively used as a filter benchmark [11]. They correspond to a lowpass filter. The passband ripple is of ±0.125 dB from 0 up to 3 kHz. The stopband attenuation is -14 dB from 4 kHz to 4.6 kHz and -32 dB higher.

As most implementations of this benchmark, a fifth order elliptic function is chosen for the filter approximation. This function is mapped into the lattice structure of figure 2.

Coefficient quantization has provided a suitable coefficient set: [9/16, 1/4, 1, 1/2, 9/16]. In this context, it is interesting to point out that the coefficient quantization does not affect the size of the RMs; the precision of the coefficients actually determines the size of the logic providing the pulse rate coefficients. In this point of view, it is also favorable to have identical coefficients.

Analysis of the transfer function deviation due to the quantization of the variables requires a 10 bit dynamic range for the counter outputs to keep the transfer function and its expected deviation within the specifications. Due to different DC working points of the state variables, this necessitates 11 bit operators. This choice of 11 bit operators leads to a sampling rate of 78 MHz for a cutoff frequency of 8 kHz.

![Figure 3 Transfer function deviation.](image-url)

Simulations with sine waves of different frequencies applied to the ODLF input are used to estimate the transfer function deviation of the device and to compare it with the expected values provided by the analysis. Figure 3 allows to compare the frequency responses issued from analysis and simulation.
The transfer function deviations of figure 3 have been estimated for a 10 bit device clocked at 39 MHz and there is a slight overlap of the frequency response on the specifications. The choice of the simulation of a simpler device arose from the large calculations and memory requirements which are bound to the simulation of any oversampled device.

Nevertheless, the comparison shows a satisfactory correspondence between the theoretical transfer function deviation provided by the analysis and the actual deviation obtained from the simulations.

7. CONCLUSIONS

Based on methods used for analog integrator based active filters, the design procedure of a new oversampled digital filter structure has been investigated.

In summary, the design methodology covers the topics of synthesis, analysis and realization of the filter class. It provides a valuable insight to the matter of oversampled digital filters.

The present work has focused on the topic of filters with variables coded on a single bit. The leading goal was the hardware simplicity of the device. As it is the case for ΣΔ modulators, an optimum can be found between the number of bits of the signals, which determines the oversampling, and the cost of the realization.

The derivation of a LF structure results in a low coefficient sensitivity and the canonic realization of the filter with respect to the number of operators. The description of a signal processing model of the RM allows to provide a valuable estimation of the effects of quantization inside the ODLFs.

The high oversampling is the major limitation to the applicability of the ODLFs. As a result, they are appropriate for the processing of low frequency signals. Important to note is that the oversampling of the ODLFs depends exponentially from the number of bits of the operators, this is from the required signal to noise ratio. For this reason, the use of higher order filters is mostly of interest for specifications bearing narrow transition bands rather than severe attenuations. In addition, the ODLF delivers in its passband an output frequency corresponding to the input rate. As such, it can be regarded as a digital Phase Locked Loop (PLL) and applied for tasks such as digitally controlled frequency synthesis or the demodulation of frequency modulated signals.

ACKNOWLEDGEMENTS

This work has been supported by the Swiss National Foundation for Scientific Research under project numbers FN 20-5627.88 and FN 20-25536.90.

REFERENCES


USING THE ROOT DATASIEVE TO RECOGNISE AND
LOCATE PATTERNS IN A DATA SEQUENCE

J. Andrew Bangham

Schools of Biology and Information Systems, University of East Anglia,
Norwich, NR4 7TJ, UK.

Abstract
The root datasieve is an arrangement of median filters that is suitable for
decomposing a digital signal to a set of rectangular pulses or rects. These
characterise a one dimensional pattern in terms of spatial, as opposed to
amplitude, features. It performs a multiscale decomposition. This is illustrated by
analysing a simple spatial pattern, a set of fixed width bands, and using the result
to design a non-linear matched filter, or sieve. The matched sieve is remarkably
tolerant of variations in band amplitude but can, never-the-less, select out the
pattern from a data signal that includes non-stationary and random noise.

Introduction
Median [1] and related stack [2,3,4] filters are attractive because of their edge
preserving and statistically robust properties. Recent analysis [5] has shown their
relationship with morphological filters [7,8] and positive boolean functions applied to the
thresholded binary data streams [6]. The datasieve is based on median filters.

It has recently been argued [10,11] that the root datasieve is an effective method for
removing random noise, $E$, from a signal $D$ whilst also preserving an underlying
pulsatile signal, $U$, where $D=U+E$. The critical feature of the root datasieve is the fidelity
with which it preserves edges and discriminates between pulses of different
lengths. Let the length be $m$. In this simple
one dimensional case, $m$ is a measure of a
spatial characteristic.

The datasieve is a multistage device in which successive stages, each is a median
filter, filter out rectangular pulses (rects) of
increasing $m$. However, it is not immediately
clear from a theoretical viewpoint that the
resulting rects would be sufficiently
independent of each other to make them a
reliable 'signature' for a spatial pattern. So a
practical test has been devised.

This paper shows that by matching the
passband properties of a root datasieve to the
spatial properties of a test signal, a matched
filter can be designed that selectively passes
only the specified patterns. To distinguish it
from linear matched filters this matched non-
linear filter will be called a 'matched sieve'.

Methods
Imagine a pattern, $U$, of thick and thin
bands on a cloth fabric, a tartan perhaps as
illustrated in Fig.1G. Within each repeating
unit the bands are of fixed but differing
width and are uniformly distant from one another. If a one dimensional signal is
obtained by making a single scan of the pattern, the bands form a signal such as that
shown in Fig. 1F. The band amplitudes vary
but the width and spacing does not. Fig. 1A.
shows, by way of a synthetic example, the
intensities that may occur as result of
scanning an image with non-uniform lighting,
sheen and dirt. The associated amplitude
signal is shown by Fig. 1B. It is the spatial
characteristics and edges of this pattern unit
that make it distinctive, not the intensity.
Selecting this pattern is, therefore, analogous
to recognising features in an image.

Spatial analysis is performed by a root
datasieve. It is cascade of root median filters.
A standard median filter runs the data
through a window of $2m+1$ samples and finds
the median at each position and a root is
achieved by repeatedly filtering through a
$2m+1$ median filter until there is no further
change. A root datasieve is an arrangement of
root median filters in which the signal is
median filtered with window of 3, then the output is filtered through a median filter with a window of 5 and so forth to a window of 2m+1. Despite being superficially complex, this strategy has an order complexity than depends on m^3 [10,11] because each stage implicitly subsamples the previous stage.

By subtracting the output of each stage (m) of the root dataseive from its input, a profile of what is 'removed' by each stage is obtained (Fig.2). These are rectangular pulses called, for the moment, rects(). Rects are characterised by the relative position at which they occur (x), an amplitude (y) and width (m). In other words, each stage of the sieve removes rects of a different scale, where scale is represented by m. This is analogous to multiscale analysis by wavelets [12].

Results

Fig.2 A shows the amplitude profile of the underlying signal U. Fig.2 B shows the corresponding spatial decomposition using the root dataseive. Each unit of the pattern is represented by six rects. If the rect at m=21, rect(x,y,21), is used as a positional reference the others occur offset and with spatial characteristics, m, as follows: rect(x+4, y, 12), rect(x+21, y, 9), rect(x-7, y, 7), rect(x-12, y, 5) and rect(x+30, y, 4). Fig. 2 B shows the result of decomposing the raw test data (shown in Fig. 1B) into rects. A matched sieve has to select from these the appropriate subset

\{(rect(x1,y1,m1), (rect(x2,y2,m2),
  (rect(x3,y3,m3), (rect(x4,y4,m4),
  (rect(x5,y5,m5)); x2=x1+4, x3=x1+21,
  x4=x1-7, x5=x1-12, x6=x1+30, m1=21,
  m2=12, m3=9, m4=7, m5=5, m6=4)\}

Notice that the pulse amplitude is not constrained, but it could have been. The result of filtering the underlying signal Fig. 1F through this device is shown in Fig.1 E. As expected they are identical. The question is how well does the device distinguish this pattern from others?

A preliminary result from an exact match sieve is shown in Fig.1 D. The result is similar to E but two units of the pattern are missing. However, it is possible to increase the tolerance of the matched filter. The simplest method is to accommodate a match if at any position at least, say, four rects are present. It is also possible to accept a slight mismatch in x and m. The result of accepting a match within x±2, m±2 for only four out of six rects is shown in Fig. 1C. To ease comparison with the original underlying patterns the output from the tolerant matched sieve is plotted as a dotted line over the original in Fig. 2 A. They are very similar.

Conclusions

It has been shown that the properties of the non-linear root dataseive, and by implication the single pass dataseive [10], can be used as a pattern recognizer. In this example the filter design was manual however adaptive sieves are also possible [cf. 12]. These results also lead to the possibility of designing two dimensional matched sieves that may take the place of linear spatial filters in models of primate vision processes [14].

References


Figure 1. Amplitude traces B to F are all to the same scale, likewise the two intensity profiles. A linear intensity bar is shown in H. A) Shows the intensity profile of the test signal. It could be a section through an image. B) Shows the amplitude of the test signal. Notice that the underlying patterns are interspersed both with noise patterns that are similar to the test patterns and with random and impulsive noise and a changing baseline. C) Shows the result of filtering the test signal (B) through the matched sieve accepting a loose match of m±2, x±2 and n-1 matching features. D) The same as (C) but using a matched sieve that only passes features that match exactly. E) Shows the result of filtering the underlying patterns through the matched sieve (exact match). F) Shows the underlying pattern. G) Similar to A but showing the underlying intensity pattern. The arrows point to patterns with bands of different amplitudes (1,2,3,5) and to noise (4,6).

Figure 2 on next page.
Figure 2. A) Shows a one dimensional pattern. The solid line shows the original underlying pattern U and the hatched line shows the result of filtering the test signal D through a matching sieve. B) Shows a multiscale root data sieve decomposition of the original set of patterns, U. The abscissa is the same as in A. Scale, m, is plotted logarithmically down the ordinate and intensity represents rect amplitude. Grey is zero. C) Similar to B, but shows the result of decomposing the test signal. Notice that the underlying patterns can be discerned from the noise. The intensity of
A BEZOUT RESULTANT BASED STABILITY TEST FOR 2-D DIGITAL RECURSIVE FILTERS

M. BENIDIR* and M. BARRET**

* Université de Paris-Sud, L2S-ESE, Plateau de Moulon, 91192 Gif-sur-Yvette, France
** ESE, Établissement de Metz, 4 Rue É. Belin, Technopôle 2000, 57078 Metz, France

A simple algorithm based on the Bezout resultant of two polynomials is proposed for testing the stability of 2-D digital recursive filters. The method consists in the calculation of the determinant of a matrix whose elements are polynomials. This determinant is 1-D polynomial and we have to test if it has zeros on the unit circle or not. The proposed method is compared to those given in recent works.

1. INTRODUCTION

The bidimensional (2-D) recursive digital filters have a wide range of applications in different areas such as geophysics, processing of radar and sonar data, medical imagery, etc... The stability constraint of these filters is a major problem and there is a growing need for the development of procedures for testing this stability. The fundamental stability theorems (necessary and sufficient conditions) have been established by Huang, Justice, Shanks, Anderson, Jury, DeCarlo, Fettweis [1]–[10]. The procedures for testing these conditions introduce many computations and numerical algorithms that reduce the volume of numerical computations have recently been introduced. In this paper, we propose a method based on the computation of the Bezout resultant of two bidimensional polynomials for testing the stability of 2-D digital recursive filters. The complexity of the proposed method is studied and several examples are discussed.

2. THE BEZOUT RESULTANT

We consider a first-quadrant 2-D linear shift-invariant digital filter, described by its transfer function \( G(z_1, z_2) = A(z_1, z_2) / B(z_1, z_2) \) where \( A(z_1, z_2) \) and \( B(z_1, z_2) \) are polynomials in the independent complex variables \( z_1 \) and \( z_2 \). In the following, we assume that the transfer function \( G(z_1, z_2) \) is irreducible and \( G(z_1, z_2) \) has no non essential singularities of the second kind (NESSK). In this case, the filter is stable if, and only if, [1] [2] its denominator satisfies the following condition:

\[
B(z_1, z_2) \neq 0, \quad (z_1, z_2) \in \tilde{U}^2
\]

where \( \tilde{U} = \{ z / |z| \leq 1 \} \) denotes the closed unit disk. However, condition (1) is complicated to test and it is shown [1] [2] [8] that this condition can equivalently be replaced by the conjunction of the three following conditions (Strintzis theorem [5]):

\[
B(z_1, 0) \neq 0, \quad z_1 \in \tilde{U}
\]

\[
B(1, z_2) \neq 0, \quad z_2 \in \tilde{U}
\]

\[
B(z_1, z_2) \neq 0, \quad (z_1, z_2) \in T^2
\]

where \( T = \{ z / |z| = 1 \} \) denotes the unit circle.

Conditions (2) and (3) are very simple to test by using classical procedures, well-known for the 1-D case. Condition (4), however, is still complicated to test. For practical applications, we have to reduce the volume of computations introduced by (4). For simplicity we suppose that \( B(z_1, z_2) \) has real coefficients, the proposed method can easily be extended to any complex polynomial. A classical method to determine if two 1-D polynomials have any common factors consists in computing the resultant of these polynomials. In order to extend this method the 2-D case, we have to define the Bezout resultant. For this, let us associate to

\[
B(z_1, z_2) = \sum_{k=0}^{m} \sum_{l=0}^{m} a_{k,l} z_1^k z_2^l = \sum_{i=0}^{m} a(i) z_2^i
\]

its reciprocal, defined by

\[
B^*(z_1, z_2) = z_1^n z_2^m B(z_1^{-1}, z_2^{-1}) = \sum_{i=0}^{m} a_{-i,i} z_1^i z_2^l
\]
where for \(0 \leq l \leq m\)
\[
a_k(z_1) = \sum_{k=0}^{m} a_{k,l} z_1^k \quad \text{and} \quad a_1^z(z_1) = z_1^n a(z_1^{-1}). \quad (7)
\]

We can suppose without loss of generality that \(m \leq n\). The zeros of \(B(z_1, z_2)\) on the unit bidisc \((\mathbb{T}^2)\) are common zeros of \(B(z_1, z_2)\) and \(B^*(z_1, z_2)\). Then it is natural to introduce the Bezout resultant \(R(z_1)\) of
\[
B(z_1, z_2) = 0 \quad \text{and} \quad B^*(z_1, z_2) = 0. \quad (8)
\]

This method has already been exposed by Lefnord-Chipart [5] for monodimensional polynomials. In this paper, we extend their results to bidimensional polynomials. Consider the \(2m\) polynomials constructed from \(B(z_1, z_2)\) and \(B^*(z_1, z_2)\) as follows: for \(1 \leq k \leq m\)
\[
B_k(z_1, z) = a_{m,k} z_1^k + \ldots + a_0(z_1) \quad \text{(9a)}
\]
\[
B_k^*(z_1, z) = a_{0,k} z_1^k + \ldots + a_{m,k}^*(z_1). \quad (9b)
\]

**Comment** Relations (9a-b), extended to \(k=0\), give
\[
B_0(z_1, z) = B(z_1, z) \quad \text{and} \quad B_0^*(z_1, z) = B^*(z_1, z). \quad \text{When} \ z_1 \quad \text{is a constant, (9a-b) are the Bezout polynomials of (8).}
\]

Now consider the polynomial expanded in power of \(z_2\)
\[
F(z_1, z_2, z) = B(z_1, z_2)B^*(z_1, z) \div B^*(z_1, z_2) \div B(z_1, z_2) = F_2(z_1, z_2)z_2 + \ldots + F_m(z_1, z_2)z_2^{m-1} \quad (10)
\]

**Lemma 1.** (Proof in the Appendix) The degrees with respect to \(z\) of polynomials \(F_k(z_1, z), 1 \leq k \leq m, \) (10) are not greater than \(m - 1\) and the \(F_k(z_1, z)\) are given by
\[
F_k(z_1, z) = B_k(z_1, z)B^*(z_1, z_2) = B_k^*(z_1, z)B(z_1, z_2). \quad (11)
\]

The \(c_k(z_1)\) appearing in the following form of \(F_k(z_1, z)\):
\[
F_k(z_1, z) = c_k(z_1) + c_{k-1}(z_1)z + \ldots + c_{m,n}(z_1)z^{m-1} \quad (12)
\]
satisfy
\[
c_{m,n}(z_1) = c_{1,n}(z_1) \quad \text{for} \ 1 \leq i, k \leq m. \quad (13)
\]

**Definition** The Bezout resultant \(R(z_1)\) of (8) is defined by
\[
R(z_1) = \begin{vmatrix}
 \cdots & a_{1,m}(z_1) \\
 \cdots & \cdots \\
 c_{m,1}(z_1) & \cdots & c_{m,n}(z_1)
\end{vmatrix}. \quad (14)
\]

The determinant \(R(z_1)\) is a polynomial and the location of its zeros is related to this of \(B(z_1, z_2)\) as follows.

**Lemma 2** (Proof in the Appendix) With the previous notations, let \(\lambda\) be a complex number such that either \(a_{m}(\lambda) \neq 0\) or \(a_0^*(\lambda) \neq 0\). Then, \(R(\lambda) = 0\) if, and only if, \(B(\lambda, z)\) and \(B^*(\lambda, z)\) have a common zero. The hypothesis \(a_{m}(\lambda) \neq 0\) or \(a_0^*(\lambda) \neq 0\) can be omitted in the only if part.

**Theorem 1** The polynomial \(B(z_1, z_2)\) doesn't vanish on the closed unit bidisk \(\overline{U}^2\) if, and only if, the three following conditions hold.
\[
B(z_1, 0) \neq 0 \quad \text{for all} \ z_1 \in U \quad (16)
\]
\[
B(1, z_2) \neq 0 \quad \text{for all} \ z_2 \in U \quad (17)
\]
\[
R(z_1) \neq 0 \quad \text{for all} \ z_1 \in T. \quad (18)
\]

**Proof** Assuming that (1) holds, (16) and (17) are obvious. Let us suppose there exists \(\lambda \in T\) such that \(R(\lambda) = 0\). If \(a_m(\lambda) = a_0^*(\lambda) = 0\), then \(a_0(\lambda) = \lambda a_k(\lambda) = 0\). Thus \(B(\lambda, 0) = 0\) and this contradicts the hypothesis. If either \(a_m(\lambda) \neq 0\) or \(a_0^*(\lambda) \neq 0\), then (Lemma 2) there exists \(\zeta\) such that
\[
B(\lambda, \zeta) = B^*(\lambda, \zeta) = 0. \quad (19)
\]

Thus
\[
B^*(\lambda, \zeta) = B(\lambda, \zeta^{-1}) = 0. \quad (20)
\]

As either \((\lambda, \zeta)\) or \((\lambda, \zeta^{-1})\) \(\in \overline{U}^2\) there is a contradiction. Conversely, according to Lemma 1, if \(B(z_1, z_2) = 0\) on \(T^2\), then \(R(z_1) = 0\) on \(T\). The end of the proof is a direct consequence of abovementioned Sturitzis theorem.

The first two conditions of theorem 2 are easy to test using classical stability tests for 1-D digital filters. When \(R(z_1)\) is expanded in powers of \(z_1\), the third condition can...
easily be tested by the extended Schur-Cohn procedure proposed in [13]. In another way, we will see in appendix that \( R(z_1) \) has a degree not greater than \( 2nm \) and satisfy:

\[
R(z_1) = z_1^{2nm} R(z_1^{-1})
\]  

(21)

Then it is possible without many computations to find the polynomial \( Q(u) \) of degree \( nm \) which satisfies:

\[
z_1^{2nm} R(z_1) = Q(u) \quad \text{where} \quad u = z_1 + z_1^{-1}.
\]  

(22)

It is well-known that \( R(z_1) \) doesn't vanish on the UC if, and only if, \( Q(u) \) doesn't vanish on the real segment \([-2; 2]\). This last condition can be tested by Sturm theorem [11]. The main difficulty is to compute \( R(z_1) \). We can easily extend the results of [10] to show that the \( z_1 \)-matrix \( M(z_1) \) is equal to the product of the \( z_1 \)-matrix \( F(z_1) \) of dimension \( m \times 2m \) and \( G(z_1) \) of dimension \( 2m \times m \), given by

\[
F(z_1) = \begin{pmatrix}
a_1(z_1) & -a_{m-1}(z_1) & a_2(z_1) & -a_{m-2}(z_1) & \cdots & -a_0(z_1) \\
a_2(z_1) & -a_{m-2}(z_1) & a_3(z_1) & -a_{m-3}(z_1) & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\
a_m(z_1) & -a_0(z_1) & 0 & 0 & \cdots & 0
\end{pmatrix}
\]  

(23)

and

\[
G(z_1) = \begin{pmatrix}
a_m(z_1) & a_{m-1}(z_1) & \cdots & a_0(z_1) \\
a_0(z_1) & a_1(z_1) & \cdots & a_{m-1}(z_1) \\
0 & a_1(z_1) & \cdots & a_{m-2}(z_1) \\
0 & a_0(z_1) & \cdots & a_{m-2}(z_1) \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & a_0(z_1)
\end{pmatrix}
\]  

(24)

The column \( k, k > 2, \) of \( F(z_1) \) is obtained from its column \( k - 2 \) by shifting one row to the top and adding a zero on the last row. The row \( k, k > 2, \) of \( G(z_1) \) is obtained from its row \( k - 2 \) by shifting one column to the right and adding a zero at the left of the row \( k.\)

The symmetries of \( M(z_1), \) with respect to its two diagonals (see the appendix), can reduce the volume of its computation. Below, we will propose a method to determine \( R(z_1). \) First, let us consider some examples.

3 APPLICATIONS

Example 1: Let \( B(z_1, z_2) = \sum_{k=0}^{n} a_{k,0} z_1^k + z_2 \sum_{k=0}^{n} a_{k,1} z_1^k. \)

Then

\[
R(z_1) = \sum_{l=0}^{n-1} \left( \sum_{m=0}^{l} a_{l-m,0} a_{n-l,1} - \sum_{m=0}^{l} a_{l-m,1} a_{n-l,0} \right) z_1^l + z_1^{2nm-1} + \left( \sum_{m=0}^{n-1} a_{l-m,1} - \sum_{m=0}^{n-1} a_{l-m,0} \right) z_1^n
\]  

(25)

Example 2 [8]: \( B(z_1, z_2) = 8 + 2z_1 + (6 + 5z_1)z_2 + (1 + 2z_1)z_2^2 \)

\( Q(u) = -2944 (6 - u) , \) then the 2-D filter is stable.

Example 3: \( B(z_1, z_2) = 36 + 60z_1 + 37z_1^2 + 13z_1^3 + z_1^4 \)

\( + (1 + 3z_1 + 2z_1^2 + 5z_1^3 + 4z_1^4) z_2 \)

\( Q(u) = -2196 - 3302u - 1996u^2 - 511u^3 - 32u^4. \)

\( Q(-2) = 0 \) and \( B(-1, 1) = 0 : \) the 2-D filter is unstable.

Example 4 [6]: \( B(z_1, z_2) = z_1^3 z_2^3 + (z_1^2 + 2z_1) z_2^2 + (z_1 + 2z_1^2 + 5z_1^3 + 4z_1^4 + 8z_1^5 + x_1^6) \)

\( Q(u) = -87616 + 89898u + 14522u^2 + 13997u^3 + 10332u^4 + 7052u^5 + 3392u^6 + 1392u^7 + 448u^8 + 64u^9 \) and the 2-D filter is stable.

4 CALCULATION OF \( R(z_1) \)

B. T. O'Connor and T. S. Huang [7] state, a congruence method is the most efficient to compute \( R(z_1). \) The total number of calculations needed is approximately \( O((2mn)^2 + m^2) \). If \( n \) is close to \( m \) this gives \( O(6m^2) \) operations. This is really too much for large values of \( m. \) But, for \( m = 2, 3 \) or \( 4, \) the proposed method is competitive. For \( m = 2, \) the number of operations for computing \( M(z_1), \) using the symmetries of this matrix is \( O(3m^2) \) multiplications and \( O(3m^2) \) additions. To compute the autoreciprocal polynomial \( R(z_1) \) it takes \( O(3m^2) \) multiplications and \( O(3m^2) \) additions. To compute \( R(z_1) \) from \( B(z_1, z_2) \) it takes \( O(12n^2) \) operations. For \( m = 3, \) the calculation of this matrix needs \( O(12n^2) \) operations, to compute \( R(z_1) \) from \( B(z_1, z_2) \) it needs \( O(58n^2) \) operations. For \( m = 4, \) the calculation of the matrix needs \( O(20n^2) \) operations. To compute \( R(z_1) \) from \( B(z_1, z_2) \) it needs \( O(238n^2) \) operations.
Appendix

Proof of Lemma 1: From (10) we deduce

\[ F(z_1, z_2, z) = B^*(z_1, z) \frac{B(z_1, z_2) - B(z_1, z)}{z_2 - z} \]

\[ B(z_1, z) \frac{B^*(z_1, z_2) - B(z_1, z)}{z_2 - z} = \sum_{k} b_m(z_1, z)z_2^{-k} \]

(26)

(27)

We obtain a similar relation replacing \( B \) by \( B^* \), then (11) results from (10) and (27). We see in (10) that \( F(z_1, z_2, z) = F(z_1, z, z_2) \), then we deduce (12) and (13) from (27).

Proof of Theorem 1: Replacing the first column of (15) by the linear combination, given in (12), implies that \( R(z_1) \) is equal, for all complex \( z \), to the determinant of the matrix (15) when the first column is replaced by \( \{ F_1(z_1, z), ..., F_m(z_1, z) \} \). If \( B^*(z_1, z) \) and \( B(z_1, z) \) have a common zero \((\lambda, \xi)\) then \( F_1(\lambda, \xi) = 0 \) for \( 1 \leq k \leq m \). Thus \( R(\lambda) = 0 \). Conversely, if \( a_m(\lambda) = 0 \) then the first column of \( A(\lambda) \) is null and \( R(\lambda) = 0 \). Assume either \( a_m(\lambda) = 0 \) or \( a_m(\lambda) \neq 0 \). If \( R(\lambda) = 0 \), we deduce of (12) and (15) that there are \( m \) complex numbers \( \mu_1, ..., \mu_m \), which depend on \( \lambda \) and are not all equal to zero such that for all complex \( z \mu_1 F_1(\lambda, z) + ... + \mu_m F_m(\lambda, z) = 0 \).

From (11) we can deduce that there exist two polynomials in \( z, P_2(z) \) and \( Q_2(z) \), with degrees not greater than \( m - 1 \), such that \( P_2(\lambda)B(\lambda, z) + Q_2(z)B^*(\lambda, z) = 0 \), then (Gauss theorem) \( B(\lambda, z) \) and \( B^*(\lambda, z) \) have a common prime factor. As the coefficient at the intersection of the \( k \)th row and the \( (l + 1) \)th column of \( R(z_1) \) is equal to:

\[ c_{k+l+1}(z_1) = \sum_{i=0}^{\min(k-l, l)} a_{k+l+1}^{(i)}(z_1) - \sum_{i=0}^{\min(k-l, l+1)} a_{k+l+1}^{(i)}(z_1) \]

(28)

Then \( c_{k+l+1}(z_1) \) is of degree \( \leq 2n \), so \( R(z_1) \) is of degree \( \leq 2mn \). We now show that \( R(z_1) \) is an autoreciprocal. If \( a_m(z_1) \) and \( a_m(z_2) \) have no common zero, and if \( R(0) \neq 0 \), then \( R(z_1) = 0 \), and only if, there exists \( \xi_2 \neq 0 \) such that \( B(z_1, \xi_2) = B^*(z_1, \xi_2) = 0 \). It is clear that \( B(z_1, \xi_1) = B^*(z_1, \xi_1) = 0 \) and then \( R(z_1, \xi_1) = 0 \). As the determinant is continuous, this result can be extended to any polynomial \( B(z_1, z_2) \). We can also verify from (28) that \( c_{m-k+l+1}(z_1) = \xi_2^2 c_{k+l+1}(z_1) \).

References


FIR DIGITAL FILTERS USING A LEAST-SQUARES DERIVED RECURSIVE SECTION

D. H. Horrocks and D. R. Bull

School of Electrical, Electronic and Systems Engineering, University of Wales, P.O. Box 904, Cardiff CF1 3YH, UK.

This paper presents a new approach to the realisation of finite impulse response digital filters. The technique utilises a transversal section in cascade with a small recursive section designed such that the preceding non-recursive part has minimum computational complexity in the least squares sense. This provides flexibility in filter response characteristics not offered by other similar methods. The paper outlines the design methodology and presents an example filter designed using this method.

1. Introduction

FIR digital filters have the advantages of linear phase and guaranteed stability. The former is achieved with symmetry in the filter impulse response while the latter is due to the fact that in a non-recursive realisation, all poles lie at the origin in the z-plane. In spite of these advantages, the large number of coefficient multiplications often required in such filters result in conventional direct-form implementations which are computationally complex.

Over the past few years numerous structures and design methods have been proposed which reduce or eliminate the multiplication burden in FIR filters. Such approaches are appropriate for application specific systems or filters which form a component in a larger integrated circuit with a requirement for high throughput coupled with low implementation complexity. Approaches based on the conventional transversal architecture, shown in figure 1, include the use of signed digit coding for coefficient representation [1]. Other methods have relied on benefits obtained from restricting coefficient values, either by limiting wordlength or, more usefully, by allowing an increased dynamic range but restricting the number of non-zero digits available [2]. These approaches do however often result in a deterioration in the filter frequency response characteristics. A powerful technique which results in the complete elimination of multipliers with no loss of coefficient accuracy has been presented by Bull and Horrocks [3] and involves the use of a primitive operator graph.

![Figure 1 Conventional FIR structure](image)

Competing approaches to FIR filter implementation have attempted to reduce implementation complexity through the use of a recursive realisation. These include the familiar frequency sampling structure where a comb filter is cascaded with a bank of resonators whose poles selectively cancel the zeros of the preceding section. If appropriately positioned, the poles can result in simple resonator coefficients. An alternative method employs a single low order recursive section, $B(z)^{-1}$, as shown in Figure 2. This is chosen such that the modified FIR section
H'(z) is realised with coefficients which are reduced to small integers while the cascade provides the same overall transfer function H(z), where

\[ H'(z) = H(z) \cdot B(z) \] (1)

Benvenuto et al [4] have described such an arrangement with simple recursive accumulator sections of the form \( B(z) = (1 - z^{-1})^k \). Coefficients, \( h'_j \), are limited to \{-1,0,+1\} and optimised using dynamic programming techniques. In earlier work by Van Gerwen et al [5], B(z) is selected from a family of five resonators with poles placed around the unit circle and with coefficient values for both non-recursive and recursive sections limited to small powers of 2. Such structures are economical in hardware but suffer from reduced performance if the centre frequency is not close to that of the cascaded resonator. This family was later extended to eleven by Lynn [6], who fills the gaps present in centre frequency coverage while still retaining the advantages of small integer coefficients.

![Figure 2 Modified FIR Structure](image)

The aim of this contribution is to present a new approach for the choice of B(z) based on a least-mean-square (LMS) approach. In this sense the recursive section is optimal for the given overall filter response and does not depend on the identification and selection from a limited set of pre-defined B(z) functions.

2. The Design Method

The required overall transfer function for an Nth order FIR filter is given by:

\[ H(z) = h_0 + h_1 z^{-1} + \cdots + h_N z^{-N} \] (2)

and the Mth order recursive section has the transfer function:

\[ B(z)^{-1} = 1/(b_0 + b_1 z^{-1} + \cdots + b_M z^{-M}) \] (3)

where \( b_0 = 1 \) is assumed without loss of generality.

Thus, to achieve the required overall transfer function, (2), the recursive section, (3), must be placed in cascade with a modified FIR section, \( H'(z) \), that satisfies the following relationship:

\[ H'(z) = H(z) \cdot B(z) = h'_0 + h'_1 z^{-1} + \cdots + h'_{N+M} z^{-N-M} \] (4)

The aim is to choose the coefficients \( \{b_j\} \) of the recursive section so that the coefficients, \( \{h'_j\} \) in the modified FIR section have minimum amplitudes. A convenient way to measure these amplitudes is by the sum of squares,

\[ P = \sum_j (h'_j)^2 \] (5)

Differentiating this expression with respect to each of the recursive-section coefficients \( \{b_j\} \) produces a set of linear equations involving the autocorrelation coefficients, \( r_k \), of the required coefficient sequence \( \{h_j\} \). The equations are similar in form to the normal equations arising from linear prediction theory and other optimal signal processing problems [7]. To illustrate, for the case \( N=4 \), they are:

\[
\begin{bmatrix}
    r_0 & r_1 & r_2 & r_3 & r_4 \\
    r_1 & r_0 & r_1 & r_2 & r_3 \\
    r_2 & r_1 & r_0 & r_1 & r_2 \\
    r_3 & r_2 & r_1 & r_0 & r_1 \\
    r_4 & r_3 & r_2 & r_1 & r_0 \\
\end{bmatrix}
\begin{bmatrix}
    b_0 \\
    b_1 \\
    b_2 \\
    b_3 \\
    b_4 \\
\end{bmatrix} = \begin{bmatrix}
    0 \\
    0 \\
    0 \\
    0 \\
    0 \\
\end{bmatrix} \] (6)

where

\[ r_k = \sum_{i=0}^{N-k} h_{i+k} \cdot h_i \] (7)

and \( b_0 = 1 \).
These equations could be solved for the optimal \{b_j\} and then the associated \{h_j\} obtained from equation (1). However most FIR filters are symmetric or antisymmetric and permit a folded direct-form that saves on multipliers. In order to preserve this desirable property \(B(z)\) must correspondingly be symmetric or antisymmetric. The effect of this constraint on the set of equations for \(M=4\) is:

\[
\begin{bmatrix}
2r_0 + 2r_4 & 2r_1 + 2r_3 & r_2 + r_2 & \vdots & b_0 \\
2r_1 + 2r_3 & 2r_0 + 2r_2 & r_1 + r_1 & \vdots & b_1 \\
r_2 + r_2 & r_1 + r_1 & r_0 & \vdots & b_2
\end{bmatrix} = \begin{bmatrix}
0 \\
0 \\
0
\end{bmatrix}
\] (8)

where the signs +/- are chosen according to whether the filter is symmetric or antisymmetric respectively. The form of these equations for other values of \(M\) is obvious.

Because of symmetry/antisymmetry, \(b_M=\pm 1\), or -1 and because \(b_M\) equals the product of the roots of \(B(z)\) it follows that at least some of the poles of the recursive section will lie on or outside the unit circle. However, because poles from the recursive section are exactly cancelled by corresponding zeros in the modified FIR section, \(H'(z)\), instability due to excitation of unstable poles by the filter input signal can be avoided. This can be ensured simply by placing the \(H'(z)\) section before the recursive section, \(B(z)^{-1}\), in the cascade. To prevent instability in the recursive part due to noise induced by signal quantisation, a further condition is that coefficients \(b_j\) be integer. As a consequence, coefficients \(h_j\) are also integer. In this contribution simple rounding is performed to convert \(b_j\) to integers.

Given the desired FIR response, \(H(z)\), and resonator order \(M\), the design method proceeds as follows:

i) calculate the autocorrelation coefficients \(r_k\) from equation (7),

ii) solve the set of equations (8) for \(b_j\) and round to integer values.

iii) Calculate the modified FIR coefficients, \(h_j\), using equation (1).

3. A Design Example

The impulse-response coefficients for a typical bandpass filter with \(N = 120\) and maximum coefficient magnitude of 64 are illustrated in figure 3. Resonator sections were generated over the range \(M=1\) to 15. The recursive section resulting in \(H'(z)\) with the smallest coefficient dynamic range is given in equation (9),

\[
B(z)^{-1} = \frac{1}{(1 - 2z^{-1} + 2z^{-3} - 2z^{-5} + z^{-6})}
\] (9)

This section increases the order of the system slightly to 126, but produces coefficients in \(H'(z)\) which have a maximum value of only 3. A comparison between the coefficient dynamic ranges of \(H'(z)\) and \(H(z)\) can be observed in figure 3.

Figure 3 FIR Coefficients

4. Conclusions

A method has been presented for the design of resonator structures for the difference coefficient approach to reducing the coefficient amplitudes in
FIR digital filter structures. Since FIR structures are typically of high order any reduction in computational complexity obtained is significant. The method presented here produces a recursive section that is optimal for the given overall transfer function, \( H(z) \), and does not depend on selection from predefined functions as do other methods.

References


EXPLICIT DESCRIPTION OF LATTICE STATE-SPACE WAVE DIGITAL FILTERS

M. Ansorge and F. Pellandini

Institute of Microtechnology, University of Neuchâtel, Rue Tivoli 28,
CH-2003 NEUCHATEL-Serrières, Switzerland.

This paper provides explicit formulas for Lattice State-Space Wave Digital Filters derived from reference filters based on cascaded all-pass sections and on cascaded unit elements. The achieved results show that the latter filters are the most interesting from an implementational point of view. Their complexity increases only linearly with the filter order compared to other state-space Wave Digital Filters, and their structure is particularly well suited for efficient realizations on vector processors and array processors.

1. INTRODUCTION

Wave Digital Filters (WDFs) are obtained from analog reference filters and are characterized by excellent numerical properties with respect to coefficient accuracy requirements, dynamic range, and all aspects of stability under finite-arithmetic conditions [1].

WDFs can be designed using many different filter structures [1], among which the lattice configurations show to be especially interesting for a broad range of applications, since they provide canonic and regular filter structures which can be designed in a straightforward way [2]. In addition, Lattice WDFs are well suited for the realization of half-band (or birciprocald) [2, 3], branching, and quasi linear phase filters.

The WDFs can advantageously be transformed into an equivalent state-space form leading to efficient hardware and software implementations (high regularity, maximum parallelism, high throughput, etc.) with a reduced round-off noise. By properly deriving the state-space form, so-called Essentially Equivalent State-Space WDFs can be obtained which preserve all the properties of the initial filter [4].

The present paper provides the explicit (symbolic) formulas of Lattice State-Space Wave Digital Filters (LSSWDFs) derived from reference filters where both lattice-branches are realized either by cascaded all-pass sections or by cascaded unit elements [1].

The usefulness of explicit filter descriptions is twofold. First, they give an essential insight into the properties of the filters to determine their overall performance and complexity characteristics. Second, they can efficiently be used for the finite wordlength optimization of filter coefficients, leading to an improvement of the computation accuracy and a reduction of the processing time.

The explicit formulas of a given filter can usually be obtained using either general purpose algebraic computation tools like Mathematica [5], or specific signal flow graph processing tools as SFD [6]. However, in case of regular filter structures such as lattice WDFs, it is even more effective to determine the formulas in a generic form by a proper analysis of the wave flow graph.

In the following sections, the basic properties of lattice WDFs are first reviewed. Then the explicit formulas of LSSWDFs are derived from the state-space realization of the lattice-branches. Finally, the achieved symbolic filter descriptions are discussed and compared with respect to their complexity.

2. LATTICE WAVE DIGITAL FILTERS

Lattice Wave Digital Filters (LWDFs) are basically derived from lossless symmetrical lattice reference filters [1]. A classical lattice two-port inserted between terminations with equal resistance \( R \) is represented in Fig. 1, \( Z_1 \) and \( Z_2 \) being the canonic lattice impedances. For reasons of simplicity, only the case with a single voltage source \( E \) applied to the two-port will be considered.

![Figure 1: Symmetrical lattice reference filter.](image)

The transmittance \( S_{21} \) of the two-port is given by:

\[
S_{21} = B_2 / A_1 = (S_2 - S_1) / 2
\]

(1)

\( A_1 \) being the incident wave at port 1, and \( B_2 \) the reflected wave at port 2. The terms \( S_1 \) and \( S_2 \) are the (canonic) reflectances of \( Z_1 \) and \( Z_2 \) respectively. \( S_1 \) and \( S_2 \) correspond to all-pass functions, since \( Z_1 \) and \( Z_2 \) are pure reactances. They can be expressed as:
\[ S_i = \frac{Z_i - R}{Z_i + R} ; \quad i = 1, 2. \]  
(2)

The realization of the lattice filters depends essentially on the realization of the canonic lattice impedances for which many structures have been proposed in the literature, e.g. [1]. Two configurations based on cascaded all-pass sections and on cascaded unit elements have shown to be particularly interesting and will be discussed in this paper.

In the next sections, the realization of a single lattice reflectance will be considered and all terms (i.e., variables, coefficients, indices, reactance order N) are implicitly related to it.

3. LATTICE STATE-SPACE WDFs

LSSWDFs are derived from reference WDFs by compacting the initial wave flow graph in function of the signal dependencies and by keeping only the signals corresponding to the filter delays [7, 4, 6]. These signals are then referred to as the state variables.

3.1 State-space description

The state-space description is defined in (3, 4), where the vector \( \mathbf{W} \) contains the state variables, while \( X \) and \( Y \) specify the input and output variables. \( A, B, C^T, \) and \( D \) are the usual state-space terms which can be merged into a single state-space matrix \( \mathbf{M} \) [1, 4].

\[
\begin{align*}
W(k+1) &= A \cdot W(k) + B \cdot X(k) \quad (3) \\
Y(k) &= C^T \cdot W(k) + D \cdot X(k) \quad (4)
\end{align*}
\]

\[
\begin{bmatrix}
W_1(k+1) \\
W_2(k+1) \\
W_3(k+1) \\
Y(k)
\end{bmatrix} =
\begin{bmatrix}
A & B \\
C^T & D
\end{bmatrix}
\begin{bmatrix}
W_1(k) \\
W_2(k) \\
W_3(k)
\end{bmatrix} +
\begin{bmatrix}
\alpha_{1,1} & \alpha_{1,2} & \alpha_{1,3} & \alpha_{1,4} \\
\alpha_{2,1} & \alpha_{2,2} & \alpha_{2,3} & \alpha_{2,4} \\
\alpha_{3,1} & \alpha_{3,2} & \alpha_{3,3} & \alpha_{3,4} \\
\alpha_{4,1} & \alpha_{4,2} & \alpha_{4,3} & \alpha_{4,4}
\end{bmatrix}
\begin{bmatrix}
1 \\
1 \\
1 \\
1
\end{bmatrix}
\begin{bmatrix}
-\gamma_m \\
(1 - \gamma_m) \\
\gamma_m \\
-\gamma_m
\end{bmatrix} \cdot \begin{bmatrix}
a_{1,m} \\
a_{2,m}
\end{bmatrix} \quad (5)
\]

3.2 Realization with cascaded all-pass sections

For LWDFs with cascaded first and second-order all-pass sections, the basic reflectances \( S_1 \) and \( S_2 \) can directly be designed using the approach proposed by Gazsi [2], leading to the structure represented in Fig. 2. The first-order section only exists for an odd reactance order \( N \), while the number of second-order sections is specified by \( m_{\text{max}} \):

\[
m_{\text{max}} = \begin{cases} 
N/2 & \text{for } N \text{ even} \\
(N-1)/2 & \text{for } N \text{ odd}
\end{cases} \quad (6)
\]

Compared to the approach given in [2], the same two-port adaptor is systematically used in the whole structure, since the scaling of state-space filters can be performed very easily in an ultimate step [4]. The considered two-port adaptor is illustrated in Fig. 3 and defined in (7), \( \gamma_m \) being the adaptor coefficient.

\[
\begin{bmatrix}
b_{1,m} \\
b_{2,m}
\end{bmatrix} = \begin{bmatrix}
-\gamma_m & (1 + \gamma_m) \\
(1 - \gamma_m) & \gamma_m
\end{bmatrix} \cdot \begin{bmatrix}
a_{1,m} \\
a_{2,m}
\end{bmatrix} \quad (7a)
\]

\[
-1 < \gamma_m < 1 \quad (7b)
\]

There exists a global signal dependency which is due to the direct link between the auxiliary variables \( V_m \) (Fig. 2). This signal dependency increases with \( N \) and has an important effect on the complexity of the state-space representation. Solving the linear recurrence defined by the variables \( V_m \), the explicit formulas of the lattice reactance can be determined for the state-space form. The final results are given in Listing 1, and the corresponding equations are provided in the appendix. As expected, the complexity of the obtained expressions grows with \( N \), see (A6). At the end, the state-space coefficients \( \alpha_i \) can be determined by matching the achieved expressions with (5).

Listing 1: Design algorithm for cascaded all-pass sect.

```plaintext
IF ODD(N) THEN
   Apply Formulas (A1, A2)
ELSE IF EVEN(N) THEN
   Apply Formulas (A3);
   \( \text{1st second-order section} \)
IF \( m_{\text{max}} \geq 1 \) THEN
   Apply Formulas (A4, A7 @ m=1);
   \( \text{2nd second-order section} \)
IF \( m_{\text{max}} \geq 2 \) THEN
   Apply Formulas (A5, A7 @ m=2);
   \( \text{Remaining second-order sections} \)
   \( m := 3; \)
   \( \text{WHILE} \ (m \leq m_{\text{max}}) \ \text{DO} \)
   \( \text{BEGIN} \)
   \( \text{Apply Formulas (A6, A7);} \)
   \( m := m + 1; \)
   \( \text{END; \ (m \leq m_{\text{max}})} \)
   \( \text{Apply Formulas (A8);} \)
```

Figure 2: Lattice react. with cascaded all-pass sect.

Figure 3: Two-port parallel adaptor
3.3 Realization with cascaded unit elements

It has been shown by Richards that any reactance of degree \( N \) can be realized by means of either an open-circuited or a short-circuited cascade of Unit Elements (UEs) [8, 9]. By properly retiming the obtained structure with an alternating delay distribution, an efficient configuration with only local signal dependencies can be derived. The resulting lattice reactance is illustrated in Fig. 4, where \( k_0 \) is defined as \( (k_0 = 1) \) for open-ended, and \( (k_0 = -1) \) for short-circuited reactances.

![Figure 4: Lattice reactance with cascaded UEs](image)

Again, the wave flow graph analysis, which is much simpler in this case, provides the explicit formulas which can be merged to get the formal design algorithm given in Listing 2. The state-space coefficients \( a_{ij} \) are then derived from the obtained expressions which are here independent of \( N \) (B1-B9).

```plaintext
IF (N = 1) THEN
  Apply_Formulas (B1,B2)
ELSE
  BEGIN (N > 1)
    { 1st UE handled separately }
    Apply_Formulas (B3,B4);
    { Intermediate sections with 2 UEs }
    m := 1;
    WHILE (2*m+1 < N) DO
      BEGIN
        Apply_Formulas (B5,B6);
        m := m+1;
      END;  { WHILE (2*m+1 < N) }
    { Final UE sections }
    IF ODD(N) THEN
      Apply_Formulas (B7,B8)
    ELSE  { EVEN(N) }
      Apply_Formula (B9);
  END;  (N > 1)

Listing 2: Design algorithm for cascaded UEs
```

4. PERFORMANCE COMPARISON

Comparing the explicit formulas which have been determined, the following observations can be made.

First, the number of significant state-space coefficients \( a_{ij} \) increases with a factor of \( (N^2 + 12N)/4 \) for lattice reactances based on cascaded all-pass sections, in contrast to a factor of \( 4N \) for lattice reactances with cascaded UEs. The situation is illustrated in Fig. 5 for complete (canonical and non-bireciprocal) state-space WDFs. For comparison purposes, the situation is also represented for ladder state-space WDFs, which have a growing factor of \( (N_F + 1)^2 \), \( N_F \) being the filter order.

Second, the complexity of the expressions defining the coefficients \( a_{ij} \) increases linearly with \( N \) for lattice reactances based on cascaded all-pass sections, while it remains constant for reactances with cascaded UEs. This aspect is important with regard to the finite wordlength optimization of the filter coefficients, as explained in [4, 6].

![Figure 5: WDF state-space matrix density](image)

Finally, Fig. 6 shows that the distribution of the significant coefficients \( a_{ij} \) within the state-space matrix is more structured and regular for lattice reactances with cascaded UEs. This topic is significant with respect to the implementation efficiency.

```
\[
\begin{array}{cccccccc}
  j=0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\
  i=0 & x & x & x & x & x & x & x \\
  1 & x & x & x & x & x & x & x \\
  2 & x & x & x & x & x & x & x \\
  3 & x & x & x & x & x & x & x \\
  4 & x & x & x & x & x & x & x \\
  5 & x & x & x & x & x & x & x \\
  6 & x & x & x & x & x & x & x \\
  7 & x & x & x & x & x & x & x \\
  8 & x & x & x & x & x & x & x \\
  9 & x & x & x & x & x & x & x \\
\end{array}
\]
```

![Figure 6: State-space lattice reactances](image)

From the former comparison criteria, the last two are the most important for practical applications, since the filter order \( N_F \) usually remains in a limited range. Clearly, LSSWDFs derived from reference WDFs with cascaded UEs are more efficient. From an implementational point of view, these filters are particularly well suited for effective realizations on vector processors [4], array processors [10], and systolic arrays for band matrix processing.
5. CONCLUSION

In this paper, the explicit formulas of Lattice State-Space Wave Digital Filters derived from reference filters based on cascaded all-pass sections and cascaded unit elements have been determined. The achieved results, which will be further refined in the future with a set of application examples, show that the second alternative is the most interesting from an implementational point of view.

Finally, it should be noticed that the presented approach can easily be extended to the design of bireciprocally LSSWDFs.

ACKNOWLEDGEMENTS

This project was supported by the Commission for the Promotion of Applied Scientific Research (Grant CERS C-2014.1).

REFERENCES


**APPENDIX**

Lattice reactances with cascaded all-pass sections

\[
W_0(k+1) = (1 - \gamma_0) X(k) + \gamma_0 W_0(k) \tag{A1}
\]
\[
V_0 = (1 + \gamma_0) W_0(k) - \gamma_0 X(k) \tag{A2}
\]
\[
V_0 = X(k) \tag{A3}
\]
\[
W_1(k+1) = (1 - \gamma_1) V_0 + \gamma_1 [1 + \gamma_1] W_2(k) - \gamma_2 W_1(k) \tag{A4}
\]
\[
W_3(k+1) = (1 - \gamma_3) \left[ \xi_{1,v} V_0 + \xi_{2,v} W_1(k) + \xi_{3,v} W_2(k) \right] + \gamma_3 \left[ (1 + \gamma_3) W_4(k) - \gamma_3 W_3(k) \right] \tag{A5}
\]
\[
W_2m(k+1) = (1 - \gamma_{2m}) W_{2m-1}(k) + \gamma_{2m} W_{2m}(k) \tag{A7}
\]
\[
Y(k) = v_{m_{\text{max}}} \tag{A8}
\]
\[
\xi_{1,m} = - \gamma_{2m-1} \tag{A9a}
\]
\[
\xi_{2,m} = -(1 + \gamma_{2m-1}) \gamma_{2m} \tag{A9b}
\]
\[
\xi_{3,m} = (1 + \gamma_{2m-1})(1 + \gamma_{2m}) \tag{A9c}
\]

Lattice reactances with cascaded unit elements

\[
W_0(k+1) = (1 - \gamma_0) X(k) + k_0 \gamma_0 W_0(k) \tag{B1}
\]
\[
Y(k) = - \gamma_1 X(k) + k_0 (1 + \gamma_1) W_1(k) \tag{B2}
\]
\[
W_1(k+1) = (1 - \gamma_1) X(k) - \gamma_1 \gamma_2 W_2(k) + \gamma_1 (1 + \gamma_2) W_2(k) \tag{B3}
\]
\[
Y(k) = - \gamma_1 X(k) - \gamma_2 (1 + \gamma_1) W_1(k) + (1 + \gamma_1) (1 + \gamma_2) W_2(k) \tag{B4}
\]
\[
W_2m(k+1) = -(1 - \gamma_{2m}) \gamma_{2m} W_{2m-1}(k) + \gamma_{2m} W_{2m}(k) \tag{B5}
\]
\[
W_2m+1(k+1) = (1 - \gamma_{2m+1}) (1 - \gamma_{2m}) W_{2m}(k) + \gamma_{2m} (1 - \gamma_{2m+1}) W_{2m}(k) + \gamma_{2m+1} W_{2m+1}(k) \tag{B6}
\]
\[
W_{N-1}(k+1) = - \gamma_N (1 - \gamma_{N-1}) W_{N-2}(k) - \gamma_{N-1} \gamma_N W_{N-1}(k) + k_0 (1 + \gamma_N) W_{N-1}(k) \tag{B7}
\]
\[
W_N(k+1) = (1 - \gamma_N) (1 - \gamma_{N-1}) W_{N-2}(k) + \gamma_{N-1} (1 - \gamma_N) W_{N-1}(k) + k_0 \gamma_N W_{N-1}(k) \tag{B8}
\]
\[
W_N(k+1) = k_0 \left[ (1 - \gamma_N) W_{N-1}(k) + \gamma_N W_{N-1}(k) \right] \tag{B9}
\]
DESIGN OF TWO-DIMENSIONAL VIDEO FILTERS WITH SPATIAL CONSTRAINTS

Valéry OUVRARD, Pierre SIOHAN

CCETT, 4 rue du Cloe Courtel, BP 59, 35512 Cesson-Sévigné, FRANCE
Tel: (33) 99.12.43.05 - Téléx: 740 284 F - Fax: (33) 99.12.40.98

A method based on linear programming is used to design optimal minimax 2-D FIR filters which have to satisfy frequency and space-time constraints. The ringing is controlled for edges having different shapes and orientations.

1. INTRODUCTION

Owing to the current fast technological evolution the use of multidimensional digital filters is becoming, nowadays, more and more frequent, even for high speed processing such as those involved in video applications. In most cases the video filters have to be designed in order to achieve an acceptable trade-off between different types of degradations. For instance when designing low-pass decimation filters one must to keep a maximum resolution while producing a minimum aliasing [1]. Another disturbing distortion is related to the oscillations called “ringing” that appear at the filter output when the input signal contains sharp transitions [2]. These ringing effects are directly related to the filter step responses. Thus, for Finite Impulse Response (FIR) filters, the interest of which is to allow a perfect linearity of the phase-frequency characteristic, the control of the step response results in a simple set of linear constraints. Therefore linear programming formulations can be derived for minimax FIR design problems with constraints on the step response. Originally proposed for 1-D FIR filter design [3, pp.180-183], this technique has been also applied to 2-D FIR filters [4]-[5]. However if the mathematical extension to the 2-D case of the linear programming formulation is straightforward an important difference between the problems in one or two dimensions comes from the fact that the 2-D unit step does not represent the whole set of all the possible sharp transitions. In this paper we consider step functions which not only correspond to a quadratic edge, but also to horizontal, vertical and diagonal edges. An analysis and design results corresponding to these different step functions are given.

2. PROBLEM STATEMENT

2.1. Edge orientation and 2-D unit sequences

Sharp transitions in a video signal may cause ringing [2], to avoid or, at least, limit this visually disturbing effect, filter design techniques including a control of the step response can be used. A straightforward solution to this problem has already been proposed for 1-D FIR digital filters [3]. The extension to 2-D FIR non-separable filters proposed in [4]-[5] has globally the same features, and both problems can be formulated as a linear program. However an important difference between the two problems is related to the definition of the step function. If in 1-D, the unit step is the unique mathematical model to describe a sharp transition this is no longer true in the 2-D case for the usual definition of the 2-D unit sequence [6, pp. 8], which only corresponds to a specific shape of transition.

<table>
<thead>
<tr>
<th>2-D Step functions</th>
<th>2-D Step responses</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Horizontal step</strong></td>
<td>( v_1(h) = \sum_{n_0=0}^{h_1} \sum_{n_2=0}^{h_2} h(n_0, n_2) )</td>
</tr>
<tr>
<td>( u_1(n_1, n_2) = 1 ) if ( n_2 \geq 0 )</td>
<td></td>
</tr>
<tr>
<td>( u_1(n_1, n_2) = 0 ) if ( n_2 &lt; 0 )</td>
<td></td>
</tr>
<tr>
<td>( A_1 = { [1], 1 \leq 1 \leq N_1 } )</td>
<td></td>
</tr>
<tr>
<td><strong>Vertical step</strong></td>
<td>( v_2(h) = \sum_{n_0=0}^{h_1} \sum_{n_2=0}^{h_2} h(n_0, n_2) )</td>
</tr>
<tr>
<td>( u_2(n_1, n_2) = 1 ) if ( n_1 \geq 0 )</td>
<td></td>
</tr>
<tr>
<td>( u_2(n_1, n_2) = 0 ) if ( n_1 &lt; 0 )</td>
<td></td>
</tr>
<tr>
<td>( A_2 = { [1], 1 \leq 1 \leq N_1 } )</td>
<td></td>
</tr>
<tr>
<td><strong>Quadrantal step</strong></td>
<td>( v_3(h, i) = \sum_{n_0=0}^{h_1} \sum_{n_2=0}^{h_2} h(n_0, n_2) )</td>
</tr>
<tr>
<td>( u_3(n_1, n_2) = 1 ) if ( n_1 \geq 0 ) and ( n_2 \geq 0 )</td>
<td></td>
</tr>
<tr>
<td>( u_3(n_1, n_2) = 0 ) otherwise</td>
<td></td>
</tr>
<tr>
<td>( A_3 = { [1], 1 \leq 1 \leq 1 \leq N_1 } )</td>
<td></td>
</tr>
<tr>
<td><strong>Diagonal step</strong></td>
<td>( v_4(h) = \sum_{n_0=0}^{h_1} \sum_{n_2=0}^{h_2} h(n_0, n_2) )</td>
</tr>
<tr>
<td>( u_4(n_1, n_2) = 1 ) if ( n_1 + n_2 \geq 0 )</td>
<td></td>
</tr>
<tr>
<td>( u_4(n_1, n_2) = 0 ) otherwise</td>
<td></td>
</tr>
<tr>
<td>( A_4 = { [1], 1 \leq 1 \leq N_2-1 } )</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Examples of 2-D step functions, and their corresponding step responses.

In Table 1, where it assumed that the 2-D FIR filter has support over the region \((n_1, n_2), 0 \leq n_1 \leq N_1-1, 0 \leq n_2 \leq N_2-1\), the 2-D unit step is denoted by \( u_3(n_1, n_2) \) and we call it the quadrantal step because it is non-zero over one quadrant in the \((n_1, n_2)\)-plane. Besides the 2-D unit step we have added three other step functions, the common characteristics of which are, firstly, to be non-zero over one half-plane in the \((n_1, n_2)\)-plane, and,
secondly, to correspond, for the horizontal and vertical steps, to directions to which the human visual system is particularly sensitive. Let card \( \Lambda_k \) with \( 1 \leq k \leq 4 \), denote the number of samples for the four different step functions. From Table I, it can be seen that the step response control will, in general, yield a greater number of constraints for the quadrantal step, for which card \( \Lambda_3 = N_1 N_2 \), than for the three other step functions, which yield respectively: card \( \Lambda_1 = N_1 \), card \( \Lambda_2 = N_2 \), card \( \Lambda_4 = N_1 + N_2 - 1 \). In Section III it will be shown that, if we choose to limit the ripples for the half-plane step functions, which yield a smaller number of constraints, it is easier to reach an acceptable trade-off between ringing and frequency selectivity.

2.2. Linear programming formulation

Let \( h(n_1, n_2), 0 \leq n_1 \leq N_1 - 1, 0 \leq n_2 \leq N_2 - 1 \), denote the impulse response of the 2-D FIR digital filter. If \( h(n_1, n_2) \) is symmetrical the filter has a linear phase characteristic with respect to the frequency and its frequency response can be written as:

\[
H(\omega_1, \omega_2) = H(\omega_1, \omega_2) e^{j\frac{N_1-1}{2} \omega_1 + j\frac{N_2-1}{2} \omega_2}
\]  

(1)

In the following we will assume that the filter has an eightfold symmetry and that, consequently, \( N_1 = N_2 = N \), \( H(\omega_1, \omega_2) \) being a real trigonometric polynomial such as:

\[
H(\omega_1, \omega_2) = \sum_{i=1}^{R} b(i) g_i(\omega_1, \omega_2)
\]  

(2)

The \( R \) independent coefficients \( b(i) \) depend linearly from the \( h(n_1, n_2) \)'s, these relations and the expression of the basis functions \( g_i(\omega_1, \omega_2) \) can be found in [6, pp. 128]. The use of symmetry allows us to reduce the number of variables which thus decreases from \( N^2 \) to a number \( R \) given by:

\[
R = \begin{cases} 
\frac{(N+1)(N+3)}{8} & \text{if } N \text{ odd} \\
\frac{N(N+2)}{8} & \text{if } N \text{ even}
\end{cases}
\]  

(3)

In the frequency domain we, similarly, assume that the desired frequency characteristic \( D(\omega_1, \omega_2) \) is octagonally symmetric, i.e. \( D(\omega_2, \omega_1) = D(\omega_1, \omega_2) \). We, now, want to obtain a set of coefficients \( b(i) \) which minimizes the weighted Chebyshev norm between the filter and the desired frequency response. Let \( W(\omega_1, \omega_2) \) be a positive weighting function the problem is to find the coefficients \( b(i) \) which minimize the following expression:

\[
\|W(\omega_1, \omega_2)|D(\omega_1, \omega_2) - H(\omega_1, \omega_2)| (\omega_1, \omega_2) \|_W
\]  

(4)

where \( F \) is a compact subset of the 2-D frequency plane.

After a discretization of the frequency domain \( F \) using a finite number of \( K_p \) discretization points the continuous minimax problem (4) may be approximated by a discrete problem, for which, a linear programming formulation is given by:

\[
\begin{aligned}
\text{Find the R coefficients } b(i) \text{ minimizing } & \delta (\delta > 0) \\
\text{Under the } 2K_p \text{ following constraints (} 1 \leq k \leq K_p \text{)} & \\
\sum_{i=1}^{K_p} b(i) g_i(\omega_{1k}, \omega_{2k}) - \frac{\delta}{W(\omega_{1k}, \omega_{2k})} & \leq D(\omega_{1k}, \omega_{2k}) \\
- \sum_{i=1}^{K_p} b(i) g_i(\omega_{1k}, \omega_{2k}) - \frac{\delta}{W(\omega_{1k}, \omega_{2k})} & \leq -D(\omega_{1k}, \omega_{2k})
\end{aligned}
\]  

(5)

If we want to obtain an accurate result for this later problem, the number of discretization points, \( K_p \), has to be much greater than the number of variables \( R \). Therefore, as in [1], [4] and [5], we choose to solve the dual problem of (4), the result of which can be obtained more quickly.

In Figure 1 the desired function for a diamond-shaped low-pass filter is represented. This typical shape of filter may be required in video applications such as scan conversion or HDTV encoding [1].

\[\text{Fig. 1. Desired function } D(\omega_1, \omega_2) \text{ for the diamond-shaped filter.}\]

2.3. Linear program including step response constraints

Let \( L_k(1) \) and \( U_k(1) \) denote the lower and upper bounds, respectively, for the functions \( v_k(1) \), \( 1 \) being, as shown in Table I, an integer index \( I_1 \) or \( I_2 \) or an integer vector \( (I_1, I_2) \). For a given step function \( v_k(1) \) the step response constraints are such as:

\[
L_k(1) \leq v_k(1) \leq U_k(1), I_k 1 \leq I_k \subseteq \Lambda_k
\]  

(6)

\( I_k \) is the subset of \( \Lambda_k \) where the inequalities (6) have to be satisfied. It can be seen from the different expressions of \( v_k(1) \) that they are linear with respect to the filter coefficients \( h(n_1, n_2) \). Therefore as the variables \( b(i) \) are such as \( b(i) = c.h(n_1, n_2) \)
with \( c = 1/2, 2 \) or 4, the linear constraints we add to the linear program (5) are such as:

\[
\begin{aligned}
&- \sum_{i=1}^{R} \alpha(i, k, l) b(i) \leq - L_k(I) \\
&\sum_{i=1}^{R} \alpha(i, k, l) b(i) \leq U_k(I)
\end{aligned}
\]

(7)

where \( \alpha(i, k, l) \) are real numbers and \( k \) is an integer value less or equal 4.

A graphical illustration of the step response constraints is given in Figure 2 for \( k = 1 \) or 2. In this example it can be noted that owing to the impulse response symmetry it is sufficient to limit the constraints to the \((N-1)/2\) first samples. Furthermore if we do not specify the raise time period we can take \( I_{Sk} = \{1, 2, \ldots, J_1\} \). In the examples presented in the next section we allow identical ripple values for the overshoot and undershoot.

![Fig. 2. One-dimensional step response constraints.](image)

**3. EXAMPLES**

The method used to control ringing outlined from the description occurring in Section 2, in other words one justify carrying out the filter design without spatial constraints, once this result is obtained we can determine the ripple for each step response given in Table 1. Therefore we can advocate several solutions in order to carry out a test, for example to determine which response produces the maximum ringing, the next stage is to reduce it, for example by a factor of two, however at the same time trying to avoid an increase in the maximum value for the other step response ripples. We can also imagine the case for lowering the maximum oscillations of each response. We should also emphasize that for each filter obtained, the optimal frequency error should not go beyond a fixed limit. The used filters will be diamond-shaped filters (cf. Fig. 1), the value \( x \) is the width of the transition band along the horizontal direction. Table II shows, for a filter size 13x13 and for filter specifications corresponding to \( \delta_{q} = 0.6\pi, x = 0.4\pi \), the maximum ripple for each step response (horizontal and vertical responses are the same) and the filter optimal approximation error. The weights \( W_1, W_2 \) in the passband and stopband regions, respectively, are constant and chosen such as \( W_1 = 1, W_2 = 0.1 \); thus the optimal error, denoted \( \delta_{opt} \), is given by \( \delta_{opt} = \max(\delta_1, 0.1\delta_2) \) where the \( \delta_i (1 \leq i \leq 2) \) represent the approximation error in each of the two specified regions. The first line of Table II contains the results without spatial constraints, while for the three following examples constraints are imposed, either to keep a maximum ripple which is less or equal than the initial value, or to reduce it to a given inferior value which is indicated by an asterix (*)

<table>
<thead>
<tr>
<th>( \delta_{opt} )</th>
<th>( \delta_{v} )</th>
<th>( \delta_{q} )</th>
<th>( \delta_{qb} )</th>
<th>( \delta_{qb} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.23E-3</td>
<td>8.04E-2</td>
<td>9.17E-2</td>
<td>6.11E-2</td>
<td>1.65E-1</td>
</tr>
<tr>
<td>1.78E-2</td>
<td>8.2E-2</td>
<td>4.5E-2</td>
<td>*</td>
<td>6.2E-2</td>
</tr>
<tr>
<td>2.94E-2</td>
<td>8.2E-2</td>
<td>8.2E-2</td>
<td>*</td>
<td>8.2E-2</td>
</tr>
<tr>
<td>1.34E-2</td>
<td>6.2E-2</td>
<td>6.2E-2</td>
<td>*</td>
<td>1.65E-1</td>
</tr>
</tbody>
</table>

Table II. Results for a 13x13 diamond-shaped filter with and without spatial constraints.

- \( \delta_{v} \): maximum ripple for the vertical step response
- \( \delta_{q} \): maximum ripple for the diagonal step response
- \( \delta_{qb} \): maximum ripple for the quarter step response (lower part)
- \( \delta_{qb} \): maximum ripple for the quarter step response (higher part)

**Fig. 3. Frequency approximation error obtained by varying the ripple for the filter of Table II.**

The results represented in Fig. 3 are obtained with the same filter and an identical simulation procedure, i.e. the maximum step response ripples we authorize are those given in the first line of Table II, then we compare the evolution of the approximation error with respect to the quadrantal and diagonal step response ripples. It can be seen that, owing to the smaller number of constraints involved, the approximation error is always less
higher with the diagonal step than with the quadrantal step. It can be added that the results corresponding to the horizontal (and vertical) step, not reported here, are quite near to those obtained with the diagonal step. The frequency responses and the diagonal step responses corresponding to the first two lines of Table II are reported in Fig. 4-7.

Fig. 4. Frequency response for the filter of Table II without spatial constraints (line 1 of Table II).

Fig. 5. Frequency response for the filter of Table II (line 2 of Table II).

Fig. 6. Diagonal step response for the filter of Table II without spatial constraints (line 1 of Table II).

Fig. 7. Diagonal step response for the filter of Table II (line 2 of Table II).

4. CONCLUSION

A 2-D FIR filter design method is proposed which takes into account frequency and space-time constraints. A control of the ringing is ensured for edges having different shapes and orientations.

REFERENCES


Improved Frequency-Sampling Two-Dimensional FIR Filter Design

MICHAEL LIGHTSTONE and SANJIT K. MITRA

Department of Electrical and Computer Engineering
University of California at Santa Barbara
Santa Barbara, CA 93106, USA

A new method is presented for the fast and efficient design of two-dimensional (2-D) linear phase FIR filters. In this approach, the concept of the structural subband decomposition is applied to the frequency sampling design of 2-D FIR filters. The filter is implemented as a parallel connection of \(KL\) sparse subfilters cascaded with interpolators. An algebraic relationship is then developed to determine the frequency samples of the \(KL\) subband filters from the original specified \(M \times N\) frequency samples. An efficient overall filter design results by individually designing the shorter length subfilters.

1 Introduction

Recently it has been shown that any FIR filter can be decomposed into a parallel connection of cascades of an interpolator and a sparse subfilter with each cascade contributing to the overall frequency response only within a given band of frequencies [1]. In the 2-D case, an FIR filter of size \(M \times N\) is decomposed into a parallel implementation (bank) of \(KL\) sparse 2-D FIR subfilters cascaded with interpolators. An elegant algebraic method is presented to determine the frequency samples of the subfilters in terms of the original specified frequency samples. The impulse responses of the subfilters can then be determined with much shorter length inverse DFT’s using the conventional frequency sampling approach [2]. In addition, for very narrow-band or very wide-band filters, many of the subbands contribute little to the overall frequency response. Thus, a large number of these subbands may be dropped from the overall structure with the remaining branches still matching the desired frequency specifications reasonably well, resulting in a significant reduction of the hardware requirements. If the original response is not narrow-band or wide-band, the number of discardable branches is smaller. However, the contributions of the remaining branches are not equal, and significant hardware reductions can result from selectively quantizing the wordsize of the coefficients for the subfilters.

2 2-D Structural Subband Decomposition [2]

Consider a 2-D sequence \(h(m, n)\) of size \(M \times N\) with z-transform \(H(z_1, z_2)\). If \(M = KP\) and \(N = LQ\), then the \(K \times L\) structural subband decomposition of \(H(z_1, z_2)\) is given by

\[
H(z_1, z_2) = \begin{pmatrix} \frac{1}{z_1} & \frac{1}{z_2} \end{pmatrix} \cdot \begin{pmatrix} F_1(z_1) & F_2(z_1) \\ F_1(z_2) & F_2(z_2) \end{pmatrix} \cdot \begin{pmatrix} \frac{1}{z_1} & \frac{1}{z_2} \end{pmatrix}
\]

where \(F_1(z_1)\) and \(F_2(z_2)\) are given by

\[
F_1(z_1) = \sum_{k=0}^{K-1} r_{ik} z_1^{-k}, \quad i = 0, 1, \ldots, K - 1, (3)
\]

\[
F_2(z_2) = \sum_{k=0}^{L-1} s_{kj} z_2^{-k}, \quad j = 0, 1, \ldots, L - 1. (4)
\]
Realizations of \( H(z_1, z_2) \) based on the structural subband decomposition given by Eqs. (1) and (2) are shown, respectively, in Figures 1 and 2. Note that the delays in the implementation of the subfilters in Figure 1 can be shared, leading to a canonic realization of the overall structure. As can be seen from Figure 2, the structural subband decomposition can be viewed as a generalization and 2-D extension of the interpolated FIR structure [3], where \( I_i(z_1) \) and \( F_j(z_2) \) are the interpolators, and \( G_{i,j}(z_1, z_2) \) is a 2-D shaping filter of size \( P \times Q \). Thus, the transfer function of an arbitrary 2-D FIR filter of size \( M \times N \) (where \( M \) and \( N \) are divisible by \( K \) and \( L \), respectively) can be decomposed into a bank of \( KL \) subbands and structurally implemented as a parallel combination of the \( KL \) subfilters with \( PQ \) nonzero coefficients and their respective interpolators.

**3 Determination of the Subband Frequency Samples**

A very attractive method for 2-D frequency-sampling design is introduced by transforming the 1-D analytic response given by Järvis et al [4] to a 2-D analytic response using the McClellan frequency transformation [5]

\[
\cos(\omega) = \frac{1}{2} \left( 1 + \frac{1}{2} \cos(\omega_1) + \frac{1}{2} \cos(\omega_2) + \frac{1}{2} \cos(\omega_1 \cos(\omega_2)) \right).
\]

This function, when sampled, provides approximately circularly symmetric 2-D FIR filters with passband and stopband characteristics very near that of the 1-D design with minimal computation. Though similar, this procedure is not equivalent to the frequency transformation method proposed by McClellan which is applied to an already designed 1-D impulse response and guarantees the amplitude characteristics of the 1-D filter will be preserved in the 2-D filter [5]. However, this approach does guarantee that the 1-D amplitude characteristic will be preserved along the horizontal and vertical frequencies, and in practice actually improves the amplitude characteristic along the diagonal frequencies.

In order to develop the frequency-sampling design of 2-D FIR filters based on the structural subband decomposition, the \( P \times Q \) frequency samples for each of the \( KL \) shaping filters \( G_{i,j}(z_1^M, z_2^N) \) must be determined in terms of original \( M \times N \) frequency samples of \( H(z_1, z_2) \). The corresponding impulse responses of the shaping filters may then be computed separately. To determine the frequency samples of each shaping filter, the \( z \)-transform of each shaping filter must be evaluated at \( P \times Q \) evenly spaced points in the \( \omega_1, \omega_2 \) plane. If we define

\[
\tilde{H}(m, n) = H(W_M^m, W_N^n), \quad m = 0, 1, \ldots, M - 1 \\
\quad n = 0, 1, \ldots, N - 1
\]

\[
\tilde{G}_{i,j}(k, l) = H(W_P^k, W_Q^l), \quad k = 0, 1, \ldots, P - 1 \\
\quad l = 0, 1, \ldots, Q - 1
\]

where \( W_N = e^{-j2\pi/N} \), we can obtain from Eq. (1) for \( k = 0, \ldots, P - 1 \), and \( l = 0, \ldots, Q - 1 \):

\[
\tilde{H} = W_R \tilde{G} S V^T
\]

where \( \tilde{H} \) and \( \tilde{G} \) are \( K \times L \) matrices whose \((i,j)\) elements are given by \( \tilde{H}(k + (i-1)P, l + (j-1)Q) \) and \( \tilde{G}_{i,j}(k, l) \), respectively, and \( V \) and \( W \) are \( L \times L \) and \( K \times K \) matrices whose \((i,j)\) entries are given by \( W_N^{(j-1)(k+(j-1)Q)} \) and \( W_N^{(j-1)(k+(j-1)P)} \), respectively. Note that \( V \) can be written as \( V = AB \) where \( A \) is an \( L \times L \) DFT matrix and \( B \) is the diagonal matrix given by

\[
B = \text{diag} \left( 1, W_N^1, \ldots, W_N^{(L-1)} \right).
\]

Similarly, \( W \) can be written as \( W = CD \) where \( C \) is a \( K \times K \) DFT matrix, and \( D \) is given by

\[
D = \text{diag} \left( 1, W_M^1, \ldots, W_M^{(K-1)} \right).
\]

The frequency samples of the subfilters can now be solved in terms of the frequency samples of the original \( M \times N \) filter by solving the matrix equation of Eq. (7) for \( k = 0, \ldots, P - 1 \), and \( l = 0, \ldots, Q - 1 \), resulting in:

\[
\tilde{G} = R^{-1}B^{-1}A^{-1}HC^{-1}D^{-1}S^{-1}.
\]

Note that because \( A \) and \( C \) are DFT matrices, and because \( B \) and \( D \) are diagonal, their inverses are easy to determine. In addition, \( A, C, S, \) and \( R \) are all independent of \( k \) and \( l \), so they need only be computed once. Furthermore, instead of actually computing \( \tilde{A}^{-1} \) and \( \tilde{C}^{-1} \) and performing matrix multiplications, a standard inverse FFT algorithm can be utilized. Thus, Eq. (11) provides a simple algebraic method to find the frequency samples of each shaping filter for the 2-D structural subband decomposition.
It is evident from the design procedure that several possibilities exist for the choices of the subband decomposition transform matrices, \( S \) and \( T \) with the only restriction being that the matrices be nonsingular. Two possibilities with integer valued coefficients are the Hadamard transform and binomial transform [5]. Their corresponding advantages and disadvantages are examined in [7,8]. A third choice for the transform matrix examined in this paper is the DCT. Its primary advantage is that the energy compaction property that is useful in image coding can be utilized in filter design as a method for reducing the average wordsize allocated to the filter coefficients of specific subfilters with little effect on the overall filter performance. Thus, for many filters (not necessarily narrow-band or wide-band), the DCT-based subband decomposition provides an implementation with an overall savings in the bit allocation of the filter coefficients as compared with a direct-form realization (as exemplified in Section 4). While the DCT does not have integer-valued elements, in practice, the wordsize allocated to each matrix element can be significantly reduced to just a few bits with little effect on the energy compaction property of the transform.

4 A Design Example

A design example is now presented to illustrate the advantage of DCT-based subband structures and variable bit allocation in the subfilters for 2-D frequency-sampling filter design. The original 1-D filter specifications are for a lowpass filter with \( \omega_s = 0.4\pi \), \( \omega_a = 0.52\pi \), \( \delta_p = 0.067 \), and \( \delta_s = 0.018 \). From these specifications the length of the 1-D filter is determined, and simple analytic functions are obtained for the passband, the transition band, and the stopband. Using the standard frequency-sampling approach, the 1-D impulse response of the designed filter is found from an inverse DFT of 24 samples evenly spaced from 0 to \( 2\pi \). Next, the 1-D analytic function is mapped to a new 2-D analytic function using the frequency transformation of Eq. (5) and sampled at \( 24 \times 24 \) evenly spaced points in the \( \omega_1, \omega_2 \) plane. The 2-D impulse response is then calculated from a \( 24 \times 24 \) 2-D inverse DFT. The resulting 2-D circularly symmetric filter specifications are \( \omega_p = 0.42\pi \), \( \omega_s = 0.56\pi \), \( \delta_p = 0.067 \), and \( \delta_s = 0.018 \). Next, using the DCT-based subband decomposition, \( 6 \times 6 \) frequency samples are calculated for each of the 16 subfilters for a \( 4 \times 4 \) band decomposition. In order to take advantage of the DCT's energy compaction property, the fixed bit allocation scheme for the subband coefficients in Figure 3 was used. Figure 4 shows a 2-D mesh plot of the resulting frequency response which was obtained with an average of 6.3 bits per filter coefficient and the addition of two \( 4 \times 4 \) DCT matrices quantized to 4 bits per matrix element. An equivalent response from a direct implementation of the \( 24 \times 24 \) impulse response requires a 13-bit wordsize for each filter coefficient. A 1-D plot of the worst case magnitude response at each radial frequency is shown in Figure 5 for both a direct implementation and the DCT-based subband approach.

References


Figure 1: Realization of 2-D structural subband decomposition based on the decomposition of Eq.(1) for $K=L=3$.

Figure 2: Realization of 2-D structural subband decomposition based on the decomposition of Eq.(2) for $K=L=3$.

Figure 3: Bit allocation scheme for subband filters.

Figure 4: Frequency response of 2D filter obtained using DCT-based decomposition and implemented with an average of 6.3 bits per filter coefficient.

Figure 5: Frequency response comparing the worst case magnitude response at each radial frequency for direct form (solid) implementation with 13 bits per coefficient and DCT-based decomposition (dashed) with 6.3 bits per coefficient.
OPTIMAL IMPLEMENTATION OF MULTI-CHANNEL FILTER BANKS

Erik I. Verriest\(^1\) and Gang Li\(^2\)

School of Electrical Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

A new parametrization, called implementation, of a matrix fraction description in terms of certain polynomial operators is proposed. Its parameters appear in a straightforward way in the implementation of the given input-output behavior. The effective computational sensitivity with respect to these implementation parameters is easily analyzed, and in fact optimized.

1 Introduction

This paper extends the results of the polynomial operator representation, introduced in [1] to the matrix fraction descriptions [2]. The latter give a useful description of multi-channel processors, which lead to state space implementations. The filter-banks are of wide interest in multi-scale signal processing, among other applications. It is well known that state space implementations for a given system are not unique. However, in a finite precision computational environment, all theoretically equivalent state space realizations may have very different performance. We have addressed these problems from a differential geometric point of view, and the optimal realizations were characterized in terms of geometric quantities [3]. For a uniform metric we were led to the so called class of generalized balanced realizations as optimal. This is consistent with the work of Mullie and Roberts [4], who used different probabilistic arguments. This differential geometric framework, although very general and at the same time very concise has the serious disadvantage that it is based on infinitesimal analysis. This language is not quite right for discrete valued problems, which are intrinsically of a combinatorial nature. In particular, there is no reason to assume that a coefficient equal to 0 or 1 can only be imprecisely implemented digitally. In fact a large class of rational numbers can be implemented exactly, and this is not taken into account with an infinitesimal sensitivity analysis. For this reason there is a need to separate precisely implementable coefficients, and imprecisely implementable ones.

An alternative theory, in the frequency domain, was introduced by us [1] which can take such a dichotomy between exactly implementable and imprecisely implementable numbers into account. It extends in turn

\[^1\]Presently at Georgia Tech Lorraine, F-57070 Metz, France
\[^2\]Université Catholique de Louvain, B-1348, Louvain-la-Neuve, Belgium

an idea by Middleton and Goodwin [5] of implementing systems by using a representation in the \(\delta\)-operator rather than the shift operator. They showed superior numerical properties. The proposed generalization expresses both the numerator and denominator polynomial for the siso-case, not in terms of one new indeterminate (the \(\delta\)), but in terms of \(n\) optimally chosen polynomials.

This paper will extend these ideas further to the multi-variate or multi-channel processors, by deriving the generalized implementations from the matrix fraction description of the system (or filter) to be implemented. For clarity, we describe below the methodology for the single input filter bank, i.e. a single-input multi-output system, but in a framework that makes its multi-variate extension transparent.

2 Polynomial Operator Representations

For discrete time systems, the transfer matrix or matrix fraction description (MFD) is usually specified in terms of the delay operator \(z^{-1}\), from which implementations directly follow in the so-called canonical form (sometimes referred to as the "direct" forms). We shall have a closer look at such realization schemes, and generalize these concepts.

Consider the strictly proper \(p \times m\) right matrix fraction description [2]

\[
H(z) = N(z)D(z)^{-1} = [N_{0}\Psi(z)][D_{0}S_{0}(z) + D_{0}\Psi(z)]^{-1}
\]

where \(S_{0}(z) = \text{Diag}[s_{0}^{i}, \ldots, s_{0}^{m}]\), and \(\Psi(z) = \text{Blockdiag}([s_{i}^{n}, \ldots, s_{i}^{n}], i = 1, \ldots, m)\), and \(D_{0}\), \(D_{0}\) and \(N_{0}\) are coefficient matrices of size respectively \(m \times m, m \times n\), and \(p \times n\). The \(n_{i}\) are the column degrees
of the MFD, and $\sum_{i=1}^{n} n_i = n$. Moreover, we shall assume that the denominator matrix $D(x)$ is in column-reduced form. A preliminary transformation to column reduced form can always be performed. The higher order coefficient matrix $Z_m$ is then invertible. Without loss of generality, we shall assume that this coefficient matrix is the identity matrix (which can be achieved by an invertible transformation of the inputs). In this representation, the polynomial matrix $\Psi(z)$ is associated with the partial states. The states are represented by the operators $x^i$ acting on these partial states. We shall now consider a polynomial representation of the transfer MFD, by selecting a different set of representations of polynomials. So let $T$ be the matrix transforming the polynomial matrix $[S_0(x), \Psi(x)]$ to $[Z_0(x), \tilde{Z}(x)]$, where we require for certain reasons of 'causality' that the column degrees of $Z_0(x)$ are the same as those of $S_0(x)$, and that the higher order coefficient matrix of $Z_0(x)$ is also the identity. Let also the column degrees of $\tilde{Z}(x)$ be strictly less than the corresponding column degrees of $Z_0(x)$. With these constraints, $T$ is of the form

$$T = \begin{bmatrix} I_m & \tau \\ T \end{bmatrix}$$

so that the higher order coefficient matrices of the numerator and the denominator polynomial matrices are respectively $0_{nxm}$ and $I_m$, and hence are redundant. The transformation matrix is fully parametrized by the $m \times n$-vector $\tau$ and the $n \times n$ matrix $T$. It is not necessary that the submatrix $T$, and hence $T$ be invertible, as was assumed in [6].

Example: The scalar transfer function (a trivial MFD)

$$H(x) = \left\{ \begin{array}{c} x + 1 \{ x^2 + 2x + 3 \}^{-1} \\ (0, 1, 1)(x^2 + 2, z, 1) \end{array} \right\}$$

is also represented by $\{ (0, 1, 0)(x^2 + 2z + 3, z + 1, 0) \}$: $\{ (1, 0, 0)(x^2 + 2x + 3, z + 1, 0) \}^{-1}$ corresponding to a singular transformation

$$T = \begin{bmatrix} 1 & 2 & 3 \\ 0 & 1 & 1 \\ 0 & 0 & 0 \end{bmatrix}.$$ 

At this point we should remark that with each transfer matrix, several right MFDs may correspond, all satisfying the assumptions. Their numerator and denominator matrices must then be related by a unimodular transformation on the right. Should one want this, a unique MFD can be obtained for each transfer matrix by instance in Popov- or polynomial echelon form [2]. The generalized right MFD is thus specified in terms of coefficients (acting as coordinates) and polynomials (acting as frames). Let $H_0$ be the class of these representations. A particular element will be denoted by the quadruple $H_0((P, M), (Z_0(x), \tilde{Z}(x))]$ with $(P, M) \in \mathbb{R}^{mxn} \times \mathbb{R}^{nxm}$ and $(Z_0(x), \tilde{Z}(x)]$ as specified.

A multi-channel filter bank is a system with one input and $p > 1$ outputs. Let $n$ be its McMillan degree. Its right fraction description involves only a scalar denominator, i.e. $N(x)\bar{d}(x)^{-1}$, where the degree of $\bar{d}(x)$ is $n$. Denoting the $n$ dimensional coefficient-vector of the denominator and the $p \times n$ numerator coefficient matrix in generalized polynomial operator form respectively by $\alpha$ and $\beta$, we get for a particular filter bank representation:

$$H_0((\alpha, \beta), (z_0(x), \tilde{Z}(x))] \in \mathbb{R}^n \times \mathbb{R}^{nxp} \times M_{\mathbb{R}^n}(x) \times (\mathbb{R}_{n-1}[z])^n$$

where $M_{\mathbb{R}^n}(x)$ is the set of monic polynomials of order $n$ in $x$. For the above scalar example, the two generalized polynomial representations are in the standard form:

$$H_0((1, 2, 3), (1, 1)], (x^2; z, 1)], \text{ and in the generalized right fraction description:}$$

$$H_0((0, 1, 0), (0, 1)], (x^2 + 2x + 3; x + 1, 0)].$$

Clearly, the transformation to a rational vector function in $z$ is in terms of the coordinates and the polynomial frame simply

$$[\beta' \tilde{Z}(x)]z_0(x) + \alpha' \tilde{Z}(x)]^{-1}$$

In the terminology of [3], this map $H$

$$H : H_0 = \mathbb{R}^{n+p} \times M_{\mathbb{R}^n}(x) \times (\mathbb{R}_{n-1}[z])^n \rightarrow (\text{Rat}_{n}[z])^p$$

where $\text{Rat}_{n}[z]$ is the set of the strictly proper rational vector functions of $z$ of degree $n$, defines an observable. Alternatively, rather than specifying the polynomials $z_0(x)$ and $\tilde{Z}(x)$, one could also specify the transformation $T$, i.e. the matrix $T$, together with the vector $\tau$, which generate $[z_0(x), \tilde{Z}(x)]^T$ from the "standard" $[z, \Psi(z)]^T$. This class of parametrizations will be denoted by $H_0((\alpha, \beta), (\tau, T))$.

An equivalence, for which the observable $H(x)$ (the filter bank right MFD) is invariant, can be defined in the class $H_0$ of polynomial operator representations.

Definition: Equivalence in $H_0$:

$$H_0((\alpha_1, \beta_1), (z_1, Z_1)] \sim H_0((\alpha_2, \beta_2), (z_2, Z_2)]$$

iff there exist vectors $\tau_i$ and matrices $T_i$, for $i = 1, 2$ such that

$$z_{\alpha_i}(x) = x^n + \tau_i^T \Psi(x)$$

$$z_i(x) = T_i \Psi(x)$$

$$T_i^\prime \alpha_i + \tau_i = T_i \Psi_2 + \tau_2$$

$$T_i^\prime \beta_i = T_i \beta_2$$

It is clear that the common vector $d = T_d \alpha + \tau$ and matrix $N = T_d \beta \Psi$ uniquely specify the right MFD $H(z) = N(z) d(z)^{-1}$. In the next section, the key role of $\alpha$, $\beta$, $z_{\alpha}$, and $Z$ will be demonstrated in the implementation of the i-o functions represented by $H_0$. 


3 Generalized Reachable Implementations

Consider the polynomial operator parametrization of the filter bank \( H_0(\alpha, \beta, \{z_n, Z\}) \) as a right fraction description [2]. The relation between input and outputs is then

\[
y_i = [\beta' Z(x)]z_0(x) + \alpha' Z(x)u(t)
\]

(8)

It will be shown that this leads to a natural implementation, that is akin to the construction of the canonical realization via Kelvin's principle. First separate Eq. (8) into

\[
y_i = \beta' Z(x)\xi + [z_0(x) + \alpha' Z(x)^{-1}u(t)]\xi
\]

(9)

In turn (10) leads to

\[
z_0(x)\xi = u_1 - \alpha_1 z_1(x)\xi - \ldots - \alpha_n z_n(x)\xi
\]

(11)

The polynomial \( z_n(x) \) plays a privileged role, and is referred to as the core-polynomial. Its has degree \( n \), which is higher than the degree of the polynomials \( z_1(x), \ldots, z_n(x) \) by assumption. Consequently, with this core-polynomial, a core-realization of

\[
z_0(x)\xi = p_i
\]

(12)

is derived in a feed-out companion form. Specifically, this is obtained by writing (12) as

\[
z^n\xi = p_i - (r_n z^{n-1}\xi + \ldots + r_0\xi)
\]

(13)

This "core of the core" is implemented as the usual chain of \( n \) delay-elements. The chain output, \( \xi_i \), is the partial state, and the quantity \( x_i = [z_i^0, \ldots, z_i^1] \xi_i \) is the core state. It follows from (13) that the core-realization of (11) consists of a weighted feedback of the core-states \( x_i \) superposed on \( p_i \), and fed to the input of the core-core (the chain of delay-elements).

The state-space representation of this feed-out core implementation with core-polynomial \( z_0(x) \) is then given in reachable canonical form

\[
X_{t+1} = \begin{bmatrix}
-\tau_1 & \ldots & \ldots & -\tau_n \\
1 & 0 & & \\
0 & \ddots & \ddots & \\
0 & \ldots & 1 & 0
\end{bmatrix}
\begin{bmatrix}
x_t \\
0 \\
0 \\
0
\end{bmatrix} + \begin{bmatrix}
1 \\
0 \\
0 \\
0
\end{bmatrix} p_t
\]

(14)

\[
\xi_t = [0, \ldots, 0, 1]x_t
\]

(15)

Next, \( p_i \) is assembled by identifying the right hand sides of (11) and (12)

\[
p_i = u_i - \alpha_1 z_1(x)\xi - \ldots - \alpha_n z_n(x)\xi
\]

(16)

Defining the implementation-state \( x_i \) by

\[
\begin{bmatrix}
z_1(x) \\
z_2(x) \\
z_{n-1}(x) \\
z_n(x)
\end{bmatrix} = T
\]

with

\[
x = T_{X_t} = x_i
\]

(17)

yields the core-input \( p_t \) as a linear combination of the input and the implementation states

\[
p_t = u_i - \alpha' x_i
\]

(18)

Finally, the filter bank outputs follow from (15) and (17)

\[
y_t = \beta' Z(x)\xi_t = \beta' x_i
\]

(19)

The full implementation of the right fraction description \( H_0(\alpha, \beta, \{z_n, Z\}) \) is then given by (14), (16), (17), and (19), which are summarized below, using the equivalent parametrization \( H_0[\{\alpha, \beta, \{z_n, Z\}\}] \)

Core Update \( \chi_{t+1} = A_{\xi}(r)\chi_t + b_{p}\delta \)

State Transformation \( z_t = T\chi_t \)

Input Assembly \( p_t = u_t - \alpha' x_t \)

Read Out \( y_t = \beta' x_t \)

where \( \chi_t \) is the core state, \( x_t \) the realization state, \( p_t \) the core input, \( A_{\xi}(r) \) a top companion form matrix with characteristic polynomial \( z^n + r'\Psi(x) \), and \( b_{\xi} = [1, 0, \ldots, 0]^T \).

All elements of the parametrization \( \alpha, \beta, r \) and \( T \) are found as coefficients in the implementation. We further exercise care in the use of the term implementation, in order to distinguish it from the usual notion of a realization, which is specified by a triple, rather than a quadruple.

Discussion: i) The implementation requires \( p + 2 \) inner products: \( < \alpha, x > \), \( < \beta, x > \), and \( < r, x > \), together with the matrix vector product \( T\chi \). The latter should have as many "fixed" (i.e. zeros or ones) elements as possible, since these do not contribute any perturbation terms. ii) The implementation and its states can be interpreted as follows: The core consists of a bank of \( n \) filters, with transfer functions \( z_1, z_1^2, \ldots, z_n^{-1} \). All these are driven by \( p \). Their outputs are the implementation-states. Feedback of the implementation-states is performed around the core with gains \( \alpha \), while the read-out of the implementation-states is accomplished with the multi-channel linear combiner \( \beta \). The actual number of independent filters in the core filter bank equals the rank of \( T \).

4 Implementation Sensitivity

The reason for distinguishing between realization and
implementation is that the latter allows a more direct analysis of the sensitivity, as it involves directly the parameter values that are implemented, and not a function thereof. Conversely, changes in these parameters show up directly in the associated observable (the transfer function vector $H$).

Indeed, consider the set $H_1$, and the associated observable $H$, the multi-channel MFD (or transfer function vector). Since $H_1$ is parametrized by $\alpha, \beta, \tau$, and $T$, a frequency independent sensitivity measure is obtained by the sum of the $H_2$-norms of the gradients of the transfer matrix. For a parameter $\rho$ this $H_2$ norm gives the measure [6]

$$M_\rho = \frac{1}{2\pi j} \text{Tr} \left\{ \int \frac{\partial H(z)}{\partial \rho} \frac{\partial H(z)}{\partial \rho}^H dz \right\} \quad (20)$$

The contour integral is over the unit circle $|z| = 1$, and the superscript $^H$ denotes the Hermitian conjugate. In fact, this $M_\rho$ is the sum of the sensitivities in each channel, with scalar transfer function $H_i(z) = N_i(z)/\rho(z)$. Noting that on the unit circle, $z^* = z^{-1}$, and that $\Psi(z)'\Psi(z^{-1}) = n$, one finds for the sensitivity in each channel (the detailed derivations are in [1])

$$M^{(i)} = \text{Tr} W^{(i)}T' + \text{Tr} \left[ c_k^{(i)} \beta \beta - 2c_k^{(i)} \beta \alpha' + c_k^{(i)} \alpha \alpha' \right] + \text{Tr} W_0^{(i)} \quad (21)$$

where $W^{(i)}$ and $W_0^{(i)}$ are certain gramian matrices computable from the transfer function $H_i(z)$. The constants $c_k^{(i)}$ are also invariant to the particular implementation. The last sensitivity term $M^{(i)}$ is implementation independent, and therefore just adds a constant term to the overall sensitivity.

The optimal implementation is obtained by constant minimization of the quantity $M = \sum_{i=1}^{\infty} M^{(i)}$ which can be rewritten in the form $M = \text{Tr} (TWT') + [\beta, \alpha] C \begin{bmatrix} \beta' \\ \alpha' \end{bmatrix} + k$ with the constraints

$$d_k = T' \alpha + \tau \quad (22)$$

$$N_k = T' \beta \quad (23)$$

where $W$ and $C$ are positive definite matrices, and $k$ a positive constant, all depending only on the i-o characteristics of the filter bank. This formulation avoids the necessity of the existence of $T^{-1}$. The solution (e.g. based on SVD) to this standard constraint matrix minimization problem gives the optimal implementation parameters, which are for the size-case (so that $N$ is now an $n$-vector). The $c_i$ are the elements of $C$.

$$\beta' = \left[ \frac{N'W'N}{L} \right]^\dagger \epsilon_1 \quad (24)$$

$$\alpha' = \frac{c_2 \beta'}{c_3} \quad (25)$$

$$\tau' = \frac{d - c_1 N}{c_3} \quad (26)$$

$$T' = \left[ \frac{N'W'N}{L} \right]^\dagger N' \quad (27)$$

$$\Gamma = \frac{c_1 c_2 - c_3^2}{c_3} \quad (28)$$

It follows from this that the extremal implementation corresponds to a controller canonical realization of the scalar transfer function $\frac{C}{R}$, with its output fed back over a gain $\alpha_1$ to the input. Moreover this output is then scaled by $\beta_1$ to form the system output. It is easily checked that the optimal filter bank implementation does not coincide with the optimal implementation of each individual channel transfer function. In fact, it is well known that the latter is not even a minimal realization in general.

References


Computing the cardinality of root signal sets of morphological filters

Qiaofei Wang, Moncef Gabbouj, and Yrjö Neuvo
Signal Processing Laboratory
Tampere University of Technology
Tampere, Finland

Abstract: Morphological opening, closing, opening-closing, and clos-opening are idempotent operations, i.e., they reduce an input signal to a root in a single pass. In this paper, we apply the theory of state description for the root signal set of these morphological filters. Some properties of root signals are discussed. A complete system of equations has been established which allows us to calculate the number of root signals for morphological filters by structuring element of width \( k \) and signals of length \( L \) with \( m \) quantization levels. For binary signals, simplified recursive equations have been obtained.

1. INTRODUCTION

The basic morphological operations are dilation, erosion, opening, and closing [1,2]. Morphological opening and closing are important algorithms in signal processing. They eliminate specific signal structures smaller than some structuring element without distorting the global features of the signal. Of particular significance is the fact that morphological opening and closing possess idempotency property, i.e., their reappplication does not cause further changes to the previously filtered result. In other words, one pass of opening or closing yields a root signal which is invariant to further passes of the same filter. Moreover, the operations employing iteratively applied openings and closings are also idempotent. They are open-closing (opening followed by closing by the same structuring element) and clos-opening (closing followed by opening by the same structuring element). They in general behave similarly to the median filter. The root signals of median filters have been extensively studied [3-5]. It has been proved [2] that the root of the median is bounded below by the corresponding opening-closing and above by the corresponding clos-opening; and the roots of open-closing and clos-opening are roots of the median filter.

A state model describing the binary root signal set of median filters was proposed by Arce and Gallagher [5]. The states in a median root propagate following a tree structure as the signal length increases. Summing over all states at certain stage gives the number of roots associated with the filter for the specified window width and the signal length. By developing a state model in the space of multi-level signals, the authors of [5] have established a complete system of equations for finding the number of root signals associated with any number of quantization levels and any window width of a median filter.

For morphological opening and closing, however, the set of the states is not complete, which means that the set of the states can not generate itself in the following stage when the signal length increases. The problem has been solved for binary signals in [6] by taking the root number as a special state and recursive equations have been obtained for finding the number of binary roots of opening, closing, opening-closing, and clos-opening. For multilevel signals, we will solve this problem in this paper by introducing the concept of virtual states.

2. SOME PROPERTIES OF 1D ROOT SIGNALS

2.1. Definitions

In this paper, morphological operations are considered as function-processing filters. In the function-processing system, it is convenient to define the signal length which is a key parameter in the analysis of root signals. Definitions are given as follows [1],

\[
\begin{align*}
\text{DILATION:} & \quad (f \oplus \hat{K})(x) = \sup\{f(y) \mid y \in K_x\} \\
\text{EROSION:} & \quad (f \ominus \hat{K})(x) = \inf\{f(y) \mid y \in K_x\} \\
\text{OPENING:} & \quad f \circ \hat{K} = (f \ominus \hat{K}) \oplus \hat{K} \\
\text{CLOSING:} & \quad f \bullet \hat{K} = (f \oplus \hat{K}) \ominus \hat{K}
\end{align*}
\]

(1)

where \( \hat{K} = \{-z : z \in K\} \) denotes the symmetric set of \( K \) with respect to the origin, and \( K_x = \{z + \sigma : z \in K\} \) denotes the translation of \( K \) by \( \sigma \). For sampled signals, the structuring element \( K \) is a discrete and finite set which is viewed as a moving window. Hence, erosion (dilation) of a sampled function by a finite set \( K \) is equal to the moving local minimum (maximum).

2.2. Threshold Decomposition Property and PBF Expressions

Let \( \vec{X} = [X_1, X_2, \ldots, X_d]^T \) (where \( T \) denotes transpose) be a multilevel, nonnegative signal vector. Without loss of generality, we require that \( X_i \in \{0, 1, \ldots, m- \)}
1). This signal vector can be decomposed into $m - 1$ binary vectors $x'_t$, $t = 1, 2, \ldots, m - 1$, by thresholding. This thresholding operation is $T^t$, so that

$$x'_t = T^t(x_t) = \begin{cases} 1 & \text{if } x_t \geq t; \\ 0 & \text{otherwise.} \end{cases}$$

(2)

Note that

$$x = \sum_{t=1}^{m-1} T^t(x_t) = \sum_{t=1}^{m-1} x'_t.$$  

(3)

**Definition 1:** An ordered set of sequences $X_1, X_2, \ldots, X_k$, is said to obey the stacking property if

$$X_1 \geq X_2 \geq \cdots \geq X_k.$$  

(4)

**Definition 2:** A binary function $g$ is said to possess the stacking property [7] if and only if

$$g(x) \geq g(y) \text{ whenever } x \geq y.$$  

(5)

It has been shown [7] that a necessary and sufficient condition for a binary function to possess the stacking property is that it is a positive Boolean function (PBF).

**Definition 3:** The stack filter $S_g$ is defined by a binary filter $g(x)$ as follows,

$$S_g(x) = \sum_{t=1}^{m-1} S_t(y'_t) = \sum_{t=1}^{m-1} g(y'_t).$$  

(6)

We say that $S_g$ obeys the threshold decomposition [4] or the weak superposition property. On the other hand, a stack filter $S_g$ is determined and, hence, represented by the PBF $g$.

Since morphological filters by flat structuring elements are stack filters, we can find their PBF expressions. Let $g_0$, $g_{oc}$, and $g_{oc0}$ denote the PBF of opening, closing, open-closing, and close-opening by structuring element $K$, respectively. Then we have [8],

$$
\begin{align*}
g_0(x) &= \bigvee_{a \in K} \bigwedge_{x \in K} x \\
g_0(x) &= \bigwedge_{a \in K} \bigvee_{x \in K} x \\
geoc(x) &= \bigwedge_{a \in K} \left( \bigvee_{b \in K} \left( \bigwedge_{x \in K} x \right) \right) \\
geoc0(x) &= \bigwedge_{a \in K} \left( \bigvee_{b \in K} \left( \bigwedge_{x \in K} x \right) \right)
\end{align*}
$$

(7)

It is important to note that $g_0$ is a dual function of $g_0$ and $g_{oc}$ is a dual function of $g_{oc0}$.

Example: Consider morphological opening by structuring element

$$K = (-1, 0, 1).$$  

(8)

Then the corresponding Boolean function is,

$$g_0(x) = (x_{-1} \land x_0 \land x_1) \lor (x_{-1} \land x_0 \land x_1) \lor (x_0 \land x_1 \land x_1).$$  

(9)

2.3. 1D Root Structures

In 1D case, assume the structuring element $K$ is a convex set of length $k$. Some important 1D structures are defined as follows.

**Constant Neighborhood:** It consists of at least $k$ consecutive identically valued points.

**Edge:** A monotonic region which contains no constant neighborhood. If the region is monotonic non-decreasing, it is called a positive edge. If the region is monotonic non-increasing, it is called a negative edge.

Based on the results found in [2, 6, 9], the following properties can be proved using the threshold decomposition property [10].

**Property 1:** A root of opening (closing) is composed of constant neighborhoods and edges. A positive (negative) edge and a following negative (positive) edge must be separated by a constant neighborhood.

**Property 2:** If $R$ is a root signal of opening, then $E = (m - 1)I - R$ is a root of closing by the same structuring element. Hence, the numbers of the roots of opening and closing by the same structuring element are equal to each other for a given signal length.

**Property 3:** A root of open-closing is a root of clos-opening by the same structuring element and vice versa. Hence, the number of the roots of open-closing is equal to the number of the root of clos-opening for a given signal length.

**Property 4:** A root of open-closing and clos-opening is composed of constant neighborhoods and edges. Two edges must be separated by a constant neighborhood.

**Property 5:** A root of opening, closing, open-closing, and close-opening by a particular structuring element is a root of opening, closing, open-closing, and close-opening by a smaller structuring element, respectively.

3. OPENING AND CLOSING

Suppose the last $k$ digits of the signal are $a_1, a_2, \ldots, a_k$ and the $k$ digits preceding the last $k - j$ ($1 \leq j \leq k - 1$) digits are $b_1, b_2, \ldots, b_k$. The states are defined as follows.

Static state $S_0(i)$: When the values of the last $k$ digits of the signal are identical and equal to $i$, i.e., $a_1 = a_2 = \cdots = a_k = i$, for $0 \leq i \leq m - 1$, we say that it is in a static state.

Downward transition state $D_k(j, i)$: If and only if $a_{k-j} > a_{k+1-j} = a_{k+2-j} = \cdots = a_k = i$, for $1 \leq j \leq k - 1$ and $0 \leq i \leq m - 1$, we say that it is in a downward state.

Upward transition state $U_k(j, i)$: If and only if $b_{k+1-j} = b_{k+2-j} = \cdots = b_k = m - 1 - i$, for $1 \leq j \leq k - 1$, $0 \leq i \leq m - 1$, and $1 \leq Q \leq L - k + j$, we say that it is in an upward transition state.

Since the upward transition state $U_k(j, i)$ does not appear at the end of a root signal, we call it a virtual state. The recursive relationship among the states can
be obtained by inspection.

\[
\begin{align*}
S_{L+1}(i) &= S_L(i) + D_L(k-1, i) + \\ 
&\sum_{r=1}^{m-1} (S_{L+1-k}(m-1-r) + \\ 
&\sum_{l=1}^{k-1} (U_{L-k-l}(l, r) + D_{L-k-l}(l, m-1-r))) \\
D_{L+1}(1, i) &= \sum_{r=1}^{m-1} (S_L(r) + D_L(l, r)) \\
D_{L+1}(j, i) &= D_L(j-1, i), \text{ for } 1 < j < k \\
U_{L-k-1}(1, i) &= \sum_{r=1}^{m-1} (S_{L-k}(m-1-r) + \\ 
&\sum_{l=1}^{k-1} (U_{L-k-l}(l, r) + D_{L-k-l}(l, m-1-r))) \\
U_{L-k-1}(j, i) &= U_{L-k}(j-1, i), \text{ for } 1 < j < k
\end{align*}
\]

The total number of static states and downward states is equal to the number of roots.

\[
R^m(L+1) = \sum_{i=1}^{m-1} (S_{L+1}(i) + D_{L+1}(l, i))
\]

Some numerical results are given in Fig. 1.

For binary signals \((m = 2)\), we can simplify (10) and (11) to get a difference equation for \(R^2(L+1)\). In binary case, we note that \(D_L(j, 1) = 0\), for \(1 \leq j \leq k - 1\). By (10) and (11),

\[
\begin{align*}
U_L(k-1, 0) &= S_{L-k}(0) + D_{L-k}(k-1, 0) + \\ 
&\sum_{l=1}^{k-1} D_{L-k-l}(l-1, 0) + S_{L-k}(1) \\
&= R(L-k)
\end{align*}
\]

where we note that \(D_{L-k}(k-1, 0) + \sum_{l=1}^{k-1} D_{L-k-l}(l-1, 0) = \sum_{l=1}^{k-1} D_{L-k-l}(l, 0)\). By (11),

\[
\begin{align*}
R^2(L+1) &= S_{L+1}(0) + S_{L+1}(1) + \sum_{l=1}^{m-1} D_{L+1}(l, 0) \\
&= S_L(0) + D_L(k-1, 0) + S_L(1) + \\ 
&U_L(k-1, 0) + \sum_{l=1}^{k-1} D_{L-k-l}(l-1, 0) + S_L(1) \\
&= R^2(L) + R^2(L-k) + S_L(1)
\end{align*}
\]

Thus we can calculate the difference,

\[
\begin{align*}
R^2(L+1) - R^2(L) &= R^2(L) + 2R^2(L-k) + S_L(1) - \\ 
&(R^2(L-1) + R^2(L-1-k) + S_{L-1}(1)) \\
&= R^2(L) + 2R^2(L-k) - R^2(L-1)
\end{align*}
\]

In the above equation, we have used the relation \(S_{L}(1) - S_{L-1}(1) = U_L(1, 0) = R^2(L-1-k)\). Adding \(R^2(L)\) to both sides of (14), we have,

\[
R^2(L+1) = 2R^2(L) + R^2(L-k) - R^2(L-1)
\]

This equation checks with the result found in [6].

4. OPEN-CLOSING AND CLOS-OPENING

It has been proved that a 1D root of open-closing or close-opening is also the root of the corresponding median filter [2]. However, a root of median filter is not necessarily a root of open-closing or close-opening, if we use first and last sample carrying on appending strategy for median filters. Therefore, the positive or negative impulse structures at ends will be removed by morphological filters but preserved by median filters because of the appending strategy. For example, "101101001" is a root of window width three median filter, but it is not a root of open-closing by structuring element of length two.

Suppose the last \(k\) digits of the signal are \(a_1, a_2, \cdots, a_k\) and the \(k\) digits preceding the last \(k-j\) \((1 \leq j \leq k-1)\) digits are \(b_1, b_2, \cdots, b_k\). The states are defined as follows.

Static state \(S_L(i)\): When the values of the last \(k\) digits of the signal are identical and equal to \(i\), i.e., \(a_1 = a_2 = \cdots = a_k = i\), for \(0 \leq i \leq m-1\), we say that it is a static state.

Downward transition state \(D_0(j, i)\): If and only if \(b_{k-j} > b_{k-j-1} = \cdots = b_{k-1} = i\), for \(1 \leq j \leq k-1\), \(0 \leq i \leq m-1\), and \(1 \leq Q \leq L - k + j\), we say that it is in a downward state.

Upward transition state \(U_L(j, i)\): If and only if \(b_{k-j} < b_{k-j+1} = \cdots = b_k = m-1-i\), for \(1 \leq j \leq k-1\), \(0 \leq i \leq m-1\), and \(1 \leq Q \leq L - k + j\), we say that it is in an upward state.

The upward and downward transition states are not allowed to appear at the end of the root signal. Both of them are virtual states.

Since the root structures of open-closing and close-opening are same as those of median filters (except the end points), the mechanism of deriving the recursive relationships among the states are similar to that found in the state model of median filters [3].
the root signal, the number of roots is equal to the sum of them at the given stage.

\[ R^m(L + 1) = \sum_{i=0}^{m-1} S_{L+i}(i) \]  

(17)

Some numerical results are given in Fig. 2.

For binary signals \((m = 2)\), we can simplify (16) and (17) to a single difference equation for \(R^2(L + 1)\). In binary case, we note that \(D_L(j,1) = U_L(j,1) = 0\), for \(1 \leq j \leq k\). By (16) and (17),

\[ R^2(L + 1) = S_{L+1}(0) + S_{L+1}(1) = S_L(0) + S_{L+1-k}(1) + S_L(1) + S_{L+1-k}(0) = R^2(L) + R^2(L + 1 - k) \]  

(18)

This equation checks with the result found in [6].

System symmetry:

We note that \(S_L(i) = S_L(m-1-i)\) for \(i = 0, \ldots, q\), where \(q\) is equal to \((m-2)/2\) for \(m\) even and \((m-1)/2\) for \(m\) odd. Therefore, the number of downward states \(D_L(j,i)\) and upward states \(U_L(j,i)\) are equal for any specified \(j\) and \(i\). The symmetries can be exploited to reduce the amount of computation.

5. CONCLUSIONS

In this paper, we have constructed the 1D root structures of morphological filters and obtained some root properties. The state models for morphological filters have been developed and a complete system of equations has been established by which we can calculate the number of roots of opening, closing, open-closing, and close-opening for a given signal length.

References


A NEW MOMENT-BASED METHOD FOR LOW NOISE ORDERING OF CASCADE FORM FIR DIGITAL FILTER

1. Anil KHARE
2. T. YOSHIKAWA

Nagaoka University of Technology
Dept. of Electrical Engineering
Kamitomioka 1603-1,
Nagaoka-shi, Niigata-ken 940-21, JAPAN

The effect of the finite word length on the moment of the impulse response of a digital filter in investigated. It is shown that the quantization error in the moment values is almost linear with a slope of 3dB/bit and is well related to the output roundoff noise. An algorithm is also proposed to order the cascade sections of a FIR filter with low output roundoff noise.

1. INTRODUCTION

Realization of digital filters using finite word length leads to deviation in the filter characteristics from the designed values[1-6]. In signal processing, moment is known as one of the effective methods to express the various properties of the continuous and discrete time signals in time domain[8,7,9]. In this paper, we investigate the effect of the finite word length on the moment of the impulse response of a digital filter. It is shown that the quantization error in the moment values is almost linear with a slope of 3dB/bit and is well related to the output roundoff noise. An algorithm is also proposed to order the cascade sections of an FIR filter with low output roundoff noise using the deviation in moment criterion.

2. THE MOMENTS OF THE IMPULSE RESPONSE

2.1 Basic Definitions

For a general discrete signal and finite duration signal of duration N, x(n), the respective moments of order r are defined in time domain as

\[ m_r = \sum_{n=-\infty}^{\infty} (n)^r x(n) \]  

(1)

\[ m_r = \sum_{n=0}^{N-1} (n)^r x(n) \]  

(2)

If \( X(e^{j\omega}) \) represents the Fourier transform of \( x(n) \), r-th order moment is given by the r-th order derivative of \( X(e^{j\omega}) \) with respect to \( \omega \), evaluated at \( \omega = 0 \).

\[ m_r = \frac{j^r}{\Theta} \left. \frac{\Theta^{r} X(e^{j\omega})}{\Theta \omega^r} \right|_{\omega=0} \]  

(3)

2.2 Moments of direct form and cascade form FIR filter

Let \( h(n) \) be the impulse response of an FIR digital filter of order N-1 and let \( H(z) \) denote its system function. If this filter is assumed to be realized in direct form, the moments of the impulse response are given by the following expression.

\[ m_r = \sum_{n=0}^{N-1} (n)^r h(n) \]  

(4)

For cascade form of realization, the system function \( H(z) \) can be expressed as the product of second order sections as follows:

\[ H(z) = \prod_{i=1}^{N_S} H_i(z) \]  

(5)

Here \( N_S \) is the number of second order sections[8].

If we express \( H_i(z) \) as
\[ H_i(z) = K (1-a z^{-1})(1-a^* z^{-1}) \]
\[ h_i(n) = K(n + a^n) \]
\[ h_i(2) = Ka^* \]

Obviously, the length of the impulse response for each second order section is 3. Using the above values for the impulse response of the individual sections, we can easily obtain the moments of the second order section as
\[ m_r(i) = \sum_{n=0}^{2} (n)^r h_i(n) \]

Here \( m_r(i) \) is the \( r \)-th order moment of the \( i \)-th section of the filter realized in cascade form. Even if the \( i \)-th order section is not a second-order section, it is possible to consider moments up to order two. We have observed that the results obtained even by considering moments only up to order two do not differ greatly when all the moments up to the required order are considered.

2.3 Coefficient quantization and deviation in moment values

If we assume the finite precision arithmetic for the filter coefficients, obviously, the moment values given by eq.(4) and eq.(10) will deviate from their ideal values. We define the deviation in moment values \( \Delta m_r \) as follows.
\[ \Delta m_r = \left| \frac{Q[m_r]}{m_r} \right| \]

Here \( Q[m_r] \) is the moment of order \( r \) when the coefficients are assumed to be quantized. \( \left| \frac{Q[m_r]}{m_r} \right| \) indicates the absolute value. Fig.1 shows the deviation in moment for different values of the coefficient word length. We note from this figure that the deviation in the moment values is almost linear with a slope of 3dB/bit. Fig.2 shows (figures on right) the deviation in moment vs. output roundoff noise variance for the direct-form realization. From the above relationships, the deviation in moment is seen to be a good indication of the deterioration in the filter characteristics in time-domain for finite word length of the coefficients.

For the \( i \)-th second order section in cascade form of realization, we define \( \Delta m(i) \) as the overall indication of the change of characteristics in the following way.
\[ \Delta m(i) = \sum_{r=0}^{2} \Delta m_r(i) \]

From the frequency domain definition of the moment, we see that \( \Delta m(i) \) provides a measure for the change in the sum of zeroth, first and second order derivatives of the Fourier transform of each section, evaluated at zero frequency. Since \( \Delta m(i) \) has a simple relationship to the impulse response coefficients in time domain and \( \Delta m(i) \) is almost linearly related to the roundoff noise variance for direct-form realization, we conclude that \( \Delta m(i) \) is also a good time-domain measure of the roundoff noise for the \( i \)-th section.

\[ \text{deviation in moment [dB]} \]

\[ \text{word length [bit]} \]

Fig.1 Deviation in moment vs. word length

\[ \text{deviation in moment [dB]} \]

\[ \text{noise variance [dB]} \]

Fig.2 Deviation in moment vs. noise variance

3. NOISE FIGURE CRITERION

In communication theory, we have the concept of noise figure which relates to the internally-generated noise of a two-port analog receiver[10]. If the spectral densities are uniform over the frequency range of interest, the average noise figure \( F \) is defined as
\[ F = \frac{(S/N)_i}{(S/N)_o} \]  

where \((S/N)_i\) is the ratio of total input available signal and noise power and \((S/N)_o\) is the ratio of total output available signal and noise power.

For a cascade of several stages of gain \(g_a\)'s, the overall noise figure of the cascade structure is given by

\[ F = F_1 + \frac{F_2-1}{g_a_1} + \frac{F_3-1}{g_a_1g_a_2} + \ldots \]  

For ideal systems with no internally-generated noise, \(F\) becomes equal to unity or 0 dB. The above criterion in terms of the equivalent noise temperature leads to the ordering which has the lowest equivalent temperature section in the front-end. While this concept of noise figure cannot be completely adapted to the cascade of digital filters, there are certain noteworthy similarities between the two systems. We have seen that each second order section of the filter realized in cascade form with fixed-point finite precision arithmetic, there exists a roundoff noise due to rounding after multiplication by the filter's coefficients. This noise is totally internal to the cascade section and there are several such sections to realize the filter. Therefore, if we can suitably define an equivalent measure of the noise figure for the sections of the cascade form digital filter, we can use the same concept in ordering these sections according to their "noise figures". We define such a measure as deviation in moment \(\Delta m(i)\) for the \(i\)-th section of the cascade form digital filter.

4. AN ALGORITHM FOR ORDERING THE CASCADE SECTIONS OF FIR FILTER

If we now consider the quantization of the impulse response coefficients, a qualitative measure of the roundoff noise is claimed to be the deviation in moment, \(\Delta m(i)\), defined by Eq. (12). The total output roundoff noise depends upon the method of scaling and the ordering and is very complicated to analyze theoretically. However as is well-known, most orderings have low output roundoff noise and it is possible to find out a good ordering which may not be the "best" yet may have roundoff noise comparable to the lowest possible roundoff noise in the best ordering. The fact that our algorithm does not involve any explicit scaling method which ensures the proper operation of the filter, our method will give a good ordering in terms of the output roundoff noise.

The proposed algorithm is as follows:

1. Given a system function \(H(z)\), decompose it into second order sections \(H_j(z)\).

2. From the knowledge of zeros of the system, calculate the impulse response for each section, using the highest precision available in a computer.

3. Using the impulse response coefficients of step 2 and the definition of moments given by Eq. (10), calculate the moments of each second order section.

4. Repeat step 2 for the desired word length and calculate the quantized impulse response coefficients.

5. From the knowledge of quantized coefficients, calculate the quantized values of moments for all the sections.

6. Calculate \(\Delta m\) defined by Eq. (12) for all the sections.

7. Perform the ordering using the rule that if \(\Delta m(i) < \Delta m(j)\), section \(i\) precedes section \(j\). In the case of equality, choice is arbitrary.

5. EXAMPLE CALCULATION

We consider the same example filter as discussed in reference [5]. Fig. 3 shows the zero locations for this filter. Each section is given a number for identification for the purposes of ordering.

![Fig. 3 Zero locations of example filter](image)

The specifications of this extraripple filter are as follows:

- \(\text{duration} = 13\)
- \(f_p = 0.279\)
- \(f_s = 0.376\)
- \(\delta_p = 0.1\)
- \(\delta_s = 0.01\)

Here \(f_p\) and \(f_s\) are normalized pass and stop band frequencies. \(\delta_p\) and \(\delta_s\) are the deviations of the magnitude of its frequency response from the desired values.
While many different word lengths have been considered to obtain the low-noise ordering according to our method, we discuss the representative results only. Deviation in moment values for each section for 8 bit, 16 bit and 18 bits against 64 bit word length which for all practical cases is equivalent to infinite word length, have been calculated. Using the algorithm described above, we obtain the ordering as shown in Table 1.

**Table 1 Low Noise Ordering Based on Deviation in Moment**

<table>
<thead>
<tr>
<th>word length/ordering</th>
<th>roundoff noise</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 bit (2, 5, 6, 1, 3, 4)</td>
<td>1.5778Q^2</td>
</tr>
<tr>
<td>16 bit (2, 5, 6, 1, 3, 4)</td>
<td>1.5778Q^2</td>
</tr>
<tr>
<td>18 bit (6, 2, 5, 1, 3, 4)</td>
<td>3.0945Q^2</td>
</tr>
</tbody>
</table>

The best ordering as obtained by considering all possible combinations is (2 6 3 4 5 1) which has roundoff noise 1.0983Q^2. The highest roundoff noise value for this filter is 192.2610Q^2 which corresponds to the ordering (6, 4, 5, 3, 1, 2). It is clear from these results that the roundoff noise for the ordering found by the proposed method is very close to the best ordering and is almost one-tenth of the highest roundoff noise. The fact that the proposed method may give different orderings for different word lengths is due to the reason that the deviation in moment is not exactly linear to the changes in word length. So different word lengths may produce different changes in the moment values resulting in slightly different orderings but the values of the output roundoff noise for these orderings are maintained at a low level. As mentioned before, these results are also verified for several different filters and in each case, the resulting ordering from our new algorithm gives very low roundoff noise, within ten percent of the worst-case value of the roundoff noise.

6. CONCLUSIONS

Due to quantization moments of all the individual impulse responses deviate from their ideal values. Though using a rigorous mathematical analysis, this deviation in moments may be shown as related to the roundoff noise. This deviation has been utilized as a qualitative measure for the output roundoff noise and cascade sections have been ordered according to their deviation in moments. We have used the concept of noise figure in analog signal processing and proposed an equivalent argument to find out a low noise ordering based on the deviation in the moment values of each individual section. The resulting ordering then has been compared with the actual analysis and it has been found that the ordering according the new algorithm indeed results in very low roundoff noise. The method has also been verified for several different FIR filters and results for one such filter have been discussed mainly.

**REFERENCES**


MULTIRATE DISCRETE FOURIER TRANSFORM

Yoshiaki Tadokoro, Shin-Ichi Yamakata and Atsuhiko Ishihara
Dept. of Inf.& Comp. Sciences, Toyohashi University of Technology, Toyohashi, 441 Japan.

This paper proposes a new computational algorithm which enables us to compute the discrete Fourier transform (DFT) faster than the fast Fourier transform (FFT). In this algorithm which is referred to as the multirate discrete Fourier transform (MR-DFT), the extreme values of each signal frequency component are sampled and accumulated by alternate additions and subtractions. Since the main operations are additions and subtractions, we can implement this algorithm simultaneously with the sampling operation.

I. INTRODUCTION

The DFT is one of the most important mathematical aids to signal processing [1]. Various algorithms for the DFT have been developed [2], [3], [4]. In many applications it is required that the DFT is computed as fast as possible. The FFT approach is an effective algorithm and widely used [2]. But generally the FFT can be computed only after the end of the sampling operation as shown in Fig.1(a). If we want to compute the DFT faster than the FFT approach, the DFT must be computed simultaneously with the sampling operation as shown in Fig.1(b). If the N-point DFT is implemented with parallel computation, N multipliers and N accumulators are necessary.

We propose a new approach for the computation of the DFT, that is, MR-DFT, without any multiplications of twiddle factors. In this algorithm, the extreme values of each signal frequency component are sampled and accumulated by alternate additions and subtractions. This algorithm can be implemented with N accumulators and one multiplier using parallel processing.

The basic-type MR-DFT algorithm mentioned above has compensation terms. This algorithm has the same features as the arithmetic Fourier transform (AFT) [5], but the MR-DFT is developed from other viewpoint, that is, the multirate signal processing [6].

To exclude the compensation operations of the basic-type MR-DFT, the frequency-shift-type MR-DFT algorithm is developed. In the MR-DFT, the generation of sampling pulses is important. Two methods for the generation of sampling pulses are proposed and the effects of the sampling time errors are considered.

II. ALGORITHM OF THE MR-DFT

A. Principle of the MR-DFT

First, we consider an input signal x(t).

\[ x(t) = a_0 + \sum_{k=1}^{N-1} \left( a_k \cos 2\pi f_k t + b_k \sin 2\pi f_k t \right) \]  \hspace{1cm} (1)

where \( f_k = pf \) and \( f_1 \) shows a fundamental frequency.

The extreme values of cosine and sine waveforms of each frequency component in the input signal x(t) are sampled. Then these sampled values are added and subtracted alternately, and accumulated. The accumulated values are shown in the right edge of Fig.2. From these accumulated values, Fourier coefficients can be computed. But the direct component \( a_0 \) can be computed by only additions as the same manner in the DFT.

\[ a_0 = \frac{1}{N} \sum_{n=0}^{N-1} x(n) \]  

\[ a_1 = \frac{1}{2} \sum_{n=0}^{N-1} (-1)^n x(n) \]
\[ b_k &= \frac{1}{2} \sum_{n=0}^{2^k-1} (-1)^n x(n/2^k) + \frac{1}{4f_k} + b_n \quad (2) \\
\alpha_k &= \frac{1}{2^{k-1}} \sum_{n=0}^{2^k-1} (-1)^n x(n/2^k) \quad (k = 2, 3, 4) \\
b_n &= \frac{1}{2} \sum_{n=0}^{2^k-1} (-1)^n x(n/2^k) \quad (4) \\
\]

B. Basic-type MR-DFT

We consider a band-limited periodic signal \( x(t) \).

\[ x(t) = a_0 + \sum_{p=1}^{P} (a_p \cos \frac{2\pi t f_p}{f_s} + b_p \sin \frac{2\pi t f_p}{f_s}) \quad (3) \]

To compute Fourier coefficients \( a_k \) and \( b_k \), first of all, \( x(t) \) is sampled at the sampling time \( t_n = n/(2f_s) \) and \( t_n = n/(2f_s) + 1/(4f_k) \) \((n = 0, 1, \ldots, 2^k-1) \), respectively.

\[ x(n/2^k) = a_0 + \sum_{p=1}^{P} (a_p \cos \frac{n \pi |p|}{2^k} + b_p \sin \frac{n \pi |p|}{2^k}) \]

Next, these sampled data are added and subtracted alternately and these accumulated values are divided by the sample number 2k.

\[ \frac{1}{2^k} \sum_{n=0}^{2^k-1} (-1)^n x(n/2^k) = \frac{1}{2^k} [2k a_0 + 2k \sum_{q=1}^{q=2^k} x(q) (2^k-q)] \quad (6) \]

\[ \frac{1}{2^k} \sum_{n=0}^{2^k-1} (-1)^n x(n/2^k) = \frac{1}{2^k} [2k a_0 + 2k \sum_{q=1}^{q=2^k} b(q) (-1)^q] \quad (7) \]

Where \( u = \lfloor \frac{n}{2^k} \rfloor \), (integer).

From Eqs. (6) and (7), the algorithm of the basic-type MR-DFT can be derived.

\[ a_0 = \frac{1}{2^r} \sum_{n=0}^{2^r-1} x(n/2^r) \]

\[ a_k = \frac{1}{2^k} \sum_{n=0}^{2^k-1} (-1)^n x(n/2^k) - \frac{1}{2^r} \sum_{q=1}^{q=2^k} x(q) \quad (k = 1, 2, \ldots, r) \quad (8) \]

\[ b_k = \frac{1}{2^k} \sum_{n=0}^{2^k-1} (-1)^n x(n/2^k) - \frac{1}{2^r} \sum_{q=1}^{q=2^k} b(q) \quad (10) \]

where \( j = k + m \quad (k = 1, 2, \ldots, r) \),

\[ x'(t) = x(t) \cos (2\pi f_k t) \quad (12) \]

III. IMPLEMENTATION OF THE MR-DFT

If \( f_{m} = f_{\text{center}} \), as shown in Fig.4, the lower half part of signal frequency components can be computed by the frequency-shift-type MR-DFT and another higher half part can be computed by the basic-type one. The frequency modulation can be carried out by an analog-digital multiplication using a digital-to-analog converter as shown in Fig.5. Alternate operations of additions and subtractions can be realized in the following computation:

![Fig.2. Principle of the multirate-DFT (MR-DFT).](image)

![Fig.3. Principle of frequency shift.](image)

![Fig.4. Implementation of the MR-DFT without compensation.](image)
Acc(n) = x(n) - Acc(n - 1),  \quad (13)

where Acc(n) and x(n) show the values of accumulator and sampled data, respectively. After the last accumulation, the value of accumulator can be written by

\[ \text{Acc}(N-1) = \sum_{k=0}^{N-1} (-1)^k x(n). \quad (14) \]

The multiplier for mean operation can be used in time sharing, because the last operations of each accumulator end at almost different times.

Two methods for the generation of sampling pulses are shown in Fig.6. One of them is based on using read only memories (ROMs) and its resolution time is determined by the clock frequency of an address counter. Another is realized by the consumed time of the program [7]. The minimum sampling period is 20 states and the sampling periods greater than 20 states can be made with the resolution of 4 states when the CPU of Z80 is used.

![Fig.5. Modulator using a digital-to-analog converter](image)

![Fig.6. Generation of the sampling pulses (In the case of Fig.2.](image)

**IV. PERFORMANCE OF THE MR-DFT**

**A. Frequency responses**

The operations of the basic-type MR-DFT can be represented by the following an FIR digital filter.

\[ H(z) = (1 - z^{-1} + z^{-2} - \cdots + z^{-(N-1)})/N \quad (15) \]

Fig.7 represents the amplitude characteristics of the frequency responses of the MR-DFT and the conventional DFT in the case \( r = 4 \). The amplitude characteristics for computing the frequency component \( f_0 \) in the MR-DFT are equivalent to those of the conventional DFT with the sampling frequency \( f_{s0} = 2f_0 \).

**B. Effects of sampling time errors**

The performance of the MR-DFT is determined by the accuracy of sampling time when the amplitude quantization errors can be ignored.

For the case of \( f_s = 100 \) Hz and \( f_r = 6.4 \) kHz, we evaluate the amplitude errors \( \Delta a'_{\Delta a_k} \) and the orthogonality errors \( a'_{\Delta a_k} / a_{\Delta a_k} \), \( p = k \) using sampling pulses in Fig.6(a) and four sampling-time patterns shown in Fig.6, where \( a_{\Delta a_k} \) is the true amplitude, \( a'_{\Delta a_k} \), \( a'_{\Delta a_k} \) are calculated ones and \( \Delta a'_{\Delta a_k} = a'_{\Delta a_k} - a_{\Delta a_k} \).

The relationship between the maximum amplitude errors \( \Delta a'_{\Delta a_k} / a_{\Delta a_k} \) and the clock frequencies \( (f_s) \) of the ROM address counter is shown in Fig.5. The computer simulations are

![Fig.7. Comparison of the MR-DFT and the conventional DFT spectra](image)
made with the arithmetic operation of 32 bit floating point. On the other hand, Fig. 10 shows the relationship between the maximum orthogonality errors \( \frac{a_n}{a_m} \) and the clock frequencies \( (f_c) \). In these figures, the sampling pattern (4) shows the best results.

C. The number of operations

Table 1 shows the number of operations when an input signal composed of \( r \) frequency components is analyzed and all Fourier coefficients are computed. Those of the FFT are also written, but generally the FFT cannot be computed if the sampling operation does not end. Comparing to the conventional DFT, the MR-DFT has smaller additions by the magnitude of 3/4 and smaller multiplications by that of 1/2\( r \). But the number of sample data in the MR-DFT is larger, that is, about \( 3r^{2}/2 \), comparing to \( 2r \) in the DFT.

\[
\begin{array}{lll}
\text{addition} & 3r^{2}+4r & 4r^{2}+2r \\
\text{multiplication} & 2r & 4r^{2}+2r
\end{array}
\]

Table 1 Comparison of the number of operations

\( (r: \text{number of frequencies}) \)

V. CONCLUSIONS

We propose a new computational algorithm of the DFT that enables us to compute Fourier coefficients simultaneously with the sampling operation. The MR-DFT is implemented by both the basic-type MR-DFT and the frequency-

- \text{round down}: pulse A, \text{round up}: pulse B
- \text{round off}: pulse A or B (nearest pulse)

\[
\begin{array}{ll}
\text{correct sampling time} & 0 \quad 0.5T \quad 1T \\
\text{clock pulse} & 0 \quad 1T
\end{array}
\]

four sampling patterns

1. round down, round off
2. round down, round up
3. round down, round down
4. round off, round down

Fig. 8. Four sampling-time patterns.

Fig. 9. Relationship between the maximum amplitude errors \( \Delta a_n/a_m \) and clock frequencies \( (f_c) \) of the ROM address counter.

REFERENCES


Fig. 10. Relationship between the maximum orthogonality errors \( \frac{a_n}{a_m} \) and clock frequencies \( (f_c) \) of the ROM address counter.
ANALYSIS OF QUANTIZATION NOISE IN FREQUENCY SAMPLING FIR FILTERS AND ITS REDUCTION VIA ERROR FEEDBACK

Daniel Gold and Hanoch Ur

Faculty of Engineering, Department of Electrical Engineering - Systems, Tel-Aviv University, Tel-Aviv 69978, Israel

This paper considers first the problem of analysis of quantization noise in frequency sampling complex FIR filters. The error feedback method, in which the discarded part of the word is appropriately feedback to the filter, so as to reduce the total noise quantization, is then applied to these filters. Reduction is indeed achievable under certain conditions and the amount of improvement is a function of the filter characteristics and the specific implementation. This is in contrast to the conventional, non recursive FIR filters.

1. INTRODUCTION

In this paper we first examine the quantization noise in frequency sampling FIR filters and then show that it can be reduced via error feedback (EFB).

The quantization operation, in multiplication in digital filters, usually done by rounding the product to the filter wordlength, causes a roundoff error. This roundoff error is usually described as a white noise sequence, which satisfies appropriate assumptions and its probability density function is assumed to be uniform over the quantization interval $\Delta = 2^{-b}$, in a $b$-bit system [1,2,3,4]. The nonlinear quantization effect is then replaced by a statistical linear model. The variance of the noise signal $e(n)$ is $\sigma^2 = 2^{-2b}/12$ for roundoff. In general the signal and the multipliers may be complex (see later), thus four noise sources are created. The noise which was created at the multipliers is filtered by the filter and may cause a strong noise at the filter output.

The EFB technique suggested to reduce this effect is as follows: the error (the least significant bits), created after the multiplication, by quantizing the double length intermediate result to the original wordlength of the digital filter, is saved in order to correct the product at the following sampling instants. Here insight can be obtained by considering the technique in the frequency domain and regarding it as error spectrum shaping (ESS). This shaping is done by introducing zeroes into the noise transfer function from its (equivalent) sources to the output. These zeroes are located at the passband of the filter, and decrease the poles' influence. In this way, the quantization noise (which is no longer white) is decreased in the pass band, possibly increased in the stop band, where, however, it is attenuated.

The technique, EFB, or ESS, has been analyzed in various digital systems: IIR filters, ROM filters [2,5,6,7,8,9], and state space digital filters, [9,10].

The corresponding technique for reducing errors in structures containing zeroes is feed-forward [13]. It is not done, however, in conventional FIR structures because there it is tantamount to double precision arithmetic. Nevertheless, FIR filters of the frequency sampling type, considered in this work have a recursive structure and are thus amenable to EFB. The frequency sampling FIR filter [1], Fig. 1, has a transfer function:

$$H(z) = (1 - Z^{-N}) \cdot \frac{1}{N} \sum_{k=0}^{N-1} \frac{H(W_k^N)}{1-W_k^N \cdot Z^{-1}}, \quad (1)$$

$W_k^N = e^{j2\pi k/N}$ are equidistant frequencies (frequency samples) at which the transfer function of the filter $H$ is specified. They, as seen, are also poles of $H(z)$, cancelled by the zeroes of $1-Z^{-N}$. In order to avoid stability problems the pole locations and the zeroes are moved from $W_k^N$, on the unit circle, to $rW_k^N$, $r$ is usually taken as $1-$, $0< r < 1$, (often $r=2^{-m}$), so $H(z)$ becomes:

$$H(z) = (1 - rN \begin{bmatrix} 1 \end{bmatrix}) \cdot \frac{1}{N} \sum_{k=0}^{N-1} \frac{H(rW_k^N)}{1-rW_k^N \cdot Z^{-1}}, \quad (2)$$

Hereafter $H(rW_k^N)$ will be denoted, for convenience, by $H(k)$.

This paper deals with the general case of complex (signals and coefficients) frequency sampling FIR filters. This complex representation is very useful in applications involving signals and their quadrature, such as in communication systems, radar systems, electronic warfare systems, etc. [14]. Investigation of the special, but common, case of linear phase real coefficients structures indicates that EFB doesn't significantly improve the quantization error behavior. The analysis for this special case will be published later.
2. The quantization noise in Frequency Sampling realization of complex FIR filters

After each multiplication, in the filtering process, a roundoff operation is performed, this operation will be represented by a noise source [1]. Fig. 1) where,

- \( n_A(n) \) is the noise source created by the \( r^N \) multiplication and roundoff.
- \( n_B(k,n) \) is the noise source created by the \( r^{W_1^k} \) multiplication and roundoff in the \( k \)-th parallel branch.
- \( n_c(n) \) is the noise source created after rounding the sum of the multiplications by \( H(k)/N \). Note that by rounding after summation in double precision of the intermediate multiplication results, only one noise source is introduced. This is instead of \( N \) noise sources, which is the case when rounding in each parallel branch before summation. This scheme is not always implementable, depending on the system.

![Fig. 1: A frequency sampling FIR filter and its quantization noise sources.](image)

We now calculate the contribution of the various noise sources to the overall output noise. We will assume that the input sequence is complex. Let us note that usually the input signal is scaled in order to avoid the risk of overflow. Absence of overflow is assumed in the following:

The contribution of \( n_A(n) \) to \( \sigma_E^2 \), the output noise variance is:

\[
\sigma_A^2 = \frac{2\sigma_e^2}{N(1 - r^{2N})} \left( \sum_{k=0}^{N-1} |H(k)|^2 \right). \tag{3}
\]

It is seen that the relative contribution of \( \sigma_A^2 \) depends on the filter characteristics and thus may be significant. One way to eliminate the effect of \( n_A(n) \) is to feed \((-) n_A(n) \) directly to the adders in the junctions of the feedback loops, using the principle of error feedforward [15]. It should be noted that this operation is equivalent to double precision processing until the recursive elements, there is no need, however, to implement the noise feedforward technique in branches, where the coefficients \( H(k) \), are equal to zero, or small enough, thus saving hardware/software. Thus the effect of \( n_A(n) \) is eliminated and it will not be taken further into consideration.

The total output noise variance is:

\[
\sigma_E^2 = \sigma_c^2 + \sum_{k=0}^{N-1} \sigma_{B(k)}^2 = 2\sigma_e^2 \left[ 1 + \frac{1}{N(1-r^2)} \sum_{k=0}^{N-1} |H(k)|^2 \right]. \tag{4}
\]

3. Error Feedback Implementation

In this section we will apply error feedback to the frequency sampling filter. As our structure contains first order loops (one delay) only, first order error feedback is used [15, 11], to compensate each parallel path pole. The quantization noise is feedback to the multiplier input, while delayed and multiplied by a coefficient, which may be complex. The feedback is implemented only for the quantization noises in the loops, and not for the \( n_c(n) \) source which is not part of a recursive circuit.

The error feedback coefficients can be complex, and expressed as \( r_k e^{i\phi_k} \), \( k=0, \ldots, N-1 \), Fig. 2. For the error feedback realization let us denote the quantization noise after the feedback has been added by \( n_{B(k)}(k, n) \), Fig. 2, presents the error feedback realization. The total output noise variance is:

\[
\sigma_E^2 = 2\sigma_e^2 \left[ 1 + \frac{1}{N(1-r^2)} \sum_{k=0}^{N-1} |H(k)|^2 (1-r^2) \right] \nonumber \\
+ \left[ r^2 + r^2 \cos(2\pi k/N - \phi_k) \right] \tag{5}
\]

The next step is to insure the quantization noise reduction by properly choosing the error feedback coefficients: \( r_k \) and \( \phi_k \), \( k=0, \ldots, N-1 \). By comparing the output noise variance in error feedback configuration (5) to the noise variance in the original configuration (4), it can be shown that

\[
r_k < 2\cos(2\pi k/N - \phi_k) \tag{6}
\]

(6) is the first condition for ensuring noise reduction.

In order to avoid additional quantization which can be created
in the error feedback circuitry by error feedback coefficient multiplication, we form another condition:

\[ \text{Re}(r_k e^{j\phi_k}) = r_k \cos \phi_k = \pm 2^n \text{ or } 0, \]

\[ \text{Im}(r_k e^{j\phi_k}) = r_k \sin \phi_k = \pm 2^n \text{ or } 0, \text{ with } n \text{ integer}. \]  \hspace{1cm} (7)

We note that \( r_k \) and \( \phi_k \) appear in (5) as part of the expression \( p_k \) which is defined as:

\[ p_k = r_k^2 + 1 - 2r_k \cos (2 \pi k/N - \phi_k). \]

The output noise variance is decreased as \( p_k \) becomes small (for each \( k \)).

The necessary procedure for choosing the error feedback coefficient \( (r_k, \phi_k) \) is to find \( (r_k, \phi_k) \) which satisfied conditions (6) and (7). If several pairs of \( (r_k, \phi_k) \) are found, then choose the pair which minimizes \( p_k \). It should be noted that adding error feedback is worthwhile only in the pass band branches. There is no practical need to add error feedback in the stop band branches, where the noise is well attenuated and its contribution is small due to the small value of the corresponding terms of \( H(k) \).

4. Simulation Results

The theoretical amount of improvement due to ESS implementation is defined as the ratio between the output noise variance without the error feedback (4) and with the error feedback (5). Let this improvement factor be denoted as \( P_\text{T} \). In order to confirm the theoretical results, a simulation program was written in Pascal language. During the simulation, the actual improvement factors, \( P_\text{a} \), were calculated.

Figure 3, a and b present an example of graphic outputs of the error output without and with error feedback. Since the real and imaginary parts look alike, only the real part is presented. The filter was a BPF with \( N = 128 \), \( r = 0.999 \), passband = 25% and 12 bit wordlength.

Extensive simulation results confirmed the theoretical ones.

**Note:** proper scaling of the input signal is required to avoid overflow. "Overscaling", however, results in a reduced signal, while the quantization noise remains the same, thus the signal to noise ratio, with or without error feedback, is reduced. A large dynamic range, i.e. no "overscaling", is implied [3] in the assumptions mentioned earlier for error feedback analysis. During simulation a proper scaling was generated for each filter.

Fig. 3: Simulation results for a BPF \( N = 128 \), \( r = 0.999 \), 25%, 12 bits, real part only.
REFERENCES


SENSITIVITY AND ROUNDOFF NOISE IN WAVE DIGITAL FILTERS VERSUS OPTIMAL STATE-SPACE REALIZATIONS

José L. Sanz-González, Enrique Rodrigo-Aparicio and Francisco Lema
Universidad Politécnica de Madrid
ETSI Telecomunicación-UPM
Ciudad Universitaria, 28040 Madrid, Spain.

This paper compares under quantization conditions the performances of elliptic, Chebyshev and Butterworth filters realized by wave digital filters and by parallel structures composed of second order state-space sections (optimal and direct sections). The classical LC ladder filters were the reference analog structure for the wave digital filters.

1. INTRODUCTION.

As is well known, a great number of structures exists for realizing linear discrete-time systems by digital implementation. However, most structures can be included in two families: the state-space structures and the wave digital filters.

The state-space structures [1] are derived from the state equations

\[ x(k+1) = A \cdot x(k) + b \cdot u(k) \]
\[ y(k) = c \cdot x(k) + d \cdot u(k) \]

where \( u(k) \) is the scalar input, \( y(k) \) is the scalar output and \( x(k) \) the \( n \)-th order state vector; matrices \( A \), \( b \), \( c \) and \( d \) are \( m \times n \), \( n+1 \), \( 1 \times m \) and \( 1 \times 1 \) real constant matrices, respectively.

The transfer function \( H(z) \) is expressed by \( H(z) = c \cdot (zI-A)^{-1}b + d \) and the structure coefficients are the elements of \( A \), \( b \), \( c \) and \( d \). The application of similarity transformations modifies structure coefficients, but the transfer function \( H(z) \) remains invariant.

The wave digital filters [2] are derived from the analog filters, specially from the classical LC filters. Several of the good properties of classical filters are transferred to the corresponding wave digital filters.

The transfer function \( H_n(z) \) of the analog filter is related to the transfer function \( H(z) \) of the corresponding wave digital filter by means of the bilinear transformation:

\[ s = (z-1)/(z+1) \].

The signal variables considered in the analog filter are voltage (or current) wave quantities; the corresponding discrete-time signal flow graph relates incident and reflected wave signals. The discrete-time structure is obtained directly from the analog filter structure with some specific and precise rules of transformation [2].

2. QUANTIZATION.

The finite register length of digital realizations modifies the characteristics of the filters designed for infinite precision arithmetic. The transfer function degrades caused by quantization of filter coefficients, and the roundoff noise due to the operation result quantizations, are usually studied independently. However, some researchers concluded (by theoretic and experimental proofs) that low-sensitivity structures with respect to the coefficients have also low roundoff noise power, and vice versa.

For state-space structures and fixed-point numerical representation, there is a tradeoff between minimum roundoff noise power and minimum number of multiplications in the implementation of a specific transfer function. Roughly speaking, for an \( n \)-th order filter, \( (n+1)^2(n+1) \) is the number of multiplications per output sample for the minimum roundoff noise structure, and \( 2(n+1) \) for the direct structure (minimum number of operations); however direct structures have high roundoff noise power when poles are close to \( z=1 \) (or \(-1\)). In [3] and [4] results of roundoff noise power were given for several kinds of filters realized in parallel and cascade connections of second order sections (direct sections against optimal state-space sections); in [5] results are given for floating-point arithmetic.

For wave digital filters, it is well known the low sensitivity of the transfer function with respect to the coefficients. Also, the number of coefficients can be minimum (n coefficients for an \( n \)-th order filter), and the dynamic range acceptable. Consequently, it is expected that the roundoff noise will be low.
3. COMPUTER RESULTS.

We have done a comparative analysis of the quantization effects on wave digital filters and state-space structures in the case of fixed-point numerical representation. The filters under consideration were Butterworth, Chebyshev and elliptic low-pass filters, and the cutoff frequency was varied from 0.05π to 0.95π. The reference analog structures for wave digital filters were the classical ladder LC-filters (LC chains). The results were obtained by formulas evaluation and by computer simulation.

3.1 Sensitivity.

Considering the sensitivity of the transfer function amplitude with respect to the filter coefficients, the number of bits for wave digital filter coefficients ranges from 6 to 8 with filter orders from 5 to 13. On the other hand, for the parallel structures composed of optimal state-space sections, the number of bits ranges from 10 to 16, under the same conditions (see Fig.1 and Table 1). The number of bits required for the coefficients increases as the order of the filter increases and, also, as the cutoff frequency approaches 0 or to π but this number is strongly insensitive to the other filter characteristics. The number of bits required for the wave digital filter coefficients is less sensitive to the filter parameters than the number of bits for state-space structures, as it can be seen from Table 1.

3.2 Roundoff Noise.

Now, we are referred to Fig.2. The roundoff noise power of optimal state-space filters is independent of the bandwidth (or cutoff frequency value). On the other hand, for wave digital Butterworth and Chebyshev filters, the roundoff noise power depends on the bandwidth and increases as the cutoff frequency approaches 0 (narrow bandwidth) or to π (broad bandwidth), being the minimum in the interval (0.4π, 0.7π); for elliptic filters, the noise power are flat and minimum from 0 and 0.4π. Considering the same overflow probability (about 10^-6), these minimum roundoff noise power values for wave digital filters are similar to those of the optimal state-space filters (e.g. about -82 dB for 5th-order filters and register wordlength of 16 bits). However, for wave digital filters, the roundoff noise power increases monotonically as the cutoff frequency decreases from 0.4π to 0 (except for elliptic filters), or increases from 0.7π to π; on the other hand, for optimal state-space filters, the roundoff noise power is maintained constant over the whole interval (0, π) of cutoff frequencies.

The dynamic range for wave digital filters (unscaled filters) decreases as the cutoff frequency (ωc) approaches to 0 or to π, but for optimal state-space filters (1- scaled filters) the dynamic range is independent of ωc.

4. CONCLUSIONS.

The most important conclusions of this paper are summarized here.

For similar transfer function degradations, the wave digital filters require less bits for the coefficients than the optimal state-space filters.

For narrow and broad bandwidth low-pass filters (except for elliptic filters), wave digital filters (obtained from LC ladder structures) have greater roundoff noise power than optimal state-space filters (it is considered the same overflow probability and white Gaussian signal at the input).

REFERENCES.

Fig. 1. Frequency response (Magnitude) of a 5th-order elliptic low-pass filter. In (a) and (c) wave digital filter responses with — ideal, — 6-bits ..., 5-bits and — 4-bits coefficients. In (b) and (d) the parallel structure responses (considering optimal state-space sections) with ideal and — 8-bits coefficients. In (a) and (b) the overall frequency responses and in (c) and (d) the passband responses.

<table>
<thead>
<tr>
<th>Filter Order</th>
<th>Elliptic Wave</th>
<th>Elliptic Parallel</th>
<th>Chebyshev Wave</th>
<th>Chebyshev Parallel</th>
<th>Butterworth Wave</th>
<th>Butterworth Parallel</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>6</td>
<td>10</td>
<td>6</td>
<td>10</td>
<td>6</td>
<td>10</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>12</td>
<td>6</td>
<td>11</td>
<td>6</td>
<td>11</td>
</tr>
<tr>
<td>9</td>
<td>8</td>
<td>13</td>
<td>7</td>
<td>12</td>
<td>6</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>9</td>
<td>15</td>
<td>8</td>
<td>13</td>
<td>6</td>
<td>16</td>
</tr>
</tbody>
</table>

Table 1. The minimum number of bits required by the coefficients of the low-pass filters with similar frequency responses in passband and stop-band than Fig. 1. The filter order depends on the transition band. The degradation of the passband frequency response delimits the lower number of bits.
Fig. 2. Output roundoff noise power ($\sigma^2$) versus cutoff frequency ($\omega_c$) for low-pass filters and 16-bit fixed-point arithmetic. In (a), (b) and (c) the noise power curves of wave digital filters, and in (d), (e) and (f) the noise curves of parallel structures with second-order sections (--- optimal sections and - - - - direct sections). Also, (a) and (d) curves correspond to 5th-order elliptic filters, (b) and (e): 13th-order Chebychev filters, and (c) and (f): 13th-order Butterworth filters.
LUMPED, SAMPLED-DATA AND DIGITAL FILTERS WITH MAXIMUM NUMBER OF CONSTRAINTS ON AMPLITUDE AND PHASE CHARACTERISTICS

TAMÁS HENK and MOHAMMED ABO-ZAHHAD

Department of Telecommunications and Telematics, Technical University of Budapest, H-1111 Budapest, Stoczek U. 2, HUNGARY
Tel. +361 1 810 300 ext 2441 and Fax: +361 1 1 666 808

For optimal filters exhibiting specified amplitude and phase (or group delay) characteristics, it is required that all the free parameters of the transfer function be used for the approximation. To achieve this requirement, the number of constraints on the amplitude and phase characteristics is defined first on general interpolation bases. Constant or arbitrarily prescribed low-pass or band-pass group delay and amplitude characteristics are approximated in the maximally-flat, ripple or mixed sense for lumped filters, whereas high-pass and band-rejection characteristics are also considered for distributed or sampled data filters. For optimal filters the relationship between the number of the free parameters and the number of the amplitude and phase constraints for a given degree is derived for the non-reciprocal and for the reciprocal lossy as well as for the reciprocal reactant cases. Some published separate and simultaneous approximation methods are referred, evaluated and compared on the above basis. It is pointed out that some of these methods don't satisfy the above requirements, although the optimal solution would exhibit higher performances.

1. INTRODUCTION

The simultaneous amplitude and phase approximation is an old filter design problem [1]-[4]. Considering the development of these methods Rhodes introduced new aspects into the construction of transfer functions exhibiting simultaneous amplitude and phase requirements [5]. On these bases a number of contributions have been published for further results [6]-[11]. Some of them however don't investigate how much constraints on the amplitude and phase characteristics can be prescribed altogether for a given degree and the number of the prescribed constraints is less than could be [7]-[10].

In this paper the relationships between the number of free parameters and the number of amplitude and phase conditions for reciprocal and non-reciprocal lumped, sampled-data and digital filters are given. On this basis a simple method is given to check the optimal property of these filters.

2. THE DESIGN CONCEPTION

The complex frequency variable $\lambda$ is introduced for the frequency variable of lumped filters or for the reference frequency variable of sampled data filters [12] or for the Richard's variable of distributed filters. The transmission of filters $S_{12}(\lambda)$ is considered for which the boundary condition $|S_{12}(\lambda)| \leq C$ holds, where $C \leq 1$ for structurally passive filters and $C$ is an arbitrary real number otherwise. As examples, reciprocal lossy (reactant) or lossy, non-reciprocal reflectional terminated lossless and also non-reciprocal lossless filters are structurally passive. Although the structurally passive condition ensures good realization properties [12], $C > 1$ is not excluded from this paper since in some referred methods $C > 1$ [7]-[9]. The scattering matrix of the lossless cases will be characterised by the canonical polynomials $f(\lambda)$, $g(\lambda)$, and $h(\lambda)$ [13], whereas the notation

$$
S_{12}(\lambda) = \frac{f(\lambda)}{g(\lambda)}
$$

(1)

will be used also for other cases. The polynomial $g(\lambda)$ is strictly Hurwitzian with degree $d$ and the degree of $f(\lambda)$ and $h(\lambda)$ is not greater than $d$, $f(\lambda)$ is even or odd for reciprocal lossless filters, whereas $h(\lambda)$ is odd or even for symmetrical lossless filters.

* On leave from Department of Electrical and Electronic Engineering, Faculty of Engineering, University of Assiut, Assiut, Egypt.
Simultaneous approximation of linear or arbitrary phase, constant or non-constant passband amplitude and zero stopband amplitude characteristics will be considered in maximally flat, combined flat and ripple, ripple, or equiripple sense for low-pass, band-pass, high-pass and band-rejection specifications. For lumped filters however, the linear phase specification can't be approximated in an infinite frequency band, therefore band-pass or combined low-pass and band-pass filters have to be applied instead of high-pass or band-rejection filters respectively.

Considering linear phase filters the specified delay for low-pass and band-pass filters and also the mid-band phase-shift in the band-pass case are often free parameters for filter design [14]. So the appropriate filter design conception is to choose a sufficiently high filter degree $d$, to fix the passband(s) for the amplitude, the band(s), for the phase-approximation, the stopband(s), the amplitude ripple in the passband(s) and in the stopband(s), whereas to have the phase-error ripple, the delay (and also the mid-band phase shift for band-pass filters) as a result of the design [11, 15-17] (first design conception).

It is more common however to fix the delay, and to have also the stopband and/or the passband amplitude ripples as a result instead [11-10] (second design conception).

3. DEFINITION OF THE NUMBER OF CONSTRAINTS

The above approximation problems can be solved by linear interpolation methods. So the number of the phase and amplitude constraints are introduced on an interpolation basis accommodating to both the first and the second design conceptions.

Let us define the sets of the interpolating frequencies $\omega_{hi}$, $\omega_{pl}$ and $\omega_{sl}$ where values and derivatives of the phase, the passband and stopband amplitude characteristics is specified respectively. The secondary set of frequencies $\nu_{hi}$, $\nu_{pl}$ and $\nu_{sl}$ are further introduced comprising each frequency $\omega_{hi}$, $\omega_{pl}$ and $\omega_{sl}$ respectively as many times as the total number of the specified value and derivatives is at this frequency.

To define the number of constraints for each characteristic, the number of the specifications, i.e. the number of the elements in the set $\nu_{hi}$, $\nu_{pl}$ and $\nu_{sl}$ are considered.

More exactly, the specified zeros of the interpolation error is written in the form:

$$\lambda \prod_{l=1}^{p} \left( \lambda^2 + \nu_{pl}^2 \right),$$

and

$$\lambda \prod_{l=1}^{s} \left( \lambda^2 + \nu_{sl}^2 \right),$$

for the phase, for the squared passband amplitude and for the zero valued stopband amplitude characteristics respectively [11]. In equations (2), $p$ and $h$ denote the number of constraints for the passband amplitude passband phase respectively. Also, denoting the degree difference of $g(A)$ and $f(A)$ by $s_1$, the number of stopband amplitude constraints is defined by $s = s_0 + s_1 + 2 s_2$.

4. THE MAXIMUM NUMBER OF CONSTRAINTS

Fixing the filter degree $d$, boundary conditions can be set up for $p$, $s$ and $h$ for different filters determining the range in which $p$, $s$ and $h$ can be chosen to satisfy a given specification. On the other hand, $p$, $s$ and $h$ have to satisfy also an optimality condition ensuring that all the free parameters of the transfer function is used for the approximation. Some previous results are available in References [11-5] for special cases. The general conditions can be stated as:

the boundary and optimality conditions on the passband and stopband amplitude and on the phase constraints for reciprocal reactant (lossless) and for reciprocal lossy or non-reciprocal filters are summarised in Table 1.

In degenerate cases not only the specified but also extra zeros of the interpolation error occur [7], [11], these zeros however are not counted in $s$, $p$ and $h$.

Observe that $f(A)$ becomes even or odd also for non-reciprocal filters if $w_d$ is odd. So the conditions become identical to the conditions of the reciprocal reactant filter. If on the other hand $w_d$ such that the specified passband amplitude is constant, then $h(A)$ becomes odd or even in the reciprocal reactant cases and so symmetrical lossless filter is involved for odd $d$. If for example a $d$-th degree Cauer filter is considered, then $s=p=d$ and $h=1$ due to the delay or the stopband ripple as a free parameter according to the first or the second design
conception respectively, both optimality conditions are satisfied and \( h(\lambda) \) is odd.

Proof: Consider first the case of \( d \)th degree non-reciprocal transfer function to be:

\[
H(\lambda) = \frac{N_d(\lambda)}{D_d(\lambda)} = \frac{\sum_{i=0}^{m} a_i \lambda^i}{\sum_{i=0}^{d} b_i \lambda^i} \tag{3}
\]

The relation between the degree of the transfer function and the number of the amplitude and phase constraints, for non-reciprocal case, is given by:

\[
s + p + h = 2d + 1 \tag{4}
\]

Equation (4) can be proved as follows.

a) The formulation given by (3) insures that there are \((d-m)\) stopband amplitude conditions satisfied at \( \lambda = \alpha \). i.e., initially \( s = d-m \).

b) There are \((d+m+2)\) coefficients in equation (3), but only \((d+m+1)\) are independent, because any of the \((d+m+2)\) coefficients may be assigned an arbitrary value without any loss in generality. The remaining \((d+m+1)\) coefficients are satisfied such that the same number of amplitude and phase conditions are satisfied.

c) From (a) and (b), the total number of the passband amplitude conditions, the stopband amplitude conditions and the phase conditions are \((2d+1)\).

Consider second the case of reciprocal reactant transfer function. It can be deduced from equation (2) with:

\[
a_i = 0, \; i=1, 3, \ldots, m-1 \text{ for } m \text{ even } \\
a_i = 0, \; i=0, 2, \ldots, m-1 \text{ for } m \text{ odd } \tag{5}
\]

In this case the relationship between the degree and the amplitude and phase constraints is given by:

\[
s + 2(p + h) = 3d + 2 \tag{6}
\]

Equation (6) can be proved as follows.

1) As in the non-reciprocal case, the number of stopband conditions results from the formulation given by equations (3) and (5), is \((d-m)\) where all are at \( \lambda = \alpha \).

2) The number of free parameters in the denominator is \((d+1)\) and the numerator has \( m \) zeros. If \( x \) from these zeros is forced to be on the \( \omega \)-axis to satisfy some stopband amplitude specifications, then the total number of stopband conditions is \((d+m+1)\). Consequently, the number of passband amplitude and phase conditions are \((d+1+a/2)\), where \(a=m-x\) is the number of numerator zeros not on the \( \omega \)-axis.

3) From (1) and (2) one can argue that \( s+2(p+h) \) is equal to \( 3d+2 \).

5. EVALUATIONS OF SOME METHODS

In previous section it has been shown that the number of free parameters for non-reciprocal transfer functions is \( 2d+1 \). This number can be distributed between the passband amplitude, stopband amplitude and the phase requirements under the conditions given in Table 1. But generally further considerations are needed for stable and bounded solutions. Evaluation of some published solutions shows that those of [7]-[9] are non-optimal. This is because not all the degrees of freedom are used to improve the required characteristics. The most severe case is that given in [9]. It is non-optimal even if no conditions are imposed on the stopband amplitude characteristic; i.e., if \( m=0 \).

Similarly it has been found that there are some non-optimum reciprocal reactant filters. The solutions given in [7]-[10] are the most severe cases. However the solution given in [10], is optimum if \( r=0 \) and this solution corresponds to the low-pass case with two transmission zeros at infinity.

6. CONCLUSION

The relationships between the number of amplitude and phase constraints and the number of free parameters for reciprocal reactant and non-reciprocal optimal filters are given. Evaluating by means of Table 1 some published simultaneous approximation methods, it turns out that some of them are not optimal in the sense that they don't use all the free parameters available in a transfer function with fixed degree, without any extra realization benefit. Examples confirm however that their features can be enhanced by using all the parameters.
Table 1. Conditions on passband-stopband amplitude and phase constraints

| Type of filter | Condition on | Specified $|S_{12}(\omega)|$ | Boundary Conditions | Optimality conditions |
|----------------|--------------|--------------------------------|---------------------|----------------------|
| Reciprocal reactant | even or odd | constant | $0 < \delta \leq \frac{3d-1}{2}$ | $1 + 2h < \delta < 3d + 2$ |
| Reciprocal lossy or Non-reciprocal | | non-constant | $0 < \delta \leq \frac{3d-3}{2}$ | $1 + 2h < \delta < 3d + 2$ |

REFERENCES


DIGITAL FILTERS BASED ON COMPLEX TRANSMISSION LINE NETWORKS

Akira TAKAHASHI, Nobuo NAGAI and Nobuhiro MIKI

Research Institute for Electronic Science, Hokkaido University, Sapporo 060, JAPAN

This paper deals with complex wave digital filters based on complex transmission line networks with imaginary resistances. Such networks can realize a single band pass response with asymmetrical transitions on the both sides of the pass band. Therefore an order reduction can be achieved if symmetrical response is not required. Even not one real, if there exist a reflection free port, a series adaptor based on voltage dimensional waves or a parallel adaptor based on current dimensional waves holds losslessness after quantization. Furthermore the structures of these two adaptors are identical. Therefore a new ladder construction is proposed for modularity as an alternative combination of voltage dimensional series sections and current dimensional parallel sections.

1. INTRODUCTION

This paper introduces some properties of complex wave digital filters based on complex transmission line networks with imaginary resistances. Complex wave digital filters can realize highly flexible responses [1], and useful to perform digital simulations of complex systems such as quantum electronics devices [2]. The incorporation of imaginary resistances [3] into transmission line networks [4] bring the following properties. 1) With a reference to such a network, a single band-pass transfer function is easily derived from a real low-pass filter through a new type of reactance spectral transformation. The transformation enables a flexibility to control the upper transition and the lower transition be arbitrary asymmetrical. If the design tolerance may accept such a response, the necessary filter order can be reduced in comparison with the frequency shifting transformation shown by Crystal and Ehrman [5]. 2) Even in complex cases, wave digital filters [6] possess structural correspondences to their reference network with wide aspects of numerical robustness. Even not one real [7], if there exist a reflection free port, a series adaptor based on voltage dimensional waves or a parallel adaptor based on current dimensional waves holds losslessness after quantization. Furthermore the structures of these two adaptors are identical. A new ladder construction is proposed for modularity as an alternative combination of voltage dimensional series sections and current dimensional parallel sections.

2. REFERENCE NETWORK

Lossless commensurate transmission line (LCTL) networks are basically constructed with elements shown in Fig.1 [4]. A 'unit element' is a lossless uniform element with two ports, which can be regarded as a delay unit in the sense of reflectance. In a 'commensurate' network, all elements have a unified 'round propagation delay time' $T$.

The correspondences of three complex frequency variables, $\lambda$ for LCTL networks, $z^{-1}$ for digital filters and $s$ for Laplace transformations are expressed as

$$\lambda = \tanh \frac{sT}{2} = \frac{1 - z^{-1}}{1 + z^{-1}}. \tag{1}$$

Because of the above relationship, a realizable function of $\lambda$ and a realizable function of $z^{-1}$ are compatible with each other, if $T$ is identified with the sampling period.

(a) Unit element

(b) Short circuited unit element.

(c) Open circuited unit element.

Figure 1: Basic elements of LCTL networks.

A short circuited unit element with characteristic resistance $R$ has a voltage-current relationship in the frequency domain, similar to a lumped inductance but in the sense of $\lambda$,

$$V = \lambda RI \tag{2}$$

on its input terminal pair. Besides an open circuited unit element with characteristic conductance $G$ follows a relationship, similar to a capacitance,
\[ I = \lambda GV. \]  

Let \( R \) and \( X \) be real part and imaginary part of impedance \( Z \). Similarly, \( G \) and \( S \) denote real part and imaginary part of admittance \( Y \). In a complex circuit, all signals are complex-valued even in the time domain. Let \( v(t) \) and \( i(t) \) correspond to time signals of \( V \) and \( I \), respectively. Then an instantaneous effective power \( p(t) \) is defined by (4). The superscript star denotes complex conjugate.

\[ p(t) = \text{Re}[v(t)i^*(t)] \]  

A complex time-invariant lumped element, characterized as (5) is named an 'imaginary resistance' [8].

\[ v(t) = jXi(t) \quad V = jXI \]  

3. ASYMMETRICAL TRANSFORMATION

If imaginary resistances are properly incorporated to low-pass LCTL networks, complex asymmetrical band-pass responses are obtained. Fig. 3 shows one of the simplest case, where

\[ jX = -jR \tan \frac{\Omega \pi T}{2}. \]  

The original response and the modified response are illustrated in Fig. 4, as a continues line and a break line, respectively. The attenuation level for \( \omega = 0 \) is mapped to \( \omega = \Omega \), while the positions of \( \omega = \pm \pi T \) are unchanged. Therefore a real low-pass response is transformed to an asymmetrical complex response.

If an imaginary conductance (8) is inserted parallel to an open-circuited unit element, the identical spectral transformation is achieved.

\[ jS = -jG \tan \frac{\Omega \pi T}{2} \]  

Where \( G \) is the characteristic conductance of the open-circuited unit element.

From (1)(2)(3)(7) and (8), such a transformation can be regarded as replacement of the complex frequency variable

\[ \lambda \rightarrow \lambda - j \tan \frac{\Omega \pi T}{2}. \]  

(a) Original filter.  
(b) Transformed filter.

\[ (T = 1, \quad \Omega = 2.) \]

Figure 3: Incorporation of imaginary resistance.

---Original response. ---Transformed response.

Figure 4: Asymmetrical spectral transformation.

A higher order example is shown in Fig. 5. Because of an imaginary resistance is lossless, the proposed transformation can be regarded as one of the special case of reactance spectral transformations. Therefore the magnitude ripples are inherited straightforward. If the initial specification may accept such an asymmetrical response, the necessary filter order is reduced in comparison with a popular symmetrical design.

The asymmetrical transformation described here, does not have the freedom itself to be shifted along the frequency axis. But the conventional shifting transformation (10) [5] can be applied to the network by replacing all the unit elements into complex ones including distributed imaginary resistances [8]. Therefore an arbi-
trary allocation of the pass band can be achieved.

\[ z \rightarrow z e^{-j\Omega z} \quad \text{or} \quad \lambda \rightarrow \frac{\lambda - j\alpha}{1 - j\alpha \lambda}, \quad \alpha = \tan \frac{\Omega x}{2}. \] (10)

At last, through the relationship (1), the identical frequency response can be reproduced as a digital transfer function. By tracing back the above chain of the spectral transformations, an asymmetrical complex design problem is reduced to a classical low-pass design problem.

\[ \lambda = 1 \quad (z^{-1} = 0) \] (12)

is called 'port impedance'.

Complex power wave quantities for a complex port are defined as

\[ A = \frac{V + ZI}{2\sqrt{R}} \quad B = \frac{V - Z^*I}{2\sqrt{R}} \] (13)

in the frequency domain, where \( Z \) and \( R \) are a port impedance and its resistance part, respectively. These wave quantities directly lead to the incident power \( P_A \) and the reflected power \( P_B \),

\[ P_A = AA^*, \quad P_B = BB^*. \] (14)

Therefore coefficient multipliers of a building block in a power wave digital filter can be easily quantized to guarantee passivity, \( P_A - P_B \geq 0 \).

The definition (13) originates from voltage dimensional waves (16). Other power waves (15) based on current dimensional waves (17) are also available.

\[ A = \frac{I + YV}{2\sqrt{G}} \quad B = \frac{I - Y^*V}{2\sqrt{G}} \] (15)

\[ Y \] and \( G \) are a complex port admittance and its conductance part.

\[ A = \frac{V + ZI}{2} \quad B = \frac{V - Z^*I}{2} \] (16)

\[ A = \frac{I + YV}{2} \quad B = \frac{I - Y^*V}{2} \] (17)

If the number of the independently quantized multipliers is equal to the number of the freedom in a reference network, the losslessness is exactly reproduced, even after the multipliers are quantized. If all the entries of a scattering matrix are purely real-valued, the corresponding adaptor is exactly lossless even after quantization in the same reason as a real wave adaptor. A restricted condition for such losslessness is known as 'one realness' [7], which excludes complex impedance under the evaluation (12). Although the networks introduced in 3. do not possess this property, at least passivity after quantization can be guaranteed easily by using power wave quantities. But in general, a power wave filter needs a greater number of multipliers in comparison with a voltage dimensional wave filter or a current dimensional wave filter because of the nonlinear relationship of the entries of each scattering matrix [6].

Even not one real, if there exist a reflection free port, a series adaptor based on voltage dimensional waves (16) or a parallel adaptor based on current dimensional waves (17) holds losslessness after quantization. In ad-

--- 5th order asymmetrical filter.

--- 9th order symmetrical filter.

Figure 5: Design example.

As a result, starting from a normalized low pass response with pass band \( \lambda = -j \sim j \), the whole transformation can be expressed as,

\[ \lambda \rightarrow \frac{1}{m}(\lambda - j\alpha - j\lambda). \] (11)

The parameter \( m \) describes the pass band expansion of normalized low pass response.

If there is any need to digitally simulate the structures of the prototype networks, such as a multiport filter, complex wave digital filters become useful.

4. Wave quantities and adaptors

Wave digital filters are digital filters which simulate LCTL networks in the sense of incident and reflected wave quantities [6]. An individual reflectance is expressed as a unimodular multiplier. An element interconnection is simulated by a digitally realized scattering matrix, called an 'adaptor'. A wave digital filter is sufficiently stable, if all the building blocks are lossless or passive.

A impedance value for a port, which is evaluated under a condition

\[ \alpha = \tan \frac{\Omega x}{2}. \] (10)

is called 'port impedance'.

Complex power wave quantities for a complex port are defined as

\[ A = \frac{V + ZI}{2\sqrt{R}} \quad B = \frac{V - Z^*I}{2\sqrt{R}} \] (13)

in the frequency domain, where \( Z \) and \( R \) are a port impedance and its resistance part, respectively. These wave quantities directly lead to the incident power \( P_A \) and the reflected power \( P_B \),

\[ P_A = AA^*, \quad P_B = BB^*. \] (14)

Therefore coefficient multipliers of a building block in a power wave digital filter can be easily quantized to guarantee passivity, \( P_A - P_B \geq 0 \).

The definition (13) originates from voltage dimensional waves (16). Other power waves (15) based on current dimensional waves (17) are also available.

\[ A = \frac{I + YV}{2\sqrt{G}} \quad B = \frac{I - Y^*V}{2\sqrt{G}} \] (15)

\( Y \) and \( G \) are a complex port admittance and its conductance part.

\[ A = \frac{V + ZI}{2} \quad B = \frac{V - Z^*I}{2} \] (16)

\[ A = \frac{I + YV}{2} \quad B = \frac{I - Y^*V}{2} \] (17)

If the number of the independently quantized multipliers is equal to the number of the freedom in a reference network, the losslessness is exactly reproduced, even after the multipliers are quantized. If all the entries of a scattering matrix are purely real-valued, the corresponding adaptor is exactly lossless even after quantization in the same reason as a real wave adaptor. A restricted condition for such losslessness is known as 'one realness' [7], which excludes complex impedance under the evaluation (12). Although the networks introduced in 3. do not possess this property, at least passivity after quantization can be guaranteed easily by using power wave quantities. But in general, a power wave filter needs a greater number of multipliers in comparison with a voltage dimensional wave filter or a current dimensional wave filter because of the nonlinear relationship of the entries of each scattering matrix [6].

Even not one real, if there exist a reflection free port, a series adaptor based on voltage dimensional waves (16) or a parallel adaptor based on current dimensional waves (17) holds losslessness after quantization. In ad-

--- 5th order asymmetrical filter.

--- 9th order symmetrical filter.

Figure 5: Design example.

As a result, starting from a normalized low pass response with pass band \( \lambda = -j \sim j \), the whole transformation can be expressed as,

\[ \lambda \rightarrow \frac{1}{m}(\lambda - j\alpha - j\lambda). \] (11)

The parameter \( m \) describes the pass band expansion of normalized low pass response.

If there is any need to digitally simulate the structures of the prototype networks, such as a multiport filter, complex wave digital filters become useful.
Figure 6: Proposed ladder realization.

Fig.6 can be performed by a complex multiplier for each direction because both of the definitions are linear combinations of the voltage and the current of a port. Including the conversion multipliers, such wave digital filter needs a smaller number of multipliers than a power wave filter, and its stability is easily guaranteed by simple truncations of conversion multipliers while all the resistant part of port impedance is positive. All the adaptors in Fig.6 have the same structure. Instead of an adaptor with no reflection free port, a combination of a matched adaptor and a quadruple pair of multipliers are used.

5. CONCLUSIONS

Properties of complex digital filters based on complex transmission line networks are shown. The described networks and the equivalent wave digital filters can accept a new frequency transformation and suitable to perform simulations of complex systems. For modularity, new ladder structure is introduced.

REFERENCES


HINTS ON THE RELATIONSHIPS BETWEEN SUBBAND CODING, WAVELET AND VAGUELETTE DECOMPOSITIONS AND COSINE SINE ALTERNED TRANSFORMATIONS

Stephane MAES (2)

Université Catholique de Louvain, Unité FYMA, Unité TELIS, chemin du Cyclotron, 2. R-1349 Louvain-la-Neuve, Belgium

The equivalence between subband coding and wavelet-vaguelette decompositions is described. For this purpose, CSAT are introduced. The approach is suitable to all perfect reconstruction subband decompositions and to hybrid subband coding schemes. Hybrid DCT-subband coding is also presented for optimal decompositions in time and frequency.

1. INTRODUCTION

The wavelets have been introduced by Norlet and Grossmann [1,2] for efficient analyses of analog signals. Subband coding schemes [3] were already widely used in signal processing. Different filter bank design have been developed for perfect (or almost perfect) reconstruction [4]. Quatadrature Mirror Filters (QMF) are, by definition, filter banks adequate for perfect reconstruction subband (PR) coding schemes. Mallat [5] has shown that discrete wavelet transforms with a dilation factor equal to 2 are hierarchical subband coding decomposition (HSD) with QMF (conjugate QMF); i.e., the decomposition filters are the conjugates of the reconstruction filters. Daubachies [6] has studied all the FIR-QMF filters in one dimension.

Non-hierarchical decomposition (uniform, best basis) are also well known in signal processing. A mathematical formulation (the wavelet packets) has been presented by Wickerhauser [7]. It is a straightforward extension of Mallat's theory.

The purpose of this paper is to show that any PR-HSD is a wavelet decomposition. Our approach requires the introduction of the CSAT (Cosine Sine Alterned Transformation) and it may be extended to analyses with a rational dilation factor. We call those decompositions "vaguelette analyses". It is a generic term which embodies all the basis (or tight frames) functions associated as coherent states [1] with PR subband coding schemes.

2. WAVELET EXTRACTION

In this section, we are concerned by analyses with an integer dilation factor. We present an original construction different from Cohen's approach [8]. The conclusions are similar for the Smith-Barwell class of filters (see [4] for classification); HSD are orthogonal decompositions into biorthogonal wavelets. However the approach is also suitable for all the lattice filter banks [4] and hybrid subband coding schemes [9].

2.1. The Smith-Barwell Class

Let us consider the Smith-Barwell class [4] which includes CQMF. The dilation factor is chosen equal to 2. The decomposition filters are $H_2^1(\omega)$ and $G_2^1(\omega)$. The reconstruction filters are $H_2^2(\omega)$ and $G_2^2(\omega)$. They are chosen to cancel at reconstruction the aliasing errors which have been introduced by down-sampling. In order to have a PR-HSD, we impose:

$$\sum_{k=-\infty}^{\infty} |H_1^2(\omega + 2\pi k)|^2 \chi(\omega) = \chi(\omega) \quad (1)$$

where $\chi(\omega)$ is the characteristic function of $[-\pi, \pi]$);

$$|H_1^2(0)| = 1 \quad (2)$$

(1) ensures the validity of a HSD for any signal with a frequency band lower than $\omega$. It is equivalent to Cohen's (P)-condition [6]. (2) ensures a minimal discrimination between the low and high frequency subbands obtained after decomposition.

For the Smith-Barwell Class, it is also mandatory to introduce the power complementarity:

(2) Research Assistant of the FNRS (National Fund for Scientific Research, Belgium).
\[
|H_1^*(\omega)|^2 + |G_1^*(\omega)|^2 = 1 \tag{3}
\]

Without (3) it would not be possible to relate easily PR-HSD and wavelet orthogonal transform:
We introduce the CSAT as defined by Yu. Meyer [10].
(3) implies that if we consider the filter bank
\[
B^* = \left\{ H_1^*(\omega), G_1^*(\omega) \right\}
\]
we may associate four different spaces:
\[
\mathcal{V}_- = \text{span} \left\{ \frac{1}{\sqrt{n}} H_1^*(\omega), \sqrt{2} H_1(\omega) \right\}
\]
\[
\text{cos } k (\omega + \frac{n}{2})
\]
\[k = 1, 2, \ldots \]
\[
\text{span } 1 \text{ stands for "generated by } 1 \text{ in the frequency domain"}.
\]
\[
\mathcal{W}_- = \left\{ \frac{1}{\sqrt{n}} H_1(\omega) | H_1^*(\omega)|, \right\}
\]
\[
\text{sin } k (\omega + \frac{n}{2})
\]
\[k = 1, 2, \ldots \]
\[
\mathcal{V}_- \text{ and } \mathcal{W}_- \text{ are similarly obtained: sine are associated with } |H_1^*(\omega)| \text{ and cosine (as well as the constant function) are associated with } [G_1^*(\omega)].
\]

If we define \( \mathcal{V}_0 \) as the space generated by the low frequency coherent states of the HSD [11], then it results from [10] that we may write:
\[
\mathcal{V}_0 = \mathcal{V}_-^{\downarrow} \oplus \mathcal{W}_-^{\downarrow} = \mathcal{V}_-^{\downarrow} \oplus \mathcal{W}_-^{\downarrow}
\]
By definition of \( \mathcal{V}_0 \), \( K = \prod_{k=2}^{\infty} |H_1^*(\omega/2^k)| \) is an isometry on \( \mathcal{V}_0 \). Therefore:
\[
\mathcal{V}_0 = KV_-^{\downarrow} \oplus KW_-^{\downarrow} = KV_-^{\downarrow} \oplus KW_-^{\downarrow}
\]
\[
\mathcal{V}_0 = KV_-^{\downarrow} \oplus KW_-^{\downarrow}
\]
\[\text{i.e. :}
\]
\[
\mathcal{V}_0 = \mathcal{V}_-^{\downarrow} \oplus \mathcal{W}_-^{\downarrow}
\]
\[
\text{For example, if we define } \phi(\omega) = \prod_{k=1}^{\infty} |H_1^*(\omega/2^k)|,
\[
\text{we have } \mathcal{V}_-^{\downarrow} = \text{span} \left\{ \frac{1}{2^n} \phi(2^n), \frac{1}{2^n} \phi(2^n), \right\}
\]
\[
\text{span } k \omega \text{ with } k = 1, 2, \ldots
\]
\[
\text{with } T_k x = e^{i k x}. \text{ span } 1 \text{ means the same as span } 1 \text{ but in the time domain. We have a similar expression for } \mathcal{V}_-^{\downarrow}
\]

The whole approach is easy to repeat for any integer dilation factor and for any type of decomposition (uniform best basis). For rational dilation factors it is not possible to go farther than (4): the HSD decomposes \( \mathcal{V}_0 \) into \( \mathcal{V}_-^{\downarrow} \) and \( \mathcal{W}_-^{\downarrow} \).
\[
\mathcal{V}_-^{\downarrow} \ominus \mathcal{W}_-^{\downarrow} = \mathcal{V}_0 \ominus \mathcal{V}_-^{\downarrow}, \mathcal{W}_-^{\downarrow}
\]
\[
\mathcal{V}_-^{\downarrow} \ominus \mathcal{W}_-^{\downarrow}
\]
\[\text{It is impossible to extract wavelet bases out of (5) and (6): orthogonal bases are not "coherent" bases. This is related to the non commensurability between } \mathcal{Z} \text{ networks and } \mathcal{A} \text{ networks with } \phi \in \mathcal{Q}, \mathcal{Z}. \text{ It is in agreement with Auscher's results} [12].
\]
2.2 Lattice filters and hybrid subband decompositions

In [4] other filter banks are introduced: the lattice filters. Most of them belong to the Smith-Barnwell class and have already been treated. Others are for example non power complementary filters. To have a PR, we impose
\[ |H_1^\ast(\omega)|^2 + |G_1^\ast(\omega)|^2 = (A^{-1}(\omega))^2 > 0 \]

AFB = \( \begin{pmatrix} A(\omega) & H_1^\ast(\omega) \\ H_1(\omega) & A(\omega)G_1^\ast(\omega) \end{pmatrix} \) belongs to the Smith-Barnwell class. Therefore, we can reproduce the analysis. \( V_0, V_{-1}, V_{-1} \) are the spaces associated to AFB. If \( F_\psi \) denotes the orthogonal projection on \( V_\psi \), we have: \( f = V_0 \sum_{n=1}^\infty (A^{-1}f) + P_\psi (A^{-1}f) \). \( \phi \) and \( \psi \) of AFB are also the vaquesettes of this RSD. (See figure 2)

If \( A^{-1} = [L A^{-1} / \psi^2] \) is reversible, we may also consider the non orthogonal vaquesettes \( A_{\lambda}^{-1} \phi \) and \( A_{\lambda}^{-1} \psi \).

It introduces hybrid subband decompositions where the filter banks are classical and a reversible transformation \( (A^{-1}) \) is applied on the resulting subband at each step of decomposition. For example, DCT-subband hybrid best-basis decomposition (i.e. \( A^{-1} = \text{DCT} \)) leads to optimal decomposition in time and frequency: decomposition of the signal or of its DCT is decided, provided that it reduces the number of associated bits. This is illustrated in figure 3. DCT is not only an excellent decorrelator, it is also a practical way to switch from the time domain to the frequency domain. The construction presented hereabove also emphasize the duality that exists between subband coding and DCT.

2.3 Other classes

In [4], "almost perfect reconstruction" filter banks obtained by different design methods are also presented. Those analyses have to be understood as approximations of the vaquesette decomposition associated to the closest (in \( L^2 \)) PR filter bank.

3. CONCLUSIONS

We have shown that any PR (hybrid) subband coding is in fact a vaquesette decomposition. Studies of the associated vaquesettes allow to better understand the subband coding performance (regularity, stability, symmetry, locality...). Second generation subband coding schemes (with edge characterization, subband prediction, non uniform sampling,...) especially require such a background theory.

ACKNOWLEDGMENTS

We want to thank all the members of TELE and FYHA especially Professor J.P. Antoine and Professor P. Delogne, for their support.

REFERENCES


Prime Factor Real-Valued Fourier, Cosine and Hartley Transforms

Shing-chow Chan* and Ka-leung Ho**

*Department of Electronic Engineering, City Polytechnic of Hong Kong, Hong Kong.
**Department of Electrical & Electronic Engineering, University of Hong Kong, Hong Kong.

Abstract: Prime Factor Algorithm is an efficient algorithm for computing DFTs of lengths which are co-primes. However, for real-valued input (RPFA), the indexing is very complicated. In this paper, we examine and solve these problems for the vertical-split FFT algorithms including the RPFA. Two FORTRAN programs implementing the RPFA-I and RPFA-II, together with their timing results on the VAX 8820 computer are presented. The approach is then applied to the Discrete Cosine and Hartley transforms to obtain new PFA-DCT and PFA-DHT algorithms. The PFA-DCT is particularly useful in the realization of multirate modulated filter banks and transmultiplexers.

I. Introduction

Prime Factor Algorithm [1,2] is an efficient algorithm for computing DFTs of lengths which are co-primes. However, for real-valued input (RPFA), the indexing is very complicated [3]. In this paper, we examine and solve these problems for the vertical-split FFT algorithms including the RPFA. Two FORTRAN programs implementing the RPFA-I and RPFA-II, together with their timing results on the VAX 8820 computer are presented. The RPFA-II which is usually believed to be faster is found to be inferior to the RPFA-I if indexing problems are taken into account.

Yang and Narasimha [10] have demonstrated that it is possible to obtain a PFA-DCT from the PFA FFT through the mapping of Narasimha and Peterson, and Makhoul. Recently, Lee [11] has presented a direct derivation of the algorithm and suggested to implement the input and output mapping in form of tables. The indexing is very complicated and a general method for obtaining the short DCT modules is lacking. Using the general approach for the RPFA, a unified treatment of the prime factor decomposition of DCT (PFA DCT) is presented in this paper which avoids all these shortcomings. By similar development, a prime factor discrete Hartley transform (PFA DHT) has also been derived.

II. Prime Factor Real-Valued FFT

The PFA decomposes a length-N (=N_1, N_2, ..., N_m & are co-primes) DFT into a m-dimensional DFT (MDFT):

\[ X(k_1, ..., k_m) = X(<k, M_1 + ... + k, M_1 >) \]

\[ = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \cdots \sum_{n_m=0}^{N_m-1} x(n_1, ..., n_m) W_1^{n_1 k_1} W_2^{n_2 k_2} \cdots W_m^{n_m k_m} \quad (1) \]

\[ x(n_1, ..., n_m) = X(<r, M_1 + ... + r, M_1 >) \]

using the Prime Factor Mapping \[ n_1 < r, M_1 + ... + r, M_1 > \] \[ k_1 < r, M_1 + ... + r, M_1 > \]

where \[ M_i = N_i / N \] and \[ \sum_{i=1}^{m} M_i = 1 \] \(<r> \) means mod N.

TheMDFT is computed in a row-column manner using the Winograd short DFT modules. At the 1st stage:

\[ X_1^{(k_1, k_2, n_1, n_2, ..., n_m)} = \frac{1}{N_1} \sum_{n_1=0}^{N_1-1} X^{(n_1)}(k_1, k_2, n_1, n_2, ..., n_m) W_1^{n_1 k_1} \quad (3) \]

\[ X^{(n_1, ..., n_m)} = X(k_1, ..., k_m); \quad X^{(n_1, ..., n_m)} = x(n_1, ..., n_m) \]

If the multidimensional indices are arranged in lexicographically, the corresponding DFT matrix W_N can be written as:

\[ W_N = (B_N \otimes B_{N_1}) \otimes ... \otimes (B_{N_m} \otimes B_{N_1}) \quad (4) \]

where W_N is the real DFT modules, B_{N_i} is the combination matrix, and \( \otimes \) is the Kronecker matrix product. The vertical split FFT can be obtained by appropriate permutating the factors in (4). The two RPFA are given by:

PRFA-I: \[ W_N = (B_{N_1} \otimes ... \otimes B_{N_1}) \quad (5) \]

PRFA-II: \[ W_N = (B_{N_1} \otimes ... \otimes B_{N_1}) \quad (6) \]

If the multiplications are nested together, we obtain two similar real-valued Winograd Fourier transform algorithms (RWFTA).

It can be shown [13] that the following conjugate symmetry exists at each stage of the algorithm

\[ X^{(k_1, ..., k_m)} = \left[ X^{(n_1, ..., n_m)}(k_1, ..., k_m) \right]^* \quad (7) \]

\[ 0 \leq k_1 \leq N_1 - 1, \quad j = 1, ..., i \quad x \leq m \]

\[ 0 \leq n_i \leq M_i - 1, \quad j = i+1, ..., m \]

The intermediate symmetry can be used to reduce the arithmetic operations and storage. Since the computation of RPFA is similar to RWFTA, only the former will be discussed here.

III. Indexing and Unscrambling for RPFA FFT RPFA-II:

In RPFA-II, both real and complex small N transforms are required. This real-valued in-place PFA was first introduced in [4] and a more detail formulation was given by Pitas and Strohzeit [5].
The indexing and storage scheme in [5] have two drawbacks:
1) An unscheduler is needed at the end because the real and imaginary parts are stored according to the PFM.
2) The real and imaginary parts are not simply related and considerable modulo operations are required in indexing the data.

Improved indexing and storage scheme can be obtained by storing the real and imaginary parts in locations which are conjugate to each other as shown in Fig. 1.

The application of the efficient indexing scheme in [6,7] to the RPFA-ll has two problems:
1) It is necessary to determine whether the real parts or the imaginary parts are being referenced during the updating.
2) It is necessary to determine whether the data to be transformed is real or complex.

For example, the real and imaginary parts of group i-j at the i-th stage are stored as follows (x is the array of storage):

\[ \text{Re}(x_i^{(k_i)}) \rightarrow 2(k_{i-1} \cdots k_1 n_{n_i-1} \cdots n_m) \]
\[ \text{Im}(x_i^{(k_i)}) \rightarrow 2(k_{i-1} \cdots k_1 n_{j-1} \cdots n_m) \]

(8)
(9)

It can be seen that
1) The transform will be real if all the first i indices are equal to k_i.
2) If the first j indices are equal to k_j, and k_j lies within [1, p_j], then the real part of the transform is being referenced. Otherwise, the imaginary part is being referenced.

It is found that the entire implementation can be greatly simplified if we restrict N_i to be the even factor whenever the transform is even.

In order to determine whether the transform is real or complex and determine whether the real or imaginary parts are being referenced, we need to maintain the values of k_g. To avoid excessive number of modulo operations, k_2 is only updated whenever k_2 is equal to k_2.

For group 1, all the data to be transformed are real and we have to perform two real transforms at the same time to ensure in-place operation. In order to reduce the length of the program, the odd-length DFT modules are modified so that they can transform complex, a pair of real transforms and a single real transform. Since the even-length modules, if invoke, must deal with real transform only, their implementation is considerably simplified.

The exchange operation between the real and imaginary part are merged with the last stage of the algorithm.

RPFA-I:

The indexing of the first stage of RPFA-I is exactly the same as the complex PFA except that only real transforms are performed. Though the combination can be carried out by successively applying the combination operators in turn to the intermediate sequence and make use of the complex conjugate symmetry to avoid redundant operations, it is inconvenient and inefficient because the index map has to be set up for each factor. Due to the simple structure of the combination operators, it has been found that the 2^n (less than 2^n) positions for combination of lower order) output points X(n_{i+1} \cdots n_{m}) can be obtained from the 2^n input points x(n_{i+1} \cdots n_{m}) where n_i = n_i or N_i - n_i, and only one pass is required. Thus only one indexing through the data is required.

Without loss of generality, let's consider the combination of the group X(n_i \cdots n_{m}) (dimension 2) as shown in Fig. 2. The procedure starts with the four initial addresses denoted by I and are incremented along the solid line followed by the dotted line. Combination subprograms of different dimensions are written and properly organized to accomplish the entire combination. This is illustrated in Fig. 3 for the RPFA FTT up to four factors. Unscrambling the PFA FTT outputs can be done by permuting the output or input pointers of the modules or by modifying their constants. For the real-valued PFA, the latter two approaches are applicable depending on whether the applications require a fixed-length transform or not. The output permutation approach will introduce a lot of inconvenience. Comparisons of the PFA-FFT, RPFA-I and RPFA-II are shown in Table 1. The RPFA-I is faster than the RPFA-II except for N = 1008, where the timings are very close to each other. The RPFA-II is also faster than the PFA FTT, except for the length N = 30. The speedup is seen to increase with the transform length because of the overheads required to support the storage scheme. For more than one transforms, these overheads can be shared using the pipeline extension technique introduced in [14].

IV. Prime Factor Hartley Transform

Since the DHT kernel is not separable, the PF decomposed DHT cannot be computed by the simple row-column method, as in the PFA FFT. Therefore, it is more appropriate to obtain the PFA DHT via its relation to the DFT. Using the relation:

\[ x(k) = x(k) = x(k) + j H(N-k) \]

and the prime factor decomposition, we obtain:

\[ x(k_{1} \cdots k_{m}) = (1+j) \sum_{n_{1} \cdots n_{m}} x(n_{1} \cdots n_{m}) W_{N_{1} \cdots N_{m}}^{n_{1} \cdots n_{m}} W_{N_{1} \cdots N_{m}}^{n_{1} \cdots n_{m}} \]

\[ + j\text{cas}(2\pi n_{1} k_{1}/N_{1}) \]

\[ \text{cas}(2\pi n_{m} k_{m}/N_{m}) \]

(10)

The equation implies that the Hartley transform can be computed via a hybrid Fourier-Hartley transform as in the power of two cases. The transform is computed in m different stages. At the first stage, we have:
where $D_N$ is a diagonal matrix corresponding to the post-multiplication with $e^{j2\pi nk/N}$. Two PFA-DCT algorithms are given by

**PFA-DCT**^4: $V_N = D_N [Q_{n, m} \cdots \otimes Q_{n, m}^t] [\begin{bmatrix} C_{N, m}^1 \alpha \cdots C_{N, m}^{1 \ell} \end{bmatrix}]$

**PFA-DCT**^8: $V_N = D_N [Q_{n, m} C_{N, m}^{1 \ell} \cdots \otimes Q_{n, m}^{1 \ell}]$ (18)

The factor $\phi_{\text{radix}}$ arises from the fact that the terms $\overline{k, M} \cdots \overline{k, M}$ can take values outside $4N$. The meaning of the multidimensional indices is arranged in lexicographical order, then the Kronecker matrix product representation of $V_N$ is:

$$V_N = D_N \left[ V_{m, n} \otimes \cdots \otimes V_{m, n} \right]$$

(17)
prime factor DFT, DHT, and DCT algorithms have been performed. All these algorithms share the same common properties and can be computed with the same general structure. Results on RPPA-I, which is usually believed to be faster, is inferior to RPPA-II if indexing consideration is taken into account. This is demonstrated by a timing comparison on the VAX 8820 computer.

REFERENCES
On the Design of Chebyshev Filters with Complex Specifications

J.S. Mason, A. Terchi and N.N. Chit
University College of Swansea
SWANSEA, SA2 8PP, UK

Two different approaches have been proposed recently offering solutions to complex Chebyshev filter design. The first is based on a Remez exchange algorithm, and the second is based on a double adaptive system identification algorithm. This paper lends practical evidence to the optimality of solutions from the DAS approach via three classic linear phase examples, designed using a completely general complex function specification and approximation.

1 Introduction

The design of FIR digital filters approximating a complex-valued frequency response in both magnitude and phase arise in circumstances, such as in the design of equalisation networks and frequency selective filters with approximately linear phase of arbitrary slope. There have been a number of proposals for designing such filters, usually based on separate approximations to the two functions, magnitude and phase, and optimising each separately. Considering the two as a complex-valued function, leads naturally to an approximation based on complex polynomials.

Such an approximation, resulting in real-valued filter coefficients, has for a long time lacked a practical solution. Rivlin and Shapiro [1] mention that in general the Chebyshev approximation is a least-squares approximation with appropriate costs. The information required is the error extrema locations and their suitable costs. Lawson [2] proposes a procedure for deriving these values for the complex case but the procedure proved to be impracticable, even after further refinement by a number of researchers.

Very recently however, two very different approaches have been proposed, almost concurrently, both offering solutions to this problem. The first of these is presented by Preuss, [3] [4], who describes an extension of the popular real-valued Remez exchange algorithm. Schulist [5] claims improved convergence of this algorithm, and in the vein of Remez exchange extensions, Alkhairy [6] reports a new algorithm appropriate for the complex case.

The second solution is attributed to the Authors, Chit and Mason, [7] [8] [9]; their double adaptive system (DAS) approach comes from the field of systems identification. The merit here is in its generality: the DAS package is successful in tackling linear-phase, minimum-phase, as well as that of complex specifications.

Both the extended Remez exchange and the DAS approaches are based on iterative procedures approximating the complex function directly, and the resultant approximations are claimed optimum in a complex Chebyshev sense. Here we focus on the DAS approach, which is an LMS procedure within an iterative loop which finds the error extrema and their suitable costs.

Optimality

All Chebyshev optimum solutions have a number of extrema that is has lower and up-
per bounds, [1]. More is known about the characteristics for the real case, and it is far easier to check whether or not a given solution approaches Chebyshev optimality or not. Less is known, however, about the complex case, and thus it is more difficult to be certain that a designed solution has a sufficient number of extrema to be a true Chebyshev approximation. It is not enough to merely derive an equiripple characteristic. An equiripple solution which includes overlapping zeros is an obvious example which is unquestionably sub-optimum in that alternative solution, with the same peak value and an equiripple characteristic but reduced extrema count, always exists.

To give an additional level of confidence that the DAS approach is capable of giving optimum complex Chebyshev approximations we design examples for which the solutions are well documented, namely that with linear phase specifications. We consider the three classic cases of optimal filters, extraripple, scaled ripple and equiripple. These are difficult to achieve in their detail, and as such provide a good test for any design procedure. Obviously it is acknowledged that such examples reduce to a real-valued approximation, and hence can be designed by the standard Remez exchange algorithm. It remains of interest however, to see how close the complex approximation procedure meets the detail of these special cases.

2 The DAS Approach

The DAS approach is shown pictorially in Fig. 1, and full details are given by Chit in [7]. The input signal \( u_n \) to both the realized and the ideal filter, and the desired output signal \( y_n \) are given by

\[
y_n = K \sum_{i=1}^{N_f} C(w_i) G(w_i) \sin(w_i n + \Theta(w_i))
\]

(3.2)

where \( K \) is a scalar controlling the absolute level of the input and output signals and \( N_f \) is the number of frequency domain samples needed to adequately represent the spectrum. The iterative update of \( C \) is designed to give meet both \( G \) and \( \Theta \) in a minimax complex sense.

3 Design Examples

We design three linear-phase filters exhibiting the classic extraripple, scaled extraripple and equiripple characteristics. These filters present a stern task to any design procedure, if for example the subtle differences at the band edges are to be revealed. These three examples are taken from Reference [11] and represent three similar lowpass filters with slightly different transition band edges, giving the three subtly different ripple characteristics.

Example (1):
extraripple, \( f_p = 0.2076 \) and \( f_s = 0.2513 \)

Example (2):
scaled extraripple, \( f_p = 0.2095 \) and \( f_s = 0.25377 \)

Example (3):
equiripple, \( f_p = 0.2280 \) and \( f_s = 0.27551 \)

and for all three the remaining design data are the same:

\[
N \rho = 24, \quad \tau = 12, \quad \delta_p = \delta_s = 0.05.
\]
4 Discussion and Conclusion

The imposition of exact symmetry on the filter coefficients leads to exact linear phase (i.e. zero phase error), and with the DAS algorithm such constraints are very easily imposed, since the algorithm actually models the filter action directly in the time domain. However, here the purpose is to not to resort to such constraints, but to demonstrate convergence towards optimum solutions using a general algorithm, and a general complex specification.

Details of the responses of the designed filters are presented below. It is found that the phase errors tend to zero, the coefficients tend to be symmetrical, and the complex errors tend to the gain errors. Responses of the designed filters are presented below. In all three cases, the maximum deviation is 0.05, as reported originally by [11]. The general unconstrained algorithm, given a specification in a complex form, is therefore capable of producing solutions which are known to be optimum.

References


Fig. 1 Structure of the Double Adaptive Systems Approach. The difference between the ideal and the realised filter is used in both loop adaptations.
Fig. 2 | Complex error and gain response of 24th order LPH filters:
(a & b) extraripple.
(c & d) scaled extraripple.
In both cases |complex error| and gain error tend to 0.05

Fig. 3 | Complex-domain representation within passband for extraripple and scaled extraripple:
(a & c) trace of $H(w)$.
(b & d) trace of $E(w)$. 
Active Noise Cancellation using a Modified Form of the Filtered-X LMS Algorithm

Elias Bjarnason

Institut für Netzwerk- und Signaltheorie
Technische Hochschule Darmstadt
Merckstr. 25, D-6100 Darmstadt, Germany

Abstract

The Filtered-X LMS algorithm [1] is an alternate form of the LMS algorithm for use when there is a transfer function in the auxiliary path following the adaptive filter and/or in the error path. To ensure convergence of the algorithm, the input to the error correlator has to be filtered by a copy of the auxiliary-error path transfer function[3]. A major drawback is that bounds for the stepsize are very difficult to calculate and that the stepsize has to be chosen small to guarantee convergence. A modification is introduced to improve the situation. Simulation results comparing the two algorithms and the results of real time measurements in a duct using a floating point signal processor system are presented.

![Diagram of Active noise cancellation system without feedback](image)

Figure 1: Active noise cancellation system without feedback

1 Introduction

The active control of sound and vibration involves the introduction of a number of controlled "secondary" sources driven such that the field generated by these sources interferes destructively with the field caused by the original "primary" source. The extent to which such destructive interference is possible depends on the geometric arrangement of the primary and secondary sources and their environment, and on the spectrum of the field produced by the primary source [2]. Due to the time-variance of the acoustic path the system has to be adaptive. Fig. 1 shows a simple active noise cancellation system without feedback using a loudspeaker as a secondary source. Here, \( w \) describes the adaptive controller and \( I \) the interpolation filter used to reconstruct the estimated signal \( y_f(n) \).

Fig. 2 shows a model of the active cancellation system. Here, \( h_x \) describes the system in the auxiliary path, modeling anti-aliasing filter, loudspeaker and acoustic path from loudspeaker to microphone. The system in the error path \( h_e \) models the path common to desired and estimated signal, including error microphone and

![Diagram of Active noise cancellation system with feedback](image)

Figure 2: (a) \( h_e \) can be moved through the summing junction to form an equivalent system. (b) \( h_x \) and \( h_e \) can be combined to form a new error path system function \( h_{x_e} \). In the same way \( w_o \) and the inverse system of the auxiliary path \( h_x^{-1} \) can be combined to \( w_o' \), which is the system to be identified.
anti-aliasing filter. Because of the system in the auxiliary path, \( w_o \) has to include the inverse system of \( h_x \) (\( W'_o(z)/H_x(z) \) is the system to be identified). Measurements show (see Fig. 3 and 4) that \( h_x \) can be a nonminimum phase system. This results in making \( w_o \) noncausal and therefore degrading the cancellation effect. The drawback of the non causal parts is decreased due to inherent delay in acoustic system. The filter \( h_x \) does not affect the required model \( w_o \) but it does affect the behavior of the adaptive algorithm as does \( h_x \).

2 The Filtered-X LMS Algorithm

2.1 The General Form

The existence of a filter in the auxiliary and/or the error path has been shown to generally degrade the performance of the LMS algorithm. Thus the convergence rate is lowered, the residual power is increased and the algorithm can even become unstable. In order to stabilize the algorithm an identical filter to the filter \( h_{se} \) in the auxiliary-error path (the combined auxiliary and error path) is used to filter the input \( w(n) \) (see Fig. 5). The recursive relation for updating the tap-weight vector in this case is:

\[
w(n+1) = w(n) + \mu \varphi_f(n)e_f(n),
\]

where \( \varphi_f \) and \( w \) are vectors of dimension \( N \) and:

\[
x_f(n) = x(n) * h_x(n) * h_x(n) = x(n) * h_{se}(n),
\]

\[
y(n) = w(n) * h_x(n),
\]

\[
y_f(n) = y(n) * h_x(n) * h_x(n) = y(n) * h_{se}(n),
\]

\[
d_f(n) = d(n) * h_x(n) = d(n) * h_{se}(n),
\]

\[
e_f(n) = d_f(n) - y_f(n).
\]

It is assumed that \( x(n) \) and \( d(n) \) are jointly stationary and that their first moments are zero.

Defining the weight-error vector at time \( n \) as

\[
\varepsilon(n) = \bar{w} - w(n),
\]

taking the expectation, using the independence assumption, and modeling the auxiliary-error path with a transversal filter \( h_{se} \) of order \( M \) results in:

\[
E[\varepsilon(n+1)] = E[\varepsilon(n)] - \mu \sum_{j=0}^{M-1} \sum_{i=j}^{M-1} h_{se}(i)h_{se}(j) E[\varepsilon(n-j)] E[\varepsilon(n-i)].
\]

Here, bounds for the stepsize have been calculated for the simple case of a delay in the auxiliary-error path \([4,5]\).
For very small $\mu$ the expectation value of the weight-error can be approximated:

\[
E[g(n + 1)] \approx (I - \mu \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} h_{se}(i)h_{se}(j)) E[g(n)] - \mu E[\xi_g(n)] E[g(n)] = (I - \mu R_{\xi_g}) E[g(n)].
\]  

(4)

This is the same equation as for the LMS algorithm with filtered input $x_f(n)$, thus guaranteeing the convergence of the Filtered X-LMS algorithm for very small $\mu$.

### 2.2 The Modified Form

Assuming that the whole system is time invariant, calculating the mean square of the filtered error results in:

\[
E[\xi_g^2(n)] = \sigma_d^2 - w^T P_{d_x} w - w^T P_{d_x} P_{d_y} w + w^T R_{d_x} w.
\]  

(5)

Then differentiating the mean square of the filtered error with respect to the weight vector $w$:

\[
\nabla = \frac{dE[\xi_g^2(n)]}{dw} = 2R_{\xi_g} w - 2P_{d_x} P_{d_y} w.
\]  

(6)

and using the method of steepest ascend, the weight update equation is given by:

\[
aw(n + 1) = aw(n) - \frac{1}{2} \mu \nabla w(n) = aw(n) + \mu P_{d_x} P_{d_y} w(n).
\]  

(7)

Replacing the unknowns with instantaneous estimates of their values results in the modified form of the Filtered-X LMS algorithm:

\[
aw(n + 1) = aw(n) + \mu x_f(n)[d_f(n) - \xi_g^T(n)] w(n).
\]  

(8)

Observing that $d_f(n)$ is not available in the case of active noise cancellation but knowing the filtered error $x_f(n)$ and the auxiliary-error path function $h_{se}$ the modified form (see Fig. 6) of the Filtered-X LMS algorithm can be rewritten to:

\[
aw(n + 1) = aw(n) + \mu x_f(n)[x_f(n) + y_f(n) - \xi_g^T(n)] w(n).
\]  

(9)

In [5] the same kind of modification is introduced for the Delayed LMS algorithm.

For the modified form the expectation of the weight-error vector is:

\[
E[\xi_g(n + 1)] = (I - \mu \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} h_{se}(i)h_{se}(j)) E[\xi_g(n)] - \mu E[\xi_g(n)] E[\xi_g(n)]
\]  

\[= (I - \mu R_{\xi_g}) E[\xi_g(n)].
\]  

(10)

From Equation (10) it can be seen, that the modified form of the Filtered-X LMS algorithm has the same behavior as the LMS algorithm with filtered input $x_f(n)$, except in the case of time variant systems.

### 2.3 Comparison between the General and the Modified Form

Comparing Eq. (3) and (10) it can be seen that in Eq. (3) the weight-error vector update is a function of weighted past values of the weight-error vector, but in Eq. (10) only the actual value of the weight-error vector and thus all information learned from past updates is used. Therefore, the modified form has a higher convergence rate and the same predictable behavior as the LMS with a filtered input. On the other hand the modified form requires a larger amount of computation.
Figure 7: Comparison of the highest convergence rate in each case between the general and the modified form for a white Gaussian input \( N = 50, M = 20, S/N = 20dB \)

\[3N + 2M \] instead of \( 2N + M \) and storage capacity. In Fig. 7 a comparison is shown between the normalized form of the two algorithms. For both algorithms the stepsize was chosen to optimize the dynamic behavior of the algorithm.

Figure 8: Realtime implementation

2.4 Implementation

The approach shown in Fig. 8 was implemented using the DSP32C signal processor from AT&T. For the acoustical system a duct with 40cm diameter and 2m length was used. Both algorithms were implemented in their normalized form, including adaptive modeling of the auxiliary-error path in series or parallel using the NLMS algorithm with independent low-level random source \( \mu(n) \) (see [3]). Typical results for the noise reduction after the adaptation using a broadband noise as input are shown in Fig. 9. The curve was obtained by subtracting the canceled from the uncanceled log spectrum. The minima at about 400 Hz and 800 Hz are due to acoustical resonances of the duct. The cancellation system showed to be effective for broadband noise as well as for narrowband noise.

Figure 9: Noise reduction with an active attenuation system using the normalized modified form of the Filtered-X LMS algorithm

3 Conclusion

A modified form of Filtered-X LMS algorithm has been introduced. This algorithm has the same properties as the LMS algorithm with a filtered input. The advantages of the modified form over the general form are the known bounds for the stepsize and the higher convergence rate. This has to be paid for with increased complexity. A complete active attenuation system has been described and implemented. The system can be operated fully adaptive employing an adaptive filter using the NLMS algorithm with independent low-level random noise source to determine the auxiliary-error path transfer function [3], and using the general or the modified form of the Filtered-X NLMS algorithm for the active noise cancellation.

References


A PIPELINED ADAPTIVE LATTICE FILTER
ARCHITECTURE: THEORY AND APPLICATIONS*

Naresh R. Shanbhag
Keshab K. Parhi

Dept. of Electrical Engineering
University of Minnesota
Minneapolis, MN 55455

Abstract

The stochastic gradient adaptive lattice filter is pipelined by the application of the relaxed look-ahead. This form of look-ahead maintains the functional behaviour instead of the input-output mapping. The sum and product relaxations are employed to pipeline the lattice filter. The hardware complexity of the proposed pipelined filter is of the same order as the sequential filter. Thus, the new architecture is attractive from an implementation point of view. Convergence analysis results are presented to illustrate the trade-off offered by relaxed look-ahead. Application of the lattice filter to pipelined video predictive coding is demonstrated.

1 Introduction

Algorithm transformation techniques [1], for increasing the concurrency in digital signal processing architectures, have been well studied. Pipelining [2] and parallel processing [3] are two major techniques for developing high-speed digital signal processing architectures. Pipelining of fixed-coefficient filters via the look-ahead computation [2] has already been developed and has also been applied to pipeline adaptive lattice filters [4, 5]. The conventional look-ahead technique transforms a given serial algorithm into an equivalent (in the sense of input-output mapping) pipelined one. This, however, results in an enormous increase in the hardware complexity. In case of adaptive filters, this complexity is even higher due to the coefficient-update loop. For pipelining adaptive filters, it is possible to give up this exact input-output mapping if the average convergence behavior is not changed substantially. In this paper, we use a relaxed form of look-ahead as opposed to the conventional look-ahead for pipelining the stochastic gradient lattice adaptive filter [6]. The relaxed look-ahead is an approximation to the conventional look-ahead which does not alter the convergence behavior substantially.

The relaxed look-ahead has been successfully employed to develop a pipelined LMS adaptive filter architecture [7] and the adaptive pulse-code modulation codec [8]. In this paper, we choose a particular stochastic gradient lattice adaptive algorithm [6] to pipeline via this technique. Note that relaxed look-ahead can be employed to pipeline any of the adaptive lattice algorithms in [6].

A brief discussion of the conventional look-ahead and its relaxed form is provided in section 2. In section 3, we derive the pipelined stochastic gradient lattice architecture (PIPSGLA). In section 4, we discuss its convergence properties. In section 5, a pipelining example and image compression using pipelined adaptive pulse code modulation (ADPCM) is demonstrated.

2 The Relaxed Form of Look-ahead

We first illustrate the conventional look-ahead [2] with the help of an example. Consider the following first-order recursion,

$$x(n+1) = ax(n) + u(n),$$

where $x(n)$ is the output, $u(n)$ is the input and $a$ is a constant. Iterating (2.1) $M_1$ times, we get

$$x(n+M_1) = a^{M_1}x(n) + \sum_{i=0}^{M_1-1} a^i u(n+M_1-1-i).$$

Equation (2.2) represents the application of look-ahead to (2.1). Due to the nature of the transformation both (2.1) and (2.2) represent the same system. However, (2.2) can process data $M_1$ times faster than (2.1). This is because the $M_1$ latches introduced into the system can be used to pipeline the multiply-add operation. The second term in (2.2) represents the extra hardware overhead necessary. This overhead can be quite expensive for higher order systems and for fine-grained pipelining (i.e. large values of $M_1$). However, if (2.1) were an adaptation algorithm, then we may apply the sum relaxation to write the second term in (2.2), as follows

$$x(n+M_1) = a^{M_1}x(n) + M_1 u(n+M_1-1).$$

The system described by (2.3) would be a close approximation of the system in (2.2) if $a$ is close to unity and

---

* This research was supported by the army research office by contract number DAAL03-90-G-0063.
u(n) is approximately constant. The $a^{M_i}$ term in (2.3) can be precomputed if $a$ is a constant. If $a$ in (2.3) is time-varying (i.e. we have $a(n)$ instead of $a$), but the magnitude of $a(n)$ is close to unity, then we replace $a(n)$ by $(1 - a'(n))$, where $a'(n)$ is close to zero. Then, we apply the product relaxation to write (2.3) as

$$x(n+M_L) = (1-M_La'(n))x(n)+M_Lu(n+M_L-1). \tag{2.4}$$

Thus, (2.3) and (2.4) represent possible application of relaxed look-ahead to (2.1). Depending on the application at hand, different types of approximations may be employed. Unlike (2.2), (2.3-2.4) are not equivalent to (2.1) with respect to its input-output mapping. Therefore, in case of adaptive algorithms, convergence analysis of (2.3-2.4) needs to be done. In general, the relaxed form of look-ahead is not an algorithm transformation technique in the usual sense [1]. This is because it modifies the algorithm to which it is applied. However, it can be considered an algorithm transformation technique in the stochastic sense if the average output profile is maintained.

3 The PIPSGLA Architecture

As usual, the indices $n$ and $m$ denote the time instance and the lattice stage number, respectively. The serial stochastic gradient lattice algorithm (SSGLA) chosen for pipelining is described as follows [6]

$$k_m(n+1) = [1 - \beta_m(n)P(n|m-1)]k_m(n) + 2\beta_m(n)Q(n|m-1) \tag{3.1}$$

$$S(n+1|m-1) = (1 - \beta)nS(n|m-1) + P(n|m-1) \tag{3.2}$$

$$\beta_m(n) = \frac{1}{S(n|m-1)} \tag{3.3}$$

$$e_f(n|m) = e_f(n|m-1) - k_m(n)e_f(n-1|m-1) \tag{3.4}$$

$$e_o(n|m) = e_o(n-1|m-1) - k_m(n)e_f(n|m-1) \tag{3.5}$$

where

$$P(n|m) = e_f^2(n|m) + e_o^2(n-1|m) \tag{3.6}$$

$$Q(n|m) = e_f^2(n|m) + e_o^2(n-1|m) \tag{3.7}$$

The reflection coefficient $k_m(n)$ is updated according to (3.1), while the input power estimate $S(n|m)$ is updated using (3.2). Thus, (3.1) and (3.2) are the two recursive loops in the algorithm. The equations (3.3) and (3.4) are the order-updates. The SSGLA architecture in Fig.1 shows intermediate outputs $e(n|m)$, which would be employed to develop an ADPCM codec. Critical path analysis of Fig.1 shows a computation time ($T_c$) of

$$T_c = 4T_m + (N+2)T_a \tag{3.8}$$

where $T_m$ and $T_a$ are two operand multiply and add times. The loop computation time $T_L$ is

$$T_L = T_m + T_a \tag{3.9}$$

The pipelining of SSGLA is done by first inserting interstage pipelining latches. This step is trivial as the stages are connected in a non-recursive fashion. If the desired speed-up is not achieved, then the second recursive loop needs to be pipelined. Let $M_S$ and $M_L$ represent the number of interstage and loop pipelining latches, respectively. To do this we apply relaxed look-ahead to (3.1) and (3.2). Like (2.1), (3.2) is a first order recursion. Therefore, applying the sum relaxation (see (2.3)) we can simply write down the final equation by inspection

$$S(n'|M_L) = (1 - \beta)^{M_L}S(n'|m-1) + P(n'|m-1) \tag{3.10}$$

where $n' = n - M_S$. Similarly, we can show that application of the product and sum relaxations (see (2.4)) to (3.1) results in the following

$$k_m(n'+M_L) = [1 - M_L\beta_m(n')P(n'|m-1)]$$

$$k_m(n') + 2M_L\beta_m(n')Q(n'|m-1) \tag{3.11}$$

Note that the loop multiplier coefficients in (3.1) and (3.2) are close to unity. Therefore, the form of relaxation used to derive (3.10) and (3.11) is justified. The rest of the equations describing PIPSGLA are identical to SSGLA and hence are not provided.

The complete PIPSGLA is shown in Fig.2, where it can be seen that the hardware increase is two multipliers per stage and the pipelining latches (see Fig.2). This is remarkably lower than that obtained by conventional look-ahead [4, 5].

4 Convergence Analysis

The convergence analysis, for a single lattice stage, was carried out under the independence assumptions [6]. These assumptions become more true as the number of lattice stages increases. The inputs to the stage under
consideration is assumed to be stationary. The results of this analysis are presented below. The bounds on $\beta$ for convergence of weights is

$$0 \leq \beta \leq \frac{2}{M_L}, \quad (4.1)$$

which is tighter than that of SSGLA (for which the upper bound is 2). However, for most applications the value of $\beta$ is smaller than the upper bound.

The convergence time constant for PIPSGLA ($\tau_{\text{PIP}}$) equals

$$\tau_{\text{PIP}} = \frac{1}{2\beta M_L} \quad (4.2)$$

and this value is $1/M_L$ times that of SSGLA. This is to be expected as (3.11) has an effective adaptation constant which is $M_L$ times that of SSGLA in (3.1).

The misadjustment for PIPSGLA ($\mathcal{M}_{\text{PIP}}$) is given by

$$\mathcal{M}_{\text{PIP}} = \frac{M_L \Gamma(k_{\text{opt}}^2)}{(2 - M_L \gamma)} \quad (4.3)$$

where $\gamma = 1 - (1 - \beta)M_0$. Assuming $\beta$ is much smaller than unity, (4.3) implies that $\mathcal{M}_{\text{PIP}}$ is $M_L^2$ times that of SSGLA.

5 Examples

In this section, we first present a pipelining example to demonstrate the increase in speed. Then, we employ image coding using ADPCM to compare the performance of SSGLA and PIPSGLA.

5.1 Pipelining Example

To illustrate the speed-up due to pipelining, we assume $T_e = 20$, $T_m = 40$ and the filter order $N = 2$. Therefore, from (3.8), it follows that $T_e = 240$. For a speed-up of $M = 6$, the clock-period of PIPSGLA should be 40. This can be achieved with $M_L = 2$ and $M_S = 6$. Retiming these latches (Fig.3) results in the desired clock-period.

5.2 Image Coding

A pipelined ADPCM codec is developed by employing PIPSGLA as a predictor [6]. The pipelined ADPCM coder architecture is shown in Fig.4, where $N_{\text{cols}}$ is the number of pixels in a row of the image. The predictor output is formed by summing the $e(n,m)$ outputs of each of the lattice stages $L_0, L_1$ and $L_2$. The input is in a row-by-row raster scan format. The $N_{\text{cols}} - M_S - 1$ latches at the input of the lattice predictor are for enabling prediction of the current pixel with the three nearest pixels from the previous line.

The input (Fig.5) is a 256 x 256 frame, 8-bit per pixel (bpp) with 256 gray levels. The quantizer has 3 bits. The adaptation constant $\beta$ was 0.009 for both the serial and pipelined architectures. The pipelined architecture (Fig.4) had $M_S = 35$ and $M_L = 3$, which corresponded to a speed-up of 12. The reconstructed images for the serial (Fig.6) and the pipelined architectures (Fig.7) show no perceptual differences. The signal-to-noise ratio (SNR) for the serial architecture was 23.39 dB, while that of the pipelined architecture was 23.33 dB.

Thus, we see that large speed-ups are achievable without any appreciable degradation in the convergence behavior.

6 Summary and Conclusions

The stochastic gradient lattice filter has been pipelined via the application of relaxed look-ahead [7]. The relaxed look-ahead results in an enormous savings in hardware. As it is not a one-to-one mapping like its deterministic
counterpart, the relaxed look-ahead offers a multitude of architectural possibilities. The PIPSSGLA is one such choice but the user may select any other form of relaxation depending on the application at hand.

References


BLIND DECONVOLUTION OF MULTIVARIATE SIGNALS USING ADAPTIVE FIR LOSSLESS FILTERS

Philippe LOUBATON*, Philip A. REGALIA**

* Ecole Nationale Supérieure des Télécommunications, Département Signal, 46 rue Barrault, 75234 PARIS CEDEX 13, FRANCE. ** Institut National des Télécommunications, Département Electronique et Communications, 9 rue Charles Fourier, 91011 EVRY CEDEX, FRANCE.

The problem of adaptive blind deconvolution of multivariate signals is considered. A deflation procedure is proposed. First, a classical contrast is maximized in order to extract the first component of the p-variate excitation. Next, an adaptive FIR lossless filter produces a (p-1)-variate signal on which the preceding algorithms can be applied to extract the other components of the excitation.

1. INTRODUCTON.

In a number of applications, a certain scalar stationary random process y = (y(n))eZ (the observation) is the output of an unknown filter driven by an independent identically distributed (iid) sequence of random variables w = (w(n))eZ representing the useful signal. The blind deconvolution of y consists of extracting the excitation w from y, or equivalently of identifying the inverse of the transfer function h(z) of the unknown filter. As it is well known, this problem does not make sense when the signal w is Gaussian, except in the case where h(z) is assumed to be minimum (or maximum) phase. By contrast, when w is assumed to be non Gaussian, it is possible to establish under very mild technical assumptions ([12]) that the representation of y as y(n) = [h(z)].w(n) is unique up to time delays and scaling factors.

Therefore, the adaptive blind deconvolution of non Gaussian scalar signals makes sense, and it has been investigated in numerous works (e.g. [1], [4], [7]). The concept of contrast (called in [4] objective function) plays a central role in the above mentioned question. A contrast is a function  in defined on an appropriate set  of random variables, which satisfies the following requirements: if xe , then  depends only on the probability distribution of x ;  for all  and xe ; if (x) are independent copies of xe , then  with equality if and only if (if)  turns out that a possible approach to perform the adaptive deconvolution of y is to adapt a filter g(z) so as to maximize  for some contrast  One of the simplest implementation is obtained when  is given by

\[ \Psi(x) = \frac{c_4(x)}{E(x^2)^2} \]  \hspace{1cm} (1-1)

where  denotes the fourth-order cumulant of x (i.e.  if  = 0). In fact, by using a classical whitening of the observations, it is possible to assume without restrictions that (y(n))eZ is a unit variance uncorrelated sequence. Then, if the excitation w is known to have a strictly negative (resp. strictly positive) fourth-order cumulant, the maximization of  is equivalent to the minimization (resp. the maximization) of  under the constraint  is 1. The key point is that this problem does not possess spurious local extrema [7], so that it is possible to use a stochastic gradient algorithm in order to adapt a unit norm filter so as to minimize or maximize  However, in practice, the optimization has to be performed in the class of FIR filters of fixed degree N, and it is known that is not large enough, then the previously introduced extremum problem may exhibit spurious local extrema.

In this paper, we address the above adaptive blind deconvolution problem, but in the case where the observation is a p-variate signal  which can be written as

\[ Y(n) = \sum_{k=1}^{p} [H_k(z)].W_k(n) \]  \hspace{1cm} (2-1)

where the  are unknown p×1 transfer functions and where the excitations  are statistically independent non Gaussian centered iid sequences of variance 1. To our knowledge, this problem has not yet been considered in previous works. However, it should be noted that some recent works devoted to the separation of statistically independent narrow band sources by means of a sensor array (e.g.[2], [3]) are directly related to the multivariate deconvolution problem considered here, but in the very special case where the unknown transfer function  is supposed to be a constant matrix. In the frame of this paper, we shall extensively use the contrast  defined by (1-1), so that we shall assume that the fourth-order cumulants (c_4(W_k))_{k=1,p} of the excitations are non
zero, and for the numbers of positive and negative cumulants are known; for ease of exposition, we shall suppose that the \((c_k(W))_{k=1}^p\) are strictly negative, and that the \((c_k(W))_{k=p+1}^p\) are strictly positive. It is however worth mentioning that this last hypothesis is not strictly necessary, since the deflation procedure to be presented below may be used to estimate the numbers of signals of positive and negative fourth-order cumulant.

The first step of our deconvolution scheme consists of a classical whitening of \((Y(n))_{n \in \mathbb{Z}}\) which can be realized by means of an adaptive multivariate linear prediction error filter. Therefore, it is possible to assume without loss of generality that \((Y(n))_{n \in \mathbb{Z}}\) is a second-order white noise sequence (i.e. that \(E(Y(n)Y^*(m)) = \delta(n-m)\)), and that the transfer function \(H(z)\) is lossless, in the sense that \(H(z)H^*(z) = I_p\). The signals \((W_i)_{i=1, q}\) whose fourth-order cumulants are negative are extracted by using the following two-step procedure. A unit norm large enough enough degree 1xp FIR filter \(g(z)\) is adapted by means of a stochastic gradient algorithm so as to minimize \(J(g) = E(g(z); Y(n))^2\) on the unit ball \(g_{\infty} = \{g \mid \|g\|_{\infty} = 1\}\). By studying the properties of \(J\) on \(g_{\infty}\), we shall show that, up to the truncation effects, the signal \(r(t) = \langle g(z), Y(n) \rangle\) converges to a signal \(W_i\) such that \(c_i(W_i) < 0\). On the other hand, a 1xp FIR filter whose transfer function \(V(z)\) satisfies \(V(z)V^*(z) = I_p\), is adapted in such a way that \(\langle V(z), Y(n) - r(n) \rangle\) converges to zero. The adaptation algorithm is based on the fact that these transfer functions admits simple parameterizations based on lossless lattice structure ([8]). Moreover, for such a transfer function \(V\), the above-mentioned lossless lattice structure makes possible to exhibit a FIR (p-1)p transfer function \(U(z)\) for which \((U(z)U^*(z) = I_p)\) is lossless. Then, the p-1 variate signal \(Y'(n) = \langle U(z), Y(n) \rangle\) converges to a p-1 variate second order white noise sequence, uncorrelated with \(W_i\) which coincides with the output of a \((p-1)(p-1)\) lossless filter driven by \(W(n) = (W_{1}(n), \ldots, W_{p}(n), W_{p+1}(n), \ldots, W_{p+1}(n))\). It turns out that the preceding algorithm can be applied to \(Y'(n)\) in order to extract an other signal with negative fourth-order cumulant. Clearly, this deflation procedure allows the extraction of the signal \(W_i\) for \(i = 1, q\). The positive fourth-order cumulant excitation are extracted similarly by exchanging minimization by maximization in what precedes.

This paper is organized as follows. The section 2 is devoted to the adaptive minimization of \(J(g)\). The deflation procedure based on the adaptive lossless filter is presented in section 3. Finally, a simple example illustrating these results is given in section 4.

2. ADAPTIVE MINIMIZATION OF \(J\).

In this section, we justify that a stochastic gradient algorithm based on the minimization of \(J(g) = E(g(z); Y(n))^2\) on the unit ball \(g_{\infty}\) allows the deconvolution a negative fourth-order cumulant excitation signal. For this purpose, we need to study the extrema of \(J(g)\) on \(g_{\infty}\). First, we remark that as \(E(g(z)); Y(n)) = 1\), if \(g_{\infty}\), then the study of \(J(g)\) is equivalent to that of the function \(g \rightarrow c_i(g(z); Y(n))\). As in [1] and [7], we will put \(f(z) = g(z)H(z)\); as \(H\) is lossless, the correspondence \(g \rightarrow f\) is a unitary operator. Therefore, the study of the extrema of \(J\) on \(g_{\infty}\) is equivalent to that of the extrema of the function \(K\) defined on \(f_{\infty}\) by \(K(f) = c_i(f(z); Y(n))\). The analytic expression of \(K\) is very simple. In fact, if \(f(z) = \sum_{k=1}^p f_k z^k\), and if we denote by \((f_k)_{k=1,p}\) the components of the row vector \(f\), then we get immediately that

\[
K(f) = \sum_{k=1}^p \sum_{l=1}^p c_k(W_l) f_k l_i  
\]  

(2-1)

In order to study the extrema of \(K\) on \(f_{\infty}\), we follow [7] and we consider the Lagrangian \(L(f, \lambda)\) defined by:

\[
L(f, \lambda) = K(f) - 2\lambda(\|f\|^2 - 1)
\]

(2-2)

The following nice result can be seen as a generalization of what is presented in [7]. Due to the lack of space, the proof is omitted.

Proposition 2-1.

Let \(f\) be an element of \(f_{\infty}\). If at least two coefficients \(f_k\) are non zero, then \(f\) is a saddle point of \(K\).

If \(f\) is the sequence defined by \(f_k = 5(k-k_0), 1-k_0\) for \(k_0 \leq k\), then \(f\) is a local minimum of \(K\); if \(k_0 \geq q+1\), then \(f\) is a local maximum of \(K\).

From this, we get immediately that the local minima of \(J\) on \(g_{\infty}\) are the filters \(z^q H(z^{q-1})\) for \(n \in \mathbb{Z}\) and \(i = 1, q\), and the local maxima of \(J\) are the filters \(z^q H(z^{q-1})\) for \(n \in \mathbb{Z}\) and \(i = q+1, p\). In practice, the deconvolution filters \(g(z)\) over which the minimization is performed are finite impulse response filters with a fixed number of coefficients \(N\); in this case, there is no restriction to assume that they are causal. Therefore, the real problem is that of the adaptive minimization of \(J\) over the class of unit norm degree \(N\) FIR causal filters, for which no positive result similar to Proposition 2-1 holds. However, Proposition 2-1 suggests that if \(N\) is chosen large enough, then the local extrema of the restricted minimization problem should represent reasonable approximations of the optimal filters \(z^q H(z^{q-1})\). Consequently, the stochastic gradient algorithm corresponding to the minimization of \(J\) on \(g_{\infty}\) should converge to a reasonably good FIR causal approximation of one of the filters \(z^q H(z^{q-1})\) for \(i = 1, q\). Finally, if we denote by \(Y(n)\) the \(p(N+1)\) dimensional vector defined by \(Y(n) = (Y(n), \ldots, Y(n-p))^T\), let us recall that the most classical stochastic gradient algorithm can be written as
\[ g(n+1) = \frac{g(n) - \mu r(n) y(n)}{l(n)} \]  
\[ l(n) = \mu r(n) y(n) \]  
where \( g(n) = (g_0(n), ..., g_j(n))^T \) represents the vector of the coefficients of the filter \( g(n, z) = \sum g_k(n) z^{-k} \) at time \( n \) and where \( r(n) \) is defined as \( r(n) = \sum _{k=0} ^{\infty} E[k] Y(n-k) \).

3. THE DEFLOATION ALGORITHM.

Before presenting the adaptive deflation algorithm, it is appropriate to recall (8) that every \( p \)-pole transfer function \( V(z) = \sum V_k z^{-k} \) satisfying \( V(z) V^T(z^{-1}) = 1 \) admits the lossless lattice realization presented fig. 1, where the \((\theta_k)_{k=1,0(N+1)/p-1} \) are uniquely defined angles belonging to \([-p/2, p/2] \). Therefore, such a transfer function is parameterized by its corresponding vector \( \theta = (\theta_1, ..., \theta_M)^T \), where we have put \( M = (N+1)/p-1 \). It is also important to note that the \((p-1)\)-pole transfer function \( U(z) \) of fig. 1 by \( Z_0(z) = U(z) Y(z) \) is such that \( U^T(z), V^T(z) \) is lossless.

Suppose that the adaptation scheme presented in section 2 converges to a FIR filter \( g_\theta(z) \) representing a good approximation of a certain filter \( e^{-mH_2(z^{-1})} \) for j sq; in this case, \( g_\theta \) will approximatively satisfy \( g_\theta(z) g_\theta^T(z^{-1}) = 1 \). Let us now adapt \( V(z) \) so as to minimize \( E[2(z(n) - r(n))]^2 \). Then, after convergence, \( V(z) \) will approximate the filter \( e^{-mH_2(z^{-1})} \), and \( z_\theta(n) \) will "coincide" with \( W_0(n-m) \). Therefore, the \((p-1)\) dimensional signal \( Z_0(n) \) associated to the evolution of \( V(z) \) through fig. 1 will converge to a signal uncorrelated with \( W_1 \), which will represent the output of a \((p-1)(p-1)\) lossless filter driven by the \((p-1)\) variate excitation \( (W_0(n), W_1(n), ..., W_{p-1}(n)) \). Consequently, it will be possible to use the previously introduced algorithms on \( Z_0(n) \). This procedure allows clearly the extraction of the signals \( W_i \) for isq.

At this point, it is necessary to derive a stochastic gradient algorithm associated to the minimization of \( E[2(z_\theta(n) - r(n))]^2 \), or equivalently to the maximization of \( E[2(z_\theta(n) r(n))] \) (recall that \( z_\theta(n) \) and \( r(n) \) have a variance equal to 1). For this purpose, let us begin by giving the state-space description of the input-output relation between \( Y(n) \) and \( (Z_0(n), z_\theta(n))^T \). If \( X(n) \) is the \((N-1)\) dimensional vector defined fig. 1, then it is easily seen that

\[
\begin{pmatrix}
Z_0(n), X^T(n+1), z_\theta(n)
\end{pmatrix}^T = Q(\theta) \begin{pmatrix}
X^T(n), Y^T(n)
\end{pmatrix}^T, 
\]  
\( (\text{3-1}) \)

where \( Q(\theta) \) represents the orthogonal matrix given by \( Q(\theta) = Q(\theta_1), ..., Q(\theta_M) \) and where \( Q(\theta_k) \) is the Givens rotation

\[
Q(\theta_k) = \begin{pmatrix}
\cos \theta_k & -\sin \theta_k \\
\sin \theta_k & \cos \theta_k
\end{pmatrix}, 
\]  
\( (\text{3-2}) \)

In particular, if \( q(\theta) \) represents the last row of \( Q(\theta) \), then \( z_\theta(n) = q(\theta)(X^T(n), Y^T(n))^T \). Consequently, the stochastic gradient algorithm can be written as \( \theta(n+1) = \theta(n) + \mu r(n) \).

But, this algorithm poses two problems. First, the calculation of \( (dz_\theta(n)/dq(\theta)) \) \( \theta \) requires the calculation of the derivative of the state-space signal with respect to each parameter, leading to an enormous computational complexity. Second, the function \( E[z_\theta(n)r(n)] \) may admit spurious local maxima, so that the stochastic gradient algorithm cannot be guaranteed to converge to the global maximum of \( E[z_\theta(n)r(n)] \). But, in that case, \( E(z_\theta(n)r(n)) \) coincides up to a diagonal matrix with positive elements with the \( M \) first rows of \( Q(\theta) \), and taking into account (3-1), we get that the modified adaptation algorithm can be written as

\[
\theta(n+1) = \theta(n) + \mu r(n) \begin{pmatrix}
Z_0(n) \\
X(n+1)
\end{pmatrix}, 
\]  
\( (\text{3-3}) \)

Now, we have to study the possible convergence points of the algorithm (3-3). For this purpose, we shall suppose that the limit filter \( g(\omega) \) of the adaptation scheme presented in section 2 coincides exactly with the optimal filter \( e^{-mH_2(z^{-1})} \); in this case, \( g(\omega) \) satisfies \( g(\omega) g(\omega)^T = 1 \). Obviously, this will not be exactly true in practice, but we think that the derivation of a positive result concerning the algorithm (3-3) under this hypothesis is sufficient to motivate its utilization. The possible convergence points of (3-3) are those filters \( V \) whose parameters \( \theta \) satisfy

\[
E[q(\theta) \begin{pmatrix}
Z_0(n), X^T(n+1)
\end{pmatrix}] = 0. \quad L. (U(z), V(z)) = (D_1 + C_1 z^{-1} A)^{-1} B, \quad D_2 + C_1 z^{-1} A)^{-1} B \]

be the minimal state-space representation of the corresponding lossless structure of fig. 1. Then, it is easily seen that \( E[q(\theta) \begin{pmatrix}
Z_0(n), X^T(n+1)
\end{pmatrix}] = 0 \) iff the \((p+1)\) dimensional row vector \( g(\omega) = (g_0(\omega), ..., g_\omega(\omega)) \) is orthogonal to the rows of the commutability matrix \( (B, A B, ..., A^{N-1} B, 0) \), and to the rows of the matrix \( (D_1, ..., C_1 B, ..., C_1 A^{N-1} B) \). But, by elementary properties of lossless filters, this holds iff \( g(\omega) \) belongs to the space generated by the data \( (V_0, ..., V_N, 0, 0, ..., V_\omega, 0) \). By using the fact that \( g(\omega) \) satisfies \( g(\omega) g(\omega)^T = 1 \), it is possible to establish the following result.

**Proposition 3-1.**

Suppose \( g_\omega = g(\omega) \) is non-zero. Then, the stationary points of the algorithm (3-3) correspond to the filters \( V(z) = g(\omega) \) and \( V(z) = -g(\omega) \).

This result implies that if the first coefficient of the adaptive filter \( g(n, z) \) defined in section 2 does not converge to zero, then the deflation procedure based on the adaptive generation of \( Z_0(n) \) will be successfull. On the other hand, if \( g_\omega = 0 \) in Proposition (3-1), then (3-3) may possess another stationary points.
means that the algorithm (3-3) should converge to an undesirable point if $\|g(n,\infty)\|$ converges to zero; however, based on simulations, we have observed that it is not the case, so that the spurious stationary points corresponding to the case where $g_{i0} = 0$ are probably unstable. Nevertheless, the convergence speed of (3-3) is degraded when $\|g(n,\infty)\|$ converges to zero. A simple solution to overcome this drawback should be to test the value of $\|g(n,\infty)\|$ and replace $r(n)$ by $r(n+1)$ in (3-3) if $\|g(n,\infty)\|$ is sufficiently small.

4. A SIMPLE EXAMPLE.

In order to illustrate what precedes, we consider the following simple example. $p$ is equal to 2, $W(z)$ takes the values $1$ and $-1$ with probability $1/2$ and $W_s(n)$ is uniformly distributed on the interval $[-\sqrt{3}, \sqrt{3}]$. The observation is the output of the 2x2 FIR filter whose transfer function $F(z)$ is given by $F(z) = (1 + F_1 z^{-1}) / (1 + F_2 z^{-1})$, where $Q(n/4)$ is the 2x2 Givens rotation of angle $n/4$, and where $F_1$ and $F_2$ are the matrices

$$F_1 = \begin{pmatrix} 2 & 0.75 \\ 0.75 & 2.25 \end{pmatrix}, \quad F_2 = \begin{pmatrix} 1 & 1 \end{pmatrix}.$$ 

The zeros of $F$ are $-2, -4, \sqrt{2} - 1$ and $-(\sqrt{2} + 1)$; therefore, $F$ has 3 zeros outside the unit disk, from which we deduce that $F$ is non minimum phase. In order to simulate the whitening filter, we have generated the signal $A(z)W(z), W(n)$, where $A(z)$ represents the order 10 normalized prediction error filter associated to the 11 first exact correlation coefficients of the observations. The degrees of the filters $g(z)$ and $V(z)$ are both equal to 7. The figure 2 represents the evolution of the 4 first angles corresponding to the adaptation of $V(z)$, and the figure 3 represents the signal $z(n)$ after convergence of the filter $V(z)$.

REFERENCES

Relation between reduced dimension time and frequency domain adaptive algorithm

G.P.M. Egelmeers  P.C.W. Sommen
Eindhoven University of Technology
P.O. Box 513, 5500 MB Eindhoven

Abstract

Convergence characteristics of update algorithms for adaptive filters are influenced by statistical properties of the input signal. This dependency can be removed by decorrelation of the input signal. Usually this decorrelation is performed, in time- or frequency-domain, by using \( N \) degrees of freedom, with \( N \) the number of adaptive weights. In literature also techniques are available that can perform this decorrelation with \( M \) \((\leq N)\) degrees of freedom, where the number \( M \) can be chosen more properly in resemblance to the statistical properties of the input signal. These techniques are the Partitioned Block Frequency Domain Adaptive Filter (PBFDAF) in frequency domain and the Block Orthogonal Projection (BOP) algorithm in time domain. The aim of this paper is to investigate the relationship between these two algorithms. One of the main novel results is that decorrelation is performed in PBFDAF with low complexity but not in accordance with the BOP approach.

1 Introduction

In this paper the signal estimation scheme, as depicted in fig. 1, is used as a vehicle. This is a form of system identification problem, from which the system to be identified is the optimal Wiener filter for the adaptive filter. It is assumed that all signals are discrete in time, having an inter sample distance of \( T \) seconds, and furthermore that the unknown system \( H \) has an impulse response \( h_i \), that can be modelled by a transversal structure with \( N \) coefficients \( h_i \); for \( i = 0, \ldots, N-1 \). The input signal \( x[k] \) produces via the unknown system \( H \) a signal \( e[k] \), that has to be estimated by the adaptive filter. A signal \( s[k] \), uncorrelated with \( x[k] \), is added to \( e[k] \) resulting in the measurable signal \( \tilde{e}[k] \). The adaptive filter makes a replica \( \tilde{e}[k] \) of the signal \( e[k] \). Theoretically the residual signal \( r[k] = \tilde{e}[k] - e[k] \) in steady state will become equal, in average, to the signal \( s[k] \).

From literature [1] it is known that convergence behavior of adaptive filters is strongly influenced by statistical properties of the input signal. This dependency can be removed by decorrelating the input signal.

In frequency domain this decorrelation is performed by normalizing the power of separate frequency components resulting in a Block Frequency Domain Adaptive Filter (BFDAF). One of the problems with this technique is the large processing delay for large \( N \). In order to reduce this delay by a factor \( Q = N/M \) the impulse response of the adaptive filter can be partitioned. This leads to the PBFDAF algorithm [2], where decorrelation is performed by normalizing over less frequency components.

On the other hand a well known method to decorrelate the input signal in time domain is the Recursive Least Squares (RLS) algorithm. This method uses an estimate of the inverse of the \( M \times M \) autocorrelation matrix of the input signal, with the dimension \( M \) equal to the number of adaptive weights \( N \). The complexity of the RLS method can be reduced by choosing the dimension \( M \) in a more resemblance to the statistical properties of the input signal, resulting in the BOP algorithm [3].

A strong relationship between the PBFDAF and BOP algorithm is expected. In order to investigate this relationship this paper is organized as follows: Section 2 describes the PBFDAF algorithm. In Section 3 the BOP algorithm is first rewritten and the result is transformed to frequency domain. From this result it follows that decorrelation of BOP and PBFDAF is slightly different. In Section 4 the decorrelation methods of BOP and PBFDAF are compared. The results are verified by simulations in Section 5.
2 PBFDAF algorithm

In order to reduce the processing delay of a BPFDAF by a factor $Q = N/M$, the adaptive weight vector is partitioned into $Q$ vectors of dimension $M$. Each of these smaller vectors is updated by separate BFDAs. Combining Discrete Fourier Transforms (DFTs) and performing the delay between different parts of the impulse response and their summation in frequency domain, results in a PBFDAF structure [2] as depicted in fig. 2.

Figure 2: Partitioned structure: $L = 2M - 1$, $Q = N/M$

In order to perform the overlap-save method with block processing techniques, a Serial/Parallel (S/P) converter fetches the input signal samples with an overlap of $M - 1$ samples. Once every $M$ samples a block of $L = 2M - 1$ input signal samples is transformed to frequency domain resulting in a vector $X_0[kM]$.

$$X_0[kM] = \mathcal{F} \cdot ([x[kM - L + 1]]', \ldots, [x[kM]])^T$$

with $(\cdot)^T$ the transpose of a vector and $\mathcal{F}$ the $L \times L$ DFT matrix, from which the $(p, q)^{th}$ element is defined as $(\mathcal{F})_{pq} = e^{-j2\pi pq/L}$. The vector $X_0[kM]$ is input to a length $Q - 1$ delay line, having delays of $T_M = M \cdot T$ seconds. The convolution of each separate part of the weight vector with the corresponding part of the input signal vector is performed in frequency domain by multiplying $X_i[kM]$ with $W_i[kM]$ for $i = 0, \ldots, Q - 1$. Furthermore the convolution of the complete weight vector with the input signal vector is given by summing up all these multiplications and transforming this result back to time domain with an inverse DFT.

Subtracting the last $M$ values of this result from the vector $\tilde{z}[kM]$ yields the residual signal vector $z[kM] = ([r[kM-M+1]], \ldots, [r[kM]])^T$. This vector is converted by a Parallel/Serial (P/S) converter to the residual signal samples $r[k-M+1]$, so the processing delay is $M - 1$ samples.

To perform decorrelation, as depicted in fig. 3, the residual signal vector $z[kM]$ is extended with $M - 1$ zeros and transformed to frequency domain as:

$$\tilde{z}[kM] = \mathcal{F} \cdot \left( \begin{array}{c} 0 \\ \mathbf{I}_M \end{array} \right) \cdot z[kM]$$

with $\mathbf{I}_M$ the $M \times M$ identity matrix and $0$ a matrix containing all zeros having appropriate dimension. The frequency domain vector $\tilde{z}[kM]$ is multiplied by the adaptation constant $2\alpha/M$, and each frequency component $l$ is multiplied by the $l^{th}$ component of the input signal power vector $L^{-1}[kM]$. This $l^{th}$ element equals $L/\mathbb{E}([X_0[kM])_l^2]$, with $\mathbb{E}$ the mathematical expectation. The normalization yields the $l$ dimensional vector

$$\mathcal{Y}_L[kM] = \frac{2\alpha}{M} L^{-1}[kM] \odot \tilde{z}[kM]$$

as depicted in fig. 3, where the operator $\odot$ is defined as an element by element multiplication of its two operand vectors.

Figure 3: Normalization of PBFDAF

This vector $\mathcal{Y}_L[kM]$ is used for the update of each weight vector $W_i[kM]$ as depicted in fig. 4. Note that in fig. 2 and fig. 4 $\mathcal{Y}_L[kM] = \mathcal{Y}_P[kM]$. These figures will be used later on for the BOP approach with $\mathcal{Y}[kM] = \mathcal{Y}_P[kM]$. The estimate of the gradient for each part $i$ of the filter is calculated by multiplying the vector $\mathcal{Y}_P[kM]$ with the conjugate $(\cdot)^*$ of the (delayed) input signal vector $X_i[kM]$. In order to obtain a correct overlap-save result a window $\mathbf{V}$ is applied to this product, defined as:

$$\mathbf{V} = \mathcal{F} \cdot \left( \begin{array}{c} \mathbf{I}_M \\ 0 \\ 0 \end{array} \right) \mathcal{F}^{-1}.$$  \hspace{1cm} (4)

Finally the update of each adaptive weight vector $W_i[kM]$ for $i = 0, \ldots, Q - 1$ is given by:

$$W_i[(k+1)M] = W_i[kM] + \mathbf{V} \cdot (X_i[kM] \odot \mathcal{Y}_P[kM]).$$

(5)
3 BOP algorithm

In [3] it is shown that the BOP algorithm uses, in contrast to the RLS method, an \( M \times M \) matrix \( \hat{R}[kM] \) to decorrelate the input signal. Furthermore, as a result of the block processing approach, the adaptive weight vector \( w[kM] = (w_{M-1}[kM], \ldots, w_0[kM])^T \) is updated only once every \( M \) samples according to:

\[
w_i[(k+1)M] = w_i[kM] + \frac{2\alpha}{M} \chi[kM] \hat{R}^{-1}[kM] z[kM]. \tag{6}
\]

In this update equation the input signal samples are collected in the \( N \times M \) matrix

\[
\chi[kM] = (z[kM-M+1], \ldots, z[kM])
\]

containing \( M \) vectors of dimension \( N \) defined for \( i = 0, \ldots, M-1 \) as:

\[
z[kM-i] = (z[kM-i-M+1], \ldots, z[kM-i])^T. \tag{8}
\]

Furthermore the estimate of the \( M \times M \) autocorrelation matrix is defined as:

\[
\hat{R}[kM] = \frac{1}{N} \chi^H[kM] \cdot \chi[kM]. \tag{9}
\]

The aim of this paper is to investigate the relationship between the BOP and PBFDAF algorithm. For this the \( N \) dimensional adaptive weight vector \( w[kM] \) is first partitioned in \( Q = N/M \) separate weight vectors \( w_i[kM] = (w_{(i+1)M-1}[kM], \ldots, w_{iM}[kM])^T \), of dimension \( M \). The update equation (6) can now be rewritten for \( i = 0, \ldots, Q-1 \) as:

\[
w_i[(k+1)M] = w_i[kM] + \frac{2\alpha}{M} \chi_i[kM] \hat{R}^{-1}[kM] z[kM] \tag{10}
\]

with the "partitioned" \( M \times M \) matrix \( \chi_i[kM] \) containing the appropriate part of the \( N \times M \) input signal matrix \( \chi[kM] \). In order to transform this update equation to frequency domain the following two steps are used:

1. With \( L = 2M - 1 \) the \( L \times L \) matrix \( \chi_i^H[kM] \) is defined as a mirrored circulant expansion of the \( M \times M \) matrix \( \chi_i[kM] \) and these matrices are related as:

\[
\chi_i[kM] = J_M \cdot (I_M \ 0) \cdot \chi_i^H[kM] \cdot (0 \ I_M) \tag{11}
\]

with \( J_M \) the \( M \times M \) mirror matrix containing only ones on the mirrored main diagonal, thus \( (J_M)_{p,q} = 1 \) for \( |p-q| = M-1 \).

2. The \( L \times L \) circulant matrix \( \chi_i[kM] \) can be diagonalized with the DFT matrix \( F \) as follows:

\[
F^{-1} \cdot \text{diag}(\chi_i^H[kM]) \cdot F = \chi_i[kM] \tag{12}
\]

with \( \text{diag}(\cdot) \) representing the diagonal matrix with as diagonal the vector between the curly brackets.

Using these steps it is possible to transform the update equation (10) to the frequency domain. The transformation is constructed as to obtain an equal residual signal vector \( z[kM] \) for both algorithms. This results in the following transformation \( T \):

\[
T = F \cdot (I_M \ 0) \cdot J_M. \tag{13}
\]

Applying \( T \) to both sides of (10) results in the following update equation:

\[
W_i[(k+1)M] = W_i[kM] + V \cdot (\chi_i[kM] \odot Y_p[kM])
\]

\[
Y_p[kM] = \frac{2\alpha}{M} F^{-1} \cdot (0 \ I_M) \hat{R}^{-1}[kM] z[kM]. \tag{14}
\]

From these equations it follows that the BOP algorithm can be implemented with the structure as depicted in fig. 2 using the update as given in fig. 4. The difference between the BOP and PBFDAF algorithm is the way of normalization to obtain vector \( Y_p[kM] \).

4 Normalization methods

In this section the normalization methods of the BOP and the PBFDAF algorithm are compared to obtain insight into their differences. The PBFDAF normalized vector \( Y_p[kM] \) is obtained by a component-wise multiplication in frequency domain according to eq. 3. Such a component-wise multiplication is the frequency-domain equivalent of the product of a circulant matrix and a vector, so:

\[
Y_p[kM] = \frac{2\alpha}{M} F \cdot R_p^{-1}[kM] \cdot (0 \ I_M) \cdot z[kM] \tag{15}
\]

with \( F \cdot R_p^{-1}[kM] \cdot F^{-1} = \text{diag}(P^{-1}[kM]) \).
The $L \times L$ circulant pseudo-autocorrelation matrix $\mathbf{R}_P[kM]$ contains the (pseudo-)autocorrelation vector $\rho_p[kM] = (\rho_p[0], \ldots, \rho_p[L - 1])^T$. The elements of $\rho_p[kM]$ are given by $\rho_p[r] = E[x[r]x[k-r]]$ and $\rho_p[r] = E[x[r]x[k-r]]$.

Equation 14 can now be written as:

\[
\gamma_B[kM] = \frac{2\alpha}{M} \cdot \mathbf{R}_P[kM] \cdot \begin{pmatrix}
\mathbf{0} \\
\mathbf{I}_M
\end{pmatrix}
\]

\[
\gamma_B[kM] = \mathbf{R}_P[kM] \cdot \begin{pmatrix}
\mathbf{0} \\
\mathbf{I}_M
\end{pmatrix}
\]

The difference between the normalization of BOP and PBFDAD is concentrated in equation 17. BOP normalization uses this equation, while PBFDAD approximates the matrix $\mathbf{I}_M$ by the $M \times M$ identity matrix $\mathbf{I}_M$ and $\mathbf{0}$ by the zero matrix $\mathbf{0}$.

5 Simulation results

In order to gain insight into the magnitude of the approximation made in the PBFDAD case of the previous section, both algorithms were simulated. Some typical results are presented in figure 5. As input an Auto-Regressive signal of order 1 (AR1) signal is used: $x[k] = 0.3122 \cdot n[k] + 0.95 \cdot x[k-1]$ with $n[k]$ a zero mean (white) noise signal with variance equal to 1. The additive noise signal $\tilde{n}[k]$ is also (zero mean), with variance 0.01. The adaptive filter length $N$ is 32 and the adaptation constant $\alpha = 0.0001$. Choosing the parameter $M = 1$ results in the NLMS algorithm (no decorrelation), while choosing $M = N$ and using the BOP algorithm results in the IDEAL case (full decorrelation). Values of $M$ between 1 and $N$ yield no relevant differences between the BOP and PBFDAD algorithm. The case of $M = 4$ is depicted in figure 5 for both algorithms.

6 Conclusions

The approximation of PBFDAD with respect to BOP (section 4) does not seem to introduce relevant deviations in the convergence curves of the PBFDAD algorithm compared to the BOP algorithm. As the complexity of the PBFDAD algorithm is for most practical cases smaller than the complexity of the BOP algorithm, the first one seems to be preferable. However, as the fraction $M$ has to be a divisor of the total adaptive filter length $N$ in the PBFDAD case, it is not always possible to replace the BOP algorithm by a PBFDAD algorithm with equal parameters. More investigations of the exact nature of the difference between both algorithms have to be carried out in future.

References


Figure 5: Results BOP and PBFDAD
AN ALTERNATIVE SCHEME FOR ADAPTIVE NONLINEAR PREDICTION USING RADIAL BASIS FUNCTIONS

J. Vesein
Laboratoire de Traitement des Signaux
Ecole Polytechnique Fédérale de Lausanne, 1015 Lausanne Switzerland

Abstract. Radial basis functions (RBF) constitute a valuable alternative to polynomials and multi-layer perceptrons in the context of nonlinear signal prediction. They are characterized by the fact that they are linear in some of their parameters (the external parameters), and nonlinear in the other ones (the internal parameters). This paper describes an adaptive RBF prediction scheme, based on LMS, but incorporating a recursive form of the internal parameter selection technique used in batch mode prediction.

1. Introduction

There has been a recent interest in the nonlinear predictive analysis of time series, for instance in domains such as speech processing and the study of chaotic signals. Although polynomials possess the universal approximation capability, their use in nonlinear prediction presents some problems such as lack of stability and difficulties in the choice of the relevant terms [1]. Some authors [3, 4] have proposed the use of neural networks, mainly in the context of multilayer perceptrons (MLP) equipped with the well-known back-propagation algorithm, due to the fact that they can approximate any continuous multidimensional function. However, one of the major drawbacks of this scheme is its very slow speed of convergence. More recently Radial Basis Functions (RBF), which have been used for some time in the context of multidimensional interpolation [4], and possess too the universal approximation capability [5], have attracted attention for nonlinear prediction purposes, for the first time to our knowledge in [6] on chaotic time series and speech data, and in [7] on speech data. In this last paper, a significant improvement upon linear prediction was reported in an experiment. Known procedures exist in the literature to select the internal parameters of the RBF so that their amplitude parameters can be estimated in a least squares sense in batch mode prediction. However, poor speed of convergence is observed in the case of adaptive prediction if a nonlinear LMS scheme [8] is used, due to the fact that the internal and amplitude parameters get mixed in the update equations. We first summarize briefly in this paper some known results about RBF, then present an adaptive scheme which includes the advantages of RBF internal parameter selection. It is shown on real speech data to provide a faster convergence with a smaller computation load when compared to a classical nonlinear LMS algorithm.

2. RBF parameter selection

\{x_n\}, \ n = 1, ..., N, being the time series of interest, and defining \ X_{n-1} = [x_{n-1}, x_{n-2}, ..., x_{n-p}]^T, \ the prediction estimate \ \hat{x}_n \ of \ x_n \ of \ order \ p \ with \ a \ RBF \ network \ will \ be \ expressed \ as

\[ \hat{x}_n = f(x_{n-1}) = \sum_{\mu=1}^{m} \lambda_{\mu} \tilde{f}(\|X_{n-1} - C_{\mu}\|) \] (1)

that is, as the weighted sum of nonlinear functions of the Euclidian distance of the data vector \ X_{n-1} \ to \ a \ set \ of \ predetermined \ vectors \ (centers) \ \{C_{\mu}\}, \ \mu = 1, ..., m. \ Several \ choices \ are \ possible \ for \ the \ functions \ \tilde{f}(\cdot) \ [9]. \ The \ most \ common \ one \ is

\[ \tilde{f}(r) = \exp \left( -\frac{r^2}{\beta^2} \right) \] (2)

If prediction is done in block mode, one generally tries to minimize the squared error \ E \ over \ N \ samples, that is

\[ E = \sum_{j=1}^{N} (x_j - \hat{x}_j)^2 \] (3)

In [7] the estimation of the parameters \ \{\lambda_{\mu}\}, \ \{\beta_{\mu}\}, \ and \ \{C_{\mu}\}, \ \mu = 1, ..., m, \ was \ performed \ by \ a \ gradient \ descent \ over \ E. \ But \ due \ to \ the \ nonlinear \ dependence \ of \ E \ on \ the \ \{\beta_{\mu}\} \ and \ \{C_{\mu}\}, \ convergence \ to \ a \ global \ minimum \ is \ not \ assured. \ A \ better \ strategy \ [9] \ seems \ to \ select \ first \ the \ centers \ \{C_{\mu}\} \ using \ a \ data-dependent \ clustering \ algorithm \ such \ as \ Kohonen \ self-organizing \ procedure \ [11] \ which \ in \ its \ simplest \ form \ can \ be \ expressed \ as

\[ C_k(n) = C_k(n-1) + \alpha(n)[x_{n-1} - C_k(n-1)] \] (4)

where the center \ \ C_k \ at \ step \ (n-1) \ is \ such
\[ \|X_{n,1} - C_{i}\| = \min_{i}\{\|X_{n,1} - C_{i}\|\} \]  

and \( \alpha(n) \) is a small coefficient decreasing with \( n \).

The operation described by eqs. (4) and (5) can be re-iterated on the set of data until convergence is obtained. Once the centers are selected, the \( \{\beta\} \) parameters can be chosen in one of the two following ways: 1) if \( \beta_i = \beta \) for all \( i = 1, \ldots, m \), then a convenient rule-of-the-thumb value for \( \beta \) is \[ \beta = \max_{i,j}\{\|C_i - C_j\|\} \]  

2) otherwise, the \( \{\beta\} \) can be obtained by minimizing with respect to them the following quantity [10]

\[ D = \sum_{l=1}^{m} \left[ \sum_{j=1}^{m} \left( \frac{\|C_l - C_j\|^2}{\beta_{j}} \right) d_l(\|C_l - C_j\|) - P \right]^2 \]  

with \( P \) an overlap parameter. The minimization of \( E \) in eq. (3) with respect to the \( \{\lambda\} \) can then be performed in an optimal way using an efficient method such as the singular value decomposition. We have found experimentally by Monte-Carlo tests on real and simulated data (results are not shown here due to the lack of space) that another good choice for the \( \{\beta\} \) is

\[ \beta_i = \frac{1}{N} \sum_{j=1}^{N} \|X_{j} - C_{i}\| \]  

Intuitively, this formula is justified by the fact that the clustering algorithm described by eqs. (5) and (6) will concentrate the centers in the vicinity of the modes of the probability density of \( \{X_{k}\} \). The range of action, parametrized by the \( \beta_{i} \) of the functions \( f(\cdot) \) corresponding to these clustered centers should be smaller in order not to "blur" their contributions in that part of the state space. Eq. (8) expresses this dependence of the \( \{\beta_i\} \) upon the relative location of the \( \{C_i\} \) with respect to the data vectors \( \{X_{n}\} \).

### 3. Proposed adaptive algorithm

For the time being, let us denote the global parameter vector at time \( n \) by \( \theta(n) = [\lambda_{1}, \ldots, \lambda_{m}, \beta_{1}, \ldots, \beta_{m}, c_{11}, \ldots, c_{1p}, \ldots, c_{mp}]^{T} \), where \( c_{ik} \) (dependence on \( n \) not shown) represents the \( j^{th} \) coordinate of the center \( C_{k} \). In a classical LMS frame, the objective is to minimize \( \mathcal{E}[\{\epsilon_{n}\}^{2}] \), where \( \epsilon_{n} = x_{n} - f(X_{n,1}) \). The gradient of this statistical quantity is replaced by its instantaneous estimate, namely

\[ \nabla(n) = \frac{\partial f}{\partial \theta(n)} = 2 \epsilon_{n} \frac{\partial f}{\partial \theta(n)} = -2 \epsilon_{n} \frac{\partial f(x_{n,1})}{\partial \theta(n)} \]  

and the parameter vector update equation is

\[ \theta(n+1) = \theta(n) - \mu \nabla(n) = \theta(n) + 2 \mu \epsilon_{n} \frac{\partial f(x_{n,1})}{\partial \theta(n)} \]  

with \( \mu \) a coefficient governing the rate of convergence.

Let us split now \( \theta(n) \) into two vectors \( \lambda(n) = [\lambda_{1}, \ldots, \lambda_{m}]^{T} \) and \( \gamma(n) = [\beta_{1}, \ldots, \beta_{m}, c_{11}, \ldots, c_{1p}, \ldots, c_{mp}]^{T} \). The idea is to update \( \gamma(n) \), that is the internal parameters of the RBF, using recursive schemes corresponding to the "good" choices presented in the section 2. Thus only the center \( C_{k} \) closest to \( X_{n,1} \) will be updated with eq.(4), which is already in a recursive form (now a constant update coefficient, \( \alpha_{C} \), instead of \( \alpha(n) \), will be employed).

If a unique parameter \( \beta \) is considered, it is then updated using eq.(6). Note that only the distances involving the updated center \( C_{k} \) have to be recomputed and compared to the old value of \( \beta \). If different \( \beta \) are considered, eq.(8) is easily put in the following recursive form

\[ \beta(n+1) = (1 - \alpha_{B}) \beta(n) + \alpha_{B} \|X_{n,1} - C_{i}\| \]  

with \( 0 < \alpha_{B} \ll 1 \). The objective is still to minimize \( \mathcal{E}[\{\epsilon_{n}\}^{2}] \), but now \( \epsilon_{n} = x_{n} - [f(X_{n,1}) + \Delta f(X_{n,1})] \), where the term \( \Delta f(X_{n,1}) \) stands for the modification of \( f(\cdot) \) due to the modification \( \Delta \gamma(n) = \gamma(n+1) - \gamma(n) \). Using

\[ \Delta f(X_{n,1}) = (\Delta \gamma(n))^{T} \frac{\partial f(x_{n,1})}{\partial \gamma(n)} \]  

the instantaneous gradient estimate becomes:

\[ \nabla(n) = \frac{\partial f}{\partial \lambda(n)} = 2 \epsilon_{n} \frac{\partial f}{\partial \lambda(n)} = -2 \epsilon_{n} \left[ \frac{\partial f(x_{n,1})}{\partial \lambda(n)} + (\Delta \gamma(n))^{T} \frac{\partial^{2} f(x_{n,1})}{\partial \gamma(n) \partial \lambda(n)} \right] \]  

and the update equation for \( \lambda(n) \) remains:

\[ \lambda(n+1) = \lambda(n) - \mu \nabla(n) \]
This scheme can be interpreted in the following way: If the centers \( \{C_k\} \) and the parameters \( \{\beta_l\} \) were held constant (\( \Delta f(X_{n-1}) = 0 \)), then eqs. (13) and (14) would correspond to a classical LMS scheme on \( \lambda(n) \), that is, to an approximate gradient descent on a paraboloid. Since \( \{C_k\} \) and \( \{\beta_l\} \) vary, the paraboloid moves slowly and the additional term in the right-hand side of eq. (13) constitutes a correction to the update value taking account of this displacement. Note that, due to the fact that \( \Delta y(n) \) is a sparse vector and that the coordinates of one center only have to be updated, eq. (14) necessitates a lot less computations than eq. (10) does.

4. Experimental illustration

Nonlinear prediction with a RBF network has been performed on two speech signals sampled at 20 KHz corresponding to the sounds /el/ and /al/. These signals are shown respectively in figure 1 and 3. For both of them, a prediction order \( p=3 \) was chosen by inspection of the singular values of their estimated covariance matrix. In all experiments a RBF network of \( m=10 \) functions was used. The same value \( \mu=0.2 \) was used for the gradient coefficient in the standard LMS scheme (eq. (10)) and in the proposed scheme (eq. (14)). In all experiments the center update coefficient was taken to be \( \alpha_c=0.2 \). In the case of multiple \( \{\beta_l\} \) the update coefficient value was \( \alpha_{\beta}=0.02 \).

In figures 2, 4, and 5 the prediction error signal obtained with the classical LMS scheme is represented by a dashed line and the prediction error signal obtained with our method by a solid line.

Figure 2 shows the prediction error signals corresponding to the signal of figure 1 with a single parameter value \( \beta_i \).

Figure 4 shows the prediction error signals corresponding to the signal of figure 3 with a single parameter value \( \beta_i \).

Figure 5 shows the prediction error signals corresponding to the signal of figure 3 with multiple parameter values \( \{\beta_l\} \).

Faster convergence and smaller residual error for our algorithm are clearly visible in all examples.

5. Conclusion

We have presented here an adaptive algorithm for time series prediction using RBF that incorporates an interesting feature of these functions in block mode prediction, namely the possibility to separate the computation of their external and internal parameters. It was shown on examples that this algorithm leads to faster convergence and lower computational load when compared to classical LMS. An interesting improvement would be a strategy to switch off the internal parameter update when the prediction power is below a fixed threshold.

References

Figure 1
Sound /f/

Figure 2
Prediction error signals
unique value $\beta$
dashed line: classical LMS
solid line: our method

Figure 3
Sound /a/

Figure 4
Prediction error signals
unique value $\beta$
dashed line: classical LMS
solid line: our method

Figure 5
Prediction error signals
multiple values $\{\beta\}$
dashed line: classical LMS
solid line: our method
Convergence analysis of stochastic gradient adaptive algorithms for arbitrary error surfaces

José M. Páez Borralo and Francisco J. Casajús Quirós
ETSI Telecommunicación, Universidad Politécnica de Madrid
Ciudad Universitaria s/n, 28040, Madrid, SPAIN

Abstract. The paper presents a general analysis of the second-order convergence of gradient adaptive algorithms for arbitrary error surface without considering the reason for its choice. It shows how the absence of any design criteria of the adaption step may cause undesired misconvergence, or even divergences, for the cases of non-quadratic error surfaces. The convergence analysis is then particularized for some members of the L1 family (LMS and LMF) and a special case of decision-directed algorithm for Binary statistic. Some computer simulations are also given.

I. Introduction

As it is widely known, the performance of any stochastic gradient adaptive algorithm is strongly linked to the shape of its error cost function. This is usually a memory-less, even-shaped and convex function of the measured error signal. Therefore, its minimum is supposed to be reachable following a steepest descent line (its negative gradient) conveniently scaled by a constant adaption step. This step is generally the only user-dependent parameter involved in the algorithm and plays an important role in the design and satisfaction of the convergence conditions. In fact, it controls simultaneously the speed and degree of convergence of the algorithm (unfortunately two opposite objectives). In summary, the family of stochastic gradient adaptive algorithms is generally expressed by:

$$w(n+1) = w(n) - \mu \text{grad}_w[C(e(n))]$$

(1)

where w is the vector of parameters to update in each iteration n, μ, the adaption step or convergence parameter, z the current measured error signal and C(z) the associated cost function. In our work, we will make use of a transversal adaptive filter working in a plant identification environment (see figure 1). With this structure, expression (1) simplifies to

$$w(n+1) = w(n) + \mu f(e(n))x(n)$$

(2)

where f(z) is an odd-shape function of (z) and x is the vector of input data. Here, it is important to remark the separability of the instantaneous gradient into two factors f(z(n)) and x(n). This separation will be helpful in the study of final stages of the convergence where both processes are generally assumed to be nearly uncorrelated. This assumption allows us to focus the second-order convergence analysis (disregarding the input power) on the arbitrary shape of the cost function.

Perhaps, the most relevant member of such family of algorithms is the extensively studied LMS algorithm due to Widrow and Hoff, which searches for the unique minimum of an instantaneous quadratic error surface (the best possible choice for Gaussian statistics) leading to a linear gradient. However there exist other possibilities in the choice of the error criterion (more complex than the quadratic one) depending on several aspects such as noise statistics, presence of inhomogeneities in the data, or the use of some problem-oriented techniques to exploit features and symmetries of the signals in order to improve the original error reference. Examples of these arise in communication applications, for instance, the CMA family of adaptive algorithms that exploits signals with constant modulus, the L1-norm family (k>2), better suited to signals with short-tail probability density functions, the decision-directed algorithms, which try to improve the error reference removing part of the interfering far-end signal, and some others. These non-standard adaptive algorithms have in common non-linear gradients of the error reference that leads, in general, to more complex convergence conditions than that of the LMS algorithm (just a constant upper bound for the adaption step).

![Figure 1](image-url)

To give some insight into the behavior of algorithms with non-linear gradients, consider, as an example, the
unidimensional curve $C(z) = z^3$ as an error surface of $z$ (figure 2). The maximum allowed correction, before divergence, for an updating algorithm at any generic point $z$ is $C_{r_{\text{max}}} = 2\mu|z|$ (the maximum distance to the opposite side) and the absolute gradient at $z$ is $\text{grad}C(z) = 3z^2$ (greater than $C_{r_{\text{max}}}$ for certain values of $z$). It implies a convergence dependence on the starting position $z(0)$ when using an algorithm based on the gradient of this deterministic cost function to update any set of parameters. That is, the gradient has to be sufficiently scaled to satisfy the following convergence condition:

$$C_{r_{\text{max}}}(z) = 2|z| > \mu|\text{grad}C(z)| \quad \forall z \quad (\mu > 0) \quad (3)$$

![Figure 2](image)

It shows that $\mu$ has, in general, to be selected according to the $2\mu|z(0)|/|\text{grad}C(z(0))|$ starting ratio. For the quadratic surface, both functions, $C_{r_{\text{max}}}(z)$ and $\text{grad}C(z)$, are linear, and therefore only a slope adjustment is needed ($\mu$ is therefore independent of the starting condition).

Thus, having in mind the possibility of a undesired convergence behavior shown in the latter example, the aim of this paper is to present, regardless of the error surface shape or special error measure, a general formulation for the second order statistics of the convergence that allows to establish recommendations in the choice of a constant adaption step that guarantees convergence to the desired steady-state value. We will make use of usual approximations and assumptions (zero mean i.d.d. inputs, slow convergence, uncorrelation between tap vector and input, etc) in our analysis.

2. Second order analysis

Denoting $\theta(n)$ as the identification error, that is, $\theta(n)$ is splitted as $\theta(n) = \theta(n) + \epsilon(n)$, we can write:

$$\epsilon(n) = [w^* - w(n)]'x(n) = v'(n)x(n) \quad (4)$$

and using (1) and (4) we arrive to

$$v'(n+1)v(n+1) = v'(n)v(n) + \mu f(n)e(n) + \mu^2 f'(n)x'(n)x(n) \quad (5)$$

Considering now the usual assumptions above mentioned, we can derive:

$$\sigma^2(n) = E\left[\epsilon'(n)x(n)\right] = \sigma^2(n)e(n)v(n) \quad (6)$$

Therefore, taking expectations in (5) and multiplying both sides by $\sigma^2$, we obtain the following recursive equation for the identification error variance

$$\sigma^2(n+1) = \sigma^2(n) + \mu^2 \sigma^2(n) \quad (7)$$

where $P(\mu, n)$ is a time-dependent quadratic polynomial in $\mu$ (required to be positive defined in $\mu$) given by:

$$P(\mu, n) = 1 - \mu \sigma_{1}(\epsilon) + \mu^2 \sigma_{2}(\epsilon) \quad (8)$$

Polynomial $P(\mu, n)$ depends on the statistics of the involved signals and shape of the measured gradient. Here $S_1$ and $S_2$ are the following mean values

$$S_1(n) = 2\sigma^2 \frac{E\left[(\epsilon(n)f(n))\right]}{\sigma^2(n)} \quad (9a)$$

$$S_2(n) = N\sigma^2 \frac{E\left(f^2(n)\right)}{\sigma^2(n)} \quad (9b)$$

The main subject under study is to guarantee the global second order convergence, that is, $\sigma^2(n+1) < \sigma^2(n)$. The analysis, here presented, is therefore focused on a systematic analysis of the positiveness and local boundeness of $P(\mu, n)$, which, assuming the surface shape (or its derivatives) are known in a stationary environment, allows us to establish general conclusions about local and global convergence conditions.

Thus, since $S_1(n)$ is a quotient of quadratic means, and therefore, it is always positive, (8) turns out to be a positive parabolic function of $\mu$. Nevertheless, to show its validity in any positive range of $\sigma^2$, we have also to prove that its discriminant, defined as $D = S_1^2(n) - 4S_2(n) \leq 0 \quad \forall \sigma^2 > 0$ or equivalently $S_1^2(n)/4S_2(n) \leq 1$. Hence, considering the ratio

$$\beta = \frac{S_1(n)}{4S_2(n)} = \frac{E\left[(\epsilon(n)f(n))\right]}{N\sigma^2 \frac{E\left(f^2(n)\right)}{\sigma^2(n)}} \quad (10)$$

and according to the Cauchy-Schwarz inequality, $E[XY] \leq E[X^2]E[Y^2]$, we obtain $\beta \leq 1/N$, where $\forall N \geq 1 \Rightarrow \beta \leq 1$ and therefore, it proves that (8) is not negative.

3. Convergence analysis

$P(\mu, n)$ allows an easy analysis of two important aspects of any adaptive algorithm:
a) the convergence condition, by means of solving for μ the inequality:

\[ P(\mu, n) < 1 \quad \forall n \in [0, \infty) \]  

(11)
b) the final misadjustment knowledge (σ2(∞)) by considering the convergence stopped when:

\[ 1 - \mu S_1(\infty) + \mu^2 S_2(\infty) = 1 \Rightarrow \sigma_2^2(\infty) \]  

(12)

From (11) the convergence condition follows

\[ 0 < \mu < \min \left[ \frac{S_2(n)}{S_1(n)} \right] \quad \forall n \in [0, \infty) \]  

(13)

that is, μ has to be upper bounded by a function of σ2(n) in the whole convergence range of the algorithm.

Regarding (12), the constant adaptation step μ is obtained as

\[ \mu = \frac{S_2(\infty)}{S_1(\infty)} \]  

(14)

and therefore from (13) and (14) we arrive to a general dynamic mean condition which guarantee convergences to any selected final value σ2(∞):

\[ \frac{S_2(n)}{S_1(n)} < \frac{S_2(\infty)}{S_1(\infty)} \quad \forall n \]  

(15)

That is, with a priori knowledge of the μ-bounding function, B(n)=S_1(n)/S_2(n), in the desired convergence range, σ2(0)→σ2(∞), it is possible to predict, in a mean sense, any possible misconvergence of the adaptive algorithm.

To illustrate the last assertion, we make use of a graphic example. Let figure 3 be a generic bounding function B[σ2(∞)], where σ2(0) and σ2(∞) are the initial and the steady state values of the considered convergence range.

![Figure 3](image)

For this particular situation the practical convergence analysis should be as follows: a) σ2(0) is known (or at least an accurate estimate or guess of it). It determines the convergence starting point. b) The desired σ2(∞) is selected. It fixes μ=B(∞). c) The inequality B(∞)<B(n) ∀ n is satisfied.

Thus, we can conclude here that mean convergence from σ2(0) towards σ2(∞) is guaranteed. However, if we tried any other choice of σ2(∞) located between the local minimum and maximum of figure 3, in a non-monotonic part of the curve, the algorithm would lead to a possible local misconvergence because the condition B(∞)<B(n) is violated in some parts of the curve.

4. Some particular error surfaces

Let us consider now the three following error criteria:

a) C(ε)=ε^2 \quad \rightarrow \quad \text{LMS algorithm}

b) C(ε)=|ε|^4 \quad \rightarrow \quad \text{LMF algorithm}

c) C(ε)=(ε-\text{sgn}(ε))^2 \quad \rightarrow \quad \text{Decision-directed algorithm}

Their corresponding μ-bounding functions are shown in figure 4. We have assumed Binary statistic in the analysis for both processes, x(n) and u(n), because only in this case does the use of decision-directed algorithms make sense. Here we can observe the increasing monotonic character of B(σ2) for the LMS algorithm (solid line). It implies that the convergence condition will be always satisfied for any selected steady state value σ2(∞)<σ2(0). Although not shown in the figure, the corresponding bounding function for the LMA algorithm (C(ε)=|ε|) also has monotonic character with σ2. In the case of the LMF algorithm (dashed line), if the initial error variance σ2(0) is located to the left of the local maximum, the effective bounding function behaves as a monotone curve again and the convergence condition will be always satisfied. Further, if σ2(0) is on the right of the maximum, the steady state value σ2(∞), had to be selected in such a way that μ=B(σ2(∞))<B[σ2(0)]. It means that the convergence of this kind of algorithms (termed generally as Lk) for k>2, is conditioned by the initial settings of the adaptive filter. Finally, the bounding function corresponding to the decision-directed algorithm shows an inflection that could lead to undesired misconvergences before reaching the selected steady state.

![Figure 4](image)
To show the real (strictly speaking, simulated) behavior of such misconvergence in the case of the above mentioned decision-directed algorithm, consider the situation of figure 5. Here, the designer would like a steady state value of $\sigma^2(\infty) = -30$ dB (this choice for $\sigma^2(\infty)$ fixes $\mu$). Since this steady point is located on the left of the local minimum of $B(\cdot)$, there exists a risk of stopping the convergence around $\sigma^2(\infty) = 3$ dB whenever the initial settings of the adaptive filter, $w(0)$, forces $\sigma^2(0)$ to be on the right side of the local minimum in the bounding curve.

![Graph of $[e - \text{sgn}(e)]^2$](image)

Figure 5

Figure 6 shows three simulated realizations (#1, 2 and 3 in dashed lines against the theoretical result, solid line) of the above situation. The initial error power, $e^2(0)$, is around 12 dB for all of them. After the initialization of the adaptive process, the convergence first stops in the vicinity of $\sigma^2(\infty) = 3$ dB after an undetermined period of time. Suddenly, without any kind of indication, and because of the noisy character of the stochastic gradient, it leaves that pseudo-stationary point to converge to the selected steady state $\sigma^2(\infty)$.

![Graph of Risk of stopping convergence](image)

Figure 6

These results, for the decision-directed algorithm, are quite representative of the behavior of adaptive algorithms with non-monotonic $\mu$-bounding functions in the desired convergence range $\sigma^2(0) \rightarrow \sigma^2(\infty)$.

In the latter example, the designer should know both, the position of the local minimum and an accurate guess of $\sigma^2(0)$, before proceeding with the choice of $\mu$. In this way it is possible to maintain the inequality $B(\sigma^2(\infty)) > B(\sigma^2(0))$ vs to guarantee the desired convergence.

5. Conclusions

The work here summarized presents how particular shapes of non-quadratic error criteria can lead to second-order misconvergence of their corresponding stochastic gradient adaptive algorithms. In some ill-conditioned cases these misconvergences will even result in proper divergences. The work shows that this undesired behavior for any adaptive algorithm can be predicted and avoided, in a mean sense, by making prior use of a running bounding function for the adaptation step $\mu$. This function depends on the shape of the selected error criterion in the involved data statistic. It is presented versus the identification error variance, $B(\sigma^2)$, and allows one to design the right adaptation step in order to guarantee the convergence from the initial guessed situation $\sigma^2(0)$ to the selected steady state value $\sigma^2(\infty)$. In other words, it would detect possible misconvergence when $\mu$ is selected without considering any design criteria related to the steady state value.

6. References

NONLINEAR EFFECTS IN ADAPTIVE FILTERS OF THE LMS TYPE

H. J. BUTTERWECK

EE-Department, Eindhoven University of Technology, Eindhoven
P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands

An adaptive filter can be viewed as a nonlinear, time-invariant system with a deterministic input-output relation. For the LMS algorithm various exact and approximative forms of this relation are discussed. Typical nonlinear effects such as the generation of combination frequencies under time-harmonic excitation are predicted.

1. INTRODUCTION

In a widely adopted view, adaptive filters are treated as linear systems with time-varying parameters. The parameter variations are strong during an adaptation phase and approach a weak, but nonzero amplitude under the stationary conditions of statistical equilibrium.

Viewed as a system whose input-output relations are governed by deterministic laws, an adaptive filter should more correctly be described by the attributes time-invariant and nonlinear. Indeed, applying an arbitrary common delay to the input signals amounts to an equal delay in the output signal without any change of its waveform. Further, the input-output relations contain cross-products between pairs of input signals and can exhibit a nonlinear behaviour in terms of the individual input signals.

In this paper we examine the LMS algorithm for an M-length adaptive filter with input signal \( z_k \), reference signal \( d_k \), and output signal \( y_k \). It is governed by the relations

\[
\begin{align*}
\dot{y}_k &= w_k^2 \dot{z}_k, \\
\dot{w}_k &= \text{function of } w_k, z_k, d_k,
\end{align*}
\]

where \( z_k = (z_k, z_{k-1}, \ldots, z_{k-M+1})^T \) and \( w_k = (w_{k0}, w_{k1}, \ldots, w_{k,M-1})^T \) stand for the input vector (representing a finite past of the scalar input signal \( x_k \)) and the weight vector. The adaptation mechanism (2) will be discussed in Section 2.

Like many other adaptive structures the LMS-filter can be viewed as a black box with two inputs \((x_k, d_k)\) and one output \((y_k)\), whose signal transformation can be formally described in the form

\[
y_k = F(x_k, d_k; x_{k-1}, d_{k-1}; \ldots; x_{k-i}, d_{k-i}; \ldots).
\]

The function \( F \) of the present and past values of \( x \) and \( d \) has a nonlinear character. Furthermore it does not explicitly depend upon the time \( k \) of observation so that, if \( x \) and \( d \) are delayed with a certain time, \( y \) undergoes the same delay under preservation of its waveform. These properties find expression in the attributes nonlinear and time-invariant for the systems under discussion. It should be mentioned that many adaptive systems do not admit a closed-form solution for \( F \). The LMS algorithm belongs to the class for which \( F \) can be formulated, cf. (7)-(10) and (1)-(2).

2. DETERMINISTIC ANALYSIS

For the LMS algorithm the updating of the weight vector is governed by the relation

\[
\begin{align*}
\dot{w}_{k+1} &= \dot{w}_k + \epsilon_k \dot{z}_k, \\
\epsilon_k &= d_k - y_k,
\end{align*}
\]

where \( \epsilon_k \) is called the error signal. With (1) we then have

\[
\dot{w}_{k+1} = \dot{w}_k - \dot{z}_k \dot{y}_k + \epsilon_k \dot{z}_k.
\]

Observe that (4) differs from the usual notation in current literature, where \( \epsilon_k \dot{z}_k \) is replaced by \( 2 \mu \epsilon_k \dot{z}_k \). In our notation the adaptation parameter \( \mu \) and the factor 2 are absorbed (after splitting into equal factors) into \( \dot{z}_k \) and \( \epsilon_k \). This implies that the rate of adaptation is directly determined by the amplitude of \( \dot{z}_k \).

The whole dynamics of an LMS filter is governed by (6). In accordance with common terminology in the theory of dynamic systems, (6) is called a state equation, while its solution \( w_k \) is referred to as a state vector. In this terminology the variable \( y_k \) is an output signal which is determined with the auxiliary relation (1).

Although the operator \((d_k, x_k) \rightarrow y_k \) is basically nonlinear, this is not the case for the difference equation (6), that for a given \( z_k \) is linear with time-varying coefficients and can thus be solved in closed form. We
study the evolution of the system between \( k = 0 \) and some \( k = n \) \((n > 0)\), starting with an initial state \( \mathbf{w}_0 \) and ending up with a final state \( \mathbf{w}_n \). Repeated application of (6) for \( k = 0, 1, 2, \ldots \) yields the superposition sum

\[
\mathbf{w}_n = C_{n-1} \mathbf{w}_0 + \sum_{i=0}^{n-1} C_{ni} \mathbf{z}_i.
\]

where

\[
C_{ni} = B_{n-i} B_{n-2} \cdots B_{i+1} \quad (i \leq n - 2),
\]

\[
C_{n,n-1} = I \quad \text{(unit matrix)}.
\]

\[
B_i = I - z_i \mathbf{z}_i.
\]

Observe that \( z_i \mathbf{z}_i \) is a dyadic product and that, in general, the order of the B-matrices in (8) is not interchangeable. Due to the time dependence of the coefficients \( z_i \mathbf{z}_i \) in (6), the solution (7) is not of the convolution type.

3. STOCHASTIC ANALYSIS

Let \( d_k \) and \( x_k \) now represent stationary stochastic signals and let the adaptation process be finished such that \( \mathbf{w}_k \) is stationary, too. Let further the pair \( d_k, x_k \) be partially correlated according to

\[
d_k = \mathbf{w}_k \mathbf{x}_k + n_k,
\]

where \( n_k \) is a stationary signal uncorrelated with \( x_k \) and \( \mathbf{w}_k \) is an \( M \)-length fixed vector. Thus \( d_k \) can be decomposed into a part uncorrelated with \( x_k \) and another one fully correlated through a causal convolutional operator, whose memory has a maximum length equal to \( M \). The more general case of an infinite memory and also that of a noncausal operator are not treated here.

The above decomposition implies a relation between the autocorrelation \( r_k^{xx} = E[x_k x_{k-j}] \) of \( x \) and the cross-correlation \( r_k^{dx} = E[d_k x_{k-j}] \) of \( d \) and \( x \). This reads

\[
r_k^{dx} = \sum_{j=0}^{M-1} r_j^{xx} r_{k-j}^{dx}.
\]

Often the inverse problem has to be solved, in which \( \mathbf{w}_k \) has to be determined from given correlations. For this purpose only the first \( M \) values of \( r_k^{xx} \) and \( r_k^{dx} \) \((l = 0, 1, \cdots, M - 1)\) are required which are conveniently represented by the \( M \times 1 \) correlation vector \( \mathbf{r} = (r_0^{xx}, r_1^{xx}, \cdots, r_{M-1}^{xx}) \) and the \( M \times M \) correlation matrix \( \mathbf{R} \) (with \( R_{mn} = r_{m-n}^{xx} \)). Then we arrive at the Wiener-Hopf relation

\[
\mathbf{p} = \mathbf{R} \mathbf{w}
\]

with the \textit{Wiener solution} \( \mathbf{w} = \mathbf{R}^{-1} \mathbf{p} \).

In our context it is important that under the assumptions stated at the outset, the weight vector \( \mathbf{w}_k \) oscillates around the \textit{Wiener solution} \( \mathbf{w}_0 \). This assertion can be verified by considering the state equation

\[
\mathbf{w}_{k+1} = \mathbf{w}_k - \mathbf{R}_k \mathbf{w}_k + n_k \mathbf{w}_k
\]

for the difference vector \( \mathbf{u}_k = \mathbf{w}_k - \mathbf{w}_0 \) which can be derived from (6) using (11). The mathematical formulation of the assertion then reads \( \mathbf{w}_k = \mathbf{w}_0 + \mathbf{u}_k \). Its proof is facilitated by adding the (rather mild) condition that \( n_k \) is symmetrically distributed (implying \( \mathbf{E} \mathbf{u}_k = 0 \)) and statistically independent of \( x_k \) (which is more than uncorrelated with \( x_k \)). Then from (14) one readily concludes that also \( \mathbf{u}_k \) is symmetrically distributed (i.e., a certain \( \mathbf{u}_k \) and its sign-reversed counterpart are equiprobable) so that \( \mathbf{E} \mathbf{u}_k = 0 \).

4. APPROXIMATE SOLUTION FOR THE WEIGHT FLUCTUATIONS

In this section the weight fluctuations around the Wiener solution are analyzed. Since the exact solution of (14) in a form similar to (7) is difficult to handle, several small-signal approximations are discussed. With increasing amplitude of \( x_k \) these have to be more and more refined \((0^\text{th}, 1^\text{st}, 2^\text{nd}, \text{order solutions})\). With the exception of the zero-order solution they appear in a mixed deterministic-stochastic form and can, in a strict sense, only be used for stochastic excitations. If time-harmonic signals can be approximately viewed as narrow-band stochastic signals, the results are meaningful also in that domain.

The \textit{zero-order solution} is found by neglecting \( z_i \mathbf{z}_i \mathbf{u}_k \) in (14) which is permitted for small \( z \)-values. The solution

\[
\mathbf{u}_k = \sum_{j=-\infty}^{k-1} n_j \mathbf{z}_j
\]

of the resulting difference equation represents a system that simply adds up past values of the product \( n_z \). In this model we encounter the most elementary nonlinearity of the LMS algorithm, viz., the product operation applied to \( n \) and \( z \). For two time-harmonic signals with frequencies \( \Omega_a \) and \( \Omega_b \) the combination frequencies \( [\Omega_a \pm \Omega_b] \) are generated, of which, due to the basic periodicity of any summation of the type (15), only the difference frequency \( |\Omega_a - \Omega_b| \) has considerable amplitude and thus deserves further consideration. With this frequency \( \Omega_a \) the frequencies \( \Omega_k \pm |\Omega_a - \Omega_b| \) occur in the output signal \( y_k \) as determined by (1). While the linear theory with \( \mathbf{u}_k = \mathbf{w}_0 \) predicts the single output frequency \( \Omega_a \), the simple nonlinear model under discussion yields two concomitant side frequencies (which become sidebands in the case of a more general signal \( n_k \)) with a total bandwidth \( 2|\Omega_a - \Omega_b| \).
A drawback of this most rudimentary approach to the nonlinear behaviour of the LMS algorithm is its inherent instability. This becomes evident if \( n_g \) contains a d.c.-term or, in the case of a stochastic signal, a non-vanishing spectral contribution at zero frequency. The latter, however, is the normal situation, which does not occur if and only if the power spectra of \( n_k \) and \( z_k \) do not overlap. (In that case, however, there is no need for an adaptive filter.) In a frequency-domain description of the summation (15) we are faced with a degenerate first-order system with system function \( H(z) = (1 - z^{-1})^{-1} \), which has a pole at \( z = 1 \) and has to be rated amongst the (marginally) unstable systems.

This problem is avoided in the first-order approximation, where the pole at \( z = 1 \) is split up into a number of poles inside the unit circle. This is accomplished by replacing the dyadic product \( z_k^x z_k^y \) in (14) with the correlation matrix \( R \) being the average value of that product. Then (14) passes into a (vector) difference equation with constant coefficients which can be solved with elementary methods. The term \( R u_k \) in the difference equation obviously represents a damping mechanism.

With \( q_k \) and \( u_k \) in (7) replaced with \( n_k \) and \( z_k \) and neglecting the influence of an infinitely remote initial value of \( z_k \), we find

\[
y_k = \sum_{i=-\infty}^{k-1} (B_{k-i} \cdots B_{k+1}) n_i z_i.
\]

In the first-order approximation we have

\[
B_k = I - z_k z_k^x \approx I - R = B
\]

so that (16) passes into

\[
y_k \approx \sum_{i=-\infty}^{k-1} B^{k-i-1} n_i z_i.
\]

Thus we have an input-output relation of the convolution type, reflecting the time-invariant character of the first-order approximation. Compared with (15) past values of \( n_g \) are no longer uniformly weighted, but in accordance with an exponentially decaying memory. So, very remote input signals contribute negligibly to the output signal. This implies that now a constant signal \( n_g \) leads to a finite result and that the unstable character of the previous solution is removed.

Simulations reveal that the present approximation leads to surprisingly accurate results. This can be explained by the fact that in (16) successive values of \( B_k = I - z_k z_k^x \) have to be multiplied which, for sufficiently small \( x \) values, amounts to an addition or averaging of successive dyadic products \( z_k z_k^x \). The assumption in (17) obviously anticipates such averaging and leads to less final errors than might be expected from the rather rough approximation (17).

The vectorial nature of (18) asks for a modal treatment. Let \( \lambda_m \) denote the eigenvalues of \( R \) and \( g_m \) the orthonormal set of eigenvectors \((m = 1, 2, \ldots, M)\). Due to special properties of \( R \), the \( \lambda_m \) are real and nonnegative, while the freedom exists to choose \( g_m \) in real form [1]. The two vectors \( u_k \) and \( z_k \) are now advantageously written as linear combinations of eigenvectors, reading as

\[
y_{km} = \sum_{m=1}^{M} v_{km} g_m, \quad v_{km} = \delta_k g_m, \quad (19)
\]

\[
z_{km} = \sum_{m=1}^{M} x_{km} g_m, \quad x_{km} = \delta_k g_m. \quad (20)
\]

The coefficients \( v_{km} \) and \( x_{km} \) can be interpreted as normal-mode components of the pertinent vectors. From (18) and (17) it can be concluded that

\[
v_{km} = \sum_{i=-\infty}^{k-1} \xi_{k-i} n_i z_{im}, \quad (21)
\]

\[
\xi_m = 1 - \lambda_m. \quad (22)
\]

Obviously, in this model no mode coupling occurs, i.e. the \( m \)th component of \( u_k \) is determined only by the \( m \)th component of \( z_k \). In a more convenient notation this relationship reads as a convolution

\[
v_{km} = h_{km} * (n_k z_{km}), \quad (23)
\]

where the impulse response \( h_{km} \) of the \( m \)th mode is given by

\[
h_{km} = \begin{cases} 0 & \text{for } k < 1, \\ \xi_{k-1} & \text{for } k \geq 1. \end{cases} \quad (24)
\]

A specific mode \( m \) \((m = 1, \ldots, M)\) is thus associated with a first-order system having an exponentially decaying impulse response and a system function

\[
H_m(z) = \sum_{k=1}^{\infty} h_{km} z^{-k} = (z - \xi_m)^{-1} \quad (25)
\]

with a single pole at \( z = \xi_m < 1 \).

Observe that with decreasing amplitude of \( \xi_m \) also \( \lambda_m \) decreases while \( \xi_m \) approaches unity. Then the bandwidth of the pertinent transmission decreases, \( u_{km} \) contains less high frequencies and thus exhibits slower variations, while the impulse response and herewith the system's memory grow in duration.

For time-harmonic excitations with frequencies \( \Omega_1, \Omega_2 \) the basic results of the zero-order solution are not affected. Only for \( \Omega_1 \approx \Omega_2 \) the difference frequency \( |\Omega_1 - \Omega_2| \) becomes so low that the pole shift from the unit circle (at \( z = 1 \)) to its interior becomes tangible. Not only is the amplitude of the low-frequency sinusoidal (frequency \( |\Omega_1 - \Omega_2| \)) reduced due to the damping mechanism, but there occurs also an additional phase shift. Clearly, there remains a low-pass filtering, also in the more general case of a non-sinusoidal excitation. The
result is a slowly varying \( v_k \) on which rather weak oscillations due to the high-frequency components of \( (\hat{n}_{ek}) \) are superimposed.

5. THE SECOND-ORDER SOLUTION FOR THE WEIGHT FLUCTUATIONS

In the rare cases where, due to a relatively high input level, the first-order solution is not sufficiently accurate a further iteration can be applied to improve the result. The essence of such a second-order solution is a refinement of the damping term in (14)

\[
x_{k+1}^2 = R_{uk} + (x_1^2 - R)_{uk} \\
\approx R_{uk} + (x_1^2 - R)_{uk}^{(1)}.
\]  

(26)

Here \( x_{k}^{(1)} \) is the result of the first-order approximation. The error of the approximation (26) is small of second order because the term \( (x_1^2 - R)_{uk} \) itself is small and because the difference between \( x_{k} \) and \( x_{k}^{(1)} \) is small. Further one has to keep in mind that the term under discussion occurs successively and tends to be averaged out.

By virtue of the approximation (26) the feedback in the difference equation (14) is cut out so that the term under consideration is transformed into an additional source term. With \( \hat{u}_k \) replaced by \( \hat{u}_k^{(2)} \) we then obtain

\[
x_{k+1}^{(2)} = x_{k}^{(2)} - R_{uk}^{(2)} + [(R - \hat{x}_k^{(1)})_{uk}^{(1)} + n_k \hat{x}_k].
\]  

(27)

A detailed discussion of the solution of this equation is beyond the scope of this paper. Observe that also here modes are preserved: for \( \hat{x}_k = x_{m}^{(1)} g_m \) (\( g_m \) eigenvector of \( R \)) we have \( \hat{u}_m^{(1)} = u_m^{(1)} g_m \) (see previous section) so that the complete source term in (27) points in the direction of \( g_m \) and hence also \( \hat{u}_m^{(2)} \). Note, however, that due to the strong internal cohesion of \( \hat{x}_k = (x_1, x_2, \ldots)^T \) this is not more than a formal statement, because a pure mode cannot be excited. Indeed, even for a deterministic signal \( x_1 \), a constant direction of the vector \( \hat{x}_k \) can hardly be maintained.

More interesting is the response to two time-harmonic signals with frequencies \( \Omega_x, \Omega_x \). With \( n_k \hat{x}_k \) and \( R_{uk}^{(2)} \) essentially oscillating with frequency \( |\Omega_x - \Omega_x| \), one part of \( u_k^{(2)} \) oscillates with the same frequency. The term \( \hat{x}_k^{(1)} \) consists of a slowly varying part with frequency \( |\Omega_x - \Omega_x| \) and a rapid variation with frequency \( \Omega_x + \Omega_x \) which is weakly "transmitted" to \( u_k^{(2)} \). Thus all contributions to \( u_k^{(2)} \) essentially oscillate with the difference frequency \( |\Omega_x - \Omega_x| \), as was already found for the former, more primitive solutions.

6. CONCLUSIONS

The weight vector and the output signal of an LMS adaptive filter are shown to be nonlinearly dependent on the input and the reference signal. For small signal levels several approximations of the nonlinearities are discussed whose complexity increases with the required accuracy. Under time-harmonic excitation combination frequencies are generated in an adaptive filter which have to be reckoned with in applications (e.g. in the audio field) where a high degree of linearity is required.

ACKNOWLEDGEMENTS

The author wishes to thank A.C. den Brinker and A.W.J. Oomen for the extensive discussions which have substantially contributed to the notions developed in this paper.

FOOTNOTES

† The upper bar denotes time or (due to ergodicity) ensemble averaging.

REFERENCES

FEEDFORWARD AND FEEDBACK IN A SYMMETRIC ADAPTIVE NOISE CANCELER: "STABILITY ANALYSIS IN A SIMPLIFIED CASE"

Stefaan Van Gerven* and Dirk Van Compernolle†

K.U. Leuven - ESAT
Kardinaal Mercierlaan 94
B-3001 Heverlee (Belgium)
TEL: (+32) 16 / 22.99.31
FAX: (+32) 16 / 22.08.55

In this paper we present a stability analysis for the problem of blind signal separation using a symmetric adaptive decorrelation (SAD) algorithm. In the first part the signal separation problem is briefly stated and the feedforward implementation of the algorithm is derived from the interpretation of the classical Widrow LMS noise canceler as a decorrelator. In the new algorithm decorrelation is done between the signal estimate and a "signal free" noise estimate. In a second part we present a feedback implementation of the algorithm. In a third part we consider a simplified case and subject the derived algorithm to a more rigorous analysis. The asymptotic stability is investigated by a linearisation of the algorithm around the desired solution. Apart from limitations on the adaptation gains, convergence and stability are only feasible for a subclass of signal mixtures. These limitations seem to be inherent to the signal separation problem as they arise in both implementations.

PART I: DERIVATION

The Signal Separation Problem. The goal of signal separation techniques is to extract the unknown statistically independent signals $s_i(k)$ out of a set of measurements $y_i(k)$, being unknown convolutive mixtures of the original signals. We will restrict ourselves to a two channel formulation in which the direct couplings equal 1 (cfr. Fig.1) and hence that can be written as:

$$y_1(k) = s_1(k) + h_1(k) * s_2(k)$$
$$y_2(k) = s_2(k) + h_2(k) * s_1(k)$$

Following notations are used in this paper ($i = 1, 2$):

$$\xi_i(k) = E[y_i^2(k)]$$
$$C_{uv}(m) = E[u(k)y(k-m)]$$

Some theoretical derivations are only valid for white noise signals, i.e.:

$$E[s_i(k)s_i(k-m)] = \delta(m)\sigma_i^2$$
$$E[s_i(k)s_i(k-m)] = 0 \quad \forall m$$

The LMS Noise Canceler as a Decorrelator. The solid lines in Fig.1 show a classical adaptive noise canceler, in which a signal estimate of $s_1(k)$ is obtained in $u_1(k)$. The well known LMS performance criterion "minimizing the energy in the signal estimate" corresponds to "decorrelating signal estimate and noise reference", since it can easily be shown that:

$$\frac{\partial E(k)}{\partial w_i^2(m)} = -2C_{u_iu_j}(m)$$

The adaptation step can even be reinterpreted from "gradient descent on the quadratic error surface" to "Newton search of zeros in the crosscorrelation". For white noise signals with no signal leakage (i.e. $H_2(z) = 0$), this is illustrated by:

$$C_{uv}(m) = (h_1(m) - w_1(m))\sigma_2^2$$
$$\nabla_m = \frac{\partial C_{uv}(m)}{\partial w_1(m)} = -\sigma_2^2$$
$$w_1^{(k+1)}(m) = w_1(k)(m) - \gamma \nabla_m^{-1}(w_1(k)y_2(k-m))$$

$$= w_1(k)(m) + \mu(w_1(k)y_2(k-m))$$

The exact knowledge of the gradient sign and approximate knowledge of its magnitude allow for a robust Newton search of the zeros in the crosscorrelations, yielding an "Adaptive Decorrelation (AD)" algorithm which is identical to the LMS algorithm! The stability requirement $\gamma < 1$ allows for an appropriate choice of $\mu < 1/\sigma_2^2$. For non-white noise signals the above formulas are no longer exact but the interpretation of the algorithm is still the same.

*Research Assistant of the I.W.O.N.L.
†Research Associate of the National Fund for Scientific Research of Belgium (N.F.W.O.)
From Noise Cancelling to Signal Separation. The effectiveness of an LMS adaptive noise canceler [1] is proportional to the noise-to-signal ratio in the noise reference, limiting its applicability to "small" $H_2(z)$. Decorrelation of $y_2(k)$ with $w_1(k)$ is still obtained but no longer yields the desired solution. Decorrelating the signal estimate with a "signal free" noise estimate could extend the usage of the LMS algorithm. Looking at Fig. 1 the desirable solution is obvious. The original signals $s_1(k)$ and $s_2(k)$ are obtained in $v_1(k)$ and $v_2(k)$ if $W_1(z) = H_1(z)$ and $W_2(z) = H_2(z)$. If $s_1(k)$ and $s_2(k)$ are independent, then this will also be the case for $v_1(k)$, $v_2(k)$ and $v_3(k), v_4(k)$. The symmetric adaptive decorrelation (SAD) algorithm is directly inferred from the LMS/AD algorithm by replacing $y_3(k)$ by $y_2(k)$ and by adding symmetric formulas for $w_2(k)$. The full SAD algorithm has been presented in [2] and looks as follows:

$$w_1^{(k+1)}(m) = w_1^{(k)}(m) + \mu_1(v_1(k)v_3(k-m))$$
$$w_2^{(k+1)}(n) = w_2^{(k)}(n) + \mu_2(v_2(k)v_4(k-n))$$

Important constraints of this algorithm were discussed in [3] and some implementation requirements are listed below:

- All blocks must be realizable (causal)
- No unstabilities may occur in the postprocessing step, i.e. $(1 - W_1(z)W_2(z))$ must have all its zeros within the unit circle, this condition equally applies to the generating filters $H_1(z)$ and $H_2(z)$

In order to satisfy the first criterion we will restrict ourselves to causal finite length FIR filters, and request that the product of the $z^0$ coefficients differs from 1. The stability constraint cannot trivially be satisfied, since the coefficients must remain in the stable region with every iteration step. It could therefore be argued to omit the post-processing step but this will lead to signal distortion. It therefore looks more like a fundamental constraint in the signal separation problem. However let's consider the typical situation of an LMS noise canceler with a considerable signal leakage. The leakage function $|H_2(z)|$ will still $<$ 1 and the condition will most probably be fulfilled (if $H_1(z)$ is not an amplifier).

**Robustness of Gradient Estimates.** In [2] it is shown that an important difference exists between the one-sided (AD) and symmetric (SAD) algorithms with respect to the quality of the gradient estimate. In the AD-algorithm each crosscorrelation coefficient is function of only one filter tap and the gradient is constant for stationary interference. In the SAD-algorithm the gradient depends also on the goodness of the estimate of the coefficients in the other filter. As a consequence the coupled equations give rise to an inherently unstable region, independent of the adaptation step. This unstable behaviour will NOT occur if all coefficients were set to zero, which helps us in choosing initial values.

**PART II: FEEDBACK IMPLEMENTATION**

Implementation of the Algorithm. Fig.2 shows the feedback implementation which directly gives the solutions $v_1(k)$ and $v_2(k)$. The coefficient adaptation formula is identical to the formulas used in the feedforward case, except that $v$ appears instead of $u$:

$$w_1^{(k+1)}(m) = w_1^{(k)}(m) + \mu_1(v_1(k)v_3(k-m))$$
$$w_2^{(k+1)}(n) = w_2^{(k)}(n) + \mu_2(v_2(k)v_4(k-n))$$

**Stability.** Looking at the feedback loop in the algorithm it is clear that $|W_1(z)W_2(z)|$ must be $< 1$ $\forall z$ on the unit circle. This stability condition is identical to the one derived for the feedforward implementation. Furthermore we assume one of either $w_1(0)$ or $w_2(0)$ to be 0 for implementation of the algorithm. This approach is very similar to the "blind signal separation" described in [4], but their work has been limited to scalar mixing instead of the filter (convolution) mixing treated here. Their (scalar) adaptive connections are placed in a feedback way, resulting in an algorithm with identical unstabilities and spurious solutions, as the ones described in our work. For convolutive mixtures the problem
has also been addressed by [5] but without stability analysis. The choice of the adaptation constants $\mu_1$ and $\mu_2$ is very similar to the way this is done in the classical LMS noise canceler. For equal $\mu$-values the feedforward and feedback algorithm show the same convergence speed. Choosing $\mu$ inversely proportional to the energy of the incoming signals $y(k)$ is a somewhat conservative but a "safe" choice.

PART III: ANALYSIS OF A SIMPLIFIED CASE

Simplification. From now on we will restrict ourselves to a simplified case using only single tap filters [6], i.e.:

$$H_1(z) = h_1(m)z^{-m}$$
$$H_2(z) = h_2(n)z^{-n}$$

Stability. Unstabilities due to positive feedback in the error propagation will be analysed in this paragraph. In the neighbourhood of the desired solution one can linearise the algorithm. Defining

$$\Delta y(m) = w_1^l(m) - h_1^l(m)$$
$$\Delta y(n) = w_2^l(n) - h_2^l(n)$$
$$\theta = h_1(m)h_2(n)$$

and replacing the signal products by their expected values, the matrix formulation looks as follows:

$$\begin{bmatrix} \Delta y_1^{l+1}(m) \\ \Delta y_2^{l+1}(n) \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ \mu_1\sigma_1^2 & \mu_2\sigma_2^2 \end{bmatrix} \begin{bmatrix} \Delta y_1^l(m) \\ \Delta y_2^l(n) \end{bmatrix}$$

This linearisation is valid for both implementations of the algorithm. The absolute value of the eigenvalues of the linearised algorithm must be < 1. This leads to the following conditions:

$$\begin{align*}
\theta^2 &< 1 & \sigma_1 h_1(m)h_2(n) < 1 \\
\mu_1 &> 0 \text{ and } \mu_2 > 0 \\
2(\mu_1\sigma_1^2 + \mu_2\sigma_2^2) &< 4 + \mu_1\sigma_1^2\mu_2\sigma_2^2(1 - \theta^2)
\end{align*}$$

The third condition depends on $\theta$ but can be fulfilled if $\mu_1\sigma_1^2 + \mu_2\sigma_2^2 < 2$, independent of $\theta$. Conditions 2 and 3 then define a triangular region in the $\mu_1, \mu_2$ plane. Condition 1 is identical to the postprocessing and feedback constraint derived earlier and is a property of the mixtures. It is interesting to note that if one of the filters equals zero and therefore must not be estimated, the algorithm collapses to the classical LMS noise canceler and so do the stability conditions. This analysis is only true, however, in the immediate neighbourhood of the "desired solution", where $H_1(z)H_2(z) \approx W_1(z)W_2(z) \approx H_1(z)W_2(z) \approx H_2(z)W_1(z)$.

Alternate Equilibrium Points. The desired solution is an obvious equilibrium point of the algorithm. But in general "de-correlation" has multiple solutions and therefore the adaptive algorithm has multiple equilibrium points. An analytical expression for the cross power spectrum can easily be derived and the alternate solutions are shown to be:

$$w_1^e(m) = \frac{1}{h_1(m)\sigma_1^2 + \mu_2\sigma_2^2}$$
$$w_2^e(m) = \frac{1}{h_2(n)\sigma_2^2 + \mu_1\sigma_1^2}$$
$$w_1^e(m)w_2^e(n) = \frac{1}{h_1(m)h_2(n)}$$

The last equation reveals that $|w_1^e(m)w_2^e(n)| > 1$ whenever $|h_1(m)h_2(n)| < 1$ and vice versa. Stability analysis around this second equilibrium shows that it is UNSTABLE whenever the desired solution is stable, and that this solution is a stable equilibrium if the desired one is unstable.

Simulations. Simulations were carried out with both stable and unstable conditions for the filter coefficients $h_1(m)$ and $h_2(n)$. Different signal energy ratios were used with both $s_1(k)$ and $s_2(k)$ being white noise signals. Alternate solutions were theoretically calculated and convergence from zero initial conditions was controlled. The results of these simulations are summarized in Tab.1 and confirm:
the alternate solution is unstable whenever the desired one is stable and vice-versa

starting from zero initial conditions, the adaptive algorithm will converge to the desired solution if stable and to the alternate solution if the desired one is unstable

for an intrinsically stable situation (i.e., \(|h_1(m)h_2(n)| < 1\)) the decorrelation criterion convergences to the correct solution

<table>
<thead>
<tr>
<th>experiment</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>energy (\sigma_i^2)</td>
<td>40</td>
<td>40</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>coeff. (h_i)</td>
<td>30</td>
<td>30</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>alternate (w_i^a)</td>
<td>1.75</td>
<td>1.143</td>
<td>1.473</td>
<td>1.357</td>
</tr>
<tr>
<td>solution (w_i)</td>
<td>1.143</td>
<td>0.437</td>
<td>1.357</td>
<td>0.368</td>
</tr>
<tr>
<td>corr. solution</td>
<td>stable</td>
<td>unsta.</td>
<td>stable</td>
<td>unsta.</td>
</tr>
<tr>
<td>convergence</td>
<td>corr.</td>
<td>alter.</td>
<td>corr.</td>
<td>alter.</td>
</tr>
</tbody>
</table>

Table 1: Simulation results for SAD stability analysis

Simulations with both \(s_1(k)\) and \(s_2(k)\) being speech signals gave satisfying results. Almost 80% of the energy of the unwanted signals was cancelled, resulting in a improved intelligibility, although distortions of the desired signals were observed.

Identical simulations were carried out with the feedback structure and it shows the same convergence speed, but a slightly better steady state behavior. As for the speech signals less distortions of the desired signals were introduced.

CONCLUSION AND FUTURE WORK

In this paper we have reviewed two symmetric adaptive decorrelation (SAD) algorithms for signal separation. Important convergence and stability constraints were found, but many typical real-life applications fulfill these conditions. Therefore the algorithm is a very useful extension of the classical LMS noise canceler. For small adaptation constants, inherently unstable mixtures converge to a "phantom" solution. This non-uniqueness of the solution can probably be solved by using higher order statistics, although it is suggested in [3] that this might lead to unsurchatable unstabilities. Preliminary simulations still show the existence of a "phantom" solution, but here further work is needed.

REFERENCES


HIGH ORDER LEARNING IN TEMPORAL REFERENCE ARRAY BEAMFORMING

Miguel A. Lagunas, A. Pagès-Zamora, A. Pèrez-Neira

ETSI de Telecomunicación, Departamento de TSC
Apdo. 30002, 08080 BARCELONA, SPAIN

A filter bank able to acquire and track complex sinusoids in the presence of noise and
doppler rate is presented. Firstly, the so-called extended Kalman filter (EKF) which estimates
magnitude, phase and frequency of one sinusoid is derived. Afterwards, the final proposed
scheme is examined. It consists in EKFs whose outputs are driven to the inputs
crossconnecting them as inhibitory cells. Some simulation results are enclosed proving the
excellent performance of the procedure making it a valid alternative to classical spectral
estimation and time-frequency algorithms.

1. INTRODUCTION

The motivation of this work is to increase the efficiency of the communication link between
satellite and users. The goal was faced to be reached from the subject of adaptive array
beamforming with temporal reference [1].

![Fig.1. Basic structure of the receiver.](image)

In these methods, the receiver system forms a dedicated beam to the user which is asking for it.
For that purpose, the user sends a pure tone in a pre-assigned frequency band. From filtering the
output of the beamformer, a noisy sinusoid is obtained in the receiver. Once this signal has been
regenerated by an EKF is driven to modify the weights of the array (beamvector) by means of
different algorithms. The result is a beam which steers to the desired source and nulles the interferences as much as possible [2].

The work deals with the problem of collision produced when two users ask simultaneously for a
dedicated beam (notice that in solving this problem, the performance of the communications
link will be increased). To overcome this problem two beams are needed, time or frequency
multiplexed, each of them steering to one different user. For this reason, the receiver has to be able to
regenerate both reference signals.

Some other works can be found in the literature dealing with similar problems [3],[4].
In these works, one of the signals is assumed to be
the desired one and the other an interference. Therefore, in such a system one signal must be
stronger than the other; furthermore, normally
either magnitude or frequency are assumed to be
a priori known with no doppler rate in anyone of
them. In fact, this was not our situation since the reference signals are , actually, more or less at the
same signal to noise ratio (SNR) and they show up with unknown doppler and doppler rate.

In order to formulate the problem, when a collision appears, the signal received in the
satellite after front end amplification and filtering will be:

\[ x(n) = e_1(n) \exp(j \phi_1(n)) + e_2(n) \exp(j \phi_2(n)) + \omega(n) \]

being \( e_1 \) and \( e_2 \) the magnitude of each pure tone,
\( \phi_1 \) and \( \phi_2 \) the instantaneous phases and \( \omega \)
is a zero mean, white gaussian noise (wgn). The
frequency of both signals, \( f_1 \) and \( f_2 \), is in a range of
1KHz up to 3KHz. The video bandwidth is 4KHz
, normalized 0.5 hereafter, with a central frequency
of 2KHz, being the sampling rate of 8000 samples
per second. The doppler effect is in the range of +/- 15KHz with a maximum doppler rate of 4KHz per
second.

The SNR of each reference was defined as:

\[ SNR = \text{average}\left| e_1^2(n) \right| s^2 ; s^2 = 1 \]

It has been taken the average value because the
magnitudes may fluctuate in time +/-3dB during
tracking [5].
2. EXTENDED KALMAN FILTER

In this section, the EKF is presented as a previous step to achieve the reported scheme. In this application, the EKF is required to estimate magnitude, phase and frequency of a single source signal from a noisy measure of it. The measured signal consists in a pure tone with severe doppler plus wgn.

Readers can wonder why an analog phase-locked loop (PLL) hasn’t been chosen to acquire and track a pure tone located in a given frequency band. First of all, it is required a digital processing since the control of the beamformer is carried out by a ASIC processor together with DSPs. On the other hand, it is necessary to obtain a signal that estimates not only phase and frequency but also the magnitude.

The received signal, in a single source scenario, is assumed to be:

\[ x(n) = e(n) \exp[\phi(n)] + w(n) \]

The model of the system, from which the received signal comes, is described by the next state equations:

\[ x_{n+1} = F x_n + v_n \quad ; \quad z_n^T = [e(n), \phi(n), f(n)] \]

Being \( x_n \) the state vector which contains the discreet values of magnitude, phase and frequency of \( x(n) \), \( F \) the so-called transition matrix which governs the evolution of the state vector, and \( v_n \) the innovation vector, a random vector which introduces uncertainty in the evolution of the state parameters.

\[ F = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 2\pi \\ 0 & 0 & 1 \end{bmatrix} \quad E[v_n v_n^T] = Q(\text{diagonal}) \]

The EKF is described by the following expressions:

\[ \hat{x}_{n+1} = F \hat{x}_n + K_n \epsilon_n \]

\[ \hat{x}_n^T = [\hat{\phi}(n) \cos \phi(n) \quad , \quad \hat{\phi}(n) \sin \phi(n)] \]

being \( \hat{x}_n \) the state estimate of the system, \( K_n \) the so-called gain matrix, \( \epsilon_n \) the signal error between the measured signal, \( z_n \), and the estimated one, \( \hat{z}_n \). (Notice that \( z_n \), \( \hat{z}_n \) and \( \epsilon_n \) are 2-dimensional vectors composed by quadrature and phase components of the original signals).

Up to here, it has been exposed the general Kalman filter [6]. The version named extended Kalman filter comes from the first-order approximation done to set the signal error, \( \epsilon_n \), in a linear dependence of the state error, \( x_n \) (the difference between the actual state, \( x_n \), and the estimated one, \( \hat{x}_n \)).

\[ \epsilon_n = B v_n + \phi_n \quad ; \quad \hat{x}_n = x_n - \hat{x}_n = [\hat{\phi}(n), \hat{\phi}(n), f(n)] \]

Being \( B \) equal to:

\[ B = \begin{bmatrix} 0 & \phi(n) \sin \phi(n) & 0 \\ -\phi(n) \cos \phi(n) & -\phi(n) \cos \phi(n) \end{bmatrix} \]

This expression results from assuming small enough the parameters of state error. From the orthogonality [6] between the state error prediction, \( \hat{x}_n \), and the signal error, \( \epsilon_n \), the gain matrix, \( K_n \), is obtained and therefore, the EKF is completely defined.

In summary, the algorithm which implements the EKF is given hereafter, assuming initial values \( x_0 \) and \( \Sigma_0 \):

\[ \epsilon_n = z_n - \hat{z}_n \quad ; \quad \epsilon_n^T = [\hat{\phi}(n) \cos \phi(n) \quad , \quad \hat{\phi}(n) \sin \phi(n)] \]

\[ \Sigma_n = E[\epsilon \epsilon^T] = H_n \Sigma_n H_n^T + \Lambda_n = E[\omega \omega^T] + \Omega \]

\[ K_n = E[\epsilon \hat{x}_n^T] \Sigma_n = E[\epsilon \hat{x}_n] = E[\epsilon \hat{x}_n^T] - \Omega \]

\[ \hat{x}_{n+1} = \hat{x}_n + K_n \epsilon_n \]

where \( \Sigma_n \) is the state error covariance matrix and \( \hat{x}_n \) the signal that estimates \( z_n \) in magnitude, phase and frequency.

3. MULTITONE TRACKER.

From the EKF of the previous section working over a single line, a novel system which tracks and acquires two or more pure tones is derived.

![Fig.2. Multitone tracker for two signals.](image-url)

It is composed by two EKF in parallel, with feedback from the output of each EKF to the input of the other one. These feedback signals are multiplied by two adaptive weights, \( a_1(n) \) and \( a_2(n) \).
a2(n). Notice that the filter has a symmetry which could force both outputs, y1(n) and y2(n), to be equal. This symmetry is broken thanks to the different values of the weights.

The evolution of a1(n) and a2(n) is controlled in order to make the outputs as orthogonal as possible. Intuitively, this rule seems to be correct since the desired signals have different frequency and, therefore, they are orthogonal between each other.

In order to achieve the desired orthogonality between y1(n) and y2(n), it is necessary to minimize the scalar product between both outputs in a given interval of time T.

\[ \varphi = \left( \int_{T} y_1(t) y_2^*(t) dt \right)^2 \quad \text{Eq}(1) \]

It is desired a1 and a2 drive \( \varphi \) to a minimum. That is why it is obtained the derivative of function \( \varphi \) with respect to each weight. We will focus the case of a1 since the reader may conclude in the same manner the corresponding learning rule for a2.

\[ \frac{d\varphi}{da_1} = \int_{T} y_1(t) y_2(t) \frac{dy_1^*(t)}{da_1} dt \]

Being the output of the upper EKF denoted by \( y_1(t) \), we may say that \( y_1 \) obeys to:

\[ y_1 = g(x - a_1 y_2) = g(z) \]

being \( g(\cdot) \) a non-linear function. It is possible to express:

\[ \frac{dy_1}{da_1} = \frac{dy_1}{dz} \frac{dz}{da_1} = (-y_2) \frac{dy_1}{dz} \]

In order to obtain a closed form of Eq(1) some approximations have to be carried over. Firstly, it is considered \((dy/dz)^*\) as a constant and, secondly, it is taken the instantaneous value of the integral assuming a small interval of time T. From these approximations, a linear rule, based on gradient methods, has been chosen to update the weights:

\[ \frac{d\varphi}{da_1} = -|y_2(n)|^2 \frac{dy_1^*(n)}{y_1(n)} y_1(n) \]

\[ a_1(n + 1) = a_1(n) + m_1 |y_2(n)|^2 y_1(n) y_1(n)^* \quad \text{Eq}(2) \]

being \( m_1 \) a parameter which controls the trade-off between the speed of the weights in getting a value near the optimum and the final accuracy. From Eq(2) is possible to infer the expression from which a2 is updated:

\[ a_2(n + 1) = a_2(n) + m_2 |y_1(n)|^2 y_2(n) y_2(n)^* \]

The structure formed by the EKFs in parallel with inhibitory cells was inspired in the pioneer work of C.Jutten [7] and further works of the same author [8]. In these works the author proposed an unsymmetrical rule for learning of the inhibitory cells. Afterwards, in the field of high order signal processing applied to blind equalization [9], the four order objective for orthogonalizing two signals was reported. Herein, the authors prove that both approaches are connected and that unsymmetrical third order learning rules are strongly related with minimization of the cross-correlation between two signals.

One of the most important aspects of the method is the simplicity in the extension of the filter in order to acquire and track more than two signals. For instance, let us assume a signal which contains four different tones:

\[ x(n) = \sum_{i=1}^{4} e_i(n) \exp[i \omega_i(n)] + \alpha(n) \]

The filter can be designed as the neuron structure formed by four EKF of Fig.3.

\[ \text{Fig.3. Multitone tracker of four signals} \]

Each output is driven to the input of the other three EKFs multiplied by three different weights. The weight \( a_{ij} \) represents the weight which multiplies the output j and the result is driven to input i. Going on the same idea developed above, each weight \( a_{ij} \) is adaptive and its learning is done in order to make the output j and output i as orthogonal as possible. Therefore, as Eq(2):

\[ a_{ij}(n + 1) = a_{ij}(n) + m_{ij} |y_j(n)|^2 y_j(n) y_i(n)^* \]

4. SIMULATIONS

The first group of simulations corresponds to the filter of Fig.4. The plots, in this figure, represent magnitude and frequency versus discret time (samples), of the incoming signals. The input signal contains two tones of (-3 dB) and (0 dB) of SNR. The doppler rate was sinusoidal with a
maximum of +/- 1.5KHz and rate, on average, of 3.2KHz/sec.

Fig. 4, Magnitude and frequency versus time.

The figures have been obtained with m1=m2=1E-3 and the diagonal values of matrix Q has been set to 1E-4, 1E-5 and 1E-6. It can be observed that, in the same signal, frequency is acquired before magnitude and the estimation is also better. The reason for this behaviour is the fact that the received signal has more noise in magnitude than in phase (that’s why generally, phase modulations are more suitable than magnitude ones).

The following plots are magnitude and frequency from the filter of Fig. 3. The diagonal values of matrix Q are set to 1E-6, 1E-6 and 1E-7. The values of mij are in the range from 1.5E-2 to 2E-2.

Fig. 5, Magnitude and frequency versus time.

It is distinguished three different parts: with one, two and four incoming signals. When there are four inputs the SNR of them is of 10, 7, 4 and 1. It is important to point out that at the output are obtained as many signals as input signals are. For instance, firstly there is only one input signal which is acquired and tracked by only one output since the rest of the outputs have a null magnitude.

REFERENCES
An Adaptive Morphological Filtering Method

Pauli Kuosmanen, Lasse Koskinen, and Jaakko Astola
Tampere University
Department of Mathematical Sciences
Tampere, Finland

Abstract

This paper introduces a new type of adaptive filters. It is based on mathematical morphology, especially on soft morphological filters. In this paper, we also present a generalization of soft morphological filters. In the same way as standard and soft morphological filters this generalization can be expressed and analyzed in the framework of stacked filters. Examples demonstrating the usefulness of the new adaptive morphological filtering method are provided. Finally, a biomedical application where promising results are obtained with the new method is presented.

1. Introduction

The field of adaptive systems design has been the subject of considerable effort of research in the area of control and signal processing for more than 30 years, cf. e.g. Honig et al. During this time adaptive filters have been applied to many applications including medical ones, Linkers. In this paper we present a new adaptive morphological filtering method and a biomedical application.

As to morphological methods, they have proven useful in many image processing applications Serra, Haralick et al., Maragos et al. and Dougherty et al. The main reason for their success is that they can well take into account the geometrical shape of the analyzed objects. On the other hand they have high sensitivity to noise or defects in the image, Koskinen et al. and Stevenson et al. Recently Koskinen et al. introduced soft morphological operations, which have a better performance than standard morphological operations in noisy conditions.

The idea of soft morphological operations is to relax the standard definitions a little, in such a way that a degree of robustness is achieved while most of the desirable properties of standard morphological operations are maintained. Standard discrete morphological filters, which process signals by sets, are based on local maximum and minimum operations. In soft morphological filters maximum and minimum are replaced by more general weighted order statistics.

2. Generalized Soft Morphological Operations

We now define generalized soft morphological operations. They are most naturally defined in the framework of weighted order statistics.

Let $A$ and $B$, $A \subset B$, be finite subsets of $Z^2$ and let $k$ and $i$ be natural numbers such that $1 \leq k < |B|$, $1 \leq i < k$. For $x \in Z^2$ and $T \subset Z^2$ we denote the "translated set" by $T_x$, that is,

$$T_x = \{ x + s : s \in T \}.$$

The symmetric set of $T$ is the set

$$T^* = \{ -t : t \in T \}.$$

We denote repetition operation of any object, in our case real numbers, by $\overset{k}{\circlearrowright}$, that is $k\overset{}{\circlearrowright}x = \overbrace{x, \ldots, x}^k$. A multiset is a collection of objects, where repetition of objects is allowed. For example $\{1, 1, 1, 2, 3, 3\} = \{3\overset{2}{\circlearrowright}1, 2\overset{2}{\circlearrowright}3\}$ is a multiset.

The basic generalized soft morphological operations are generalized soft erosion and generalized soft dilation:

Generalized soft erosion of $f$ by $[B, A, k, i]$ is denoted by $f \overset{k}{\circlearrowright} [B, A, k, i]$ and is defined by $f \overset{k}{\circlearrowright} [B, A, k, i](x) = k^{th}$ smallest value of the multiset

$$\{k \overset{}{\circlearrowright} f(a) : a \in A_x \} \cup \{i \overset{}{\circlearrowright} f(b) : b \in (B - A)_x \}.$$
Generalized soft dilation of $f$ by $[B, A, k, i]$ is denoted by $f \odot [B, A, k, i]$ and is defined by

$$f \odot [B, A, k, i](x) = \max \{k \odot f(a) : a \in A\} \cup \{i \odot f(b) : b \in (B - A)\}.$$ 

The set $B$ is called the structuring set, $A$ its centre, $B - A$ its boundary, $k$ the order index of its centre and $i$ the order index of its boundary.

Thus, the definition of generalized soft morphological operations is similar to the definition of soft morphological filters. The differences are that the condition $1 \leq k \leq \min\{|B|/2, |B - A|\}$ is relaxed and that the soft boundary has also an order index.

**Example 1.** Let the structuring set be $B = \{(-1, 0), (0, 1), (0, 0), (0, -1), (1, 0)\}$, $A = \{(0, 0)\}$ and the order indexes $k = 4, i = 1$. Then the generalized soft erosion by $[B, A, k, i]$ is defined by $f \ominus [B, A, k, i](x) = \min \{f(x_1 - 1, x_2), f(x_1, x_2 + 1), f(x_1, x_2 - 1), f(x_1 + 1, x_2)\}$.

The output of this filter at point $x = (x_1, x_2)$ is $f(x)$ unless all the values of set $\{f(b) : b \in (B - A)\}$ are smaller than $f(x)$, when the output is the largest value of set $\{f(b) : b \in (B - A)\}$.

Generalized soft opening and closing are defined in the usual way:

Generalized soft opening of $f$ by $[B, A, k, i]$ is denoted by $f \oplus [B, A, k, i]$ and is defined by

$$f \oplus [B, A, k, i](x) = (f \ominus [B, A, k, i]) \ominus [B^*, A^*, k, i](x).$$

Generalized soft closing of $f$ by $[B, A, k, i]$ is denoted by $f \oslash [B, A, k, i]$ and is defined by

$$f \oslash [B, A, k, i](x) = (f \oplus [B, A, k, i]) \ominus [B^*, A^*, k, i](x).$$

Generalized clos-opening and open-closing are defined to be closing followed by opening and opening followed by closing respectively.

3. Some Properties of Generalized Soft Morphological Filters

Generalized soft morphological filters have some attractive properties. Kuoosman et al. constructed a class of idempotent generalized soft morphological filters: generalized soft morphological operations by $B, A, [B] - 1, 1]$, where $B$ is the structuring element and $A = \{(0, 0)\}$ (two-dimensional) or $A = \{0\}$ (one-dimensional). Kuosman et al. also showed that generalized soft morphological filters preserve shapes and details of the original image well and filter out noise at the same time.

---

Figure 1.
Figure 1 illustrates differences in 2-dimensional generalized soft closing when B and A are the same as in our example 1. (a) is the contour plot of the original image, (b) is the contour plot of a corrupted image. In (c) – (f) are images after closing by B, where the order index k varies from 1 to 4 respectively and i = 1 in all cases. Thus (c) is the image after standard closing, which is known to be idempotent. Generalized soft closing in (f) belongs in the class of idempotent generalized soft morphological filters Kuosmanen et al. constructed.

We can see from figure 1, that the same structuring set with the same centre acts differently when the order index of the centre varies. Koskinen et al. pointed out that soft closing is less sensitive to impulsive noise than standard closing, which can be seen by comparing (c) and (d). Both standard closing (c) and soft closing with index k = 2 (d) had difficulties with valleys of width 1, which they filled. Soft closing (d) also lost the corners of the original image.

4. A New Adaptive Morphological Filter

Once the structuring element and its hard centre are fixed the parameters which determine the generalized soft morphological filter are the order indexes k and i. The structuring element and its hard centre are determined using a priori knowledge about the filtering problem. Also the allowed variation on order indexes can be determined before if one aim of the filtering is to preserve some details of the input. As an analytical optimization of the indexes k and i would require unreasonably careful modelling of the problem (data permitting such modelling is seldom available) an adaptive scheme for choosing the indexes is proposed.

The scheme is based on minimization of a cost function over the filters having the same structuring set. The choice of the cost function depends on the problem. Our experiments have shown that mean absolute error or Hamming distance between the filtered and desired signal perform well while mean squared error tends to destroy shapes. Also cost fuctions which reveal the difference of the shape of the desired and filtered data/image can be used.

One aim of the filtering process is often to remove outliers. Barnett et al. divided outliers into two types. The first type is when an isolated measurement error is superimposed on an otherwise reasonable realization of the process. Its effect can be obvious and it does not reflect in the values of adjacent observations. The other type is when a more inherent discordancy arises. and is reflected by the structure of the process in neighbouring (usually later) observations. The detection of outliers becomes more problematic then.

Figure 2 illustrates also the difficulties and importance of the study of outliers. In figure 2 a part of a process is presented. Although it may look out stationary and have some outliers is does not have any. In fact the process is

\[ X_t = 0.5 * X_{t-1} + 0.9 * Y_{t-1} + Y_t, \]

where \( Y_t \) are independent \( N(0,1) \).

If the expected number of outliers in the measured data is known, it is possible to design an adaptive generalized soft morphological filtering system, which optimizes the percentage of removed possible outliers. The optimization process is performed over the class of idempotent generalized soft morphological filters.

Figure 2.
5. Application

The new filtering method has been in use in a biomedical experiment at Tampere University. The purpose of the experiment was to examine the levels and variability in blood pressure of overweight persons before and after a reducing diet. The pulse rates were also examined. Kalli et al.\textsuperscript{4} stated that blood pressure signal processing needs non-linear smoothing, which is capable of preserving sharp changes in the data and still able to filter out noise and other fast components superimposed on the data. The measurement of blood pressure has been done by using a new continous, non-invasive tonelink in which the amount of samples varies from 50 to more than 100 in a minute depending on the pulse rate.

In this experiment there exist many criteria for the goodness of the output after filtering process. For example, there is a need for such a filter, which removes the most possible outliers (of type one), when the percentage of them is known. Our experiments showed that an adaptive method is necessary for achieving this and the method presented before worked well.

The proposed adaptive morphological methods turned out to be promising biological signal processing methods, because the filters are able to preserve and identify trends and patterns of different length from longterm records. The freedom for the choice of the cost function makes it suitable for different situations where the criteria for the goodness of the output varies.

References

A REALIZATION METHOD OF 2-D SEPARABLE-DENOMINATOR DIGITAL FILTER 
BASED ON REDUCED-DIMENSIONAL DECOMPOSITION

Syu-jii Kawasaki, Nozomu Hamada

Department of Electrical Engineering, Faculty of Science and Technology, 
KEIO University, 223, Hiyoishi, Yokohama, Japan

abstract The class of 2-D separable-denominator(SD) digital filter is attractive for designing 
and implementing the 2-D signal processing. This paper presents a new technique for 
synthesizing 2-D SD digital filter based on the reduced-dimensinal decomposition(RDD).
The proposed filter structure is the cascade of a SIMO(single input/multi output) output 
 tapped 1-D filter and a MISO input tap 1-D filter. Thus the multiple outputs of the first 
filter are injected to the inputs of the second filter. The systematic synthesis procedure
is derived by giving the explicit input/output relationships for two dual 1-D filters. The 
result carried out a specific canonical structure of a RDD idea proposed by Lin, et.al[1].

1 INTRODUCTION

In digital processing of multidimensional data, such as video or image, seismic, radar, sonar, etc., multidimensional digital filter is an important tool, and many efficient filter structure have been proposed. Especially, the filter that is represented by separable-denominator(SD) transfer function is attractive in the sense that its stability is easily checked by 1-D stability criteria and it can be implemented by simple way.

In the last decade, a general treatment of 2-D digital filter is developed via the reduced-dimensionnal decomposition(RDD) by T.Lin et al[1], or several other reseachers. In those works, Roesser's state-space representation model is used and their balanced realization, low sensitivity or minimal roundoff noise structure, and controllability/observability aspects are studied.

On the other hand, Taguchi and Hamada[2] proposed lattice form structure for realizing 2-D SD digital filter. They have done the work with equivalent transformation of direct form state-space model into the lattice form realization. The obtained structure can be also derived as a specific realization model of RDD point of views.

And the conventional realization methods based on RDD stand on such concepts as, singular value decomposition[6], LU decomposition[6], etc.

This paper gives more systematic and simpler method of realizing 2-D SD digital filter than the conventional RDD realizations using simple algebraic calculations, and reveals that the practical RDD realization procedure is brought by dividing whole system into three blocks, single-input/multi-output(SIMO) 1-D filter, linear matrix operation block, and MISO 1-D filter.

2 THE REALIZATION METHOD

Any 2-D SD transfer function can be written as

$$H(z_1, z_2) = \frac{N(z_1, z_2)}{D(z_1, z_2)}$$

$$= \begin{bmatrix} z_2^{L-1} & \cdots & z_2 \end{bmatrix} \begin{bmatrix} n_{00} & \cdots & n_{0K} \\ \vdots & \ddots & \vdots \\ n_{L0} & \cdots & n_{LK} \end{bmatrix} \begin{bmatrix} z_1^{-1} \\ \cdots \\ z_1^{-K} \end{bmatrix}$$

$$= \left( \sum_{k=0}^{K} d_{2k} z_2^{-k} \right) \left( \sum_{k=0}^{K} d_{1k} z_1^{-k} \right)$$

If one would like to realize this 2-D transfer function via a method of matrix decomposition, such as the singular value decomposition[6], the Jordan decomposition[6], the LU decomposition[7], or maximum rank decomposition[1], though each algebraic meaning of these decompositions are different, all these methods will reduce (1) into the inner product form

$$\begin{bmatrix} \eta_0(z_2) \\ \eta_1(z_2) \\ \vdots \\ \eta_K(z_2) \end{bmatrix} = \begin{bmatrix} \zeta_0(z_1) \\ \zeta_1(z_1) \\ \vdots \\ \zeta_K(z_1) \end{bmatrix}$$

where $r$ is the rank of matrix $N$. These methods are called RDD, and are used to realize 2-D SD digital filter with $r$ 1-D filter pairs $(\zeta(z_1), \eta(z_2))$(Fig.1). In these methods, denominators $D_1(z_1), D_2(z_2)$ are realized within each block respectively. And (2), the inner product of two 1-D transfer function vector can be understood as the extension of simple product of two scalar 1-D transfer functions

$$H(z_1, z_2) = H_2(z_2)H_1(z_1)$$

But in these methods, the number of multipliers are relatively large.

[Fig.1]
The proposed realization method is briefly summarized as follows: that is, we realize the desired 2-D SD digital filter with two 1-D digital filters and intermediate matrix operation. Denominators \( D_1(z_1) \) and \( D_2(z_2) \) are realized in each 1-D filter block respectively, and this is the same idea as the method of inner product of two 1-D transfer function vector, as expressed in (2).

Numerator polynomial \( N(z_2, z_2) \) is, however, realized more flexible fashion; it is done such that the 2-D SD digital filter will be realized by cascade connection of three blocks: 1. the first block: 1-D digital filter with delay element only \( z_1 \), transfer function vector \( \mathbf{Z}_1 / D_1(z_1) \) where \( \mathbf{P} \) is a \((K+1)\times(K+1)\) matrix 2. the intermediate block: memoryless coefficients multiplier with a \((L+1)\times(L+1)\) matrix \( \mathbf{M} \) 3. the last block: 1-D digital filter with delay element only \( z_2 \), transfer function vector \( \mathbf{Z}_2^T Q / D_2(z_2) \) where \( Q \) is a \((L+1)\times(L+1)\) matrix

And it is illustrated in Fig. 2. In Fig. 2, \( z_k \) \((k=0,1,...,K)\) are \( l \)-th internal state-variable of the first block, and \( \mathbf{z}_l \) \((l=0,1,...,L)\) are \( l \)-th input to the last block, i.e.,

\[
\mathbf{z}(n,m) = \mathbf{M} \mathbf{x}(n,m)
\]

where \( \mathbf{x} = [x_0, ..., x_K]^T, \mathbf{z} = [z_0^T, ..., z_L^T] \).

Now, let us assume we could realize 1-D SIMO first block by using some specific structure. Then, its transfer function vector from input \( \mathbf{U}(z_1, z_2) \triangleq z_1z_2 [u(n,m)] \) to \( \mathbf{X}_1(z_1) \triangleq z_1z_2 [x(n,m)] \) can be written in the form

\[
\left[ \begin{array}{c}
\zeta_0(z_1) \\
\zeta_1(z_1) \\
\vdots \\
\zeta_K(z_1)
\end{array} \right] = \frac{1}{D_1(z_1)} \left[ \begin{array}{c}
P_0(z_1) \\
P_1(z_1) \\
\vdots \\
P_K(z_1)
\end{array} \right]
\]

\[
= \frac{1}{D_1(z_1)} \left[ \begin{array}{c}
1 \\
p_0 \cdot z_1^{-1} \\
\vdots \\
p_K \cdot z_1^{-K}
\end{array} \right] \left[ \begin{array}{c}
P_0 \\
P_1 \\
\vdots \\
P_K
\end{array} \right] = \frac{\mathbf{P}}{D_1(z_1)} \mathbf{Z}_1
\]

Similarly, if we could realize 1-D MISO last block in some specific structure, its transfer function from \( \mathbf{X}_1(z_1) \triangleq z_1z_2 [x(n,m)] \) to \( \mathbf{Y}(z_1, z_2) \triangleq z_1z_2 [y(n,m)] \) can be written as

\[
\mathbf{z}(n,m) = \frac{1}{D_2(z_2)} \left[ \begin{array}{c}
\mathbf{Q}_0(z_2) \\
\mathbf{Q}_1(z_2) \\
\vdots \\
\mathbf{Q}_L(z_2)
\end{array} \right]
\]

\[
= \frac{1}{D_2(z_2)} \left[ \begin{array}{c}
1 \\
\mathbf{Q}_0 \cdot z_2^{-1} \\
\cdot \\
\mathbf{Q}_L \cdot z_2^{-L}
\end{array} \right] \left[ \begin{array}{c}
\mathbf{P}_0 \\
\mathbf{P}_1 \\
\vdots \\
\mathbf{P}_K
\end{array} \right] = \frac{\mathbf{Z}_2}{D_2(z_2)} \mathbf{Q}
\]

Therefore we get the realization of \( \mathbf{H}(z_1, z_2) \) of whole 2-D system as the cascade of three blocks:

\[
\left( \begin{array}{c}
\mathbf{Q}_0(z_2) \\
\mathbf{Q}_1(z_2) \\
\vdots \\
\mathbf{Q}_L(z_2)
\end{array} \right)^T \left( \begin{array}{c}
\mathbf{P}_0(z_1) \\
\mathbf{P}_1(z_1) \\
\vdots \\
\mathbf{P}_K(z_1)
\end{array} \right) = \left( \begin{array}{c}
\mathbf{P}_0(z_1) \\
\mathbf{P}_1(z_1) \\
\vdots \\
\mathbf{P}_K(z_1)
\end{array} \right)
\]

\[
= \frac{\mathbf{Z}_2^T \mathbf{M} \mathbf{Z}_1}{D_2(z_2)D_1(z_1)}
\]

Now the intermediate coefficient is determined by

\[
\mathbf{M} = \mathbf{Q}^{-1} \mathbf{N} \mathbf{P}^{-1}
\]

where we assumed both \( \mathbf{Q} \) and \( \mathbf{P} \) are regular matrices.

In the conventional inner product methods described above require \( r \times (K+L) \) multipliers in general. But in our method, \( K \times L \) multipliers are needed. If \( N \) is full rank, \( r=K \), and \( K=L \), then multipliers required in our method become as half as the required in conventional methods.

3. REALIZATION BY DIRECT AND LATTICE FORM

In this section, we realize 2-D SD digital filter applying the method described in section 1. Direct form and Gray-Markel lattice form are adopted as realization structure.

Direct form

Here we consider the mutually dual pair, controller canonical form and observer canonical form[12] shown in Fig. 3 to realize 2-D SD digital filter. And let us adopt the controller canonical form as indicated in Fig. 3(a) with dashed line for 1-D SIMO first block. Its transfer function vector become as

\[
\frac{1}{D_1(z_1)} \left[ \begin{array}{c}
P_0(z_1) \\
P_1(z_1) \\
\vdots \\
P_K(z_1)
\end{array} \right] = \frac{1}{D_1(z_1)} \left[ \begin{array}{c}
1 \\
z_1^{-1} \\
\vdots \\
z_1^{-K}
\end{array} \right] = \mathbf{I}_K \mathbf{Z}_1
\]

And, on the other hand, if we adopt observer canonical form as indicated in Fig. 3(b) with dashed line for 1-D MISO last block, its transfer function vector become as

\[
\frac{1}{D_2(z_2)} \left[ \begin{array}{c}
\mathbf{Q}_0(z_2) \\
\mathbf{Q}_1(z_2) \\
\vdots \\
\mathbf{Q}_L(z_2)
\end{array} \right] = \frac{1}{D_2(z_2)} \left[ \begin{array}{c}
1 \\
z_2^{-1} \\
\vdots \\
z_2^{-L}
\end{array} \right] = \mathbf{Z}_2^T \mathbf{I}_L
\]

where \( \mathbf{I}_K, \mathbf{I}_L \) are \((K+1)\)-th, \((L+1)\)-th order identity matrices respectively. Then in this occasion, intermediate coefficient matrix \( \mathbf{M} \) become

\[
\mathbf{M} = \mathbf{N}
\]
This realization form may be referred as 2-D direct form (SD), because all multiplier coefficients in this filter are directly correspond to the coefficients of desired 2-D transfer function. And, similar realization patterns may be obtained by adopting another dual pair, controllability canonical form and observability canonical form[12].

Lattice form

Here, we adopt ordinary ARMA type lattice filter(output-tap type)[8][9][10] for 1-D SIMO first block, and input-tap lattice filter for 1-D MISO last block (Fig 4).

According to well-known Levinson algorithm in the linear prediction theory, the transfer function vector of (1-D) ordinary output-tap type lattice filter can be written as

\[
\frac{1}{D_{k}(z_{k})} \begin{bmatrix}
    P_{k}(z_{k}) \\
    P_{k-1}(z_{k}) \\
    \vdots \\
    P_{1}(z_{k}) \\
    P_{0}(z_{k})
\end{bmatrix} = \frac{1}{A_{k}^{(1)}(z_{k})} \begin{bmatrix}
    z_{1}B_{k}^{(1)}(z_{1}) \\
    z_{1}B_{k-1}^{(1)}(z_{1}) \\
    \vdots \\
    z_{1}B_{k-K}^{(1)}(z_{1})
\end{bmatrix}
\]

\[
= \frac{1}{A_{k}^{(1)}(z_{k})} \begin{bmatrix}
    b_{0}^{(1)} \\
    b_{0}^{(1)} \\
    \vdots \\
    b_{0}^{(1)} \\
    b_{0}^{(1)}
\end{bmatrix} \begin{bmatrix}
    1 \\
    z_{1}^{-1} \\
    \vdots \\
    z_{1}^{-K}
\end{bmatrix}
\]

\[
= B_{k}^{(1)}Z_{k}^T A_{k}^{(1)}(z_{k})
\]

where \(P_{k}(z_{k})\) \((k=0,1,...,K)\) are the numerator polynomials of transfer function between input and each \(k\)-th output tap, and are \(k\)-th order in \(z_{1}^{-1}\). Here, \(b_{0}^{(1)}\) is the coefficient of \(z_{1}^{-j}\) of \(zB_{k}^{(1)}\) determined by Levinson algorithm

\[
A_{k}^{(1)}(z_{1}) = \begin{cases}
    1 & \text{if } k = 0,1,\ldots,K \\
    1 & \text{if } k = 0,1,\ldots,K
\end{cases}
\]

and especially \(b_{0}^{(1)}(k=0,1,...,K)\) are equal to unity, and

\[
z_{1}B_{k}^{(1)}(z_{1}) = 1.
\]

On the other hand, transfer function vector of input-tap type lattice filter can be written as (see Appendix for derivation)

\[
\frac{1}{D_{k}(z_{k})} \begin{bmatrix}
    A_{0}^{(2)}(z_{2}) \\
    A_{1}^{(2)}(z_{2}) \\
    \vdots \\
    A_{k}^{(2)}(z_{2})
\end{bmatrix} = \frac{1}{A_{k}^{(2)}(z_{2})} \begin{bmatrix}
    c_{0}A_{0}^{(2)}(z_{2}) \\
    c_{1}z_{2}^{-1}A_{1}^{(2)}(z_{2}) \\
    \vdots \\
    c_{L}z_{2}^{-L}A_{L}^{(2)}(z_{2})
\end{bmatrix}
\]

\[
= \frac{1}{A_{k}^{(2)}(z_{2})} \begin{bmatrix}
    a_{0}^{(2)} \\
    a_{0}^{(2)} \\
    \vdots \\
    a_{0}^{(2)} \\
    a_{L}^{(2)}
\end{bmatrix} \begin{bmatrix}
    z_{2}^{-1} \\
    \vdots \\
    z_{2}^{-L}
\end{bmatrix}
\]

\[
= \frac{Z_{k}^{T}A^{(2)}}{A_{k}^{(2)}(z_{2})}
\]

(14)

where \(A_{k}^{(2)}(z_{2})(k=0,1,\ldots,K)\) are the numerator polynomials of transfer function between \(k\)-th input tap and output, and are all \(L\)-th order. Here, \(c_{0}A_{0}^{(2)}(z_{2}) = A_{0}^{(2)}(z_{2})\), and \(a_{L}^{(2)}(k=0,1,\ldots,L)\) are equal to unity.

In this lattice form realization, we can easily calculate coefficient matrix \(M\) as \([A^{(2)}]^{-1}N[B^{(2)}]^{-1}\), because both \(B^{(1)}, A^{(2)}\) are triangular matrices.

4. Concluding remarks

In this paper, we presented an approach for realizing 2-D SD denominator digital filter. The concept of the proposed method is based on the RDD by which three blocks, i.e. \(z_{1}\)-directional operation block, intermediate block, and \(z_{2}\)-directional operation block were assembled into 2-D system.
This method can be said of the approach that most utilized the advantages of SD transfer function, because the first and the last block are constructed independently so as to realize each denominator polynomials $D_1(z_1), D_2(z_2)$.

**APPENDIX**

Derivation of (14):

Basic idea of finding transfer function between $u_i$ and $y$ is as follow. That is, we divide whole input-tap lattice filter into three parts $F_i(z), G_i(z), zB_{L-1}(z)/A_{L-1}(z)$ as shown in Fig.5.

If we take state $V_i(z)$ at the position shown in Fig.5, then

$$V_i = \frac{1}{1 - F_i zB_{L-i}} A_{L-i}$$  (15)

and

$$\frac{Q_i}{D_2(z_2)} = Y = \frac{G_i}{U_i} = \frac{G_i}{U_i} A_{L-i} zB_{L-i} A_{L-i}$$  (16)

where $zB_{L-i}/A_{L-i}$ are transfer function of $(L-1)$-th order lattice filter of right side of input $u_i$, and $zB_{L-i}$ are reciprocal polynomials of $A_{L-i}$. We can find $zB_{L-i}/A_{L-i}$ from $A_{L}(z)$ by using Levinson's stepdown recursion. Then, using Levinson's recursion, transfer function $F_i$ and $G_i$ become

$$F_i = \frac{zB_{L-i}V_i}{A_{L-i}} \quad G_i = \frac{A_{L-i}V_i}{A_{L-i}V_i} = \frac{A_{L-i}}{A_{L-i}}$$  (17)

$F_i$ and $G_i$ have recursive relation such as

$$F_i = \frac{z^{-1}(F_{i-1} - \Delta_{L-i+1})}{1 - \Delta_{L-i+1} F_{i-1}}$$  (18)

$$G_i = \frac{A_{L-i}}{A_{L-i}} = \frac{A_{L-i}}{A_{L-i}^2} = \frac{G_{i-1}(z^{-1} + \Delta_{L-i-1}(F_{i-1}))}{A_{L-i}}$$  (19)

Using (18), (19), we can exactly simplify the calculation of $F_i, G_i$. From $F_i, G_i, zB_{L-i}/A_{L-i}$ as mentioned above, we can determine transfer function $Q_i/D_2(z_2)$ of $(u_i \rightarrow y)$ by (7). For example, $Q_0/D_2, Q_1/D_2, Q_2/D_2, Q_3/D_2$ become as

$$Q_0 = \frac{A_L}{A_L} = 1$$

$$Q_1 = \frac{z^{-1} + \Delta_{L}(-\Delta_{L}z^{-1})}{1 - (-\Delta_{L}z^{-1})(A_L - \Delta_{L}zB_L)/(1 - \Delta_{L})}$$

$$Q_2 = \frac{1 - \Delta_{L}^2}{(1 - \Delta_{L+1}z^{-1})A_{L-2}}$$

$$Q_3 = \frac{1 - \Delta_{L}^2}{(1 - \Delta_{L+1})z^{-1}A_{L-2}}$$

It is easy to show

$$Q_i = \prod_{n=0}^{i-1} (1 - \Delta_{L-i}) \quad z^{-1} A_{L-i}$$  (20)

**References**


ADAPTIVE PREDICTION AND INTERPOLATION FILTERS FOR RADAR DATA PROCESSING

Michal Tuszyński and Andrzej Wojtkiewicz

Institute of Electronic Fundamentals, Warsaw University of Technology
Nowowiejska 15/19, 00-665 Warsaw, POLAND

This paper discusses application of adaptive prediction and interpolation filters for Moving Target Indicator (MTI) filtering in coherent Doppler radars. The first part of the paper considers use of adaptive interpolation filters, instead of adaptive prediction filters as MTI filters in Doppler radars. It is shown that these filters can achieve higher improvement factors (IF) than standard prediction filters. The second part of the paper presents an overview of adaptive filtering of nonuniformly sampled signals and considers use of such filters in radar systems.

1. INTRODUCTION

Suppression of clutter and man-made interference in coherent Doppler radars widely used in air traffic control systems is still an important research subject [1]. Moving Target Indicator (MTI) filters used in Doppler radars for suppression of clutter should reject echoes from stationary and slowly moving objects (ground clutter) and precipitation echoes (weather clutter) while passing through useful signals from airplanes (target signals).

Doppler frequencies of airplane signals may be much higher than the average sampling frequency. If the sampling was uniform (constant pulse repetition frequency, PRF), the MTI filter would have so-called "blind speeds" - deep nulls in the frequency response; this means that some target (aircraft) signals would be suppressed by the MTI filter and pass undetected. Periodically nonuniform sampling (staggered PRF), which "fills" these nulls and flattens the frequency response in the passband, is often used to avoid associated loss in detection probability.

As mentioned above, an MTI filter should not only reject low-frequency signals (ground clutter) but also reject weather clutter, which may have a quite considerable mean Doppler frequency. If the clutter spectrum was known exactly, an optimal filter could be designed. In practice, however, a clutter spectrum is not known and, furthermore, this spectrum changes in range, azimuth and time. Obviously, some form of adaptation to a clutter is necessary to achieve near-optimal clutter rejection. The first part of the paper introduces improvement factor and optimal MTI filters and discusses links between optimal MTI filters and Pisarenko harmonic decomposition. The next part of the paper considers use of adaptive interpolation filters, instead of adaptive prediction filters for MTI filtering. It is shown that interpolation filters can achieve higher improvement factors (IF) than prediction filters. This part also presents recursive formulas for computing interpolation filter coefficients and suggestions for realizations of adaptive interpolation filters in a given data case. The last part of the paper presents an overview of adaptive filtering of nonuniformly sampled signals and considers use of such filters in radar systems. This part presents basic properties of adaptive filters working with such signals and an algorithm for fast inversion of a time-varying correlation matrix.

2. ADAPTIVE MTI FILTERS

MTI filters are usually analyzed under assumption that the target signal may be adequately modeled as a stochastic complex harmonic signal. The stochastic clutter signal is usually modeled as an AR process of a low order [1].

The purpose of an MTI filter is to reject clutter and pass through useful target signal in such a way as to maximize the improvement factor (IF), defined as: the signal-to-clutter ratio at the output of the clutter filter divided by the signal-to-clutter ratio at the input of the clutter filter, averaged uniformly over all target radial velocities of interest [2]:

\[
IF = \frac{SCR_{out}}{SCR_{in}}
\]
where SCR (Signal to Clutter Ratio) is a ratio of powers of a target and a clutter signal at the MTI filter input and output correspondingly.

The optimal MTI filter is a filter which maximizes improvement factor defined by (1). It was proved [3] that the coefficient vector of the optimal MTI filter is equal to the eigenvector corresponding to the smallest eigenvalue \( \lambda_{\text{min}}(N) \) of the \( N \times N \) clutter correlation matrix \( R_N \). It is interesting to note that the optimal MTI filter is the same as the eigenfilter in Pisarenko harmonic decomposition (PHD) [4]. This gives better insight into behavior of an optimal filter, but it seems not has been noticed before.

The roots of the transfer function of the optimal MTI filter are located on the unit circle. The frequencies of these roots coincide with the frequencies of the sinusoids in the solution of the PHD model for the correlation matrix \( R_N \). Let us remind that the PHD technique assumes that the random process may be modelled as a sum of stochastic sinusoids in white noise.

The PHD model may also be used for extending the autocorrelation sequence, in a way similar to, for instance, maximum entropy extension. It turns out [5], that this extension by PHD is equivalent to the minimization of the maximum IF, that is to setting \( \lambda_{\text{min}}(N+1) = \lambda_{\text{min}}(N) \). It is easy to see that if indeed the clutter signal follows PHD model, then the optimal filter (eigenfilter) simply eliminates individual sinusoids; actually, the eigenfilter is a prediction polynomial associated with the positive semidefinite autocorrelation matrix \( R_N - \lambda_{\text{min}}(N) I \).

The conclusion of the above discussion is that the maximization of the IF implies, in a way, the PHD model of the autocorrelation function. However, this may not be very compatible with a physical nature of a clutter signal; it seems that the AR (or ARMA) model is nearer to the physical reality. Moreover, the implementation of an adaptive optimal MTI filter is difficult, because it requires solving an eigenvalue problem in real time.

Many researchers [1] proposed using adaptive linear prediction filters. It was found that in many cases the IF of the prediction filter is not much less than the IF of the optimal MTI filter. Nevertheless it might be useful, both from practical and theoretical points of view, to find an adaptive filter which is relatively easy to implement in real time, has computational complexity comparable to a prediction filter and to have the IF greater than a prediction filter. In the next part of the paper we show that the interpolation filter meets all of these objectives.

3. ADAPTIVE INTERPOLATION FILTERS

Prediction filters use only past samples of the signal to estimate the current sample while interpolation filters [6,7] use both past and future samples for linear estimation of the current sample. Interpolation filters, although not causal, can nevertheless be used as MTI filters, because a radar system can usually tolerate delay in the MTI filter output signal.

The linear interpolation is defined as optimal (in the mean square sense) estimation of a sample of a stochastic signal \( y(n) \) by \( K \) preceding and \( L \) following samples. The interpolation error is the output signal of a non-causal FIR interpolation filter \( G_N(z) \):

\[
\tilde{e}_{KL}(T) = y(T) - y_{KL}(T) = \sum_{i=-L}^{K} g_{KL}(i)y(T-i), \quad g_{KL}(0) = 1
\]

In a given correlation matrix case, the coefficient vector of the interpolation filter \( G_N(z) \) is easily found as a normalized \( K+1 \)-th row of the inverse of the correlation matrix [6]. It may be proved [6] that the normalization factor is equal to the interpolation error variance \( \sigma_{yKL}^2 \). In the infinite interpolation case (\( KL \to \infty \)) the frequency response of the interpolation filter \( G(\omega) \) is equal to the inverse of the input signal power spectrum \( S(\omega) \):

\[
G(\omega) = \frac{1}{S(\omega)}
\]

It is easy to see that the coefficients \( g(i) \) of the infinite order interpolation filter may be found as a convolution of the infinite order prediction filter coefficients \( a(i) \):

\[
g(i) = \frac{\sigma^2}{\sigma^2} [a(i) - a(-i)]
\]

where \( \sigma^2 \) is the interpolation error variance and \( \sigma^2 \) is the prediction error variance. The ratio of the variances is [6]:

\[
\frac{\sigma^2}{\sigma^2} = \frac{1}{\frac{1}{\frac{1}{2\pi} \int_{-\pi}^{\pi} |R(\omega)|^2 d\omega}} = 1 + \sum_{i=1}^{\infty} |a(i)|^2
\]

Let us consider now interpolation of an AR process of the order \( p \). In this case all of the higher order coefficients of the prediction filter are equal to 0, and it follows (4) that the coefficients \( g(i) \) for \( |i| > p \) are also equal to 0.

The IF of the MTI filter \( H(z) \) may be written as:

\[
IF = \left[ \frac{1}{2\pi} \int_{-\pi}^{\pi} |H(\omega)|^2 d\omega \right] \frac{\sigma^2_y}{\sigma^2_x}
\]

where \( \sigma^2_x \) is the variance of the input signal and \( \sigma^2_y \) is the variance of the MTI filter output signal. We may now use (5) and (6) to compute a ratio of improvement factors for interpolation filter \( IF_{\text{int}} \) and prediction filter \( IF_{\text{pred}} \):

\[
\frac{IF_{\text{int}}}{IF_{\text{pred}}} = \left[ \frac{\pi}{\pi} \int_{-\pi}^{\pi} |G(\omega)|^2 d\omega \right] \frac{\sigma^2_y}{\sigma^2_x} = \sum_{i=-p}^{p} |a(i)|^2
\]
Obviously, this ratio is not less than one (g(0) = 1).

In practice, this ratio of interpolation and prediction IF will depend on the nature of the input signal and orders of both filters. Simulations performed so far indicate that the IF of the interpolation filter of sufficient order is usually greater than IF of the prediction filter of the same order. It seems that in practice interpolation filters should perform better than prediction filters. For example, in one simulation we compared the IF of the optimal, interpolation and prediction filters of the fourth order for a gaussian shaped sixth order AR process. The improvement factors of the optimal, interpolation and prediction filters were 39.66 dB, 39.65 dB and 38.11 dB.

Let us also remark that the maximization of the IF is not equivalent to the maximization of the detection probability. The interpolation filters, as well as prediction filters, may also be used in different, optimal detection schemes, for instance IBDA [1].

There remain two important problems: first, how to implement adaptive interpolation filter and second, how to filter nonuniformly sampled signals; the latter problem is considered in the next section.

Coefficients of the interpolation filter in the given correlation matrix case are easily found as a row of the inverse of the correlation matrix [6]. We will show that by application of a Golubg-Semencul decomposition [8,9] it is possible to find simple order recursions for lattice implementation of the interpolation filter.

The important Golubg-Semencul theorem [9,10] states that an inverse of aToeplitz matrix may be written as a difference of two products of triangular matrices:

$$R^{-1} = A_t A_t^T - B_t B_t^T$$

(8)

where elements of $A_t$ and $B_t$ are variance-normalized coefficients of a stationary, $N$-th order prediction filter.

When the correlation matrix is not known and must be estimated from data, two correlation matrix estimates are often used: so-called prewinded and covariance matrices. As these matrices no longer have Toeplitz structure, the Golubg-Semencul theorem cannot be applied directly. In a recent paper [10] the Golubg-Semencul formula was generalized to include these estimates. Let $\hat{R}_t$ denote the prewinded covariance matrix estimated at the time instant $t$. The inverse matrix $\hat{R}_t^{-1}$ may then be written as:

$$\hat{R}_t^{-1} = A_t A_t^T - B_t B_t^T$$

(9)

where the variance-normalized matrix $A_t$ is:

$$A_t = \begin{bmatrix} a_N^1(0) & 0 & \cdots & 0 & 0 \\ a_N^1(1) & a_N^1(0) & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ a_N^1(N-2) & a_N^1(N-3) & \cdots & a_N^1(N+1)(0) & 0 \\ a_N^1(N-1) & a_N^1(N-2) & \cdots & a_N^1(N+1)(1) & a_N^1(0) \end{bmatrix}$$

(10)

and the matrix $B_t$ is:

$$B_t = \begin{bmatrix} b_N^1(N-1) & 0 & \cdots & 0 & 0 \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ b_N^1(2) & b_N^1(3) & \cdots & 0 & 0 \\ b_N^1(1) & b_N^1(2) & \cdots & b_N^1(N-1) & 0 \end{bmatrix}$$

(11)

The $a_N^j(k)$ and $b_N^j(k)$ are variance-normalized coefficients of time-varying $N$-th order forward and backward prediction filters. In case of Toeplitz matrices time indexes may be ignored. The generalized Golubg-Semencul decomposition (9) makes possible expressing coefficients of interpolation filters by coefficients of fixed order time-varying prediction filters. The transfer function of the interpolation filter may be found recursively from [11]:

$$C_{K+1,L-1}^t (z) = C_{K+1,L}^t (z) + \alpha_{K+1}(z) A_N^t (z) - \beta_{K+1}(z) B_N^t (z)$$

(12)

where $C_{K+1,L}^t (z) = C_{K,L}^t (0) 0$. $C_{K,L}^t (z)$ denotes variance-normalized time-varying interpolation filter transfer functions, $C_{K,L}^t (0)$ is a free coefficient of the $C_{K,L}^t (z)$, and where $A_N^t (z)$ and $B_N^t (z)$ are transfer functions of the variance-normalized time-varying forward and backward predictors.

Let us note that the transfer function of an interpolation filter may also be recursively computed from transfer function of a lower order interpolation filter [8,11]:

$$C_{K+1,L+1}^t (z) = C_{K+1,L}^t (z) + \alpha_{K+1}(L+1) z^{-(L+1)} A_N^t (z) - \beta_{K+1}(L+1) B_N^t (z)$$

(13)

3. ADAPTIVE PROCESSING OF NONUNIFORMLY SAMPLED SIGNALS

As mentioned in the introduction, Doppler radars often work with staggered PRF, that is with periodically nonuniform sampling. Adaptive processing of such signals is not, to our knowledge, described in the literature.

Let us assume that an analog signal $x(t)$ is sampled at time instants $t_p = nT_0 + p \tau, n=0,\pm 1, \pm 2, \ldots, p=0,1,\ldots,M-1$, $t_{p+1} > t_p > t_0 = 0$, $t_M = rT_0$, $\tau = 0,\pm 1, \pm 2, \ldots$. Sampling instants belonging to one $T_0$ period may also be written as $t_p = p\tau + \delta_p$, where $\delta_p$ is a displacement from the average sampling period $\tau = T_0/M$ ($\delta_0 = \delta_M = 0$). The clutter signal vector is:

$$x_m = [x(t_m+p), x(t_m+(m-1)p), \ldots, x(t_m)]^T$$

(14)

The time-varying $N \times N$ correlation matrix $R_{NN}^{(m)}$ is:

$$R_{NN}^{(m)} = \mathbb{E}\{x_m x_m^T\}$$

(15)

It may be proved that in this case the coefficients of the optimal, prediction and interpolation filters are expressed
as in a case of uniform sampling; the only difference is that the time-varying matrix $R_N^{(m)}$ is substituted for constant matrix $R_N$.

The periodically time-varying correlation matrix (15) is no longer Toeplitz, but the structure of this correlation matrix is very special. A displacement rank $\nabla R$:

$$\nabla R = \text{rank} (R - ZRZ^T)$$  \hspace{1cm} (16)

where $Z$ is a low shift matrix (a matrix which has ones on a first lower subdiagonal), is often used for measuring "closeness" of a matrix $R$ to a Toeplitz matrix. A displacement rank of a Toeplitz matrix is, generally, equal to 2. It is well known that there exist efficient algorithms for inverting matrices of low displacement rank.

Unfortunately, the displacement rank $\nabla R_N^{(m)}$ of the time-varying correlation matrix may be very large. We found, however, that because of the special structure of the time-varying correlation matrix, it has a very low time-varying displacement rank. The time-varying displacement rank $\nabla R_N^{(0)}$ of a time-varying matrix $R_N^{(0)}$ has been introduced in a recent paper [10], and is defined as:

$$\nabla R_N^{(0)} = \text{rank} (R_N^{(0)} - ZR_N^{(0-1)}Z^T)$$  \hspace{1cm} (17)

It is easy to see that the time-varying displacement rank of a matrix $\nabla R_N^{(0)}$ (15) is, generally, equal to 2:

$$\nabla R_N^{(0)} = \text{rank} (R_N^{(0)} - ZR_N^{(0-1)}Z^T) = 2$$  \hspace{1cm} (18)

Using techniques presented in [10] it is relatively easy to write a generalized Golberg-Semenov formula (9) for a case of nonuniform sampling, where $a_n^N(k)$ and $b_n^N(k)$ are now variance-normalized coefficients of time-varying $N$-th order forward and backward prediction filters for the time-varying correlation matrix $R_N^{(m)}$. Extension of this technique may also allow modification of given data algorithms for nonuniformly sampled signals.

We conclude this part of the paper with presentation of a lattice prediction filter for nonuniformly sampled signals. Let us assume that we know the $N-1$-th order time-varying forward and backward predictors $A_N^{(0)}(z)$, $B_N^{(0)}(z)$. The $N$-th order forward and backward predictors can be expressed as a combination of known predictors:

$$A_N^{(0)}(z) = A_{N-1}^{(0)}(z) - k_N^{(0)}z^{-1}B_{N-1}^{(0)}(z)$$  \hspace{1cm} (19)

$$B_N^{(0)}(z) = z^{-1}B_{N-1}^{(0)}(z) - k_N^{(0)}A_{N-1}^{(0)}(z)$$

These order update recursions form a basis of a lattice implementation of a prediction filter for nonuniformly sampled signals. Let us point out that, generally, it is not possible to find any time-update recursions, because there may not be any correspondence between correlation matrices for various time instants. Fortunately, in case of a periodically nonuniform sampling we have $R_N^{(m)} = R_N^{(m+M)}$, where $M$ is a length of a sampling pattern, so the coefficient vectors of the predictors are also periodical $A_N^{(0)}(z) = A_N^{(m+M)}(z)$.

4. CONCLUSIONS

This paper discussed application of adaptive prediction and interpolation filters for MTI filtering in coherent Doppler radars. We proposed use of adaptive interpolation filters, instead of adaptive prediction filters as MTI filters in Doppler radars. It was shown that these filters can achieve higher IF than prediction filters.

The second part of the paper presented basic properties of adaptive prediction and interpolation filters working with nonuniformly sampled signals. We think that this research direction is very perspective, for instance it may allow identification of wideband (wider than Nyquist range) random signals.

REFERENCES

EFFICIENT ALGORITHMS FOR SECOND ORDER POLYNOMIAL SYSTEM IDENTIFICATION

G.O. GLENTIS and N. KALOUPTSIDIS

UNIVERSITY OF ATHENS, DEPT. OF INFORMATICS
PANEPISTIMIOPOLIS, TYPA BUILDINGS, ATHENS 157 71, GREECE.

ABSTRACT. In this paper an efficient methodology for general second order system identification is presented. The proposed technique utilizes a suitable transformation to convert the problem into a multichannel FIR filtering format.

1. INTRODUCTION

Recent work has led to the development of algorithms for estimating the parameters of an unknown plant using finite Volterra polynomial input-output representations, [1]- [2].

The estimation task is to determine at each time instant a suitable estimate of a finite dimensional parameter which, once specified, completely characterizes the plant. The selection of the estimate is based on the comparison between the actual output sample and a judiciously chosen predicted value on the basis of data up to that instant. We shall use the minimum variance predictor specified by the conditional expectation of the unknown system output at each time, given all past information. In LS identification the unknown parameter is chosen to minimize the total squared error between the system output and predictor value over a finite horizon. Batch or off-line identification techniques provide the optimal solution based on a finite number of observations. Sequential or adaptive techniques adapt the optimal solution at each step, taking into account the new incoming information. Therefore, they are capable to track fast time varying dynamics.

In this paper an efficient method for general second order system identification is presented. The proposed technique utilizes a suitable transformation to convert the problem into a multichannel FIR filtering format. It then applies recently proposed fast schemes, in batch as well as in adaptive format, [3]- [8]. The performance of the proposed technique is illustrated by simulation.

2. PROBLEM FORMULATION

A second order nonlinear polynomial system has the form

\[
y(n) = - \sum_i a_y(i)y(n-i) - \sum_i a_x(i)x(n-i) - \\
- \sum_i \sum_j a_{xy}(i,j)x(n-i)y(n-j) - \\
- \sum_i \sum_j a_{yx}(i,j)y(n-i)x(n-j) - \\
- \sum_i \sum_j a_{xx}(i,j)x(n-i)x(n-j) \tag{1}
\]

where \(x(n)\) is the input, \(y(n)\) is the output and the summations are understood over a finite number of terms. The above input output expression admits a finite parametrization with parameters \(a_y(i), a_x(i), a_{xy}(i,j), a_{yx}(i,j)\) and \(a_{xx}(i,j)\).

We group together all coefficients \(a(i,j)\) of (1) associated to products \(y(n-i)x(n-j)\), \(x(n-i)x(n-j)\) and \(y(n-i)y(n-j)\) that satisfy the condition

\[
y(n-i)x(n-j) : i-j = d^{xy}, \\
x(n-i)x(n-j) : i-j = d^{xx}, \\
y(n-i)y(n-j) : i-j = d^{yy}.
\]
where \(d^{yx}, d^{xx} \) and \(d^{yy} \) are constant integers. \(d^{yx} \) takes values on the interval \([d_{pq}^{yx}, d_{pq}^{xx}]\), where \(d_{pq}^{xx} \) may be negative, while, due to parsimony, \(d^{xx} \) and \(d^{yy} \) take values on \([d_{pq}^{xx}, d_{pq}^{yy}]\) and \([d_{pq}^{yy}, d_{pq}^{yy}]\), where 0 \(\leq d_{pq}^{xx} \) and 0 \(\leq d_{pq}^{yy} \), respectively. To simplify notation we suppose that summation in (1) contains all consecutive products, \(y(n - d_{pq}^{yx} - i)x(n - i), i = 1, 2 \ldots m_{pq}^{yx}, x(n - d_{pq}^{xx} - i)x(n - i), i = 1, 2 \ldots m_{pq}^{xx} \) and \(y(n - d_{pq}^{yy} - i)y(n - i), i = 1, 2 \ldots m_{pq}^{yy} \). We reorganize coefficients through the format \(a(i, j) \to a'(k)\), where \(a'(k)\) denotes the entry of the array \(a(i, j)\) located on the \(\ell\) diagonal at the \(k\) position.

Then, model (1) can be viewed as a multichannel ARX system with inputs the signals

\[
\begin{align*}
& y(n), x(n) \\
& u^{yx}(n) = y(n - d^{yx} - i)x(n - i), 1 \leq i \leq q \\
& u^{xx}(n) = x(n - d^{xx} - i)x(n - i), 1 \leq i \leq p \\
& u^{yy}(n) = y(n - d^{yy} - i)y(n - i), 1 \leq i \leq r
\end{align*}
\]  

(2)

as

\[
y(n) = - \sum_{i=1}^{m_{pq}^{yx}} a_y(i)y(n - i) - \sum_{i=1}^{m_{pq}^{xx}} a_x(i)x(n - i) - \sum_{\ell=1}^{q} \sum_{k=1}^{m_{pq}^{yx}} a'_{yx}(k)u^{yx}(n - k) - \sum_{\ell=1}^{r} \sum_{k=1}^{m_{pq}^{xx}} a'_{xx}(k)u^{xx}(n - k) - \sum_{\ell=1}^{q} \sum_{k=1}^{m_{pq}^{yy}} a'_{yy}(k)u^{yy}(n - k)
\]

or

\[
y(n) = - \sum_{i=1}^{m_{pq}^{yx}} a_y(i)y(n - i) - \sum_{i=1}^{m_{pq}^{xx}} a_x(i)x(n - i) - \sum_{\ell=1}^{q} \sum_{k=1}^{m_{pq}^{yx}} a'_{yx}(k)u^{yx}(n - k) - \sum_{\ell=1}^{q} \sum_{k=1}^{m_{pq}^{yy}} a'_{yy}(k)u^{yy}(n - k)
\]

(3)

We rewrite (3) in compact form as

\[
y(n) = -\theta'\phi(n)
\]

(4)

where

\[
\begin{align*}
\theta &= [a_{y1}(m_{pq}^{yx})a_{x1}(m_{pq}^{xx})a_{y1}(m_{pq}^{yy}) \ldots a_{y1}(m_{pq}^{yx})] \\
a_{yx}^{\ell1}(m_{pq}^{yx}) \ldots a_{yx}^{\ell1}(m_{pq}^{yy})a_{xx}^{\ell1}(m_{pq}^{xx}) \ldots a_{xx}^{\ell1}(m_{pq}^{yy})]' \\
\phi(n) &= [y_{m_{pq}^{yx}}(n - 1)x_{m_{pq}^{xx}}(n)u_{m_{pq}^{yx}}^{yx}(n) \ldots u_{m_{pq}^{yx}}^{yy}(n)]
\end{align*}
\]

(5)

and

\[

(6)

where

\[
\begin{align*}
u_{m_{pq}^{yx}}^{yx}(n) &= [u_{m_{pq}^{yx}}^{yx}(n)u_{m_{pq}^{yx}}^{yx}(n - 1) \ldots u_{m_{pq}^{yx}}^{yx}(n - m_{pq}^{yx} + 1)]' \\
\ell &= 1, 2 \ldots q \\
u_{m_{pq}^{xx}}^{xx}(n) &= [u_{m_{pq}^{xx}}^{xx}(n)u_{m_{pq}^{xx}}^{xx}(n - 1) \ldots u_{m_{pq}^{xx}}^{xx}(n - m_{pq}^{xx} + 1)]' \\
\ell &= 1, 2 \ldots r
\end{align*}
\]

and

\[
\begin{align*}
u_{m_{pq}^{yy}}^{yy}(n) &= [u_{m_{pq}^{yy}}^{yy}(n)u_{m_{pq}^{yy}}^{yy}(n - 1) \ldots u_{m_{pq}^{yy}}^{yy}(n - m_{pq}^{yy} + 1)]' \\
\ell &= 1, 2 \ldots q
\end{align*}
\]

3. ESTIMATION

The above transformation converts the Volterra predictor estimation to a multichannel FIR filtering problem with different number of delay elements for each input channel. The resulting LS multichannel filtering has been studied in [3]- [8]. Let \(z(n) = [x'(n)y'(n)]'\) be a record of input output data over the interval 0 \(\leq n \leq N\), \(y(n) \in \mathbb{R}^q\), \(x(n) \in \mathbb{R}^p\). We seek to determine the optimal predictor \(\theta\) which minimizes the total
squared error between the estimated output $\hat{y}(n)$ and the system's output $y(n)$

$$\min_{\theta} \mathcal{V}_N(\theta) = \frac{1}{N} \sum_{n=0}^{N} \lambda^{N-n} (c(n|\theta) - c(n|\theta))$$  \tag{8}$$

where $c(n|\theta) = y(n) - \hat{y}(n|\theta)$ is the instantaneous error, and $\lambda$ is the exponential forgetting factor, $0 < \lambda \leq 1$. $\hat{y}(n|\theta)$ is given by (4), i.e., $\hat{y}(n|\theta) = -\theta' \phi(n)$.

Minimization of (8) leads to the linear system of equations

$$\mathcal{R}(N) \theta(N) = -\phi(N)$$ \tag{9}$$

where $\mathcal{R}(N)$ is the sampled autocorrelation matrix

$$\mathcal{R}(N) = \frac{1}{N} \sum_{n=0}^{N} \lambda^{N-n} \phi(n) \phi'(n) = \lambda \mathcal{R}(N-1) + \phi(N) \phi'(N)$$ \tag{10}$$

and $\mathcal{D}(N)$ is the sampled cross correlation vector

$$\mathcal{D}(N) = \frac{1}{N} \sum_{n=0}^{N} \lambda^{N-n} \phi(n) y(n) = \lambda \mathcal{D}(N-1) + \phi(N) y(N)$$ \tag{11}$$

The least squares second order identification problem can therefore solved through conversion to the dimensions varying multichannel format.

Several algorithms can be utilized to deal with the latter problem. These can be divided into two basic classes, order recursive batch algorithms and adaptive algorithms. Efficient order recursive algorithms have been derived in [3]-[5]. They solve (9) utilizing suitable permutations that are used to unshuffling the shift invariant properties at the data level. Matrix (10) is a block matrix with entries near to Toeplitz submatrices and in fact this property serves for the development of two term Levinson recursions. The Levinson algorithm can be transformed to a Schur type counterpart which offers significant advantage if parallel processing environment is available, since it reduces processing time by an order of magnitude. Implementation of the above algorithms on VLSI array processors is discussed in [10]. In order to cope with time varying dynamics, fast adaptive transversal and lattice algorithms can be used. The derivation is based on the shift invariant properties of the pertinent data. Detailed description of the algorithmic structures is given in [6]-[9].

4. SIMULATION

Let us consider a second order polynomial filter of the form

$$y(n) = -\sum_{i=1}^{m^2} a_i x(n-i) - \sum_{i=1}^{q} a_i x(n-d_i x - i) x(n-i)$$ \tag{12}$$

where, $m^2 = 5$, $q = 5$,

$$[d_1^x d_2^x d_3^x d_4^x d_5^x] = [0 1 2 3 4]$$

$$[m_1^x m_2^x m_3^x m_4^x m_5^x] = [5 4 3 2 1]$$

We define the auxiliary signals as dictated by (2)

$$u_1^x(n) = x(n)x(n), \quad u_2^x(n) = x(n-1)x(n),$$

$$u_3^x(n) = x(n-2)x(n), \quad u_4^x(n) = x(n-3)x(n),$$

$$u_5^x(n) = x(n-4)x(n),$$

Then, eq. (12) can be transformed to a multi-channel filter of 6 input channels and varying filter orders

$$y(n) = -\theta' \phi(n)$$

where $\theta$ and $(\phi(n)$ are defined by eqs. (3) and (6).

The objective of this experiment is to identify the system parameter vector $\theta$, given the input $x(n)$ and the output $y(n) = y(n) + e(n)$, where $e(n)$ is white noise. The optimal filter is obtained as the solution of
linear system (10). A multichannel fast Kalman adaptive algorithm of the FAEST family, developed in [5]-[8] was utilized. Figure 1 illustrates the performance of the proposed method, measured by the relative estimation error (learning curve).

5. CONCLUSIONS

An efficient methodology for general second order system identification has been presented. Extensions to higher order terms is readily derived.

REFERENCES

FAST LINEAR PHASE ADAPTIVE FILTERING USING CONSTRAINED LEAST SQUARES ALGORITHM

L. S. Resende(1), J. M. T. Romano(1) and M. G. Bellanger(2)

(1) DECOM/FEE - Universidade de Campinas - C.P. 6101
13081 Campinas, SP - BRAZIL
(2) Conservatoire National des Arts et Métiers - 292, Rue Saint-Martin
75141 PARIS CEDEX 03 - FRANCE

In this work, an original and fast algorithm is proposed for linear phase adaptive filtering. The approach consists in imposing the symmetry of the impulse response of the filter by means of a set of linear constraints, and then, in treating the problem as a particular case of constrained adaptive filtering. Hence, the solution is attained by using the fast least-squares algorithm for adaptive filters with linear constraints, recently proposed by the authors in a previous work. The algorithm presents a fast and accurate convergence, as it can be observed in the simulations results obtained in simple examples of adaptive equalization and prediction.

I - INTRODUCTION

In several applications of digital signal processing techniques, it is suitable to preserve a linear phase characteristic of FIR digital filters. This characteristic prevents phase distortions in the passband and implies that the Z-transfer function of the filter is a symmetrical or anti-symmetrical polynomial.

Channel equalization, systems identification, frequency estimation and line enhancement are some of the typical applications where the linear phase property may be of great interest. In many cases, the parameters of the filter must be obtained by an adaptive procedure, in order to provide useful methods for real-time operations and non-stationary environments.

Different approaches have been presented in the literature, leading to least-squares algorithms for linear phase adaptive filters [1-5]. This paper proposes a novel algorithm, departing from an original approach. First, the symmetry of the impulse response of the filter is imposed by means of a set of linear constraints over its coefficients. Hence, the problem is reduced to a particular case of adaptive filtering with linear constraints, for which a fast algorithm has been derived in [6].

In order to well establish the proposed approach, the derivation of the constrained adaptive algorithm is briefly presented first. Afterwards, the linear phase constraint is introduced and the final result is analyzed. Finally, the performance and potential of the method is discussed and illustrated by some simulations results.

II - CONSTRAINED LEAST-SQUARES ALGORITHM

The problem may be posed by considering the scheme of the adaptive transversal filter in figure 1, where the coefficients are subject to a set of linear constraints. Then, if the least squares criterion over the error signal is assumed, the following formulation holds:

Minimize \[ J(h(n)) = \sum_{\mathbf{p} \epsilon \mathbf{Z}} \{ d(p) - x'(p)h(n) \}^2 \] (1)

such that \[ C'h(n) = f, \] (2)

where \[ x(p) = [x(p) \ x(p-1) \ldots \ x(p-N+1)]', \]
\[ h(n) = [h_0(n) \ h_1(n) \ldots \ h_{N-1}(n)]' \]
and the \(N \times K\) matrix \(C\) and the \(K\)-element vector \(f\) establish the set of constraints.

![Figure 1: Adaptive Transversal Filter](image)

The optimal coefficients \(h(n)\), supposing a input samples, are obtained by means of the Lagrange multipliers method [7]. After some calculations, we obtain [6]:

\[ h(n) = R_{xx}^{-1}(n)p_{ed}(n) + \]
\[ + R_{xx}^{-1}(n)C[R_{xx}^{-1}(n)C]'\left[ f - C'R_{xx}^{-1}(n)p_{ed}(n) \right] \] (3)

where \(R_{xx}(n)\) is the autocorrelation matrix and \(p_{ed}(n)\) the cross-correlation vector, estimated by:
\[
R_{xx}(n) = \sum_{i=1}^{n} W^{-1} x(i)x'(i) \quad (4)
\]
\[
p_{x'}(n) = \sum_{i=1}^{n} W^{-1} x(i)d(i) \quad (5)
\]

Using this result, our objective is to obtain a recursive solution for \( h(n+1) \), when the new data \( x(n+1) \) and \( d(n+1) \) are available. According to equation (5), the optimal coefficients at instant \( n+1 \) are given by:

\[
h(n+1) = R_{xx}^{-1}(n+1)p_{x'}(n+1) + \Gamma(n+1)C'[\Gamma(n+1)]^{-1}f - C'R_{xx}^{-1}(n+1)p_{x'}(n+1)
\]

where we have defined:

\[
\Gamma(n+1) = R_{xx}^{-1}(n+1)C
\]

(7)

Since the inverse autocorrelation matrix can be updated by \[8\]:

\[
R_{xx}^{-1}(n+1) = \frac{1}{W} \left[ R_{xx}^{-1}(n) - g(n+1)x'(n+1)R_{xx}^{-1}(n) \right]
\]

(8)

where \( g(n+1) \) is the adaptation gain in recursive least-squares (RLS) procedure, given by \[8\]:

\[
g(n+1) = \frac{R_{xx}^{-1}(n)x(n+1)}{W + x'(n+1)R_{xx}^{-1}(n)x(n+1)}
\]

(9)

From (7) and (8), a recurrence can be easily obtained for \( \Gamma(n) \):

\[
\Gamma(n+1) = \frac{1}{W} \left[ \Gamma(n) - g(n+1)x'(n+1)\Gamma(n) \right]
\]

(10)

In this expression, only the matrix \( \Gamma(n) \) depends on the constraints. In order to reduce the computational complexity, the vector \( g(n+1) \) may be calculated by means of the fast least-squares (FLS) algorithm \[8\].

Thus, an updating is attained for the coefficients, using expressions (6) and (10). In this procedure, it is necessary to calculate the inverse of the matrix \( C'[\Gamma(n)] \) at each iteration. So, the efficiency can be substantially increased if a recurrence is also derived for this inverse matrix.

Premultiplying equation (10) by \( C' \), we obtain:

\[
C'[\Gamma(n+1)] = \frac{1}{W} \left[ C'[\Gamma(n)] - C'g(n+1)x'(n+1)\Gamma(n) \right]
\]

(11)

By an appropriate use of the matrix inversion lemma in the above expression, the following recursion holds for the inverse matrix \[6\]:

\[
C'[\Gamma(n+1)]^{-1} = W \left\{ C'[\Gamma(n)]^{-1} + \frac{1}{\Gamma(n)}[x'(n+1)]^{-1} \right\}
\]

(12)

where the vector \( I(n+1) \) is defined by

\[
I(n+1) = \frac{C'[\Gamma(n)]^{-1}C'g(n+1)}{1 - x'(n+1)\Gamma(n)C'[\Gamma(n)]^{-1}C'g(n+1)}
\]

Finally, by replacing equations (5), (8), (10) and (12) in (5), we obtain, after some algebraic manipulations, the following recursive expression for the coefficients of the filter:

\[
h(n+1) = h(n) + g(n+1)e(n+1) + W \Gamma(n+1)h(n+1)e(n+1)
\]

(14)

It is interesting to observe that the above result corresponds to the least-squares adaptive procedure modified by an additional correction term depending on the constraints. The whole proposed algorithm is summarized in Table I:

- New input data : \( x(n+1), d(n+1) \)
- Output error : \( e(n+1) = d(n+1) - h'(n)x(n+1) \)
- Adaptation gain : \( g(n+1) \)

**Introduction of the constraints:**

\[
I(n+1) = \frac{C'[\Gamma(n)]^{-1}C'g(n+1)}{1 - x'(n+1)\Gamma(n)C'[\Gamma(n)]^{-1}C'g(n+1)}
\]

\[
R_{xx}(0) = W_0 \text{ diag}[1, W^{-1}, W^{-2}, ..., W^{-(N-1)}]
\]

(15)

Then, this must be considered for the variables depending on the constraints:

\[
\Gamma(0) = R_{xx}^{-1}(0)C
\]

(16)

Finally, the filter coefficients have the following initial conditions:

\[
h(0) = \Gamma(0)C'[\Gamma(0)]^{-1}f
\]

(17)

The algorithm presented in this section is based on the minimization of the error signal, obtained by \( d(n) - h'(n)x(n+1) \). The same procedure can be developed for the case of a constrained prediction error filter, where the error signal is given directly by \( h'(n)x(n+1) \) [6].

<table>
<thead>
<tr>
<th>Table I: Least-Squares Algorithm with Multiple Constraint</th>
</tr>
</thead>
<tbody>
<tr>
<td>As mentioned in [6], the computational complexity of the above procedure is proportional to ((K+1)(N+K)).</td>
</tr>
<tr>
<td>The initialization of the algorithm must be done according to the least-squares criterion [8, \text{chap.6}]. For the FLS algorithm, used to obtain ( g(n+1) ), it is reasonable to assume all variables to be null for ( n=0 ), setting the prediction error energy to a positive value ( E_0 ). This is equivalent to admit the autocorrelation matrix initialized by ( R_{xx}(0) = E_0 \text{ diag}[1, W^{-1}, W^{-2}, ..., W^{-(N-1)}] ).</td>
</tr>
<tr>
<td>Then, this must be considered for the variables depending on the constraints:</td>
</tr>
<tr>
<td>( \Gamma(0) = R_{xx}^{-1}(0)C )</td>
</tr>
<tr>
<td>Finally, the filter coefficients have the following initial conditions:</td>
</tr>
<tr>
<td>( h(0) = \Gamma(0)C'[\Gamma(0)]^{-1}f )</td>
</tr>
</tbody>
</table>
III - APPLICATION TO LINEAR PHASE ADAPTIVE FILTERING

The algorithm presented in Table I can be used to update a linear phase adaptive filter if the constraints, C and f, are chosen in such a way to impose the symmetry on the filter coefficients. This can be easily achieved by posing:

\[
C = \begin{bmatrix}
1 & 0 & \ldots & 0 \\
0 & 1 & \ldots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \ldots & 1 \\
0 & 0 & \ldots & 0 \\
-1 & 0 & \ldots & 0 \\
\end{bmatrix} = \begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]

(18)

for \(N\) odd or

\[
C = \begin{bmatrix}
1 \\
-2 \bar{J} \\
\end{bmatrix}
\]

(19)

for \(N\) even and \(r=0\) in both cases.

Thus, it is clear that the filter is linear phase if the set of constraints \(\mathbf{C}^t = \mathbf{f}\) is posed. The sign of matrix \(\mathbf{J}\) in expressions (18) and (19) can be negative or positive, providing symmetrical or anti-symmetrical coefficients, respectively.

In the particular case of linear phase prediction, it is also necessary to impose that one of the filter coefficients be non-null, in order to avoid the trivial solution \(\mathbf{h}=0\). This can be achieved, for example in forward prediction, by increasing the matrix \(\mathbf{C}\) with a \((K+1)^{th}\) column given by

\[
\mathbf{c}_{K+1} = \begin{bmatrix} 1 \ 0 \ \ldots \ 0 \end{bmatrix}
\]

(20)

and by posing:

\[
\mathbf{r}^t = \begin{bmatrix} 0 \ \ldots \ 0 \ 1 \end{bmatrix}
\]

This case is specially interesting in signal analysis, since the proposed algorithm provides an efficient way to obtain the composite sinusoidal modeling, or harmonic decomposition, of the signal under analysis.

IV - SIMULATIONS RESULTS

In order to evaluate the performance of the proposed algorithm, we have first supposed a simple case of adaptive equalization, described by the scheme of the Figure 2. The input signal \(x(n)\) is a binary sequence \(x(n) = \pm 1\) and the channel response is assumed to be raised cosine with linear phase, that is, it has a finite and symmetrical impulse response [7, chap. 5]. Additive white noise \(v(n)\), with variance \(\sigma_v^2 = 0.001\), is also considered.

![Figure 2: Adaptive Channel Equalization](image)

We have applied the algorithm of Table I to an equalizer of order 6 (\(N=7\)) where the matrix \(\mathbf{C}\) is given by (18) and \(r=0\). Since the channel is also linear phase, the equalizer must only compensate the amplitude distortion, keeping the linear phase property.

The Figure 3a illustrates the fast convergence of the equalizer coefficients, after 30 iterations. The symmetry property is also observed during the updating. The evolution of the error signal is presented in Figure 3b.

In the second example, we have considered the problem of linear phase predictors. The input signal is composed of three sinusoids in white noise, where the frequencies are given by \(\omega_1=\pi/4\), \(\omega_2=7\pi/16\) and \(\omega_3=15\pi/16\) and the signal-to-noise ratio (SNR) is 3dB. A predictor of order 6 has been used and the matrix \(\mathbf{C}\) is given by (18), where \(K=3\), increased by the \((K+1)^{th}\) column to impose the first coefficient to be one.

The Figure 4a shows the evolution of the coefficients for 150 iterations and the symmetry property can be noted during all the updating. The frequency response of the filter after convergence is presented in Figure 4b. Three notches are observed near the frequencies \(\omega_1\), \(\omega_2\) and \(\omega_3\), since the linear phase condition constrains the zeros of the filter to be on the unit circle. A bias regarding the input frequencies occurs due to the presence of noise.

In all the above simulations, we have used \(\mathbf{W}=0.99\) for the forgetting factor and \(\mu=0.01\) for the initial error energy.

V - CONCLUSION

The fundamental aim of this work is the proposition of the constrained least-squares algorithm of Table I and its application to the specific case of linear phase adaptive filters. Departing from the idea of linear constraints, we attain a simple and elegant way to preserve the linear phase property by constraining the filter coefficients to be symmetrical or anti-symmetrical during the updating process. The use of exact least-squares recursion provides an efficient dynamic behavior to the proposed method.

The performance was illustrated by simple cases of
adaptive equalization and line detection. Clearly, the algorithm may be used in a general context of system identification, where the linear phase property is desired, or in adaptive prediction, providing the harmonic decomposition of the input signal.

Finally, it is interesting to improve the performance by reducing the computational complexity of the proposed algorithm. This may be achieved by using some interesting properties of symmetry observed in the matrix involved in the algorithm. The studies concerning these aspects are now in course.

REFERENCES


Figure 3: Case of Adaptive Equalization:
(a) Evolution of the coefficients;
(b) Evolution of the error signal.

Figure 4: Case of Adaptive Prediction:
(a) Evolution of the coefficients;
(b) Frequency response after convergence.
A Blind Adaptive Equalizer Based on a Lattice/All-pass Configuration

B. Jelonnek, K. D. Kammeyer
Telecommunications Group, Technical University of Hamburg-Harburg
Eissendorfer Str. 40, D-2100 Hamburg 90, FRG

Networks for the blind channel equalization can be designed as a combination of a prewhitening filter (adjusted by means of 2nd-order statistics) and an inverse all-pass system (adjusted by 4th-order statistics). In this paper an appropriate FIR approximation of inverse all-pass systems is presented. On the other hand the inverse all-pass modelling can be avoided if the minimum phase prewhitening filter is replaced by a lattice configuration where the maximum phase (backward prediction error) output is used for decorrelation. In that case conventional recursive all-pass networks can be applied for phase correction. It will be shown that the convergence speed can be considerably increased due to the reduced number of parameters to be identified. Under additive noise conditions a solution near the optimum MSE-solution is obtained.

1 Introduction

Blind Equalization, i.e. adaptive channel correction without use of a trainings sequence or decided data is a new field of research since the recent few years. It is well-known that the traditional 2nd order statistics are "phase blind" in the sense that the transmission path is always interpreted as a pure minimum-phase system (or pure maximum phase, respectively) if only the received signal is used for channel estimation. Consider as an example the classical application of a prediction-error filter as a prewhitening system. The design of this filter is based on the autocorrelation samples of the received signal – the result is a minimum-phase inverse system model.

The fundamental idea to perform a blind adaptive equalization including the phase characteristics is the application of higher order statistics. There exist two fundamental problems: 1st channel identification (see [1],[2],[3]) and 2nd inverse channel estimation (see e.g. [4],[5]). The latter problem will be regarded in the present paper.

2 FIR-modelling of inverse systems

The basic idea for the design of a blind equalizer is to make use of all the information that can be derived from 2nd order statistics. Consider, for example, a complex-valued Moving-Average (MA-) channel model with a z-plane zero configuration shown in fig. 1a (mixed phase). If a FIR prediction-error filter is applied pole approximations (denoted as ⊙) are introduced as shown in fig. 1b. The resulting pole-zero configuration of the system composed of the channel and the prediction-error filter is demonstrated in fig. 1c. Obviously the resulting system contains a number of all-pass pole/zero pairs which are related to the non-minimum phase channel zeros: A proper equalization requires the additional application of inverse all-pass systems after the prewhitening filter which has to be designed on the basis of higher order statistics. Before the discussion of an appropriate blind adaptive algorithm let us at first discuss the problem of the FIR approximation of poles which is necessary for the prewhitening filter design as well as for inverse all-pass modelling. Consider the 1st order transfer function

\[ H_{\text{IR}}(z) = \frac{z}{z - z_{\infty}}; \quad |z_{\infty}| > 1 \]  

(1)

under the specific condition of the pole outside the unit circle. Eq. (1) can be described by the following series expansion

\[ H_{\text{IR}}(z) = -\sum_{k=1}^{\infty} z_{\infty}^{-k} z^k \]  

(2)

which converges, since \( |z_{\infty}|^{-1} < 1 \). Thus the system can be interpreted as causal and non-stable or non-causal...
and stable, equivalently. In the 2nd case we get the impulse response

$$h_{\text{IIR}}(k) = x_k^\Delta, \quad k = -1, -2, \ldots$$  \hspace{1cm} (3)

which can be approximated by a causal FIR system by time truncation and time shift. This causal impulse response (length $\ell$) can be written as

$$h_{\text{FIR}}(k) = h_{\text{IIR}}(k - \ell) - x_k^\Delta h_{\text{IIR}}(k).$$  \hspace{1cm} (4)

After the $z$-transform we obtain the transfer function of the approximated pole

$$H_{\text{FIR}}(z) = H_{\text{IIR}}(z) \left[ z^{-\ell} - x_k^\Delta \right],$$  \hspace{1cm} (5)

the second term describes $\ell$ zeros in the $z$-plane, equidistantly spaced on a circle with the radius $|x_k^\Delta|$. One of these zeros (located at the position of the approximated pole) is canceled by the term $H_{\text{IIR}}(z)$. The result of the pole approximation is shown in fig. 2a. This form of approximation is based on a rectangular windowing of the true impulse response. For comparison fig. 2b shows the result of a Least-Squares (LS-) solution where the zeros are removed from their positions on a circle. Finally fig. 2c shows the FIR approximation of an inverse all-pass system based on a LS design.

![Figure 2: Approximation of a pole outside the unit circle $\ell = 32$](image)

3 Phase Correction Networks

As discussed above a proper equalization is obtained by the combination of a prediction error filter and a phase correction FIR-filter. Fig. 3 shows this configuration. In [5] the phase correction network was realized by a conventional FIR transversal filter. In the present paper we intend to make use of the specific a-priori knowledge, that the phase correction networks is composed of at most $n$ complex-valued 1st order (inverse) all-pass systems, where $n$ denotes the order of the (MA-) channel. For low-order channels this leads to a great advantage with regard to the adaptive adjustment, since the convergence speed strongly depends on the number of parameter to be adapted: In many applications the approximation degree $\ell$ of the phase correction network is much higher than the channel order $n$.

As far as regarded the phase correction network requires the realization of inverse all-pass systems which is possible in the form of the FIR approximation discussed in the last section. On the other hand the inverse all-pass phase correction is a consequence of the minimum-phase prediction-error filter. In contrast we can apply the well-known lattice prewhitening filter and use the backward predictor output which shows maximum phase characteristics. The consequences are illustrated by fig. 4:

![Figure 4: All-pass channel correction using a maximum phase prewhitening filter](image)

The phase correction filter can be designed as a cascade of 1st order all-pass systems (instead of inverse) that can be realized by a recursive network with considerably reduced hardware complexity. Fig. 5 shows the resulting equalizer configuration.

![Figure 5: Lattice-all-pass equalizer](image)
4 Adaptive blind equalization

The adaptive design of the prewhitening filter is solved by the power minimization of the forward (and backward) prediction error; of course this is a blind algorithm. A proper phase correction can only be performed under the application of higher order statistics. In [6] a very simple cost function is derived which is based on the specific definition of 4th order cumulants $c_4^*$:

$$ c_4^* = E\{ |s(k)|^4 \} - 2 \left[ E\{ |s(k)|^2 \} \right]^2 - 2 \left[ E\{ s^2(k) \} \right]^2 = \text{max} $$

if

$$ E\{ |s(k)|^2 \} = \text{const}; $$

(6)

where $s(k)$ denotes the complex envelope of the equalizer output signal and $E\{ \cdot \}$ means the expectancy value. The criterion (6) is closely related to the self recovering equalizer introduced by Godard [4].

The cumulant criterion is used in [6] for the adaptive adjustment of the $\ell$ coefficients of the FIR phase correction filter; in the present paper it is applied for the calculation of the $n$ poles $z_{on}$ (and the related zeros) of the all-pass network. For the iterative solution the well-known stochastic gradient algorithm is applied. After some steps of calculation we obtain the following update equation:

$$ z_{on}(l+1) = z_{on}(l) - \delta \cdot \text{sgn}(c_4^*) \cdot |s(k)|^2 \left[ \alpha^*(k) \frac{\partial s(k)}{\partial z_{on}} + s(k) \frac{\partial \alpha^*(k)}{\partial z_{on}} \right] $$

(7)

where $\delta$ is a small positive constant value and $c_4^*$ is the 4th order cumulant of the input data. The symbol $\alpha^*$ denotes the complex value. In (7) the derivative $\partial s(k)/\partial z_{on}$ is still unknown. The calculation can be carried out in the z-plane (provided that transient effects are sufficiently small). Finally we obtain the gradient network shown in fig. 6. The main disadvantage of the adaptive allpass structure explained so far is the high nonlinearity of the cost function (due to the cascade realization of the phase correction filter). This results in secondary minima and thus in convergence problems under unfavourable start conditions. As a more intuitive approach we use the minimum phase solution (which can easily be derived from the lattice filter) as an initial position. Under these conditions a proper adaptive equalization is achieved for sufficiently low channel order. Fig. 7 shows another example for the adaptive equalization of a 4th order mixed phase channel (a). Fig. 7b demonstrates the trajectories of the all-pass poles during the adaption process: Obviously the poles related to the minimum -phase channel remain at their initial positions while the poles related to the maximum -phase channel are removed on the unit circle or into the origin, respectively, where their influence vanishes. For conclusion the convergence

![Figure 7: Demonstration of the adaptive process](image_url)

![Figure 8: Performance of different blind equalizers (Parameter $\delta$)](image_url)
5 Equalizer performance under additive gaussian noise

In this section the influence of additive noise is considered. It is well-known that the 2nd-order autocorrelation samples are corrupted by additive noise whereas the 4th-order cumulants are insensitive as far as the noise is gaussian. The 1st effect results in a misadjustment of the prewhitening filter; this is illustrated in fig. 9a. Fig. 9b indicates that a proper equalization is not achieved in presence of additive noise. On the other hand an exact modelling of the inverse system (which is related to the so-called "zero-forcing" solution) is not desirable because both – the residual ISI and the power of the additive noise at the equalizer output – have to taken into account. An optimum solution in this sense is the well-known "mean-square error" (MSE) solution. The following examples illustrate that the blind adaptive design of the lattice/all-pass equalizer reveals a solution near the MSE. Fig. 10a shows the resulting power of the error at the equalizer output for zero-forcing, lattice/all-pass and MSE equalization. Fig. 10b gives the related bit error rate as a function of the signal-to-noise ratio. As an example a 1st order channel was used with a zero of $z_0 = 0.9$.

Conclusions

In this paper it was shown that the blind channel equalization problem can be solved efficiently by a certain lattice/all-pass configuration. If the maximum-phase lattice output is used for decorrelation the phase correction network can be designed as a conventional recursive all-pass system. For the blind adaptation a 4th-order scalar cumulant criterion is applied. It was shown that the convergence speed is considerably increased due to the small number of parameters to be identified (all-pass poles). Some simulation results demonstrate that under additive noise conditions a solution near the optimum MSE-solution is obtained.

Mr. M. Benthin is gratefully acknowledged for his support in carrying out the simulations and for many helpful discussions.

References


ON THE REDUCTION OF NESTING ALGORITHMS ROUNING ERRORS

Ryszard Stasiński

Department of Electronics & Communications
Technical University of Poznań
Piotrowo 3A, PL-60-965 Poznań, Poland

In the paper a simple rounding error model is used for evaluating worst-case rounding errors of nesting algorithms. The list of potentially the best from this point of view nesting DFT algorithms (WFTA) is provided, showing that the best results are obtained for some of the following data sizes: \( N = 2^r \cdot 3^s \cdot 5^t \cdot 17^u, r = 1, 2, 3, 4, 5, 6; s = 1, 2; t, u = 0, 1 \). The results are compared with those for the radix-4 and split-radix FFT, and it is shown that for practical \( N \) values the error level of WFTAs need not be more than 1 bit higher than those for the FFT. The paper contains also remarks concerning the construction of low-error nesting convolution algorithms.

1 Introduction

The nesting technique is used for reducing multiplication count in separable multidimensional computations. The most important computational problems of this sort are DFT, circular and linear convolution computation for multidimensional data, or for data sizes being highly composite numbers [1], [2], [3]. Till now it has existed only one work in which a comparison of nested and not-nested (i.e., row-column) algorithms rounding errors was done [4]. The paper concerned the comparison of WFTA with FFT, and showed that WFTA requires 1-2 bits longer accumulator registers, if the same accuracy of WFTA and FFT results were needed. The results are partly confirmed in [5], a paper based on premises given in [6].

Recently it has been shown how to explain the result, and that the effect is characteristic for nested algorithms in general [7] (see also section 2 of this paper). Namely, in the case of FFT an output error is a combination of those from \( 0(N) \) error sources, while the whole FFT structure contains \( 0(N \log N) \) error sources. The situation changes dramatically when the FFT is used for computing a convolution, when all \( 0(N \log N) \) FFT sources contribute to an output error of the algorithm. Unfortunately, nesting algorithms have structures similar to those of the FFT-based convolution ones, while the row-column ones that of FFT. Nevertheless, a detailed analysis of circular convolution and DFT algorithms shows that for some small-size algorithm structures a smaller part of error sources is “visible” from its worst-case output than for other ones.

In the paper the technique from [7] is used for finding potentially the best from the point of view of rounding errors nesting DFT and convolution algorithms for practical data sizes. Section 2 contains general description of the algorithms of interest, definition of the proposed error model, and general comparison of row-column and nested algorithms errors. In section 3 the results of section 2 and of the paper [7] are used for founding the best nesting DFT algorithms, so the section provides verification of conclusions from [7]. The problem of generating the best nested convolution algorithms is outlined in section 4. The evaluations are necessary for restricting the range of algorithms for which computer simulations of their rounding-error performance are made [5]. It is shown that the choice of two algorithms classes in [5]: for \( N = 15 \cdot 2^s \), and for \( N = 255 \cdot 2^s, s = 0, 1, \ldots \), has been correct.

2 Theoretical background

Nesting algorithms are constructed from the so-called small-N modules having a particular structure [1], [2], [3]. In matrix form the structure can be described as follows:

\[
T = EHD
\]

(1)

where the \( D, E \) matrices contain only integer elements, and the \( H \) matrix is either block-diagonal, or diagonal one. Namely, all multiplications of the algorithms are concentrated in the central stage denoted by multiplication by the \( H \) matrix.

The nesting technique can be applied if a computational problem can be transformed into a separable multidimensional one. Separability means that computations can
be made in each dimension independently. So, if the transform \( T \) is used in computations in a dimension of a \( r \)-dimensional algorithm, the calculations are the same as for the matrix \( T \) by a vector product, except for the fact that each element of a vector is not a scalar but a \( r \)-1-dimensional matrix of data. For accomplish the computations the scheme is repeated for each of \( r \) dimensions. This is the so-called row-column algorithm. When nesting the computations in other dimensions are not performed after computations for the first chosen dimension, but are merged with multiplications by elements of the \( H \) matrix (1). If applied repeatedly to the following dimensions the nesting results in a sort of “double row-column” algorithm: it starts with the row-column calculations for all \( D \) matrices, then we have a stage or stages resulting from merging of all \( H \) matrices, and it finishes with the row-column structure for \( E \) matrices in all dimensions.

For evaluating rounding errors of nesting algorithms let us consider the following simple error model [7]: all quantizations after additions, subtractions, and multiplications produce the same error equal to one, once generated an error propagates unchanged across the algorithm flowgraph (hence, e.g., error damped by multipliers is neglected), finally, correlation between errors, if any, is ignored. Quantizations are here such operations as roundings, truncations, etc. The error value in an algorithm flowgraph node is then equal to the number of error sources linked with the node by a path, hence, error counting is as simple as paths counting.

Let us take into consideration the same distribution of errors as assumed in [7] — additions and subtractions linked with multiplications by matrices \( E \) and \( H \) (1) are errorless (no quantization). The overflow is avoided by quantizing results of all additions and subtractions due to multiplication by the matrix \( D \), and by proper scaling of \( H \) matrix multiplier values. Of course, this assumption is quite natural for a fixed-point arithmetic, but not for a floating-point one.

Let us analyze now the error of a row-column algorithm. Assume that the size of the whole problem is \( N = N_0 \cdot N_1 \cdot \ldots \cdot N_{r-1} \), and that the problem is transformed into an \( r \)-dimensional one of size \( N_0 \times N_1 \times \ldots \times N_{r-1} \). As it was stated above, the algorithm can be described as a series of generalized multiplications by \( T_i \) matrices (1), \( i = 0,1, \ldots, r-1 \). In this way an output error of the algorithm consists of an internal error due to multiplication by the \( T_{r-1} \) matrix being proportional to \( N_{r-1} \), and to sum of all \( N_{r-1} \) input node errors of the flowgraph depicting multiplication by the \( T_{r-1} \) matrix. Notice that the formula is true for any algorithm stage, hence, we have the following formula on the error cumulation:

\[
e_r = e_{r-1} + N_{r-1} \cdot e_{r-1} = \ldots = e_{r-1} + N_{r-1} \cdot e_{r-2} + N_{r-2} \cdot e_{r-2} = \ldots
\]

\[
e_r = e_{r-1} + N_{r-1} \cdot e_{r-2} + \ldots + N_{0} \cdot e_0 + N \cdot e_0
\]

where \( e_i \) is the error in a node at the input to stage \( i \) (i.e., \( e_i \) is an error at an output node of the algorithm), and \( e_i \) is the error linked with multiplication by the \( T_i \) matrix. Notice that the output error is

\[
e_r = 0(N_{r-1} + N_{r-2} + \ldots + N) = 0(N)
\]

while the whole number of error source in the algorithm is proportional to

\[
\sum_{i=0}^{r-1} \frac{N_i}{N_i} = 0(rN)
\]

as each generalized matrix multiplications involves \( \frac{N_i}{N} \) multiplications by the matrix \( T_i \) on “scalar” data vectors, compare [7]. Notice also that if they are any input errors \( e_0 \) to the algorithm, all of them contribute to any output error \( e_r \).

The last observation implies that nesting algorithms could not have smaller errors than the row-column ones. Namely, it seems that an output error of the nesting algorithm consists of all errors due to multiplications by \( D_i \), and combination of \( H_i \) matrices (1), compare [7]. Fortunately it appears that for some small modules the \( E_i \) matrices are sparse ones, hence, they “transmit” a part of input errors, only. The best from this point of view are output operations of circular convolution algorithms for \( N = p^s \) when \( s \) is as high as possible, while \( p \) as small as possible, \( p \) is a prime number [7]. Operations due to \( E \) matrices of small DFT modules consist mainly of output operations of circular convolutions, hence, this is an important hint for constructing of these modules, too.

3 Errors of Winograd DFT algorithms

With no doubt the most complicated nesting algorithms are those for the DFT, named Winograd-Fourier transform algorithms (WPTAs) [1], [2], [3]. In [7] the problem of constructing the best from the rounding error point of view DFT modules for these algorithms was stated. It was found that the best DFT modules are those for \( N = 2^s \), then those for Fermat prime numbers \( (3,5,17, \ldots) \) and \( N = 3^s \), then those for \( N = 5^s \) and prime numbers of the form \( N = 2^s \cdot 3^t + 1 \), while the worse results are obtained for prime numbers for which \( N - 1 \) is a product of first powers of other prime numbers. Some modifications results from [7] for the most interesting DFT modules are summarized in Table I. The modifications concentrate on the following two aspects:
The modules are modified for making $E(1)$ matrices as sparse as possible. In particular, computations linked with $x(0)$ and $X(0)$ data samples are done in accordance with Fig. 1d from [7], and not with Fig.1c as it has been assumed in the previous paper. As it was found empirically the modification indeed results in error reduction of whole nesting algorithms.

- Polynomial products in $H(1)$ matrices are computed using reasonable algorithms, and not directly from the definition formulae as in [7]. The only difference with normally applied algorithms consists in fact that polynomial products modulo $Z^2 + 1$ and $Z^2 + Z + 1$ are computed using definition formulae, hence, 4 multiplications and 2 additions, and not 3 multiplications and additions as in usual case.

The last assumption is linked with the fact that all described here algorithms are compared with the radix-4 (hence, split-radix) FFT in which complex multiplications are computed using the definition formula, too. It was found in [7] that the error for this FFT is smaller than

$$E_{FFT}(N) = \frac{5}{3} \cdot N$$

(5)

Table I

<table>
<thead>
<tr>
<th>$N$</th>
<th>Error</th>
<th>$N$</th>
<th>Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>1</td>
<td>8</td>
<td>10</td>
</tr>
<tr>
<td>3</td>
<td>5</td>
<td>9</td>
<td>21</td>
</tr>
<tr>
<td>4</td>
<td>3</td>
<td>13</td>
<td>61</td>
</tr>
<tr>
<td>5</td>
<td>11</td>
<td>16</td>
<td>22</td>
</tr>
<tr>
<td>7</td>
<td>27</td>
<td>17</td>
<td>57</td>
</tr>
</tbody>
</table>

The above mentioned DFT modules (and some worse ones) have been combined in the Winograd’s algorithms, and their errors calculated. Polynomial products in $H(1)$ matrices have been merged using polynomial transforms, of course, if applicable [3]. Similarly as in the case of $E$ matrices, it has been assumed that inverse (output) polynomial transforms following the residual polynomial products are errorless. When residual polynomial products are multidimensional ones, they are computed using the row-column method, which has resulted e.g. in reduced maximum error for the $N = 25$-point module, compare Table I in [7]. In accordance with (2) the best method of constructing row-column structures consists in taking modules with the smallest errors as the first ones. When completing computations due to $E(1)$ matrices it is also important to use “the most sparse” matrices as the first ones for separating paths in the algorithm flowgraph as early as possible.

In Table II the best received Winograd’s algorithms are shown. The best means here that they have the smallest error to $N$ ratio among all nested algorithms, see the last column. The $E_{FFT}(N)$ value is calculated on the basis of (5). As can be seen, the best are families of algorithms for data sizes of the form $N = q \cdot 2^s$, $s = (1), 2, 3, \ldots$, where $q = 3, 15, 255, \text{and } 765$. There are also sample algorithms for $q = 9, 45, \text{and } 51$. In this context the choice of algorithm to be analyzed in [5] seems to be very fortunate, although based on somewhat different premises [6].

In Figure 1 the data for the best Winograd’s algorithm families for $q = 15, 45, 51, 255$, and 765 are plotted together with data for two frequently used algorithm families — for $q = 63$, and $315$ [1], [2]. As can be seen, the error level for the latter families is approximately $\frac{1}{4}$ to $\frac{1}{2}$ bit above that for the best algorithms. It is interesting that this gap between error levels is filled with data for several other algorithm families — for $q = 75, 153, 17, 25$, and even for 27, see also Table I. When compared with FFT the error level for WPTAs grows faster, and for $N > 10000$ and the best WPTAs it becomes more than 1 bit greater than for the FFT. This $N$ value is approximately 4 times greater than for frequently used family of algorithms for $q = 015$.

### 4 Note on nested convolution algorithms

The rounding-error properties of convolution algorithms are much easier to predict than those of DFT ones. With no doubt the best modules are those for $N = 2^s$, for $s \leq 4$, and for $N = 3, 9$, then those for $N = 5$ and so on. Nevertheless, when considering convolution algorithms in general one observation must be strongly underlined here. Namely, for obtaining the best results convolution algorithms should be constructed of convolution modules rather than of DFT ones, although in many cases the use of latter modules results in more efficient algorithms. Namely, it is unlikely that a cascade of forward and inverse small-N DFT modules produces smaller rounding errors than one convolution algorithm of the same size.

### 5 Conclusion

In the paper the results of exhaustive search for the best from the rounding-error point of view nested DFT and convolution algorithms are reported. It is shown that the best DFT algorithms of this type exist for $N = 2^s \cdot 3, 2^s \cdot 15, 2^s \cdot 255, \text{and } 2^s \cdot 765$, $0 < s < 6$, and to some other data sizes composed of DFT modules for $N = \ldots$
It is shown that for practical data sizes the error level of resultant DFT algorithms need not be more than 1 bit above that for the radix-4 and split-radix FFTs. It is somewhat less than it has been obtained in simulations reported in [5]. It seems to be linked with the fact that the error model discards error damping introduced by stages of multiplications in FFTs. On the other hand, the given by the model error growth prediction for the latter algorithms seems to be correct [5]. Finally, the paper provides some hints concerning the construction of error-efficient nesting convolution algorithms — compare [7].

References


Table II

<table>
<thead>
<tr>
<th>N</th>
<th>N divisors</th>
<th>Error</th>
<th>$E_{	ext{FFT}}(N)$</th>
<th>Error/N</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>4 x 3</td>
<td>16</td>
<td>20</td>
<td>1.25</td>
</tr>
<tr>
<td>24</td>
<td>8 x 3</td>
<td>38</td>
<td>40</td>
<td>1.58</td>
</tr>
<tr>
<td>36</td>
<td>4 x 9</td>
<td>60</td>
<td>60</td>
<td>1.67</td>
</tr>
<tr>
<td>48</td>
<td>16 x 3</td>
<td>82</td>
<td>80</td>
<td>1.71</td>
</tr>
<tr>
<td>60</td>
<td>4 x 3 x 5</td>
<td>104</td>
<td>100</td>
<td>1.73</td>
</tr>
<tr>
<td>96</td>
<td>32 x 3</td>
<td>178</td>
<td>160</td>
<td>1.85</td>
</tr>
<tr>
<td>120</td>
<td>8 x 3 x 5</td>
<td>238</td>
<td>200</td>
<td>1.98</td>
</tr>
<tr>
<td>192</td>
<td>64 x 3</td>
<td>386</td>
<td>320</td>
<td>2.01</td>
</tr>
<tr>
<td>204</td>
<td>4 x 3 x 17</td>
<td>432</td>
<td>340</td>
<td>2.12</td>
</tr>
<tr>
<td>240</td>
<td>16 x 3 x 5</td>
<td>522</td>
<td>400</td>
<td>2.175</td>
</tr>
<tr>
<td>360</td>
<td>8 x 9 x 5</td>
<td>834</td>
<td>600</td>
<td>2.32</td>
</tr>
<tr>
<td>480</td>
<td>32 x 3 x 5</td>
<td>1122</td>
<td>800</td>
<td>2.34</td>
</tr>
<tr>
<td>1020</td>
<td>4 x 3 x 5 x 17</td>
<td>2950</td>
<td>1700</td>
<td>2.47</td>
</tr>
<tr>
<td>2040</td>
<td>8 x 3 x 5 x 17</td>
<td>5550</td>
<td>3400</td>
<td>2.72</td>
</tr>
<tr>
<td>3060</td>
<td>4 x 9 x 5 x 17</td>
<td>8844</td>
<td>5100</td>
<td>2.89</td>
</tr>
<tr>
<td>4080</td>
<td>16 x 3 x 5 x 17</td>
<td>12090</td>
<td>6800</td>
<td>2.96</td>
</tr>
<tr>
<td>6120</td>
<td>8 x 9 x 5 x 17</td>
<td>19218</td>
<td>10200</td>
<td>3.14</td>
</tr>
<tr>
<td>8160</td>
<td>32 x 3 x 5 x 17</td>
<td>26374</td>
<td>13600</td>
<td>3.32</td>
</tr>
<tr>
<td>12240</td>
<td>16 x 9 x 5 x 17</td>
<td>41438</td>
<td>20400</td>
<td>3.38</td>
</tr>
<tr>
<td>16320</td>
<td>64 x 3 x 5 x 17</td>
<td>56374</td>
<td>27200</td>
<td>3.46</td>
</tr>
</tbody>
</table>

1More than 1.56 for the 32-point module, see Table 1.
2and the 510-point module, for which Error= 1200.
WAVELET TRANSFORM BASED ADAPTIVE FILTERING

Nurgün Erdöl and Filiz Başbuğ

Department of Electrical Engineering, Florida Atlantic University, Boca Raton, Florida 33431

Wavelet transforms in the discrete time domain are incorporated into an adaptive filter that uses the Widrow-Hoff least-mean square (LMS) algorithm to update the weights of a linear combiner. It is shown that the use of the wavelet transform in place of the discrete Fourier transform in the transform domain adaptive structure [1,7] significantly reduces the condition number (λ_max/λ_min) of the input signal. Experiments with the D4 wavelet [12] in a predictor configuration are encouraging.

1. INTRODUCTION

The most common implementation of adaptive filters utilizes a time domain tapped-delay-line (TDL) form and the Widrow-Hoff least-mean square (LMS) algorithm [5] for the adaptation of filter weights. In case of a large spread in the eigenvalues, however, some form of orthogonalization of the signals that are inputs to the adaptive weights can result in faster adaptation than is possible with LMS alone [8]. The first orthogonalization scheme [7] was based on the discrete Fourier transform (DFT). Others use adaptive lattice filters [8] and Gram-Schmidt orthogonalization methods [9].

Here we investigate the use of wavelet transforms (WT) in adaptive filtering. The method proposed is very much analogous to the DFT based adaptive filtering of Narayan and Peterson [1,7]. However, it has all the advantages of wavelet analysis over Fourier analysis when a weighted sum of sinusoids does not adequately represent the time varying modes of the signals involved. In discrete wavelet analysis [10-13] signals are represented by a weighted sum of the translates and dilates of a mother wavelet. Multiresolution analysis [12,13] provides a means of constructing orthonormal bases of wavelets spanning the space L2(R) of square integrable functions. These wavelets can be grouped by their dilation (scaling) constant into disjoint subsets spanning proper and orthogonal subspaces of L2(R). These subspaces that correspond to different scales are said to represent signals at different resolution levels.

In the adaptive filtering scheme using wavelets the projections of the input signal on to the above mentioned orthogonal subspaces are used as inputs to a linear combiner. The weights of the linear combiner can hence be updated by the orthogonalized LMS algorithm [8] while normalizing the power at each resolution to achieve faster and uniform convergence of all weights to be optimal. Although very much like the DFT based adaptive system [1,7], the WT based system is superior in its ability to track rapid changes in the input signal. Use of real valued wavelet requires no complex arithmetic. Due to the recursive properties of wavelet analysis at different resolution levels, the amount of computation is much less than that using the FFT.

2. BACKGROUND INFORMATION

The LMS algorithm [5] is a computationally simple algorithm that is used to obtain iteratively the weight vector Wj that minimizes the difference between a desired signal Sj and its estimate yj = WTjXj in the mean-square sense. The weight vector is updated according to

\[ W_{j+1} = W_j + 2\mu e_j x_j \]  

(1)

where the subscript j denotes time and e_j is the error signal equal to the difference S_j - y_j. If the adaptation constant \( \mu > 0 \) is less than \( 1/\text{tr}(R_x) \), where \( R_x \) is the input autocorrelation matrix, then \( W_j \) converges to the Wiener solution \( W^* \) in the mean [2]. The slowest time constant for the weights is given by

\[ \tau = \frac{\text{tr}(R_x)}{2\lambda_{\text{min}}} \]  

(2)

where \( \lambda_{\text{min}} \) is the smallest eigenvalue of the autocorrelation matrix \( R_x \). This property slows down the convergence rate when there is a large spread in the eigenvalues. This deficiency can be overcome by preprocessing the input signal \( X_j \) so that it is orthogonalized [6]. Specifically, if the new autocorrelation matrix is the identity matrix, then all the eigenvalues being unity assures the only and fastest time constant

\[ \tau = \frac{1}{2\mu} \]  

(3)

The orthonormal wavelets are used to process the input signal. A function \( x(t) \in L^2(\mathbb{R}) \) can be expressed as a linear combination of the orthonormal wavelets as

\[ \psi_{j,k}(t) = 2^{j/2} \psi(2^j t - k) \quad \text{for all integers } j,k \]  

(4)

\[ x(t) = \sum_{j,k \in \mathbb{Z}} d_{j,k} \psi(2^j t - k) \]  

(5)

The restriction of \( x(t) \) to a given resolution \( j \) is

\[ x_j(t) = 2^j \sum_{k \in \mathbb{Z}} d_{j,k} \psi(2^j t - k) \]  

(6)

where the wavelet coefficients \( d_{j,k} \) are given by the inner product of \( x(t) \) and the wavelet, i.e.,
\[ d_k^j = <x(t), 2^{j/2} \psi(2^j t - k)> \] (7)

In practice the above series will usually be truncated and an approximation to \( x(t) \) will be expressed as

\[ x^{(j)}(t) = \sum_{j=0}^{J} \sum_{k} 2^{j/2} d_k^j \psi(2^j t - k) \] (8)

where \( J \) is a positive integer. Equation (8) shows that \( x^{(j)}(t) \) is the projection of \( x(t) \) onto the collection of orthogonal subspaces \( W_0 \) \( U \) \( W_{-1} \) \( U \) \( \cdots \) \( U \) \( W_{-J} \) with \( W_0 \) and \( W_{-J} \) representing respectively the highest and lowest resolution subspaces. The error incurred by such approximations is addressed in [14] and warrants further analysis. For the purposes of this work, however, we assume that \( x^{(J)}(t) \) is acceptably close to \( x(t) \).

3. "SCALE" BASED ADAPTIVE SYSTEMS

Here we outline a discrete-time adaptive system that uses the orthogonized version of an input sequence as input to a linear combiner as shown in Figure 1. For this construction we refer to the filter bank interpretation of wavelet transforms in the discrete time domain [15,16]. Let \( \psi(2^{-t}) \) be represented by its at least \( 2^J \) times oversampled values on the "v" scale as \( g(n) \). Specifically, let

\[ g(n) = \psi(2^{-j_0} n) \quad n = 0, 1, \ldots, N-1 \] (9)

where \( N = 2^K \) form some integer \( K > J \). Then

\[ g_j(n) = g(2^j n) = \psi(2^{j-j_0} n) \] (10)

represents \( g(n) \), subsampled by \( 2 \). Then let \( \{g(n) \} \), \( n = 0,1,\ldots,2^{-j_0} N-1 \) be the discrete time representation of a wavelet oversampled by a factor of \( 2^J \) and \( N = 2^K \), \( K > J > 0 \). Define

\[ \gamma_j^k = \sum_{n=-n_0}^{n_0} z_n g_j(n) \] (11)

\[ \gamma_j^k = 2^{j/2} \sum_{n=-n_0}^{n_0} z_n \psi(2^{j-j_0} t - k) \]

where \( n_0 = 2J-j_k \), and let the synthesized signal be given by

\[ x^{(j)}_n = C \sum_{j=0}^{J} \sum_{k} a_j^k \gamma_j^k \] (12)

where \( C \) is a positive constant of normalization. Substituting (11) into (12) yields

\[ x^{(j)}_n = \sum_{j=0}^{J} \sum_{k} a_j^k g_j(n) \] (13)

where

\[ r_j(n-m) = c \sum_{k} g_{j+k}(n) \psi_{j+k}(m) \] (14)

and

\[ r_j(n) = r_0(2^j n) = r_j(-n) \] (15)

From Figure 1 it is seen that \( r_j(n) \) denotes the real symmetric finite impulse response of the filter banks used to process input vector \( X_n = (x_n x_{n-1} \ldots x_{n-N+1})^T \). Filter outputs \( v_j(n) \),

\[ v_j(n) = \sum_{m} r_j(m) z_{n-m} \] (16)

are multiplied by the weights \( w_j(n) \). The output signal \( y(n) \)

\[ y(n) = \sum_{j} w_j(n) v_j(n) \] (17)

is subtracted from the desired signal \( x(n) \) to yield the error signal \( e(n) \). Weights \( w_j(n) \) are updated by the Widrow-Hoff LMS algorithm as

\[ w_j(n+1) = w_j(n) + 2 \mu d(n) z_{n-j} \] (18)

The filter outputs can be expressed in matrix notation as

\[ V_n = R X_n \] (19)

where

\[ V_n = [v_0(n), v_1(n), \ldots, v_J(n)]^T, X_n = [x_n x_{n+1} \ldots x_{n-N+1}]^T \]

and

\[ R_j = r_j(m) \quad \text{for } j = 0, 1, \ldots, J \quad \text{and } m = 0, 1, \ldots, N-1 \]

We note that the covariance of the new input is

\[ R_e = E[v v^T] = R E[X X^T] R^T = R \] (20)

If we impose that \( R_v = I \), i.e., all eigenvalues of \( R_v \) be identical and equal to 1 then we get

\[ RR^T = R^{-1} \] (21)

Indeed, equation (21) does not give us a practical way to find \( R \). However, if the discrete time wavelets are orthonormal in the scaling and the translation parameters, then \( v_j(n) \) are orthogonal in \( j \).

This can be seen by observing from equations (12-14) that \( x^{(j)}_n \) is the \( j \)-sum of \( v_j(n) \). Hence, an alternative to equation (16) is
\[ v_j(n) = c \sum_k \sigma_j(k) \beta_j(n) \]

(22)

Substituting eqn. (22) in (20) and using

\[ c \sum_k \sigma_j(k) \beta_j(n) = \tau j(k) - \tau j(k - i) \]

(23)

proves that \( R_v = \text{diagonal} \ (C \sigma^2) \) where

\[ \sigma_j^2 = \sum_k (\sigma_j(k))^2 \]

(24)

The new eigenvalues are hence given by \( \sigma^2 \) which represent the signal energy at the \( j \)th resolution level or the energy of the output of the \( j \)th filter in Figure 1. These filters denoted by their impulse responses \( r_j(n) \) are related to each other by scaling specifically

\[ r_{j+1}(n) = r_j(n) e^{j\omega_1} + r_j(n) e^{-j\omega_1} \]

(25)

where \( r_j(e^{j\omega}) \) is the Fourier transform of the sequence \( r_j(n) \) and \( r_j(e^{j\omega}) = 0 \).

From eqn. (25) (see also [15,16]) it can be seen that the frequency domain support of the filters gets larger and shifts to the high frequency ranges as the resolution level increases. For a class of signals with \( 1/f \) like spectra the constant Q characteristics of the above filters implies that the energy of the filter outputs will tend toward a uniform distribution. Our initial experimental results supporting this argument are given in Table 1.

4. EXPERIMENTAL RESULTS

The condition number \( \lambda_{\max}/\lambda_{\min} \), is computed as a measure of the eigenvalue spread for the input signal \( x(n) \) and the transformed signal \( v_j(n) \). The condition number is indicated by \( \xi_{S} \) and \( \xi_{VS} \) for the input and the transformed input respectively and \( s \) denotes the size of the correlations matrix. The results are given below

<table>
<thead>
<tr>
<th>s</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>s</td>
<td>9.875</td>
<td>31.674</td>
<td>60.1742</td>
<td>91.9855</td>
</tr>
<tr>
<td>VS</td>
<td>8.0154</td>
<td>8.0375</td>
<td>8.0564</td>
<td>8.0564</td>
</tr>
</tbody>
</table>

Table 1
Condition Numbers as a Function of the Order of the Adaptive Filter

It is clear that the transform decreases the eigenvalue spread for the experimental signal.

5. CONCLUSIONS

A wavelet transform based adaptive filter is proposed and analyzed for its property to orthogonalize an input data sequence. The orthogonal sequence produced by a bank of real symmetric FIR filters are applied to a linear combiner whose weights are updated by the LMS algorithm for Widrow and Hoff. Such orthogonalization has the effect of fast and uniform convergence to the optimal (Wiener) solution.

![Figure 1. Wavelet Transform Based Adaptive System.](image)

REFERENCES


An Adjustable Constraint Approach for Robust Adaptive Beamforming

Luis Castaño, Ching-Yih Tseang and Lloyd J. Griffiths

1DSSR, ETSI Telecomunicación-UPM. Ciudad Universitaria s/n, 28040 Madrid.
2University of Southern California, Los Angeles, CA 90089-2564.

This paper presents a new beamforming scheme which uses a reference signal in linearly-constrained minimum variance (LCMV) adaptive beamformers. The proposed approach is able to track variations in the signal of interest (SOI) steering vector and incorporates additional constraints that can be used to control the sidelobe level. Since the constraints are adjusted to match the SOI steering vector, the signal cancellation behavior due to perturbation errors in the conventional LCMV beamformers is alleviated. An efficient implementation structure is also presented to update the beamformer weights in an adaptive fashion. The method is intended for communications applications where a reference signal can be extracted from the array output.

1 Introduction

In Linearly Constrained Minimum Variance (LCMV) beamformers a single point constraint which fixes a unity gain at the desired signal direction is typically used. This constraint makes use of the steering vector at the signal direction. However, such a beamforming system is known to be very sensitive to perturbation errors because the steering vector used in the constraint design does not match the true steering vector perfectly. In practice, it is almost impossible to precisely obtain the true steering vector because the element locations are only known within a certain degree of accuracy. Other uncertainty factors may also include random gain and phase errors occurring at the antenna elements and the variation of the desired signal direction. Even with only a small amount of perturbation errors, the output signal-to-interference plus noise ratio (SINR) can be severely degraded due to signal cancellation. Feldman and Griffiths [1] pointed out that the problem of signal cancellation can also occur if there is not sufficient amount of snapshots to estimate the covariance matrix, even though the true steering vector is perfectly known. It was shown that this finite sample data effect in fact represents a kind of perturbation error similar to other earlier mentioned errors. When the input SNR is high or the number of array elements is large, an excessively number of snapshots is generally required so that the sample data effect can be ignored.

This paper describes a new method for LCMV beamforming under perturbed conditions in which an adjustable constraint is used. It is assumed that the beamformer contains a fixed constraint that determines the quiescent response (the optimal response under white noise input). A typical requirement is to impose a low sidelobe level so that interferences can be nullled out in a short period of time. The adjustable constraint, on the other hand, is continuously updated according to the changes in the estimation of the signal of interest (SOI) steering vector. The estimate is computed using the input data and a reference signal which is recovered from the array output. The advantages of the method are that a lesser number of snapshots is required to calculate the optimum weights, pointing and calibration errors are compensated and the beamformer is able to track the desired signal source.

2 Background

The beamformer output y(n) can be expressed as the inner product of a data vector x(n) and a weight vector w(n),

\[ y(n) = w^H(n)x(n) \]

where the superscript \( H \) denotes Hermitian transpose. For a narrowband array with \( N \) elements, \( x(n) \) and \( w(n) \) are \( N \)-dimensional complex vectors. The observed data vector which contains SOI, noise and interferences can be expressed as

\[ x(n) = d(n)a(\theta) + u(n) \]

where \( d(n) \) is the SOI, \( a(\theta) \) is the corresponding steering vector and \( u(n) \) represents the undesired signal portion that may include both noise and interferences.

In LCMV beamformers, the optimization problem is formulated as

\[ \min_w \ w^H R_{xx} w \ \text{subject to} \ C^H w = f \]

where \( R_{xx} \) denotes the covariance matrix of \( x(n) \), the constraint matrix \( C \) contains \( M \) column vectors while the response vector \( f \) specifies the corresponding constraint value for each vector. The well known optimal solution to this problem is given by

\[ w_{\text{opt}} = R_{xx}^{-1} C (C^H R_{xx}^{-1} C)^{-1} f \]

Under white noise only conditions, the optimal response is known as the quiescent response which is specified by the weights

\[ w_q = C (C^H C)^{-1} f \]

An alternate solution to (3) which is the basis of the Generalized Sidelobe Canceller (GSC) structure [2] is also known as

\[ w_{\text{opt}} = w_q - B (B^H R_{xx} B)^{-1} B^H R_{xx} w_q \]
where $\mathbf{B}$ is the blocking matrix of size $N \times (N - M)$ whose columns span the orthogonal subspace of the column space of $\mathbf{C}$.

In this paper, we consider a two constraint system in which the first constraint is used to impose a desired quiescent response and the second to prevent SOI cancellation at a given direction of arrival. These two constraints are outlined below while the details of the constraint design process can be found in [9]. We first assume that a desired quiescent response with low sidelobes has been designed using the algorithm in [4] and the corresponding weights are specified by $w_d$. Also, let $a(\theta)$ be the steering vector in the SOI direction. The two constraints are given by

$$ w_d^H w = w_d^H w_d $$

$$ a(\theta)^H (I - w_d(w_d^H w_d)^{-1} w_d^H) w = 0 $$

Since both $w_d$ and $a(\theta)$ are obtained based on an assumed array, they are inaccurate when there is a mismatch between the true and assumed arrays. The error induced by $w_d$ is generally negligible since it only alters the desired quiescent response slightly, provided that the amount of perturbation is not enormous. The error caused by $a(\theta)$ is more significant, however, since even a small perturbation can result in a complete cancellation of the SOI, especially for the cases of high SNR. As shown in [1], signal cancellation can also occur, even when the array is ideal, if the optimal weights are computed based on insufficient number of snapshots. The purpose of this paper is to present an approach which prevents signal cancellation by iteratively estimating $a(\theta)$ from the data and continuously updating the second constraint.

3 Adjustable Constraint

In certain communications applications, reference signals may be transmitted for different purposes such as acquisition, synchronization, or user identification. These signals are commonly multiplexed with the transmitted information either in time, frequency, or code [5]. At the beamformer output, the reference and information signals can be separated using any conventional demultiplexing technique. The reference signal is then regeraged which can be used to estimate the steering vector. The p-vector (that is, the cross-correlation) vector between the reference signal $r(n)$ and the array input $x(n)$ is

$$ p = E[r^*(n)x(n)] = E[r^*(n)d(n)]a(\theta) + E[r^*(n)u(n)] $$

If $r(n)$ is correlated with $d(n)$ and uncorrelated with the noise and the interferences, the second term on the right hand side of (9) equals zero. Hence, (9) reduces to be

$$ p = \gamma_{rd} a(\theta) $$

where $\gamma_{rd}$ is the correlation between $r(n)$ and $d(n)$. It should be noted that $r(n)$ does not need to be a perfect replica of $r(n)$ in order to get a p-vector proportional to the steering vector. $r(n)$ which is correlated with $d(n)$ and uncorrelated with $u(n)$ is sufficient.

For the purpose of this paper, we consider a simple case where an unmodulated carrier is transmitted in addition to the information signal [5]. Thus, the desired signal is

$$ d(n) = A_d e^{j(w_d^* n + \phi_d)} $$

where $A_d$, $\omega$, and $\phi_d$ are respectively the amplitude, frequency, and phase of the carrier. In this expression, the frequency is assumed known and fixed while $A_d$ and $\phi_d$ are random. A reference signal can be obtained if we can generate a sinusoid whose phase is correlated with the phase of the unmodulated carrier. One way of doing this is by means of a band-pass filter and a Phase-Locked-Loop (PLL). The PLL basically consists of a multiplier, a loop filter and a voltage-controlled oscillator (VCO). The output of the VCO constitutes the reference signal

$$ r(n) = A_r e^{j(w_d^* n + \phi_r)} $$

where $A_r$ and $\phi_r$ are the amplitude and the phase of the VCO. Due to the feedback loop structure of the PLL, $\phi_r$ continuously tracks the input carrier phase $\phi_d$.

To incorporate the reference signal into the LCMV beamformer, $a(\theta)$ is replaced by the p-vector in equation (8)

$$ p^H (I - w_d(w_d^H w_d)^{-1} w_d^H) w = 0 $$

Making use of the constraint in (7), (13) reduces to be

$$ p^H w = p^H w_d $$

The scale factor $\gamma_{rd}$ is irrelevant because the constraint value equals zero. Formulating the second constraint in terms of the p-vector as in (13) has a significant advantage that it can be updated easily by use of the maximum likelihood estimate of $p$ from the data

$$ \hat{p} = \frac{1}{N} \sum_{n=1}^{N} r(n)x(n) $$

The problem of signal cancellation due to perturbation errors in LCMV beamformers becomes worse as the SNR increases. In that case, the utilization of the p-vector instead of the steering vector improves the beamformer performance. This is because the p-vector involves an implicit estimation of the perturbation errors and this estimate becomes more accurate as the SNR increases. Consequently, an accurate knowledge of the array characteristics and SOI direction is not necessary if the adjustable constraint approach is used. In addition, the use of the p-vector also reduces the finite data sample effect since the errors in the estimate of the covariance matrix are correlated with the errors in the p-vector estimate. [8]. The number of snapshots which is required to reach within 3 dB of the optimal SNR is subsequently reduced.

4 Adaptive Implementation

A direct way to implement the adjustable constraint is to compute the optimal weights using the closed-form solution in (4) where $R_{xx}$ and $p$ are estimated from the data. For
adaptive implementation, it is more desirable to update the weights in a simple fashion which does not require matrix inversion in each iteration. In this section, such an adaptive implementation structure is derived.

The adaptive implementation is based on solving two optimization subproblems in which the first is defined as

\[
\min_{w} \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w} \text{ s. t. } \mathbf{w}^H_0 \mathbf{w} = w_d^2
\]  

and the second is

\[
\min_{w} \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w} \text{ s. t. } \mathbf{w}^H_0 \mathbf{w} = 0 \text{ and } \mathbf{p}^H \mathbf{w} = 1
\]  

It was shown in [7] that the optimal solution to the minimization of \( \mathbf{w}^H \mathbf{R}_{xx} \mathbf{w} \) subject to (7) and (14) can be expressed as

\[
\mathbf{w}_\text{opt} = \alpha_1 \mathbf{w}_\text{opt1} + \alpha_2 \mathbf{w}_\text{opt2}
\]  

where \( \mathbf{w}_\text{opt1} \) and \( \mathbf{w}_\text{opt2} \) are the optimal solutions to (16) and (17), respectively. The combination coefficients \( \alpha_1 \) and \( \alpha_2 \) can be determined by noting that \( \mathbf{w}_\text{opt} \) has to satisfy both (7) and (14)

\[
\alpha_1 = 1
\]

\[
\alpha_2 = \mathbf{p}^H (w_0 - \mathbf{w}_\text{opt1})
\]  

With this decomposition, \( \mathbf{w}_\text{opt} \) can be implemented via computing \( \mathbf{w}_\text{opt1} \) and \( \mathbf{w}_\text{opt2} \).

Adaptive implementation of \( \mathbf{w}_\text{opt} \) can be readily obtained through the use of the GSC structure in which

\[
\mathbf{w}_\text{opt1} = \mathbf{w}_e - \mathbf{B} \mathbf{w}_\text{opt1}
\]  

where \( \mathbf{B} \) is the blocking matrix of size \( N \times (N - 1) \) and \( w_0 \) equals \( w_d \) in this case. The \( \mathbf{w}_\text{opt1} \) vector is the only data dependent part which is given by

\[
\mathbf{w}_\text{opt1} = \mathbf{R}_{xx}^{-1} \mathbf{p}_e
\]  

where

\[
\mathbf{R}_{xx} = \mathbf{E}[\mathbf{x}(n) \mathbf{x}^H(n)], \quad \mathbf{x}(n) = \mathbf{B} \mathbf{x}(n),
\]

\[
\mathbf{p}_e = \mathbf{E}[\mathbf{p}(n) \mathbf{x}(n)], \quad \mathbf{p}(n) = \mathbf{w}^H_{opt} \mathbf{x}(n)
\]  

Since (21) has the form of the Wiener solution, any unconstrained adaptive algorithm such as the LMS or RLS is directly applicable to implement \( \mathbf{w}_\text{opt1} \) adaptively.

Adaptive implementation of \( \mathbf{w}_\text{opt2} \) is somewhat more complicated since the second constraint is also data dependent. However, because the first constraint simply requires that \( w \) be orthogonal to \( w_0 \), this constraint can be eliminated by letting

\[
w = \mathbf{B} \mathbf{w}_e
\]  

and reformulating the minimization in terms of \( \mathbf{w}_e \)

\[
\min_{\mathbf{w}_e} \mathbf{w}_e^H \mathbf{R}_{xx} \mathbf{w}_e \text{ s. t. } \mathbf{p}^H \mathbf{B} \mathbf{w}_e = 1
\]  

This is a single constraint system and its solution is given by

\[
\mathbf{w}_\text{opt2} = \frac{\mathbf{R}_{xx}^{-1} \mathbf{p}_e}{\mathbf{p}^H \mathbf{R}_{xx}^{-1} \mathbf{p}_e} \mathbf{w}_\text{opt2}
\]  

As a result, \( \mathbf{w}_\text{opt2} \) can be expressed as

\[
\mathbf{w}_\text{opt2} = \frac{\mathbf{B} \mathbf{w}_\text{opt2}}{\mathbf{p}^H \mathbf{w}_\text{opt2}}
\]  

where

\[
\mathbf{w}_\text{opt2} = \mathbf{R}_{xx}^{-1} \mathbf{p}_e, \quad \mathbf{p}_e = \mathbf{E}[\mathbf{p}(n) \mathbf{x}(n)]
\]  

From this equation, it is clear that \( \mathbf{w}_\text{opt2} \) can be implemented in an adaptive fashion. Finally, substituting (19), (20) and (26), into (18) we can express \( \mathbf{w}_\text{opt} \) in the decomposed form

\[
\mathbf{w}_\text{opt} = \mathbf{w}_e - \mathbf{B} (\mathbf{w}_\text{opt1} - k \mathbf{w}_\text{opt2})
\]  

where

\[
k = \frac{\mathbf{p}^H \mathbf{w}_\text{opt1}}{\mathbf{p}^H \mathbf{w}_\text{opt2}}
\]  

Figure 1 shows the implementation scheme which is derived from this decomposition. Note that according to equations (21) and (27), \( \mathbf{w}_\text{opt1} \) and \( \mathbf{w}_\text{opt2} \) are the optimal solutions to two unconstrained standard adaptive filter problems. The input data vector is \( \mathbf{x}(n) \) for both of them, while the desired response is \( \mathbf{p}(n) \) for \( \mathbf{w}_\text{opt1} \) and \( \mathbf{r}(n) \) for \( \mathbf{w}_\text{opt2} \).

The scheme of figure 1 can be seen as a LCMV beamformer with a TR beamforming connected in a parallel form. The role of the variable parameter \( k \) is to ensure that the output of both beamformers are summed appropriately regardless of the reference signal amplitude and the norm of \( \mathbf{w}_e \). This implementation structure also suggests that the underlying problem can be considered either as a LCMV beamformer where a point constraint is designed using a reference signal, or a TR beamformer where one constraint is used for the quiescent pattern control.

5 Simulation Results

Computer simulations were carried out to show the performance of the proposed approach. We considered a 25-element linear equispaced array where each element has a random phase perturbation uniformly distributed from \(-2.5^\circ\) to \(2.5^\circ\). The signal environment consisted of a 10 dB desired signal in the broadside direction, a strong interference of 30 dB from 20° and a 0 dB white noise background.

Figure 2 shows the radiation pattern for a LCMV beamformer with two constraints as mentioned earlier. The desired quiescent response in this case was selected to be a beam pattern with uniform 30 dB sidelobe. Notice that the antenna response has a null in the interference direction. However, it exhibits an unacceptable sidelobe level due to the signal cancellation phenomenon caused by the perturbation errors.

Better performance was achieved using the adjustable constraint approach. The p-vector was estimated using the reference signal extracted from the PLL. Since this estimate took perturbation errors into account, signal cancellation was greatly reduced. The corresponding optimal beamformer response is shown in figure 3. Notice that the beamformer can now steer a main lobe towards the SOI direction and also place a null in the jammer direction. Simulations show a larger sidelobe level than the specified 30 dB level due to errors in the covariance matrix and noise in the PLL output.

The convergence curves of the SINR using recursive im-
plemention for the proposed approach and the conventional LCMV beamformer are shown in Figure 4. In this implementation, a recursive least-squares (RLS) adaptive algorithm with a forgetting factor of 0.995 is used for both adaptive loops. Notice how phase perturbation errors as small as 2.5° can seriously degrade the LCMV beamformer adaptive response while the adjustable constraint approach is insensitive to these errors.

References


Figure 1: Scheme for recursive adaptation

Figure 2: Antenna response with an LCMV beamformer

Figure 3: Antenna response with an adjustable constraint

Figure 4: SINR behavior: a) Adjustable constraint b) LCMV beamformer
TRANSIENT AND STEADY-STATE BEHAVIOR
OF THE 2D FAST RLS LATTICE ALGORITHM

X. LIU, L. MANDRIDAKE, M. NAJIM

Equipe Signal et Image and GRECO_TDSI
ENSERB, 351, Cours de la Libération, F33405 TALENCE Cedex, France

This paper mainly deals with the analysis of the performances of a new algorithm: 2D Fast, Recursive-Least-Squares Lattice (2D FLRLS) algorithm. The transient performances, initialization and steady-state finite-precision effects are investigated. The preliminary results compared with 2D fast RLS transversal algorithm are included in this paper.

1. INTRODUCTION

The application of adaptive algorithms for the two-dimensional problems have been more and more noticeable. Because of complexity of the 2D problems, the 2D adaptive algorithms, which require quite limited prior knowledge, show a wide application foreground. Many successful extensions of 1D linear prediction, autoregressive and lattice modeling to the 2D case have been reported [8][10][11]. The latest developments in VLSI technology which had a significant impact on the realization of real time applications of 1D adaptive and lattice algorithms have also made it possible to consider architectures and implementation for two or higher dimensional signals and suggest the feasibility of massive parallelism in the implementation schemes.

Recently, we presented a lattice version of the 2D Recursive Least Squares Algorithm (2D FLRLS) [13], which provides an exact growing-order least squares solution to the deterministic normal equations for the AR and MA models. It can be also extended to ARMA model. We have reported an application for the restoration of noisy images in the GRETISI91[12]. It showed improved performances compared to the TDJPL-NLMS [8][9].

This paper is organized as follows: the summary of 2D FLRLS algorithm is included in section 2. In section 3, we examine the performances of the 2D FLRLS and 2D FRLS[11] algorithms. The conclusions are included in section 4.

2. 2D PREWINDOWED RLS LATTICE ALGORITHM

2.1. 2D Prewindowed RLS Estimation

We consider a general 2D adaptive digital filter which is acting on the input 2D L×K data x(n1,n2) and provide an optimum estimation of the desired response d(n1,n2). The d(n1,n2) is assumed to be the sum of a Gaussian noise N(n1,n2) and output of the unknown filter. The output of the adaptive filter is as follows:

\[ \hat{y}(n_1,n_2) = \sum_{i,j \in \mathbb{N}} a_{i,j} x(n_1-i,n_2-j) \]

R is the filter support region. Here, we choose a quarter plan (as shown in fig. 1). It is defined as:

\[ R(N,M) = \{i,j : (0 \leq i \leq N, 0 \leq j \leq M) \cap ((i,j) \in (O,0)) \} \]

\[ \begin{array}{c}
\text{L-1} \\
\text{L-1} \\
0 \\
0 \\
M+1 \end{array} \]

\[ \text{K-1} \]

Fig. 1 First quadrant filter support region
The partial order of computation is defined as a linear scanning index along column where the data is considered to update in a heliocoidal fashion. So, we introduce a linear scanning index \( n \) such that:
\[
n = n_2 \cdot L + n_1
\]
We consider the 2D pre-windowed least-squares prediction problem, which estimates the coefficients of the filter by minimizing the accumulated squared error with exponential weighting:
\[
\xi(n) = \sum_{k=0}^{n} \left[ e^{\lambda_k} \cdot e(k_2) \right]^2
\]
where \( e(k_1) \cdot d(k_2) \cdot \bar{Q}(k_3) \)

The minimum mean-square error (MMSE) becomes \( E[N(n_1, n_2)]^2 \).

We exploit the local analogy of multichannel within the QP support region. Then, the mathematical concepts of vector space, orthogonal projection, and subspace decomposition can be used to derive this new algorithm. The detailed derivation can be found in [13].

2.2 Summary of 2D FLRLS Algorithm

We start recursions by setting all initial conditions to zero except the error covariance matrix as \( \frac{1}{\delta} I(M+1 \times M+1) \) (\( \delta \) is a small positive constant) and angle parameters as 1.

For \( n = (n_1, n_2) \):
\[
\chi^f(n) = [x_2(n+1) \ldots x_{M_1}(n+1) \times_1(n)] \\
\chi^b(n) = [x_1(n) \times_2(n) \ldots x_{M_1}(n)] \\
\bar{x}(n) = [x_2(n+1) \ldots x_{M_1}(n+1)] \\
\bar{z}(n) = [x_2(n) \ldots x_{M_1}(n)]
\]

For \( m = 0 \),

The gain of the for/backward reflection coefficients are:
\[
\hat{g}^f_0 = \frac{(\chi^b_0)^{-1}(\chi^b_0)^{-1})^T}{1 + \chi^b(n)(\chi^b_0)^{-1}(\chi^b(n))^T} \\
\hat{g}^b_0 = \frac{(\chi^f_0)^{-1}(\chi^f_0)^{-1})^T}{1 + \chi^f(n(\chi^f_0)^{-1}(\chi^f(n))^T)}
\]

The inverse of the for/backward error covariance matrix are:
\[
(\hat{F}_0^f)^{-1} = (\chi^b_0)^{-1}(\chi^b_0)^{-1} \\
(\hat{F}_0^b)^{-1} = (\chi^f_0)^{-1}(\chi^f_0)^{-1}
\]

The for/backward reflection coefficients matrix are:
\[
\begin{align*}
\hat{K}^f_0(n) &= \hat{K}^b_0(n-1) + \hat{g}^b_0(\bar{z}(n) \bar{z}(n)^T)(\hat{K}^b_0(n-1)) \\
\hat{K}^b_0(n) &= \hat{K}^f_0(n-1) + \hat{g}^f_0(\bar{x}(n) \bar{x}(n)^T)(\hat{K}^f_0(n-1))
\end{align*}
\]

The angle parameter is:
\[
\gamma_0(n-1) = 1 - \frac{\hat{g}^b_0(n)(\hat{F}_0^b(n)^{-1} - \hat{g}^b_0(n))^T}{\hat{g}^b_0(n)(\hat{F}_0^b(n)^{-1} - \hat{g}^b_0(n))^T}
\]

The for/backward error are:
\[
\begin{align*}
\hat{e}^b_0(n) &= \hat{z}(n) - \hat{F}_0^b(n)(\hat{F}_0^b(n)^{-1} \hat{F}_0^b(n)^{-1}) \\
\hat{e}^f_0(n) &= \hat{x}(n) - \hat{F}_0^f(n)(\hat{F}_0^f(n)^{-1} \hat{F}_0^f(n)^{-1})
\end{align*}
\]

The joint process updates as follows:
\[
H_0(n) = H_0(n+1) + \hat{g}^f_0(\bar{x}(n) \bar{x}(n)^T)(H_0(n-1)) \\
e_0(n) = e(n) + \hat{F}_0^f(n)^{-1} \hat{F}_0^f(n)^{-1} e_0(n)
\]

For \( m = 1, \ldots, N \),

The gain of the for/backward reflection coefficients are:
\[
\begin{align*}
\hat{g}^f_m &= \frac{(\chi^b_{m-1})^{-1}(\chi^b_{m-1}(n-1))}{\gamma_m(n-1)(\hat{e}^f_{m-1}(n-1))} \\
\hat{g}^b_m &= \frac{(\chi^f_{m-1})^{-1}(\chi^f_{m-1}(n-1))}{\gamma_m(n-1)(\hat{e}^b_{m-1}(n-1))}
\end{align*}
\]

The inverse of the for/backward error covariance matrix are:
\[
\begin{align*}
(\hat{F}^f_m)^{-1} &= (\chi^b_{m-1})^{-1}(\chi^b_{m-1}(n-1)) \hat{e}^f_{m-1}(n-1) \\
(\hat{F}^b_m)^{-1} &= (\chi^f_{m-1})^{-1}(\chi^f_{m-1}(n-1)) \hat{e}^b_{m-1}(n-1)
\end{align*}
\]

The a priori for/backward predicted error are:
\[
\begin{align*}
\hat{e}^b_m(n-1) &= \hat{e}^f_{m-1}(n-1) - \hat{g}^f_{m-1}(n-1) \hat{K}^b_m(n-1) \\
\hat{e}^f_m(n-1) &= \hat{e}^b_{m-1}(n-1) - \hat{g}^b_{m-1}(n-1) \hat{K}^f_m(n-1)
\end{align*}
\]

The for/backward reflection coefficients matrix are:
\[
\begin{align*}
\hat{K}^f_m(n) &= \hat{K}^b_{m-1}(n-1) + \hat{g}^b_{m-1}(n-1) \hat{e}^f_{m-1}(n-1) \\
\hat{K}^b_m(n) &= \hat{K}^f_{m-1}(n-1) + \hat{g}^f_{m-1}(n-1) \hat{e}^b_{m-1}(n-1)
\end{align*}
\]

The angle parameter is:
\[
\gamma_m(n) = \gamma_{m-1}(n-1) - \hat{g}^f_{m-1}(n-1)(\chi^f_{m-1}(n-1) \hat{e}^f_{m-1}(n-1))^T
\]

The for/backward error are:
\[
\begin{align*}
\hat{e}^b_m(n) &= \hat{e}^f_{m-1}(n-1) - \hat{g}^f_{m-1}(n-1) \hat{K}^b_m(n-1) \\
\hat{e}^f_m(n) &= \hat{e}^b_{m-1}(n-1) - \hat{g}^b_{m-1}(n-1) \hat{K}^f_m(n-1)
\end{align*}
\]

The joint process updates as follows:
\[
H_m(n) = H_{m-1}(n) + \hat{g}^f_{m-1}(n-1) \hat{e}^f_{m-1}(n-1)(H_{m-1}(n-1)) \\
e_0(n) = e_{m-1}(n) - \hat{e}^f_{m-1}(n-1) H_{m}(n-1)
\]
2.3. Limited-Precision Effects of 2D FRLS Algorithms

The limited-precision effects in digital implementation of 1D adaptive algorithms have been researched [14][15]. Though 2D RLS algorithms have more complex forms compared with their 1D forms, they also include the matrix inverse implicit. It is possible for this matrix to become indefinite from accumulation of quantization errors as in 1D cases. For example, effects of initialization, when the parameter \( \delta \) is too small make this matrix close to singularity at \((n_1,n_2)=(N,M)\). Another effect involves the exponential increase of the effects of the numerical errors when exponential weighting is used. 2D FRLS [11] diverge in both cases (as we will see later).

For the 2D FLRSLs, the recursions at spatial position are also recursive in order, and quantities are constructed from lower order quantities until the desired order is achieved. It increases the magnitude of quantization errors as the recursions progress and are limited in a finite number. It can be propagated in a numerically stable fashion, except in cases where the order is very large.

Due to the limited space in this paper, we will not give a detailed analysis. We only provide in the next section some simulations.

3. SIMULATIONS AND RESULTS

We implement the 2D FRLS [11] and 2D FLRSL algorithms in a VAX Station 3100. We use a 32-bit floating arithmetic and 64-by-64 2D artificial data in all cases.

Test results are presented for transient performances, initialization and steady-state finite-precision effects.

Fig. 2 shows the basic test configuration. The input \( x(n_1,n_2) \) is generated by passing white noise through a 2D low-pass digital filter. The unknown system chosen is a 5x5 filter. We select the relative error between the estimated coefficients and coefficients of "unknown" system as the performance indicator. It is defined as follows:

\[
Q(n_1,n_2) = \left\| H(n_1,n_2) - \hat{H}(n_1,n_2) \right\| / \left\| H(n_1,n_2) \right\|
\]

Fig. 3 shows the convergence of both algorithms after proper initialization (\( \delta = 10^{-3} \)).

Fig. 2 Basic test configuration

Fig. 3 Convergence of the algorithms

A poor choice of the initialization constant \( \delta \) leads the 2D FRLS [11] to instability on step \((n_1,n_2)=(M,N)\). Fig. 4 gives an example of such a phenomenon. The 2D FLRSL algorithm remains to converge in this case.

Fig. 5 compares both algorithms when the exponential weighting is used. The 2D FRLS algorithm exhibits an effect of steady-state divergence. The 2D FLRSL remains stable.

Fig. 4 Influence of the initialization parameter
4. CONCLUSIONS

The new 2D FLRLS algorithm requires about \(11(M+1)K_1\) operations per iteration, where \((M+1)\) is the number of channels, \(K_1\) is the total number of data used in the 2D filter. A reduction in the computational cost is obtained when compared with the standard RLS algorithm\((1.5K_1^2)\).

Simulations show that the 2D FLRLS algorithm can update in numerical stable fashion, while the 2D FRLS\([11]\) exhibits a numerical unstable behavior. Though the new algorithm has more complexity in computation than its transversal form\((6(M+1)K_1\) operations)\([11]\), we can expect better numerical properties, stage independence and pipelineable implementation. The research on the effects of limited-precision is under way and we will report further results.

REFERENCES

A COMPARATIVE FIXED-POINT ERROR ANALYSIS OF THE SCHUR AND THE SPLIT SCHUR ALGORITHMS

By Nicholas Glaros and George Carayannis*

A new analytical methodology is introduced here for fixed-point error analysis of two Toeplitz solving algorithms, the Schur algorithm and the newly introduced split Schur algorithm. The theoretical results obtained are consistent with experimentation. Besides the intrinsic symmetry of the error propagation recursive formulas, the technique presented here is capable of explaining many practical situations: For signals having a small eigenvalue spread the Schur algorithm behaves better than the split Schur in the fixed-point environment. The intermediate coefficients of the split Schur algorithm leading to the PARCORs cannot serve as alternatives to the reflection coefficients in error sensitive applications, etc. It is demonstrated that the error-weight vectors of the Schur propagation mechanism follow Levinson-like (2nd order) recursions, while the same vectors of the split Schur propagation mechanism follow split Levinson-like (3rd order) recursions.

I. INTRODUCTION

The Schur algorithm [1] is used for solving symmetrical Toeplitz systems, particularly those related to the autocorrelation method of linear prediction and the least squares FIR filtering techniques. A less redundant in complexity version of the Schur algorithm is the split Schur algorithm [2]. Both algorithms can be realized using fixed-point (FP) arithmetic as they process a priori bounded variables. In [3], [4] some first-order FP error analysis of the Schur and the split Schur algorithms can be found. However, the results given cannot be directly applied for comparing the two algorithms, because they are restricted to the first two reflection coefficients. In this paper, the numerical performance under FP conditions of the Schur and the split Schur algorithms for computing the reflection coefficients is examined by introducing an appropriate set of error-weight vectors and error related scalar quantities. These parameters provide a tool for an in-depth analytical study of the FP error propagation mechanism of each algorithm, and demonstrate a data-dependent superiority of the Schur algorithm over the split Schur algorithm.

II. FP ERROR ANALYSIS OF THE SCHUR AND SPLIT SCHUR ALGORITHMS

A Schur-type algorithm, which is a slightly modified version of the Le Roux-Gueguen algorithm [1], [2] has been used as a basis to analyze the FP effects of the Schur algorithm. This algorithm is outlined in (2.1) [5]

\[ e(1,1) = r(1) \]  \hspace{1cm} \text{(2.1a)}

\[ f(1,1) = r(1-1) \]  \hspace{1cm} \text{(2.1b)}

\[ e(m+1,1) = e(m,1) + k(m) \cdot f(m,1) \]  \hspace{1cm} \text{(2.1c)}

\[ f(m+1,1) = f(m,1-1) + k(m) \cdot e(m,1-1) \]  \hspace{1cm} \text{(2.1d)}

\[ k(l) = -e(l,1)/f(l,1) \]  \hspace{1cm} \text{(2.1e)}

for \( i = 1, 2, \ldots, p \)

\( r(l) \) is the \( i \)th normalized autocorrelation coefficient, \( k(l) \) stands for the \( i \)th reflection coefficient, and \( f(l,1) \) indicates the \( i \)th stage mean square prediction error. The FP computation of the products and quotients involved in recursions (2.1c), (2.1d) and (2.1e) produces the local rounding errors \( \gamma_{e}(m,1), \gamma_{f}(m,1) \) and \( \delta(l) \) respectively. The round-off error \( c \) due to the quantization of an infinite precision real number \( \chi \) into \( B \)-bits is assumed to be a zero-mean independent random variable with variance \( \sigma_{c}^{2} = q^{2}/12 \), \( q = 2^{-q} \), \( q \) being the quantization width and \( F \), \( (0 \leq q \leq B-1) \), being the number of fractional bits in the \( B \)-bits FP representation of \( \chi \). This statistical model will be adopted for all local rounding errors in the algorithms and the error in the autocorrelation coefficient FP computation. Thus, if \( \sigma_{e}^{2}, \sigma_{f}^{2}, \sigma_{r}^{2} \) and \( \sigma_{\delta}^{2} \) denote the variance of \( \gamma_{e}(m,1), \gamma_{f}(m,1), \delta(l) \) and the autocorrelation coefficient FP error, then

\[ \sigma_{e}^{2} = \sigma_{f}^{2} = \sigma_{r}^{2} = \sigma_{\delta}^{2} / q^{2} / 12, \quad q = 2^{-B+1} \]  \hspace{1cm} \text{(2.2)}

\( B \) being the word length used in the FP implementation of the algorithms. The quantities defined in Table I have been found very useful in formulating general closed-form expressions for the reflection coefficient FP error variances. By means of these parameters

*The authors are with the Division of Computer Science, Department of Electrical Engineering, National Technical University of Athens, GR-157 73 Zographou, Athens, GREECE.
and by using simple algebra, it can be shown that the reflection coefficient first-order FP error variances in the Schur algorithm satisfy [5]

\[ Y_{ij}^2 + \text{Var}(R_{ij}) = 0, \quad 1 \leq i < j \leq n \]  

(2.3)

where the \( i \times i \) vector \( \mathbf{y}_i = [y_{1,1}, y_{2,1}, \ldots, y_{i,i}]^T \) displays the statistical correlation of \( \mathbf{A}k(\lambda) \), i.e., the FP error in the \( i \)-th reflection coefficient, with all the other reflection coefficient FP errors up to the \( i \)-th stage, that is, \( \mathbf{y}_i = E[\mathbf{A}k(\lambda) \mathbf{A}k(\lambda)] \). \( u_{i,i} \) is a scalar function of all rounding errors local to the Schur algorithm as well as the autocorrelation coefficient FP errors. Complicated relationships exist, which when combined with the recursions shown in Table I, compute the components of \( y_i \) and the variance of \( R_{ji} \). These relationships are given in [5], but will be presented here for the sake of brevity.

The symmetric form of the split Schur algorithm can be described as follows [5]

\[ y(0,0) = r(0)/2 \quad (2.4a) \]
\[ k(0) = 1 \quad (2.4b) \]

for \( i = 1, 2, \ldots, p \)

\[ y(0,i) = r(i) \quad (2.4c) \]
\[ y(1,i) = r(i) + r(i-1) \quad (2.4d) \]

for \( m=1, 2, \ldots, l-1 \)

\[ y(m,1) = y(m,1) + y(m-1,1) - \lambda(m) \cdot y(m-1,1-1) \quad (2.4e) \]

\[ \lambda(1) = y(1,1)/y(1-1,1) \quad (2.4f) \]
\[ k(1) = 1 - \frac{\lambda(1)}{1 + k(1-1)} \quad (2.4g) \]

The FP realization of the relations (2.4e), (2.4f) and (2.4g) results in the local quantization errors \( \delta(m,1) \), \( \delta(1) \) and \( \delta(1) \) respectively. The variances \( \sigma^2 \) and \( \kappa^2 \) of the \( \gamma \) and \( \delta \) are expressed as

\[ \kappa^2 = \frac{\lambda^2}{12}, \quad \gamma^2 = \frac{2^{-2\alpha}}{12} \quad (2.5a) \]
\[ \kappa^2 = \frac{\delta^2}{12}, \quad \gamma^2 = \frac{2^{-3\beta}}{12} \quad (2.5b) \]

In Table II the FP error-related parameters for the split Schur algorithm are introduced. With Table II's definitions, analytical expressions for the evaluation of the first-order FP error variance in the computation of the \( i \)-th \( \lambda \)-coefficient and the \( i \)-th reflection coefficient by the split Schur algorithm can be derived [5]. The final results are

\[ Y_{ij}^T \cdot \mathbf{y}_i + \text{Var}(Y_{ij}) = 0, \quad 1 \leq i < j \leq n \quad (2.6) \]
\[ \mathbf{y}_i^T \cdot \mathbf{y}_j + \text{Var}(Y_{ij}) = 0, \quad 1 \leq i < j \leq n \quad (2.7) \]

The various terms appearing in (2.6), (2.7) have the same significance with the ones involved in (2.3), namely: \( \mathbf{y}_i \), \( \mathbf{x}_i \) are \( i \times 1 \) error-weight vectors, as the \( y_i \) one, with the convention \( T_{1,1} = 

III. COMPARISON OF THE TWO ALGORITHMS

In (2.3), (2.6), (2.7), the error variances \( m_{1,1}, \gamma_{1,1}, \theta_{1,1} \) exhibit moving average dependence on the \( i-1 \) past covariances \( m_{n-1,n}, \gamma_{n-1,n}, \theta_{n-1,n} \), with coefficients \( u_{n,n}, \tau_{n,n}, \phi_{n,n} \) and direct dependence on the variance of the error variables \( u_i, \gamma_i, \theta_i \), respectively. Let us compare the error variances in the two algorithms by considering specific values for the reflection coefficients and studying their effects on the \( u_{i,1,n}, \tau_{i,1,n}, \phi_{i,1,n} \) coefficients as well as the variance of the \( R_{1} \), \( \gamma_{1} \), \( \theta_{1} \) error quantities. As shown in Table I the order \( m \times 1 \) error-weight vectors in the FP Schur algorithm are defined on the basis of the order \( m \) similar vectors by using Levinson-like recursions. On the other hand, in Table II the order \( m \times 1 \) error-weight vectors in the split Schur algorithm are defined on the basis of both the order \( m \) and the order \( m-1 \) similar vectors by using a split Levinson-like recursions. Given that property of the error-weight vectors in the FP Schur and FP split Schur algorithm, the potential numerical behaviour of the vectors \( u_i, \gamma_i, \theta_i \) and the variances of \( u_i, \gamma_i, \theta_i \) as all reflection coefficients tend to zero simultaneously (with \( |k(n)| \) → 0, 1 ≤ i ≤ n) is found to be [5]

\[ u_{1,1}, \tau_{1,1}, \phi_{1,1} \rightarrow 0, 1 \leq i < n \quad (3.1) \]

\[ \text{Var}(R_{1}) \cdots \text{Var}(\gamma_{1}) \cdots \text{Var}(\theta_{1}) \cdots \text{Var}(u_{1}) \cdots \text{Var}(\gamma_{1}) \cdots \text{Var}(\theta_{1}) \]  

\[ \text{Var}(u_{1}) \cdots \text{Var}(\gamma_{1}) \cdots \text{Var}(\theta_{1}) \cdots 1 \leq i \leq 3 \quad (3.2) \]

\[ \text{Var}(\gamma_{1}) \cdots \text{Var}(\gamma_{1}) \cdots \text{Var}(\theta_{1}) \cdots \text{Var}(u_{1}) \cdots (3.3) \]

The previous relations state that when the first \( i \) reflection coefficients are too small, then, for both algorithms, the error variance in the \( i \)-th stage depends mostly on the variance of the error expressions \( R_{1}, \gamma_{1}, \theta_{1} \).
which involve only the local rounding errors and the FP errors in the autocorrelation coefficient computation. This is because the weight of each of the covariance vectors \( \mathbf{w}_i \), \( \mathbf{u}_i \), \( \mathbf{v}_i \) in this case is quite close to zero.

Therefore, the reflection coefficient and the \( \lambda \)-coefficient FP errors generated in earlier stages of the algorithms propagate too little into later stages and the variances of \( \mathbf{R}_i, \mathbf{U}_i, \mathbf{V}_i \) become then very good approximations of the true reflection coefficient and \( \lambda \)-coefficient error variances. Comparing (3.2) to (3.3), it turns out that the variance of \( \mathbf{V}_i \), which is related to the \( i \)-th stage \( \lambda \)-coefficient computation, is much bigger than the variance of \( \mathbf{R}_i \) corresponding to the \( i \)-th stage reflection coefficient in the Schur algorithm. In particular, \( \text{Var}(\mathbf{V}_i) \) is \( \mathcal{O}(I^2) \) dependent from the order of the algorithm's stage, whereas the respective dependence of \( \text{Var}(\mathbf{R}_i) \) is \( \mathcal{O}(I) \). As it is concluded from (3.4), the FP error variances in the \( \lambda \)-coefficient computation are approximately the same as the ones in the reflection coefficient computation for the split Schur algorithm, if the previous signal environment is taken into account. Consequently, the Schur algorithm produces smaller reflection coefficient FP error variances than the split Schur algorithm when the signal under process is noise or it has low correlation between successive samples. It is important to note that (3.4) appears to be valid only when the condition about the smallness of the reflection coefficients is strictly met, since the variance of the \( \Delta(1) \) error produced by the FP split Schur algorithm has been found to be substantially different from that of the \( \Delta(1) \) error in most of the problems arising in practice. Let us now consider the case of \( |k(1)| \rightarrow 1 \), \( i=2 \), with all the reflection coefficients of the \( i=2 \) previous stages properly valued within the \((-1,1)\) range. This situation arises in real applications where sinusoidal signals or signals with highly low-pass characteristics are processed. Then, during the \( i \)-th stage of computations, the mean square prediction error \( f(1,1) \) will arbitrarily approach to zero. Since all the left-hand quantities of (3.1)-(3.4) depend on the reciprocal of \( f(1,1) \), it follows that the \( i \)-th reflection coefficient FP error variance becomes extremely large for both algorithms when the reflection coefficient of the previous stage is absolutely close to 1.

The previous condition on the reflection coefficients values also produces very high \( \lambda \)-coefficient FP error variances. Some numerical examples, which validate the previous findings are given in Fig. 1, 2. Two standard types of input signals were used: a zero mean, white Gaussian pseudorandom noise sequence with unit variance, and a pure sinusoidal signal of radian frequency 0.85\( \pi \).

The exact mean-squared error of each coefficient, averaged over ten sets of data, is depicted in Fig. 1a and 2a for each type of the signals used respectively. The corresponding theoretical values, obtained by means of (2.3), (2.6), (2.7), are plotted in Fig. 1b and 2b. It is observed that the theoretical and experimental results show an adequate match, with the first ones being pessimistic in all cases. The superiority of the Schur over the split Schur algorithm also appears to be decreasing as the autocorrelation matrices corresponding to the signal under process tend to become ill-conditioned.

IV. CONCLUSIONS

It was proved that the Schur algorithm is more accurate than the split Schur algorithm when the underlying signal is totally unpredictable, or, equivalently, the reflection coefficients are absolutely very close to zero. The validity of this result was experimentally extended to signals with low predictability. Both algorithms were found to perform poorly when the Toeplitz autocorrelation matrix is close to ill-conditioning. Furthermore, the FP arithmetic properties of the \( \lambda \)-coefficients were found to be inferior than those of the reflection coefficients computed by the Schur algorithms and no better than those of the reflection coefficients computed by the split Schur algorithm. This result implies that the \( \lambda \)-coefficients cannot serve as numerical substitutes of the reflection coefficients in real applications employing the split Schur algorithm.

REFERENCES


TABLE 1: Definition of the parameters used for the analytical description of the Schur algorithm's FP error propagation mechanism.

\[ \begin{align*}
\lambda_n &= \begin{bmatrix} A_{n,0} & A_{n,1} & \cdots & A_{n,n-1} \\ B_{n,0} & B_{n,1} & \cdots & B_{n,n-1} \end{bmatrix}, & n \geq 1 \\
\Delta_0 &= \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, & B_1 = [0 \\ 1] \\
\Delta_{n+1} &= \begin{bmatrix} \Delta_n \\ 0 \end{bmatrix} + k(n) \begin{bmatrix} B_n \\ 0 \end{bmatrix}, & 1 \leq n \leq p \\
E_0 &= J_n \Delta_0, & 1 \leq n \leq p+1 \\
E_n &= \begin{bmatrix} E_{n,0} & E_{n,1} & \cdots & E_{n,n-1} \\ F_{n,0} & F_{n,1} & \cdots & F_{n,n-1} \end{bmatrix}, & n \geq 0 \\
F_n &= \begin{bmatrix} G_{n,0} & G_{n,1} & \cdots & G_{n,n-1} \\ H_{n,0} & H_{n,1} & \cdots & H_{n,n-1} \end{bmatrix}, & n \geq 0 \\
G_n &= \begin{bmatrix} [0]_{n,0} & [0]_{n,1} & \cdots & [0]_{n,n-1} \\ [0]_{n,0} & [0]_{n,1} & \cdots & [0]_{n,n-1} \end{bmatrix}, & n \geq 0 \\
C_{n} &= \sum_{n=0}^{p} \begin{bmatrix} C_{n,0} \\ C_{n,1} \\ \vdots \\ C_{n,p} \end{bmatrix}, & 2 \leq n \leq p+1
\end{align*} \]

TABLE 2: Definition of the parameters used for the analytical description of the split Schur algorithm's FP error propagation mechanism.

\[ \begin{align*}
\gamma_{n+1} &= \gamma_n + \Delta (n+1), & 1 \leq n \leq p \\
\xi_n &= \xi_{n-1} + \Delta (n), & 1 \leq n \leq p \\
\zeta_{n+1} &= \zeta_n + \Delta (n+1), & 1 \leq n \leq p \\
\eta_n &= \eta_{n-1} + \Delta (n), & 1 \leq n \leq p \\
\chi_n &= \chi_{n-1} + \Delta (n), & 1 \leq n \leq p \\
\delta_n &= \delta_{n-1} + \Delta (n), & 1 \leq n \leq p
\end{align*} \]
Multidimensional Wave Digital Filters by Use of Transformation Principle

Wenzhe Li

Ruhr-Universität Bochum, Lehrstuhl für Nachrichtentechnik
Universitätsstr.150, P.O.Box 102148, D-4480 Bochum 1, FR Germany

An approach has been presented to the design of multidimensional (MD) wave digital filters (WDF) with various passband behaviors by use of transformation principle. Two kinds of filters, 2-D hexagonal filter and 3-D octaeder filter have been designed by using this method. The realizations of these filters are given to show how the frequency response and the condition for massive parallelism in the direction of recursion can be satisfied at the same time. Two numerical examples are calculated also.

1 Introduction

MD WDFs enjoy popularity as 1-D WDFs for their low sensitivity and good stability properties [1, 2]. For MD WDFs with a so-called open loss-less behavior, fan filters for example, there exists already an analytical solution [3]; but for those with a closed loss-less behavior, still no general solutions have been found [1,2,4]. The approach of using complex network theory seems to be a solution for the latter [9], unfortunately difficulty arises due to the undesired parasitic passbands. Hence the problem, how such a MD WDF i.e., a suitable MD passive lossless two-port can be derived or synthesized, remains still open.

The purpose of this paper is to present an approach to the design of MD WDFs with closed loss-less behavior. This approach uses a transformation principle given in [5, 6] which has been originally used in physical systems that are simulated by MD WDF algorithm with massive parallelism. Such transformation principle then will be applied to those passive lossless two ports given in [7], with which two kinds of WDFs are designed. It shows threewith that such a MD WDF on the one hand realizes the desired frequency response and on the other hand ensures an algorithm that offers massive parallelism in the direction of recursion. Some realization considerations are discussed, and two numerical examples, a 2-D hexagonal filter and a 3-D octadedr filter are given also to show the design procedure.

2 Principle of transformation

The transformation method given in [5] can be explained as the coordinate transformation between a k-D causal system and a k-D noncausal system with k' ≥ k. In order to use such transformation for filter design, the following discussions are needed.

To design a k-D filter, one should begin with a k-D causal filter. Then, the k-D filter can be obtained by using coordinate transformation in the k'-D causal filter. The k-D filter obtained in this way is still causal, but can be considered as a k-D noncausal filter with standard sampling, i.e. sampling with sampling matrix of \( T = diag(T_1, \ldots, T_k) \). Since the transformation, as will be seen in the following, causes a change in the sampling matrix, and thus a change in the sampling lattice, the transformed k-D filter is equivalent to a k-D causal filter that works with another new sampling matrix. To this sense, the k-D filter is embedded simply in the k'-D causal filter that will be designate as the auxiliary filter also, and the passivity of the k-D filter thus obtained can be ensured if the k'-D filter itself is (internally) passive.

Let \( t' \) and \( t \) denote the coordinate vectors of the k'-D filter system and the k-D filter system respectively, with

\[
\begin{align*}
t' &= (t'_1, \ldots, t'_k)^T = T' n', \\
t &= (t_1, \ldots, t_k)^T = T n, \\
n' &\in \mathbb{Z}^{k'}, n \in \mathbb{Z}^k, k' \geq k,
\end{align*}
\]

\( p' = (p'_1, \ldots, p'_k)^T, p = (p_1, \ldots, p_k)^T \) the complex frequency vectors and \( \psi' = (\psi'_1, \ldots, \psi'_k) \), \( \psi = (\psi_1, \ldots, \psi_k) \) the equivalent complex frequency vectors, \( T' \) and \( T \) being the sampling matrix of the two systems for standard sampling. Using the known bilinear-transformation for variables \( z' = e^{p' T'} \) and \( z = e^{p T} \), it follows

\[
\begin{align*}
\psi' &= \tanh \left( \frac{z'}{2} T' p' \right), \\
\psi &= \tanh \left( \frac{z}{2} T p \right)
\end{align*}
\]

with notation \( \tanh(x) = (\tanh(x_1), \ldots, \tanh(x_n))^T, x = (x_1, \ldots, x_n)^T \). The relation between the two filter systems then can be expressed as

\[
n = M n',
\]

where \( M \) is a \( k \times k' \) integer matrix with \( \tanh(M) = k \), for which there exists at least a right inverse matrix \( M^{-R} \), such that \( MM^{-R} = 1_k \) is valid, \( 1_k \) being the unit matrix of k order. Considering (3) in (1), the coordinates of the two filter systems are related by

\[
\begin{align*}
t &= TMT^{-1} t', \\
t' &= T' MM^{-R} T^{-1} t'
\end{align*}
\]

Observing the product \( p'^T t' \), one has according to (4)
\[ p^{T}t' = (T^{-T}M^{-TR}T^{T}p)^{T}t = p^{T}t, \]  
(5)

where
\[ p = T^{-T}M^{-TR}T^{T}p', \]
(6)
or written in another form
\[ T^{T}p' = M^{T}T^{T}p. \]
(7)

Using this relation in (2) yields
\[ y' = \tanh \left( \frac{1}{2} M^{T}T^{T}p \right) = \tanh \left( \frac{1}{2} T^{T}p \right) \]
(8)

with
\[ \hat{T} = TM. \]
(9)

Accordingly, as is implied in (7) and (8), the \( k' \)-D filter is now embedded in the \( k' \)-D auxiliary filter, and in fact with a new sampling matrix \( \hat{T} \). If the \( k' \)-D auxiliary filter is (internally) passive and the transformation matrix \( M \) is well defined according to some definite criteria stated in [6], one can expect to obtain, with respect to the auxiliary filter, a \( k' \)-D filter that reflects itself in a causal behavior in the auxiliary filter. In this way, the passivity of the auxiliary filter would be preserved in the \( k' \)-D filter. This property is advantageous especially if the \( k' \)-D auxiliary filter is a WDF, since then the internal passivity (or MD passivity) [6] of the \( k' \)-D auxiliary filter and those good stability properties thus derived can be retained in the transformed \( k' \)-D filter, and the transformed \( k' \)-D filter obtained remains to be a WDF.

It can be seen from the above discussions that the matrix \( M \) plays an important role for the whole transformation. It should be chosen in a proper way such that (internal) passivity property can be preserved and massive parallelism can be achieved in the \( k' \)-D filter. In [6] there are altogether three essential criteria associated with the determination of \( M \):

1. Any change \( \Delta k > 0 \) in \( k \) alone must create changes \( \Delta k' > 0 \) in the \( t_{k}' \), \( \kappa = 1 \) to \( k' \). Moreover, there must exist at least one \( k \)-dimensional subspace of \( t_{k}' \) to \( t_{k}' \), such that the above requirement holds. In other words, there must exist at least one right inverse \( M'^{-1} \) such that any change \( \Delta k > 0 \) in \( k \) alone creates changes \( \Delta k' > 0 \) in the \( t_{k}' \), \( \kappa = 1 \) to \( k' \).
2. Any change of one of the \( t_{k}' \) for which \( \Delta t_{k}' > 0 \), \( \kappa \in \{1, 2, \ldots, k'\} \) must create a change \( \Delta k > 0 \).
3. The structure of the transformation should be as simple as is reasonably feasible.

Note that the second requirement ensures the possibility to massive parallelism. From an immediate observation, it requires that all elements of the last row of \( M \) must be positive.

3 Filter design and realizations

3.1 2-D hexagonal filter

The transformation matrix \( M \) for this filter is chosen as

\[ M = \begin{pmatrix} -1 & 1 & 0 \\ 1 & 1 & 2 \end{pmatrix}, \]
(10)

accordingly one has \( k = 2 \) and \( k' = 3 \), i.e. the design of the 2-D filter should be started with a 3-D filter. It is easy to test that \( M \) satisfies all the requirements listed in the last section. And the corresponding new sampling matrix \( \hat{T} \) to the 2-D filter is calculated according to (9) as

\[ \hat{T} = \begin{pmatrix} -T_1 & T_1 & 0 \\ T_2 & T_2 & 2T_2 \end{pmatrix} = (T_1; T_2; T_2), \]
(11)

where \( T_1, i = 1, 2, 3 \) are shift vectors to \( T \), with which the sampling lattice can be determined directly. Note that for the given \( \hat{T} \) the corresponding period lattice \( \Omega \) calculated from \( \hat{T}\Omega^{T} = 2\pi \Omega \) is not unique because of the nonuniqueness of \( T^{-R} \) for \( k' > k \), but all \( \Omega \) describe the same reciprocal lattice.

Here, the case by setting \( T_1 = \sqrt{3}T_2, T_2 = T \) in (11) is of interest because accordingly a hexagonal sampling can be derived; the sampling lattice and the corresponding reciprocal lattice are shown in Fig.1, where the Nyquist region may possess a hexagonal shape also.

![Fig.1: Sampling and reciprocal lattices according to the sampling matrix of (11).](image)

Employing (11) in (8) and considering (2), one obtains for the transformation in frequency domain

\[ \psi'_{1} = \frac{\psi_{2} - \psi_{11}}{1 - \psi_{11}} = \tanh(T^{p} - \sqrt{3}p_{1}), \]
(12a)

\[ \psi'_{2} = \frac{\psi_{2} + \psi_{22}}{1 + \psi_{22}} = \tanh(T^{p} + \sqrt{3}p_{2}), \]
(12b)

\[ \psi'_{3} = \frac{2\psi_{3} - \psi_{33}}{1 + \psi_{33}} = \tanh(p_{3}T). \]
(12c)

With this transformation the design of a 2-D hexagonal lowpass filter is considered as follows. Since quadrant symmetry is required in the frequency response, the 3-D filter needed should possess the same symmetry as [5]. In Fig.2(a) a suitable 3-D two-port whose frequency response is lowpass with an approximately cubic passband [7] has been chosen as the reference circuit for the 3-D filter. It consists of three 1-D subnets \( N'_{i}(\psi'_{i}) \) to \( N'_{i}(\psi'_{i}) \), each of which is a lossless two-port also and in fact a symmetric lattice two-port; they are isolated by gyrator-inductor circuits \( N'_{i}(\psi'_{i}) \) from each other. A good cubic passband response in the 3-D filter would be obtained if each subnetwork is designed as 1-D lowpass by use of approximations such as Butterworth, Chebyshev and so on [5].
The design procedure presented above is relatively simple and no optimization has to be needed. But this design is limited to 2-D filters whose passband extends to $\frac{2\pi}{3\sqrt{3}}$ (the maximum edge frequency of the hexagonal passband); if the edge frequency of the passband is greater than that, parasitic passbands may appear in the corner of the Nyquist region. In this case, the two-port of Fig.2(a) should be modified as a 4-D two-port with \( N'(\psi'_1) \) replaced by \( N'(\psi'_2) \) and \( \psi'_2 = \psi'_1 + \psi'_3 \). This means that the transformation should be first made in reference domain for the 4-D two-port, which then yields a new 3-D two-port. Again, the 2-D hexagonal filter can be obtained by using the transformation of (12) to this 3-D two-port.

### 3.2 3-D octaeder filter

The transformation matrix for this filter is

\[
M = \begin{pmatrix}
-1 & -1 & 1 & 1 \\
-1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 \\
1 & 1 & 1 & 1
\end{pmatrix},
\]

which yields \( k = 3 \) and \( k' = 4 \), i.e. a 4-D auxiliary filter is needed for the design of the 3-D filter. If standard sampling has been done for the 4-D filter, the sampling matrix of the 3-D filter can be derived from (9) as

\[
\dot{T} = \begin{pmatrix}
-T_1 & -T_1 & T_1 & T_1 \\
-T_2 & T_2 & T_2 & -T_2 \\
T_3 & -T_3 & -T_3 & T_3 \\
T_3 & T_3 & -T_3 & -T_3
\end{pmatrix} = (T_1; T_2; T_3; T_4),
\]

\( T_i, i = 1, 2, 3, 4 \) being again shift vectors. Consequently, the sampling lattice is shown in Fig.4(a), for which \( T_1 = T_2 = T_3 = T = \) has been assumed. Obviously, the sampling thus obtained corresponds to a so-called body-centered orthorhombic sampling [11], and the equivalent transformation in frequency domain is derived as

\[
\psi'_1 = \tanh\left(\frac{p_T - p_T - p_T}{2}\right),
\]

\[
\psi'_2 = \tanh\left(\frac{p_T + p_T - p_T}{2}\right),
\]

\[
\psi'_3 = \tanh\left(\frac{p_T + p_T + p_T}{2}\right),
\]

\[
\psi'_4 = \tanh\left(\frac{p_T - p_T + p_T}{2}\right).
\]

![Fig.3: Frequency response of the 2-D hexagonal filter.](image)

Note that the structure of this filter is advantageous to hardware realization using signal processor[8] because only two-port adaptors are needed in filter. Besides, since the direction of recursion of the 2-D filter can be chosen as \( t_2 \), the existence of the shifts between the subfilters provides more pipeline possibilities, and thus more parallelism to the realization of the filter.

![Fig.4: (a) sampling lattice according to (14); (b) passband region of the 3-D octaeder filter.](image)
4 Conclusions

k-D lowpass WDFs with various passband behaviors can be derived from a k'-D causal WDF, (lossless two-port) with (hyper-)cubic passband by use of transformation principle. Two kinds of filters, 2-D hexagonal filter and 3-D octaeder filter have been designed by this method. It shows thereby that the designed filters on the one hand realize the desired frequency response and on the other hand ensure an algorithm that offers massive parallelism in the direction of recursion. Two numerical examples are calculated and the corresponding WDF realizations have been given also which show the effectiveness of this approach.

Acknowledgement

The author is deeply indebted to Prof. A. Fettweis for his helpful suggestion and wishes to thank DAAD (Deutscher Akademischer Austauschdienst) for the support of this work.

Fig.6: Frequency response of the 3-D octaeder filter.

References


Adaptation of Grey Level Structuring Elements for Morphological filters with Application to Shape Detection

Philipp Salemblier
Harvard Robotics Laboratory
Harvard University
Cambridge, MA 02138, USA

Signal Processing Laboratory
Swiss Federal Institute of Technology
CH-1015, Lausanne, Switzerland

Abstract: This paper deals with algorithms for the adaptive optimisation of grey level structuring elements of morphological filters in discrete spaces. The adaptation process is similar to the classical least mean square algorithm used for linear filters and its goal is to minimise a statistical criterion such as the mean square (or mean absolute) error between the filter output and a desired signal. It is shown how a grey level structuring element can be optimised in the case of erosion and dilation. Then, adaptation formulas are derived for an arbitrary composition of erosions and dilations (optimisation of opening, closing, opening/closing or alternating sequential filters). Moreover, the results are further generalised to filters obtained by combination of various filters through minimum and maximum operations (optimisation of maximum of openings or morphological centres). One of the major advantages of the proposed approach is that it leads to simple adaptation formulas and allows fast convergence rate. Finally, the approach is illustrated on grey level shape detection problems.

Adaptation is used here to get the best structuring element for a particular detection problem.

1 Introduction

Morphological filters [1] have become very popular in signal and image processing because of their attractive features such as impulse noise suppression, edge preservation or shape-oriented approach to signal analysis. They analyze the geometrical structure of a signal by locally comparing it with a predefined elementary shape called a structuring element (SE). Most of the time, the choice of a specific filter is performed with the so-called structural approach. It consists in analyzing and selecting a relevant set of signal structures, and in choosing a filter which is able to properly interact with these structures. It leads very often to rely on the geometrical properties of the signal in order to choose a specific SE. Even if this approach has been successfully applied in many practical cases, a large number of researches are devoted to alternative solutions. Indeed, the structural approach often requires a fair amount of practical experience and intuition.

Recently, alternative approaches have been proposed [2,3]. They formulate the design as an estimation problem. The filter is optimised to minimise a criterion such as the mean square error (MSE) or the mean absolute error (MAE) between its output and a desired signal. In a similar way to that of linear filter optimisation, this approach yields two different types of solutions. The first one attempts to find an optimal fixed filter [2]. However, it has several drawbacks: it assumes stationarity of the various signals which is almost never the case in image processing. It also requires a certain knowledge about the noise characteristics which is often difficult to obtain. Another drawback is the computational complexity of the approach.

To avoid these difficulties of fixed optimum filters, a second approach involving adaptive optimisation problem has been proposed [3,4]. The algorithms reach the optimum point step by step and, at least partially, overcome the drawbacks mentioned above. This study is in fact an extension of an earlier work on adaptive rank order based filters [5] where algorithms have been proposed to adapt the filter mask or rank in order to minimise the MSE or the MAE. As pointed out in [3,5], this filter class includes as special case the erosion and the dilation with flat SE and all morphological filters obtained by composition of these two basic operators. The goal of this paper is to present adaptation algorithms in the context of morphological filters with grey level (not flat) SE. The solutions are valid for any composition of erosions and dilations, and also for combinations of filters with minimum and maximum operations. This allows adaptation of filters such as maximum of openings or morphological centres, etc.

The organisation of this paper is as follows: the next section defines more precisely the adaptation problem. The third section deals with formulations of the basic erosion and dilation operators with grey level SE. Then, the fourth section is devoted to the optimisation of erosion and dilation. The results are extended in the fifth section to filters obtained by composition of erosion and dilation and by combination with minimum and maximum operations. Finally, practical problems of shape detection are addressed in section six.

2 The optimisation problem

The adaptation problem can be stated as follows: let $x_i$ denote an original signal, $d_i$ a desired signal and $F$ a morphological filter with a SE defined by the set of real values $\{m_k\}_{k \in M}$ and $y_i$ the filter output. The optimisation tries to find the best set of parameters $\{m_k\}$ to minimise the MSE or MAE between the filter output and the desired process. The MSE optimisation can be statistically achieved in an iterative way using the LMS approach given by [6]:

$$m_k^{i+1} = m_k^i - \mu \frac{\partial (d_i - y_i)^2}{\partial m_k} = m_k^i + 2 \mu (d_i - y_i) \frac{\partial y_i}{\partial m_k} \quad (2.1)$$

where $\mu$ is the convergence parameter. As pointed in [5] the MAE optimisation can be approximated by replacing the $(d_i - y_i)$ term in (2.1) by its sign.

With this approach, the only difficulty consists in computing the partial derivative of the filter output with respect to the parameter to optimise. The next section is
devoted to the definitions of erosion and dilation and to the proposition of alternative formulations more suitable for the computation of the partial derivative mentioned above.

3 Morphological filters

Most morphological filters are obtained by composition of erosions and dilations and by combinations with minimum and maximum. For discrete space and grey level SE, the output $y_i$ of an erosion (resp. dilation) is the minimum value of $(x_{i+j-m_j})$ (resp. maximum value of $(x_{i+j+m_j})$):

$$E(x_i) = \min \{x_{i+m_j} \; j \in M\}$$
$$D(x_i) = \max \{x_{i+m_j} \; j \in M\}$$

To solve the optimisation problem, an alternative formulation of these basic operators is proposed now. Note that this formulation is derived in a similar way to that used in [3,5] to deal with rank order based filters and flat SE. The basic idea of the approach is to express the erosion and the dilation with easily differentiable functions. Let $S$ be the sum of signs of $(x_{i+m_j} - y_i)$ for all $j$. For an erosion, the filter output is the minimum of the set $(x_{i+m_j})$, thus all terms $(x_{i+m_j} - y_i)$ are strictly positive except one of them which is equal to zero (we assume that only one such point exists. It can be shown that this approximation allows simplification of the algorithm without significant loss in performances). If the sign of $S$ is defined as being 0, $S$ is equal to the size of the SE (N minus 1):

$$S = \sum_{j \in M} \text{sgn} (x_{i+m_j} - y_i) = N - 1$$

This equation gives the following implicit formulation of the erosion: the output $y_i$ of an erosion at location $i$ is the value of the set $(x_{i+m_j})_{j \in M}$ such that:

$$f(x_i, y_i, m_j) = \sum_{j \in M} \{\text{sgn} (x_{i+m_j} - y_i) - 1\} + 1 = 0$$

The alternative definition of the dilation can be derived in a similar way. Let us see how these implicit formulations can be used to get the derivative of the filter output with respect to the parameter to optimise.

4 Adaptive erosions and dilations

The partial derivative of the filter output with respect to the $m_k$ can be obtained by computing the total derivative of the implicit function $f$. It gives the following result:

$$\frac{\partial y_i}{\partial m_k} = \frac{\partial f}{\partial m_k} \cdot \frac{\partial f}{\partial y_i}$$

The computation of the derivative of $y_i$ with respect to $m_k$ is reduced to the computation of two derivatives of $f$ with respect to $m_k$ and $y_i$. In the case of an erosion, the first one can be extracted from (3.4):

$$\frac{\partial f}{\partial m_k} = \sum_{j \in M} \text{sgn}(x_{i+m_j} - y_i) \cdot \frac{\partial f}{\partial m_k} = - \delta(x_{i+m_k} - y_i)$$

For the second one, (3.4) gives:

$$\frac{\partial f}{\partial y_i} = \sum_{j \in M} \text{sgn}(x_{i+m_j} - y_i) \cdot \frac{\partial f}{\partial y_i} = - \delta(x_{i+m_k} - y_i)$$

This last expression can be simplified. Indeed, the case where all $(x_{i+m_j})$ are different from zero except one of them (the output is assumed to come from a single location), the summation of (4.3) reduces to a single term:

$$\frac{\partial f}{\partial y_i} = - \delta(x_{i+m_k} - y_i)$$

Finally, combining these equations gives the partial derivative of $y_i$ with respect to $m_k$:

$$\frac{\partial y_i}{\partial m_k} = - \delta(x_{i+m_k} - y_i)$$

This formula has a very simple interpretation. It means that the derivative of the filter output $y_i$ with respect to a particular parameter $m_k$ is equal to zero for all $k$ except for the (single) one which corresponds to the location which has produced the output. In this case, the derivative estimate is simply -1.

The same approach can be used for the dilation with obvious modifications. It gives:

$$\frac{\partial y_i}{\partial m_k} = \delta(x_{i+m_k} - y_i)$$

Let us see now how these formulas can be extended to more complex and useful filters.

5 Extension to morphological filters

The approach of section 4 can be generalised to deal with arbitrary compositions of erosions and dilations: suppose that a filter $F_1(\cdot)$ is the composition of $T$ elementary operators (erosions or dilations), and that the $p$th operator in the composition is denoted by $F_{p_k}(\cdot)$ where $p$ is the position of the operator in the composition and $\epsilon_p$ is it type defined by $\epsilon_p = 1$ for an erosion and $\epsilon_p = -1$ for a dilation. The global filter is therefore:

$$y_i = F_1(x) = F_{T_1}(F_{T_2}(\ldots F_p(x) F_{p_{T-1}}(x))))$$

Let $x_i^p$ denote the signal at the output of the $p$th filter:

$$x_i^p = F_{p}(x)$$

The adaptation rule involves the trajectory of the final output $y_i$ in the various filtering levels. Indeed, the output of an erosion or a dilation with grey level SE is always one of the input values at same location plus or minus the value of the SE at that location. Thus, the final output at location $i$ denoted by $y_i$ comes from a particular location $i + \Delta k_1$ at the level $T - 1$. In turn, this value comes from a particular location $i + \Delta k_2$ at the level $T - 2$. Let us denote by $i + \Delta k$ the location at level $p$ (the output value is supposed to come from a single location) which corresponds to the output value $y_i$ and $m_\alpha$ the value of the SE which was used at that location:

$$x_i^p = m_\alpha$$

With these notations, it can be shown [4] that the gradient estimate of $y_i$ is given by the following formulas:

$$\frac{\partial y_i}{\partial m_k} = \sum_{p=1}^{T} \epsilon_p \delta(x_{i+m_k} - x_i^p)$$

This rule can be used to easily derive an adaptive algorithm for an arbitrary composition of erosions and dilations. As an example, the adaptation formulas for an opening and a closing are respectively given by:

$$\frac{\partial y_i}{\partial m_k} = - \delta(x_{i+m_k} - x_i^1) + \delta(x_{i+m_k} - x_i^2)$$

$$\frac{\partial y_i}{\partial m_k} = - \delta(x_{i+m_k} - x_i^1) + \delta(x_{i+m_k} - x_i^2)$$

$$\frac{\partial y_i}{\partial m_k} = - \delta(x_{i+m_k} - x_i^1) + \delta(x_{i+m_k} - x_i^2)$$

$$\frac{\partial y_i}{\partial m_k} = - \delta(x_{i+m_k} - x_i^1) + \delta(x_{i+m_k} - x_i^2)$$
This approach already allows to deal with a large class of morphological filters. However, it cannot be directly used for more sophisticated filters such as maximum of openings or morphological centres. In the case of combinations of subfilters with min or max operations, it can be shown [4] that the derivative of the filter output is equal to the derivative of the subfilter which has produced the filter output. This very simple result is particularly useful to deal with filters such as maximum of openings, minimum of closings or morphological centres and more generally any toggle mapping.

6 Applications to shape detection

This section describes two examples of grey level shape detection where adaptation proves to be a very useful tool. The first example concerns the extraction of a given shape in a noisy environment. A typical example is presented in the upper left part of figure 1. This image is composed of black grey level shapes randomly located on a noisy background. Suppose that, the ideal shape detection result is known for one or a few original images. Then, a detection system can be optimised with these pairs of original and ideal images. Finally, the optimum detection system can be used to automatically process a very large number of original images. The goal of this section is to discuss two such examples of supervised optimisation.

In the case of the original image of figure 1, the dark grey level shapes can be extracted by a closing. However, the closing may also extract noise components. For this detection result, the most natural choice is to take as SE the shape to detect itself. The image obtained after a closing with the grey level shape to extract is shown in the lower left part of figure 1. One can see that the shapes are actually enhanced but the noise has still a very strong influence.

Let us assume that the ideal detection result is known. This desired image is shown in the upper right part of figure 1. A grey level SE is optimised in order to minimise the MSE between the image obtained after a closing on the original image and the desired image. From our experience, it seems that this optimisation problem involves several local minima, one should start the optimisation procedure with a good initial guess. In our case, the shape to detect is selected as initial SE. From the practical point of view, a grey level SE of size 15x15 is optimised following the MSE criterion and the image is scanned several times (about 10 times with \( \mu=6 \)) in order to get convergence.

Figure 2 presents the original SE, that is the shape to detect, and the resulting SE after optimisation. The optimised SE is flatter than the original one to remove noise more efficiently. However, the global shape of the original SE is preserved. The optimisation makes a compromise between shape detection and noise cancellation. The image obtained after closing with the optimised SE is shown in the lower right part of figure 1. As one can see, the contrast is efficiently improved. Finally, the MSE between the various images of figure 1 and the desired one are computed. The resulting figures are as follows:

- original image / desired image: MSE = 6438
- closing with shape to detect / desired image: MSE = 3454
- closing with optimised SE / desired image: MSE = 2846

Finally, a similar experiment is done with an original image composed of two different shapes. A typical example made of elongated dark spots and of dark rings is shown in the upper left part of figure 3. Here the goal is to eliminate as much as possible one shape while preserving the other one. The ideal result is shown in the upper right part of figure 3.

As previously, a detection can be performed by a closing with the shape to detect. The resulting image is shown in the lower left part of figure 3. The ring shapes are less dark than in the original image but are still quite visible. To improve this result, an optimisation procedure is done. The initial SE is the shape to detect and a grey level SE of size 15x13 is optimised to minimise the MSE between the output of a closing and the desired image. The initial and optimised shapes are presented in figure 4. As can be seen, the optimised SE is somewhat flatter than the original one, and there is a small depression in its centre to efficiently remove the ring shapes. The image obtained after a closing with the optimised SE is presented in the lower right part of figure 3. This result clearly demonstrates the benefit of using the optimised SE. Finally, the following table gives the MSE:

<table>
<thead>
<tr>
<th>Type of Calculation</th>
<th>MSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>original image / desired image</td>
<td>524</td>
</tr>
<tr>
<td>closing with shape to detect / desired image</td>
<td>419</td>
</tr>
<tr>
<td>closing with optimised SE / desired image</td>
<td>213</td>
</tr>
</tbody>
</table>

Conclusions

In this paper, several adaptive algorithms for the optimisation of grey level SE of morphological filters in discrete spaces have been proposed. The adaptation process is similar to the classical LMS algorithm which minimises the MSE (or MAE) between the filter output and a desired signal. It is shown how a grey level SE can be optimised in the case of erosion and dilation, arbitrary compositions of erosions and dilations (optimisation of opening, closing, open_close or alternating sequential filters) and also filters obtained by combination of various filters outputs using min/max operations (optimisation of maximum of openings or morphological centres). One of the major advantages of the proposed approach is that it leads to simple adaptation formulas and allow fast convergence rate. Finally, the approach is illustrated on grey level shape detection problems. Adaptation is used here to get the best structuring element for a particular detection problem.

References

Original image

Desired image

Closing with shape to detect

Closing with optimised SE

Figure 1: Noise cancellation with adaptive closing

Optimised SE

Original SE

Figure 2: Struct. el. used in figure 1

Original image

Desired image

Closing with shape to detect

Closing with optimised SE

Figure 3: Shape detection with adaptive closing

Optimised SE

Original SE

Figure 4: Struct. el. used in figure 3
Use of Bitonic Sorters for Rank Order Filtering

Lale Akarun, Richard A. Haddad and Cem Ersoy

Polytechnic University, Electrical Engineering Dept., 333 Jay Street, Brooklyn, NY 11201, USA

Bitonic sorters are widely used in sorting networks. We show that a new technique for rank order filtering, the decimated rank order filter (DROF), is closely related to bitonic sorters. The DROF can remove more impulses than a rank order filter of the same window size, with better edge performance and at lower computational cost.

1 Introduction

Rank order filters have been preferred to linear filters for impulsive noise removal from signals that contain sharp edges. The median filter is the most widely used rank order filter. It is noted for its ability to remove impulsive noise and to preserve sharp changes in the signal. However, there is a trade-off between noise removal capability and edge retention. When impulses occur in the neighborhood of edges, those edges are no longer preserved. Another trade-off is noise removal capability versus computational cost: complexity increases with the window size, and large window sizes are necessary for the removal of wide impulses.

The decimated rank order filter (DROF) [1] alleviates these problems: it improves noise removal while reducing edge smoothing and computational complexity. The DROF uses decimation to split long impulses into shorter ones which can then be removed by rank order filters of smaller window size. The small window sizes ensure good edge performance and low computational complexity. The decimators and the rank order filters are combined in a way that is similar to the filter-bank structure used in multirate signal processing [2].

The decimated sorter is obtained by replacing the rank order filters of the DROF by sorting elements. The decimated sorter, in general, does not sort any given sequence. However, under certain conditions, the operation performed by the decimated sorter is equivalent to sorting. We show that this condition is that the input signal be bitonic, i.e., a signal that consists of two sections, one, monotonically increasing, the other, monotonically decreasing. Therefore, the decimated sorter is a bitonic sorter. Bitonic sorters are well known in sorting networks [3], [4] and in switching [5], [6]. We show that the bitonic sorter of [5] is a special case of the decimated sorter.

Section 2 describes bitonic sorters, their structure and their areas of application. In Section 3, we introduce the decimated rank order filter and discuss its properties. In Section 4, we define the decimated sorter as an extension of the DROF. We show that the decimated sorter is equivalent to the bitonic sorter described in Section 2. Section 5 presents our conclusions.

2 Bitonic Sorters

A sequence of $W$ numbers, $(x_1, \ldots, x_W)$ is bitonic if it is first monotonically increasing up to some point $x_m$, $1 \leq m \leq W$, and monotonically decreasing after that point (or vice versa). Sorting a bitonic sequence is the same problem as merging two sorted lists: If the two sorted lists, one in increasing and the other in decreasing order, are juxtaposed, they form a bitonic sequence. Batcher devised a bitonic sorting network for merging two monotonic sequences into a single sorted sequence [5]. The structure of a bitonic sorter for $W = 8$ is shown in Fig. 1. In this structure, $W/2$ comparators (2-point sorters) are used to break the problem down. Batcher also proved that this structure is recursive, i.e., the $W$-item sorter in Fig. 1 is also a bitonic sorter. Large bitonic sorters are constructed from a number of smaller bitonic sorters: This modularity makes the architecture very suitable for VLSI applications. When the recursive structure is fully exploited and only comparators are used, $\log_2 W$ layers of $W/2$ comparators are needed. The depth of the network is $\log_2 W$. The comparators in each layer can operate in parallel, resulting in a delay of only $\log_2 W$ time units.

Cormen et al. [4] define a comparison network as a network consisting of wires and comparators. A sorting network; then, is defined as a comparison network whose outputs are sorted for every input sequence. The bitonic sorter is not a sorting network; it is a comparison network that sorts a limited class of inputs. However, since the bitonic sorter can merge two sorted lists, it is possible to construct a sorting network by recursively combining bitonic sorters [4]. The depth of a sorting network constructed in that way is $\log_2 n$.

A permutation network is a network that allows connection of its $n$ inputs to its $n$ outputs according to any
one of the $n!$ possible permutations. If each comparator in a sorting network is replaced with a 2-input, 2-output switch, a permutation network is obtained. Permutation networks are used for switching in ATM (asynchronous transfer mode) networks. Batcher [5] first proposed the use of bitonic sorters for switching. Hui [6] reviews the recent developments in this area.

3 Decimated Rank Order Filters

The output of a rank order filter $y_r = ROF\{x_1, \ldots, x_W\}$ is defined as the $r$th smallest among $\{x_1, \ldots, x_W\}$. For $W$ odd, $r = (W - 1)/2$ gives the median, whereas $r = 1$ and $r = W$ give the minimum and the maximum, respectively.

The median filter is the most commonly used rank order filter. It is preferred to linear filters when noise is of impulsive nature and the original signal contains sharp edges that are to be preserved. The noise reduction capability of a median filter increases with increasing window width. However, with non-constant background, wider window sizes also lead to smearing of details and edges in the signal. Another disadvantage of wide windows is the computational burden they bring. The decimated median filter removes a large number of impulses with better detail and edge retention than the standard median filter and with decreased computational cost.

3.1 Decimated Median Filter

We use decimation to deal with the problems induced by longer window lengths. The idea of linking decimation and rank order filtering arises from the following observation [7]: Large median filter windows are usually desired to remove impulses of longer length. However, long impulses can be split down into shorter ones by decimation. For example, if odd indexed samples and even indexed samples are separated to produce two individual subsequences, these sequences contain impulses of half the length of the impulse size in the original signal. The smaller impulses can then be removed by median filters of smaller window size, which do not cause the edge smearing problems the larger window sizes lead to. The overall computational complexity is also reduced. The structure of the Decimated Median Filter (DMF) is given in Fig. 2, where the output $z$ is defined as follows:

$$z = \text{Med}\{y_1, \ldots, y_M\}$$
$$y_i = \text{Med}\{x_j, x_{j+M}, \ldots, x_{j+(K-1)M}\}, \quad j = 1, \ldots, M$$

where the sequence $\{x_1, \ldots, x_W\}$ is obtained from the signal $x(n)$ by taking the samples inside a sliding window of $W = KM$ points centered at $n$. The decimators split up long impulses and the $K$-point median filters in the first layer remove them without detail and edge smoothing. The combining median filter in the second layer filters out any residual impulses. The $K$-point median filters in the first layer operate at a rate that is $(1/M)^{th}$ of the original sampling rate; therefore, each $K$-point filter operates once every $M^{th}$ sampling instant. At any sampling instant, only one of the $M$ filters is active, all others are idle. On the other hand, the $M$-point median filter in the second layer operates at the input sampling rate. This leads to a computational improvement factor, $I$ of the DMF over the standard median:

$$I = \frac{M^3K^2}{M^2 + K^2}$$

where the use of $O(n^2)$ algorithms is assumed. (This is a reasonable assumption since $n = W$ is small.)

The performance of the filter, on the other hand, is enhanced. The DMF removes wide impulses while leading to less edge shifting than the standard median filter. Moreover, the DMF has the potential for removing more impulses than the standard median filter if they occur in a certain pattern. The maximum number of impulses
removed by the DMF, MIR, is:

\[
\text{MIR} = \frac{(M - 1)K}{2} + \frac{(M + 1)(K - 1)}{4}
\]  

(3)

which is larger than \((KM - 1)/2\), the maximum number of impulses removed by a standard median filter of the same window length, \(W = KM\). The example in Fig. 3 illustrates these advantages. Fig. 3a shows a test signal that contains sharp edges and additive positive impulse noise. The results of filtering the noisy signal with a standard 9-point median filter (SM) and a DMF with \(M = K = 3\) are given in Fig. 3b. It is seen that the DMF removes more impulses than the standard median filter. The DMF also leads to less edge shifting. Fig. 3c displays the errors produced by both filters. The mean absolute error (MAE) figures are 0.87 for the DMF and 1.18 for the SM. The computational improvement factor, as computed in Equation 2 is \(I = 4.5\).

Since the filters considered are nonlinear, it is not possible to analyze their response to noise and predict their performance in removing noise from signals. A simplifying approach assumes that the signal will either be constant or a step edge in the given window. The output distribution is usually derived for these two cases under a white noise assumption [8]. The derivation of the output distribution of the DMF is simple for independent identically distributed (iid) inputs. The inputs of the \(K\)-point median filters in Fig. 2 are distinct; therefore, the outputs are also independent and identically distributed. The formulation of the output distributions for constant and step edge backgrounds have been detailed in [1]. Using those formulas, the output mse (mean square error) has been computed using numerical integration for iid noise with Laplacian distribution (Fig. 4). The filters compared are a DMF with \(M = K = 3\), and a standard median filter with \(W = 9\). The mse is plotted against edge height, \(h\) for a step edge at the middle of the window. When \(h = 0\), the step edge is reduced to a constant and the filters perform similarly. For edge heights \(h > 2\), the DMF shows a lower mse figure. This result agrees with our observation: The DMF performs similar to the SM in constant background regions but surpasses it near edges.

### 3.2 Decimated Rank Order Filter

To obtain decimated rank orders other than the median, we replace the median filters of Fig. 2 with other rank order filters. The output of the DROP, \(z\) is:

\[
x = \mu\{y_1, \ldots, y_M\}
y_j = \kappa\{x_{j}, x_{j+M}, \ldots, x_{j+(K-1)M}\}, \quad j = 1, \ldots, M
\]

(4)

where \((\kappa, \mu)\) denote a pair of rank order filters, the former, \(K\)-input, and the latter, \(M\)-input. With this representation, \((\text{MED,med})\) represents the filter of Fig. 2.

![Figure 3: (a) An example signal with 15% impulsive noise (b) The noisy signal as filtered by DMF \((M = K = 3)\) and standard median (SM) filter \((W = 9)\) (c) The error produced by each filter](image)

![Figure 4: mse (mean square error) figures for SM and DMF for a step edge signal and Laplacian noise (0,1)](image)
If \( M = K = 3 \), it is possible to generate all decimated rank orders in a window of nine points with 3-point minimum, maximum, and median operations. It is easy to see that \((\text{MIN}, \text{min})\) gives the minimum of the 9 points, or decimated rank order number one. Likewise, \((\text{MIN}, \text{med})\), \((\text{MIN}, \text{max})\), \((\text{MED}, \text{min})\), \((\text{MED}, \text{med})\), \((\text{MED}, \text{max})\), \((\text{MAX}, \text{min})\), \((\text{MAX}, \text{med})\), and \((\text{MAX}, \text{max})\) generate decimated rank orders 2 through 9.

4 Decimated Sorters

If the structure in Fig. 2 is generalized to obtain all decimated rank orders, the decimated sorter is obtained. The input to the decimated sorter is the sequence \( \langle x_1, \ldots, x_N \rangle \). The input to the \( j \)th K-point sorter in the first layer is the sequence \( \langle x_j, x_{j+M}, \ldots, x_{j+(K-1)M} \rangle \). The output \( \langle y^j \rangle \) of the \( j \)th sorter is such that \( y^1_j \leq y^2_j \leq \cdots \leq y^K_j \).

The output sequence \( \langle z_1, \ldots, z_{K^2} \rangle \) consists of the juxtaposition of all subsequences that are obtained by sorting the \( M \) sequences \( \langle y_1, \ldots, y^M \rangle \):

\[
\begin{align*}
\{ z_{(r-1)K+1}, \ldots, z_{rK} \} &= \{ y_1^1, \ldots, y^M_r \} \\
\quad \quad \quad z_{(r-1)K+1} &\leq z_{(r-1)K+2} \leq \cdots \leq z_{rK}, \quad r = 1, \ldots, K
\end{align*}
\]

(5)

The sequences \( \langle z_1, \ldots, z_{K^2} \rangle \) are obtained by taking the \( r \)th largest output from each of the \( K \)-point sorters and sorting them individually. For example, \( \langle z_1, \ldots, z_{K^2} \rangle \) consists of the minima from each of the \( K \)-point sorters. Note that even though the subsequences \( \langle z_1, \ldots, z_{K^2} \rangle \) sorted in themselves, \( \langle z_1, \ldots, z_{K^2} \rangle \) is not necessarily sorted. Thus, in general, the decimated sorter is not a sorting network.

**Theorem 1:** When the input sequence \( \langle x_1, \ldots, x_N \rangle \) is bitonic, the output of the decimated sorter is sorted.

This theorem tells us that the decimated sorter is a bitonic sorter. Comparison of Fig. 2 and Fig. 1 indeed reveals that the two structures are identical when \( K = 2 \), \( M = W/2 \), and the medians are replaced by sorters.

Theorem 1 also enables us to see the relationship between the root signals of the DMF and those of the SM. The root signals are defined as signals that are unchanged by median filtering [9]. The root signals of a \( W \)-point median filter consist of constant neighborhoods of at least \((W + 1)/2\) points separating monotonically increasing or decreasing edges. Inside any \( W \)-point window, such a signal can have at most three constant neighborhoods, and hence, only one minimum or maximum. Therefore, \( W \) samples of the root signals of a \( W \)-point median filter are bitonic. By theorem 1, the DMF is equivalent to the SM for a bitonic sequence. Therefore, the root signals of a median filter remain unchanged by decimated median filtering. We can state that the root signals of a median filter are also root signals of a DMF of the same window size.

**Theorem 2:** Given any bitonic input sequence, the output of the first level of the decimated sorter, \( \langle y_1 \rangle \) is also bitonic for any given \( r \).

The proofs of both theorems are given in [1]. Theorem 2 implies that we can use a recursive structure for the decimated sorter; that is, we can replace the \( M \)-point sorter with another decimated sorter. If we pick \( K \) as small as possible \((K = 2)\), we obtain the bitonic sorter of Fig. 1, in which case the depth of the whole network is \( \log W \). Alternatively, when larger values of \( K \) are used, the network is shallower but the computation at each level is more involved.

5 Conclusion

The DROF is very suitable for adaptive architectures due to its modular structure. It achieves better noise removing performance than the ROF at a lower computational cost. Parallel research in bitonic sorters gives valuable insights and suggests that the modularity can lead to successful VLSI implementations.

References


NOISE REDUCTION WITH CORRELATED SIGNALS USING HIGHER ORDER STATISTICS

Ch. SERVIERE - D. BAUDOIS

CEPHAG-ENSIIEG
BP 46 38402 Saint-Martin d’Hères cedex
FRANCE

We study noise cancelling applications when observed references contain a part of signal. We use hypothesis of decorrelation and statistical independence between sources and cancel second-order and third-order moments of linear combinations of observations. These constraints lead to solve a second order equation where solutions represent complex gains of the linear filters to estimate. Applications are reduction of rotating machines noise and separation of random signals with non zero third-order moments.

1 Introduction

This paper deals with the problem of noise cancelling in presence of noise reference with non classical hypothesis. Noise cancelling using a noise reference has been first introduced by Widrow [1]. It consists in the estimation of an useful signal [of time sequence S(k)] from two observed signals. The first one is the observed noise [X(k)=S(k)+N(k)] where N(k) is an additive noise and the second one R(k), called “noise reference”, is supposed to be correlated with N(k). S(k), N(k) and R(k) are assumed to be stationary random processes with zero-mean. Suppose the relation between N(k) and R(k) may be modeled by an unknown filter F. The solution is well-known [1] and consists first in the identification of F, by a FIR filter H. Values of coefficients of H are resulting from the minimization of a mean-square error. Their expressions only depend on second-order moments of observations X(k) and R(k). Many methods have been proposed to compute them for example with gradient methods [1] or lattice ones [2]. The estimation of N(k), is then obtained in filtering R(k) by H. [Figure 1].

 Consequently the values of h are resulting from the minimization of a mean-square error (2) at data block m and for each frequency bin i:

\[
E\left[\frac{|X_m(i) - h(i) \cdot R_m(i)|^2}{R_m(i)}\right] = \gamma_{XX}(i) - \gamma_{XR}(i) \cdot \gamma_{RR}(i)
\]

where \(X_m(i)\) and \(R_m(i)\) are obtained by p-point discrete Fourier transforms (DFT) of the m-th data block of the signals X(k) and R(k), for the i-th frequency bin. We assume that \(X_m(i)\) and \(R_m(i)\) are zeromean signals.

The solution, called "Wiener filter", is computed with the help of second-order moments. (3)

\[
h(i) = \frac{E[X_m(i) \cdot R_m(i)^*]}{E[|R_m(i)|^2]} = \gamma_{XR}(i) \cdot \gamma_{RR}(i)
\]

where \* represents the complex conjugate.

The identification of F by H is then perfect and the two complex vectors \(f\) and \(h\) are equal. In practice, \(h(i)\) is estimated at data block m, by using average periodogram methods.

2 Noise cancelling in presence of correlated inputs

In many applications, the noise reference R(k) is not accessible and a part of the useful signal S(k) is also recorded by the reference sensor. We observe then a signal Y(k) such that Y(k) is correlated with S(k). The previous method (3) replacing R(k) by Y(k) is no more usable because the two complex vectors \(f\) and \(h\) will not be equal. Performances of the estimation of S(k) will decrease and the degradation is studied in [3].

The new noise cancelling system leads to a problem of sources separation in which we generally modelized the contribution of the signal to Y(k) by an unknown linear filter G [Figure 2][4][5][6][7]

The noise cancelling system is then modelled by two inputs X(k) and Y(k) such that we have in frequency domain at data block m and for each frequency bin i:

\[
\hat{S}(k) = \frac{X(k) - h(i) \cdot Y(k)}{1 - h(i)^T \cdot G}
\]
(4) \[ X_M(i) = S_M(i) + f(i) R_M(i) \]
\[ Y_M(i) = R_M(i) + g(i) S_M(i) \]

\[ S(k) \quad + \quad X(k) \quad G \quad N(k) \quad + \quad Y(k) \quad F \]

\[ R(k) \]

Figure 2: Noise cancelling system

We now face with a problem of sources separation and have two linear filters \( F \) and \( G \) to identify, which means two complex values \( f(i) \) and \( g(i) \) to identify for each frequency bin \( i \). The proposed approach is similar to [4] [5] [6] [7].

Define first two linear combinations (5) of \( X_M(i) \) and \( Y_M(i) \).

\[ X_M(i) = X_M(i) - \alpha(i) Y_M(i) \]
\[ Y_M(i) = Y_M(i) - \beta(i) X_M(i) \]

or

\[ X_M(i) = S_M(i) [1-\alpha(i)] + f(i) [1-\alpha(i)] \]
\[ Y_M(i) = Y_M(i) [g(i) - \beta(i)] + R_M(i) [1-\beta(i)] f(i) \]

The proposed method consists in the identification of two complex coefficients \( \alpha(i) \) and \( \beta(i) \) such that each of the linear combinations (5) only depends on the signal \( S_M(i) \) or the reference \( R_M(i) \). Suppose \( \alpha(i) \) and \( \beta(i) \) solutions, we have:

\[ \alpha(i) = f(i) \] and \[ \beta(i) = g(i) \]

or \[ \alpha(i) = 1 / f(i) \] and \[ \beta(i) = 1 / g(i) \]

We remark that two couples of solutions for \( \alpha(i) \) and \( \beta(i) \) exist because the problem is totally symmetrical with regards to \( S_M(i) \) and \( R_M(i) \).

3 Case of signals with non zero third-order moments

We suppose that real and imaginary parts of the complex variables \( X_M(i) \) or \( Y_M(i) \) have non even probability densities. The third-order moments will generally not be zero. The proposed hypothesis is verified for example when one source contains sinusoids which is the case for reduction of noise of rotating machine or for particular random noises detailed in part 3.2.

3-1 Case of sinusoids in one source

We suppose that one source, \( S(k) \) for example, is the sum of a random noise \( Ns(k) \) and a sinusoid of deterministic phase \( \phi \) and deterministic frequency \( \omega \).

\( Ns(k) \) and \( R(k) \) are supposed to be random noises, third-order stationary and zero-mean. No hypothesis is made on their probability densities.

\[ S(k) = As \sin (2 \pi \omega k + \phi) + Ns(k) \]

\( R(k) \) is also supposed to be statistically independent of \( Ns(k) \).

The expression of the p-point discrete Fourier Transform of \( X(k) \) on data block \( m \) is following:

\[ X_m(i) = \sum_{k=m}^{m+p-1} \sin(\pi(\omega-\n)k) - \frac{\sin(\pi(\omega-\n)+1)p}{p} \]

\[ Y_m(i) = \sum_{k=m}^{m+p-1} \sin(\pi(\omega-\n)k) - \frac{\sin(\pi(\omega-\n)+1)p}{p} \]

We assume that \( p \) is large enough in order to approximate \( \omega \) by \( k/p \) with \( k \) integer and we propose to separate the sources for frequency bin \( i \) such that:

\[ S_m(i) = A(i) + Ns_m(i) \] with non-zero \( A(i) \).

Consequently observations at data block \( m \) are:

\[ X_m(i) = A(i) + Ns_m(i) + f(i) R_m(i) \]
\[ Y_m(i) = R_m(i) + g(i) (A(i) + Ns_m(i)) \]

We assume that \( Ns_m(i) \) and \( R_m(i) \) are statistically independent and zero-mean.

We search \( \alpha(i) \) and \( \beta(i) \) such that the two linear combinations \( X_m(i) - \alpha(i) Y_m(i) \) and \( Y_m(i) - \beta(i) X_m(i) \) only depends on signal \( S_m(i) \) or reference \( R_m(i) \).

In these conditions, the two linear combinations are uncorrelated (11):

\[ E\left\{ X_m(i) - \alpha(i) Y_m(i) \right\} \right\} \right\} = 0 \]

We verify that \( \alpha(i) \) and \( \beta(i) \) are solutions of (11) with particular form of signal \( S_m(i) \).

Unfortunately this equation is not sufficient to find the researched solution \( \alpha(i) \) and \( \beta(i) \). It only allows to obtain a family of orthogonal signals \( X_m(i) \) and \( Y_m(i) \). We will find a complementary information at third-order. Suppose \( \alpha(i) = 1 / g(i) \) and \( \beta(i) = 1 / f(i) \). Then the linear combinations \( X_m(i) \) and \( Y_m(i) \) are:

\[ X_m(i) = (f(i) - 1/g(i)) R_m(i) \]
\[ Y_m(i) = (g(i) - 1/f(i)) A(i) + Ns_m(i) \]

Then the third-order crossed moment between \( X_m(i) \) and \( Y_m(i) \) is assumed to be zero because \( R_m(i) \) and \( Ns_m(i) \) are supposed to be statistically independent and zero-mean (13).

\[ E\left\{ X_m(i) - \alpha(i) Y_m(i) \right\} \right\} = 0 \]

We will prove that these two equations are
sufficient to compute coefficients \(\alpha(0) = 1/g(0)\) and 
\(\beta(0) = 1/f(0)\) which are solutions of system (11), (13). 
Develop equations (11) and (13) in function of \(\alpha(i)\) and 
\(\beta(i)\): (14)

\[
\alpha(i) \beta(i) [\gamma_X(i) - \alpha(i) \gamma_Y(i) - \beta(i) \gamma_X(i) + \gamma_X(i) = 0 \tag{15}
\]

\[- \alpha(i) \gamma(i) E[Y(i)] X(i) + \alpha(i) \beta(i) + \gamma_X(i) \] 
\[+ \alpha(i) \beta(i) E[X(i)] X(i) + \alpha(i) \] 
\[- \beta(i) E[Y(i)] X(i) + \beta(i) E[X(i)] X(i) \] 
\[+ \beta(i) E[X(i)] Y(i) + E[X(i)] Y(i) = 0 \]

Replace in (15) \(\alpha(i)\) by its expression in (14). We 
obtain the equivalent system (16) (17) where 
equation (17) only depends on variables \(\beta(i)\) and 
\(\beta(i)\): (16)

\[
\alpha(i) \beta(i) [\gamma_X(i) - \alpha(i) \gamma_Y(i) - \beta(i) \gamma_X(i) + \gamma_X(i) = 0 \tag{17}
\]

\[
A(i) + B(i) \beta(i) + C(i) \beta(i) + D(i) \beta(i)^2 + E(i) \beta(i) + F(i) \beta(i)^2 = 0 \tag{20}
\]

where complex coefficients \(A(i), ... , F(i)\) are linear 
combinations of second and third-order moments of 
signals \(X_m(i)\) and \(Y_m(i)\):

\[
\alpha(i) = \gamma_X(i) \cdot E[Y(i) X(i)] X(i) + \gamma_Y(i) \cdot E[X(i) Y(i)] Y(i) \tag{21}
\]

\[
B(i) = \gamma_Y(i) \cdot E[Y(i) X(i)] X(i) + \gamma_X(i) \cdot E[X(i) Y(i)] Y(i) \tag{22}
\]

\[
C(i) = \gamma_Y(i) \cdot E[Y(i)] X(i) + \gamma_X(i) \cdot E[X(i)] Y(i) \tag{23}
\]

\[
D(i) = \gamma_X(i) \cdot E[Y(i)] X(i) + \gamma_X(i) \cdot E[X(i)] Y(i) \tag{24}
\]

Equation (17) may be factorized in (19):

\[
(\beta(i) - a_1(i))(\beta(i) - a_2(i))(\beta(i) - a_3(i)) = 0 \tag{19}
\]

where \(a_1(i), a_2(i)\) and \(a_3(i)\) are the three complex 
eigenvalues of equation (17). 
By identifying equations (17) and (19), we obtain 
the value of the first root \(a_1(i)\).

\[
a_1(i) = \frac{F(i)}{E(i)} \tag{20}
\]

Value of \(a_1(i)\) may be computed by replacing \(X_m(i)\) and 
\(Y_m(i)\) in function of \(A(i), N_s(i), R_m(i), f(i)\) 
and \(g(i)\). We also use hypothesis of independence 
and zero-mean of signals \(N_s(i)\) and \(R_m(i)\).

We then remark that root \(a_3(i)\) is of no interest 
because it depends on both values \(g(i)\) and \(f(i)\). 
On the contrary we will prove that one of the two 
other roots is exactly equal to \(\beta(0) = 1/f(0)\).

By identifying (17) and (19), we also remark that 
roots \(a_2(i)\) and \(a_3(i)\) may be computed by solving 
an equation of second order in \(\beta(i)\) (21):

\[
(21) \quad F(i)\beta(i)^2 + D(i)\beta(i) + E(i) = 0 \tag{21}
\]

where coefficients \(F(i), D(i)\) and \(E(i)\) are those of 
equations (18) and only depend on observations 
\(X_m(i)\) and \(Y_m(i)\). By replacing \(X_m(i)\) and \(Y_m(i)\) 
by their expressions in function of \(A(i), N_s(i), 
R_m(i), f(i)\) and \(g(i)\), we may prove that the two 
roots are (22):

\[
a_2(i) = [A(i)g(i) - \gamma_Y(i) N_s(i)] + A(i) f(i) \gamma_Y(i) \tag{22}
\]

\[
+ (g(i) - \gamma_Y(i) N_s(i) N_s(i)) (1) \tag{23}
\]

\[
\frac{1}{\gamma(i)} \gamma(i) N_s(i) (1) \tag{24}
\]

\[
\frac{1}{\gamma(i) N_s(i) (1)} \tag{25}
\]

By replacing previous roots in (16), we obtain 
two couples of solutions for \(\alpha(i)\) and \(\beta(i)\) 
\(\alpha_2(i) = a_2(i)\) and \(\alpha_3(i) = a_3(i)\). The two couples 
of solutions verify system (11),(13) but only the 
second one is able to separate sources. We 
distinguish the two solutions by adding an other 
constraint which is (24): \(E[X(i)] = 0\)

The unique solution of the new system of 
equations (11), (13) and (24) is the researched 
solution:

\[
\alpha(i) = 1/g(0) \tag{26}
\]

Consequently, values of the two complex vectors 
\(f(i)\) and \(g(i)\) may be easily computed by solving 
the equation of second order in \(\beta(i)\) (21).

In practice the coefficients \(B(i), D(i)\) and \(F(i)\) 
of equation (21) may be computed by averages on 
data blocks.

We treated the case where signal \(S(k)\) contains one 
sinusoid. The problem is similar with regards to 
both sources. So if reference \(R(k)\) contains one 
sinusoid, equation (21) is similar but one of the 
roots will be equal to \(1/f(0)\) instead of \(1/f(0)\).

If one source contain several sinusoids of different 
frequencies, the problem will be equivalent at 
previous case and will be treated independently 
on each frequency bin.

3.2 Case of random signals with non zero third-
order moments

We suppose now that \(S(k)\) and \(R(k)\) are random 
noises. In several cases one source, \(S_m(0)\) or \(R_m(0)\), 
may have non-zero third-order moments 
although p-point discrete Fourier transforms
$S_m(i)$ or $R_m(i)$ are generally supposed to lead to Gaussian complex variables if $p$ is large enough (central limit theorem see [8]). Consider for example $S(k)$ as the sum of sinusoid periods modulated by a random variable $A$ with non even probability density and zero-mean:

at data block $m$, we have $S(k) = A(m) \exp(2pkki/p)$

Consequently: $S_m(i) = p_A(m)$

If $A(m)$ is a random variable with non even probability density, non-zero third-order moment and zero-mean, so is $S_m(i)$. This kind of signals is often used in communication problems. Remark that they cannot be separated by existing methods such that [4] [5] which attempt to cancel expressions like $E[i(k)Y_m(i)]g(Y_m(i))$ with $f$ and $g$ odd: these methods necessary need the assumption that signals have even probability densities. Others methods using informations at fourth-order (moments or cumulants) are able to separate them [7] [9] but are much more complicated than the following proposed method.

As in part 3.1, we search $\alpha(i)$ and $\beta(i)$ such that the two linear combinations $[X_m(i) \cdot \alpha(i)Y_m(i)]$ and $[Y_m(i) \cdot \beta(i)X_m(i)]$ are uncorrelated and independent. Consequently $\alpha(i)$ and $\beta(i)$ are non unique solutions of (11) and (13).

$$E[X_m(i) \cdot \alpha(i)Y_m(i)] = 0$$

$$E[Y_m(i) \cdot \beta(i)X_m(i)] = 0$$

They are also solutions of (16) and (17):

$$\alpha(i)\beta(i)Y_{XY}(i) - \alpha(i)\beta(i) + \alpha(i) \beta(i) = 0$$

$$A(i) = \beta(i) + C(i)\beta^2(i) + D(i)\beta^2(i) + E(i)\beta^4(i)$$

where the complex coefficients $A(i)$, ..., $F(i)$ are developed in (18) (part 3.1). By replacing $X$ and $Y$ in function of $S_m(i)$ or $R_m(i)$ we remark that roots of (21) are equal to $\gamma(i)$ and $1/\gamma(i)$.

$$F(i) = \beta^2(i) + D(i)\beta^2(i) + B(i) = 0$$

Coefficients $F(i)$, $D(i)$ and $B(i)$ are detailed in (18).

4 Application

We validate proposed methods with following simulations. We show on figure 3 the evolution of the estimation of real and imaginary parts of complex value $f(i)$ in function of data blocks $m$. Blocks own 64 samples and are 50 per cent overlapped. Signal contains a sinusoid in frequency bin $i$, $\bar{N}(k)$ and $R(k)$ are white gaussian noises. We remark that filter converges to right values (0.35; 0.6) after 2000 samples. Similarly, we see on figure 4 the evolution of values $f(i)$ and $g(i)$. Signal and reference are white gaussian noises. We save positive values of real parts and center signals. We remark that filters converge to right values (1;0.7) and (0.2) after 3000 samples.

5 Conclusion

We study noise cancelling applications when observed reference contains a part of signal. The proposed method needs non-zero third-order moments of sources, using hypothesis of statistical independence. It is easy to apply and finally consists in solving a second order equation in frequency-domain. Applications are reduction of rotating machines noise and separation of random signals with non zero third-order moments (communication problems).

References

ON THE USE OF INSTANTANEOUS PHASE VARIANCE FOR CLUTTER SUPPRESSION

Antonio Albiol (†), Luis Vergara (‡)

(†) Departamento de Comunicaciones, ETSI Telecomunicación, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, SPAIN.

(‡) ETSI Telecomunicación, Universidad Politécnica de Madrid, Ciudad Universitaria s/n, 28040 Madrid, SPAIN.

1. INTRODUCTION

This paper presents a method to suppress the clutter corrupting a desired signal. In radar this is mainly done by considering that the desired target has a relatively high speed and can thus be detected by Doppler processing. However in the case of stationary targets this method can no longer be used. For example in ultrasounds, the same problem arises when a flaw or defect enclosed by a strongly scattering medium such as grainy metal, has to be detected. In this case Doppler processing can not be used because the desired target (flaw) has no motion.

The signal to clutter ratio can be improved by compounding statistically independent echoes from the same region of the medium. These independent echoes can be obtained via frequency diversity [1], [2], or spatial diversity scanning the transducer. Usually only envelope information of the involved signals is considered.

In this paper we present a new processor to improve the detection of flaws using ultrasounds which makes use of both the envelope and the instantaneous frequency information of the data record.

The stationary clutter problem is very important in non-destructive evaluation of materials. To prove the validity of the method, it has been applied to real ultrasonic signals.

2. SIGNAL MODEL

2.1 Grain noise

In [4] the grain clutter has been modelled to be gaussian. This is based on the central limit theorem and the fact that in the resolution cell there is a very high number of scattering centres (grains). In [3], this model is revised for the cases where it is not possible to consider such a high number of scattering centres in the resolution cell, proposing a K-distribution for the envelope in those cases. In this paper gaussian statistics for the clutter are to be considered. This implies a Rayleigh distribution for the envelope and a uniform distribution for the instantaneous phase as well as independence between envelope and phase for the clutter statistics. The method would be applicable also to K-distributed clutter although a detailed mathematical justification would be much more complicated.

The transmitted waveform is normally of the form

\[ s(t) = p(t) \cdot e^{j\omega_0 t} \]  (1)

where \( p(t) \) is a short duration envelope, and \( \omega_0 \) is the carrier frequency. The echo received from the grainy structure of the material is the convolution of the reflectivity function with the transmitted pulse. If we assume that the grains are much smaller than the pulse duration and the beamwidth then the reflectivity can be considered a white random process. The received power spectral density of the grain noise is then

\[ \Phi_s(\omega) = K \cdot |P(\omega - \omega_0)|^2 \]  (2)

So, when only grain noise is present, the received signal can be considered a gaussian band-pass random process which has a power spectral density proportional to the square of the Fourier transform of the transmitted pulse.

2.2 Signal present

When a flaw or concentrated scatterer produces an echo, the received signal is

\[ \alpha \cdot p(t-t_0) \cdot e^{j\omega_0(t-t_0)} + n(t) \]  (3)

where \( \alpha \) is the amplitude associated with the flaw echo, \( t_0 \) is the delay related to its range, and \( n(t) \) is the grain noise. We can see that locally, when a flaw is present, we have a sinusoid added to the grain noise. It is interesting to notice that a matched filter is of no use to improve the SNR in this case since the desired signal and noise share the same frequency band.
In the case of a target being present the pdf of the envelope is Rician, and the phase pdf is [5]:

\[
f_\psi(\psi) = \frac{1}{2\pi} \exp(-u) + \sqrt{\frac{u}{\pi}} \cos \psi \exp(-u \sin^2 \psi) \left[ 1 - \frac{1}{2} \text{erfc}(u \cos \psi) \right]
\]

with

\[u = \frac{\alpha^2}{2\sigma^2}
\]

where \(\alpha\) is the amplitude of the target echo and \(\sigma^2\) is the variance of the in-phase and quadrature clutter components.

Just as an initial justification of the method, we show in the next figure the variation of the envelope mean as well as the inverse of the phase variance as functions of the SNR \((s = \alpha/\sigma)\). The variations are relative to the corresponding values when no desired signal is present.

![Fig. 1-a: Variation of envelope mean with s, relative to its value when no signal is present \((s = 0)\).](image)

As can be seen from the curves in fig. 1 the inverse of the phase variance increases with s faster than the mean of the envelope. This is specially interesting for very low signal to clutter ratios and is the basic idea to be exploited in the proposed processor. However, care must be taken with the above curves, since the variance of the instantaneous phase is a quadratic parameter, and the envelope mean is a linear one. If we take this into account, we can observe that the information contained in the phase is at least comparable to that present in the envelope. What we propose in

![Fig. 2: Variation of inverse of phase difference variance with s, relative to its value when no signal is present \((s = 0)\).](image)

We can also see that the variance of the i.f. is a parameter which can be used to detect flaws in grain noise.
3. PROPOSED PROCESSOR

First of all the signal is band-pass filtered to reject unwanted out-of-band components, which may be due to thermal noise or very low frequency fluctuations. Processing of the signals begins with the computation of the analytic signal. This is done using a FIR filter of length 51. From the analytic signal (a complex signal), we obtain the envelope as the magnitude and the phase as the angle. The phase obtained from the analytic signal (in the interval \([-\pi, \pi]\)), is then unwrapped using an algorithm which removes the \(2\pi\) jumps in the phase. The i.f. is obtained by first-differencing the unwrapped phase and dividing by \(2\pi\). The mean of the i.f., \(f_0\), is obtained based on all the register length. Finally the squared error with respect to the mean is obtained. The process is summarized in figure 3.

\[ x[n] \xrightarrow{\text{Anl. Signal}} \xrightarrow{\text{Inst. Freq.}} x^2 \rightarrow \hat{f}_0 \]

*Fig. 3: Summary of processing of the received echo.*

The underlying idea of the processor is to exploit the fact that when a flaw is present, locally, the envelope mean increases and also the i.f. variance decreases.

These can be obtained from within the same record, since the variance will be low during all the pulse duration. But since the pulse amplitude varies, so will the variance vary. Also the i.f. is not a white signal, so the samples involved in the estimation will not be independent. This all leads to the conclusion that the variance estimate obtained by means of averaging the squared i.f. error in a window, is a biased estimate of the actual variance. Nevertheless it can still be useful. The window length is the only parameter of this method and the optimum value depends on the signal bandwidth of the RF signal, the envelope shape and the sampling frequency; we have found that a number of samples corresponding to approximately half duration of the transmitted pulse (approximately gaussian-shaped envelopes were used for the transmitted pulse) is a good choice. The output signal is the inverse of the local variance estimate; if we want to take into account both, the envelope and i.f. information, we can form the quotient between envelope and local variance; in this way both the envelope and phase information carried by the received signal are exploited by the algorithm.

Another way for getting the different required samples to estimate the i.f. variance is from different signal records (we will call them scans). When ultrasonic non-destructive evaluation is being carried out the transducer is scanned along a certain line. Due to the finite beamwidth a certain point in the material is illuminated from different locations getting independent echoes in each scan. If we have different scans, we can on one hand average the envelope and on the other hand estimate the variance by averaging the squared i.f. error in different scans. However it has been seen that the use of an ordered statistics filter improves the probability of detection with respect to the use of the mean squared error. The best statistic has been found to be the maximum squared error.

4. RESULTS

This section shows some of the results obtained with the proposed method applied to real ultrasonic signals. The material under test was a block of austenic steel. The defect to detect was a cut in the backside of the block under test. A 45 degrees incidence angle was used to avoid the backside specular echo. The central frequency transmitted was about 1 MHz.; the 6 dB bandwidth was 330 Khz., And the propagation velocity was 5960 m/s. The sampling frequency was 6.25 Mhz.

*Fig. 4-a: Average of the envelope of nine consecutive scans, with and without defect.*

The functional of the i.f. variance considered for figures 4 has been the mean squared frequency error. From figures 4, the detection possibilities present in the phase information become apparent and are at least as good as those of the envelope. Notice also, that if both informations, envelope and phase are somehow combined (dividing the envelope by the variance estimate for instance), the SNR can be further improved.
**5. CONCLUSIONS**

A new method for the detection of signals in clutter, which makes use of the instantaneous phase information of the received signals has been developed. The method has been successfully applied to ultrasonic B-scans signals for non-destructive testing. The method can be of interest in other related areas like medical echo-diagnostic, radar or sonar.

**REFERENCES**


Study of Generalized Linear Interpolation Approximation for Multi-Dimensional Waveforms

Takuro KIDA
Hiroshi MOCHIZUKI
Department of Information Processing, Graduate School at Nagatsuka
Tokyo Institute of Technology
Nagatsuka 4259, Midori-ku, Yokohama-shi, 227 Japan
Tel.+81-45-922-1111 ext.2522 Telefax.+81-45-921-1156

Abstract

In this paper, we consider the optimum approximation of band-limited multi-dimensional waveforms using samples of outputs of linear shift-invariant filters. We use a set of waveforms such that all the weighted integrals of the corresponding Fourier spectrums in the prescribed sub-bands are bounded. We establish a theorem showing that, among these waveforms, there exists a waveform with respect to which the sample values actually contributing to the interpolation are all zero and the absolute value of the corresponding approximation error reaches the upper limit.

As a direct consequence, we show that the presented linear approximation is superior to all other approximations using the same measure of error and sample values.

1 Introduction

Multi-dimensional signal processing is one of the important areas in digital signal processing. In the literature [1], we have proposed a certain optimum approximation for one-dimensional waveforms. When we apply the result to multi-dimensional waveforms, we may misunderstand that we can deal with the problem in the same way as the case of one-dimensional waveforms, since a multi-dimensional waveform can be expressed simply by the product of one-dimensional waveforms. However, let us consider the case of two-dimensional waveforms. As the arrangements of the sampling points, we can choose orthogonal lattice, hexagonal lattice and octagonal lattice, etc. Even if we use the orthogonal lattice, the supports of the Fourier spectrums of the waveforms and/or those of the interpolation functions are not always separable with respect to the variables. Hence, if we can transform the hexagonal lattice or the octagonal lattice into the orthogonal lattice by using a suitable transformation of coordinates, in this case we cannot use the result of the one-dimensional case. This is one of the reasons why we present a unified treatment of multi-dimensional interpolation approximation in this paper.

2 Some definitions and notations

We begin with a summary of symbols and notations used in this paper.

(a) \( X = (x_1, x_2, \cdots, x_n) \) indicates a \( n \)-dimensional real vector.
(b) \( R^n = \{ X \mid \text{all the elements are real} \} \)
(c) \( \mathbb{Z}^n = \{ X \mid \text{all the elements are integer} \} \)
(d) \( X_X = x_{k_1, k_2, \cdots, k_n} \) means a vector with the integer subscripts \( k_m \) \((m = 1 \sim n)\). Also, \( X_{KP} = x_{k_1, k_2, \cdots, a_m, a_1, \cdots, a_p} \)
where \( \mathbf{K} = (k_1, k_2, \cdots, k_n) \) and \( \mathbf{P} = (p_1, p_2, \cdots, p_p) \).
(e) \( \Sigma_{k_m \in \mathbb{Z}^n} \{ \cdot \} = \Sigma_{k_1} \Sigma_{k_2} \cdots \Sigma_{k_n} \{ \cdot \} \) indicates the summation over all the vectors in the subset \( J \) of \( \mathbb{Z}^n \).
(f) \( x + y = (x_1, y_1, x_2, y_2, \cdots, x_n, y_n) \) where \( x = (x_1, x_2, \cdots, x_n) \) and \( y = (y_1, y_2, \cdots, y_n) \).
(g) The transpose of a vector or a matrix \( Y \) is expressed as \( Y^T \). The complex conjugate of \( Y \) is expressed as \( Y^* \).
(h) \( f(X) = f(x_1, x_2, \cdots, x_n) \)
(i) \( f(X) \, dX = f(x_1, x_2, \cdots, x_n) \, dx_1 \, dx_2 \cdots dx_n \)
(j) \( D^*(\theta) = \{ f(X) \, f(X)^* \mid f(X)^* \, dX < \infty \} \) where \( \theta \) is a subset of \( R^n \).

3 Sample points and approximation formula

3.1 Set of sample points

We consider the prescribed vector \( N = (N_1, N_2, \cdots, N_n) \) whose elements \( N_i \) are integers. Further, let \( \Lambda(N) \) be the set of all the vectors in \( R^n \) whose elements \( k_i \) \((i = 1 \sim n)\) satisfy \( 1 \leq k_i \leq N_i \) \((i = 1 \sim n)\).

In the following, we assume that the symbol \( P \) always means a vector in \( R^n \). We define also that \( K \) always indicates a vector in \( \Lambda(N) \). Further, let

(a) \( H = \{ \mathbf{h} \mid \text{a given } n \times n \text{ real regular matrix} \} \)
(b) \( T = (T_1, T_2, \cdots, T_n) \) be the prescribed vector with positive elements \( T_m \) \((m = 1 \sim n)\).

Now, we consider the following set of linear equations.

\[
X_{KP} \cdot H - P \cdot T - A_k = 0 \quad (X_{KP} \in R^n)
\]  

In eq.

(1), \( X_{KP} \) is a solution vector, \( K \) and \( P \) are the subscript vectors. \( A_k \) is the prescribed vector having the form of \( A_k = (A_k^{(1)}, A_k^{(2)}, \cdots, A_k^{(m)}) \) and satisfies \( 0 \leq A_k^{(m)} < A_k^{(m)} < T_m \) \((1 \leq p < q < n; m = 1 \sim n)\).

Changing \( K \) and \( P \) one by one over all the elements of \( \Lambda(N) \) and \( R^n \), we obtain the infinite set of the \( n \)-dimensional polyhedra \( P_k \) \((k = 1, 2, \cdots)\) whose vertices correspond to the \( n \)-dimensional points \( X_{KP} \). We assume that \( C \) and \( P \) \((i \neq j) \) do not
have common vectors except their boundaries. Further, we denote by \( \Lambda \) the prescribed subset of \( \Lambda(N) \).

In this paper, we define the set of n-D sample points as the set of vertices \( X_{KP} \) \( (K \in \Lambda, \; P \in K^p) \). Selecting appropriate \( \Lambda, \, H \) and \( \Delta \), we can realize many types of arrangements of n-D sample points such as hexagonal or octagonal lattices. Letting \( H^{-1} = G^u \) in eq. (1), we get the following equation.

\[
X_{KP} = (P \ast T + A_h)G^u
\]

(2)

### 3.2 Set of waveforms

Let \( \Theta \) be a simply-connected closed domain of \( R^n \) expressed as \( \Theta = \Theta_1 \cup \Theta_2 \) \( (\Theta_1 \cap \Theta_2 = 0) \). We define positive bounded functions \( W_1(U) \) and \( W_2(U) \) as follows.

\[
W_1(U) \in L^p(\Theta_1) \quad (U \in \Theta_1) \quad (3)
\]

\[
W_1(U) = 0 \quad (U \notin \Theta_1) \quad (4)
\]

\[
W_2(U) \in L^p(\Theta_2) \quad (U \in \Theta_2) \quad (5)
\]

\[
W_2(U) = 0 \quad (U \notin \Theta_2) \quad (6)
\]

Moreover, we consider \( R_{00}(U) \) and \( R_{01}(U) \) satisfying the following conditions, where \( A_1 \) and \( A_2 \) are prescribed positive numbers.

\[
\int_{\Theta_1} \left| \frac{R_{00}(U)}{W_1(U)} \right|^p \, dU \leq A_1 \quad (7)
\]

\[
\int_{\Theta_2} \left| \frac{R_{01}(U)}{W_2(U)} \right|^p \, dU \leq A_2 \quad (8)
\]

Since \( W_1(U) \) and \( W_2(U) \) are bounded, \( F_0(U) \) and \( F_1(U) \) apparently belong to \( L^p(\Theta_1) \) and \( L^p(\Theta_2) \), respectively, where \( F_0(U) \) and \( F_1(U) \) are band-limited in \( U \in \Theta_1 \) and \( U \in \Theta_2 \), respectively.

We define \( f_0(\mathbf{X}) \) and \( f_0(\mathbf{X}) \) as the inverse Fourier transforms of \( F_0(U) \) and \( F_1(U) \), respectively. Further, we denote by \( f(\mathbf{X}) = f(x_1, x_2, \ldots, x_n) \) the sum of \( f_0(\mathbf{X}) \) and \( f_0(\mathbf{X}) \). Letting \( A_1 \gg A_2 \gg 0 \) in eqs. (7) and (8), for example, \( f(\mathbf{X}) \) corresponds to ordinary low-pass type signals. Let

\[
f(\mathbf{X}) = \frac{1}{(2\pi)^n} \int_{\Theta} F(U) e^{iU \cdot X} \, dU \quad (9)
\]

We abbreviate the relation of eq. (9) as \( f(\mathbf{X}) \rightarrow F(U) \). The set of these \( f(\mathbf{X}) \) is written as \( V(\Theta) \).

### 3.3 Approximation formula

We define the simply-connected bounded regions \( B_{KP} \( (K \in \Lambda ; \; P \in K^p) \) satisfying (a) and (b).

(a) \( B_{KP} \) contains \( X_{KP} \).

(b) For arbitrary vectors \( P \) and \( Q \) \( (P \neq Q) \) in \( K^p \), the region \( B_{KP} \) is identical to the parallel translation of \( B_{KP} \) by \( X_{KP} \).

For each \( B_{KP} \) \( (K \in \Lambda ; \; P \in K^p) \), we consider a bounded function \( \psi_{KP}(X) \) \( (K \in \Lambda ; \; P \in K^p) \) satisfying \( \psi_{KP}(X) = 0 \) \( (X \notin \Lambda_{KP}) \). Further, we consider the approximation formula of \( f(\mathbf{X}) \) belonging to \( V(\Theta) \) as follows.

\[
g(X) = \sum_{K \in \Lambda} \sum_{P \in K^p} f(X_{KP}) \psi_{KP}(X) \quad (10)
\]

In eq. (10), we call \( \psi_{KP}(X) \) the interpolation functions. Afterwards prove that these interpolation functions can be realized as the impulse responses of the finite number of linear shift invariant filters.

### 4 Integral expression of the measure of approximation error

We define the approximation error \( \epsilon(\mathbf{X}) \) between \( f(\mathbf{X}) \) and \( g(X) \) as follows.

\[
\epsilon(\mathbf{X}) = \| f(\mathbf{X}) - g(X) \| \quad (X \in \Theta) \quad (11)
\]

The definition of \( \Theta \) is given in Sec. 5.1. Make \( g(X) \) by eq. (10) and change \( f(\mathbf{X}) \) in \( V(\Theta) \) under the condition that the interpolation functions \( \psi_{KP}(X) \) are fixed. Moreover, consider an envelope \( \epsilon_{max}(X) \) of the corresponding approximation errors \( \epsilon(\mathbf{X}) \) as follows.

\[
\epsilon_{max}(\mathbf{X}) = \sup_{X \in \Theta} \{ \epsilon(\mathbf{X}) \} \quad (X \in \Theta) \quad (12)
\]

From now on, we adopt \( \epsilon_{max}(X) \) as the measure of approximation errors. The interpolation functions \( \psi_{KP}(X) \) minimizing \( \epsilon_{max}(X) \) are called the optimum interpolation functions. Moreover, at an arbitrarily specified point \( X = Y \), if \( f(\mathbf{X}) \) exists in \( V(\Theta) \) with respect to which \( \epsilon(\mathbf{Y}) = \epsilon_{max}(Y) \) is valid, we write one of these \( f(\mathbf{X}) \) as \( f(X; Y) \) and call it the extremal waveform (for the fixed point \( Y \)).

By using the well known H"older's inequality,

\[
e(\mathbf{X}) = \frac{1}{(2\pi)^n} \int_{\Theta} F_0(U) \Xi(U; \mathbf{X}) \, dU + \int_{\Theta} F_0(U) \Xi(U; \mathbf{X}) \, dU \\
\leq \frac{1}{(2\pi)^n} \left[ \int_{\Theta} |F_0(U)|^p \, dU \right]^{1/p} \left[ \int_{\Theta} |\Xi(U; X)|^q \, dU \right]^{1/q} \\
+ \frac{1}{(2\pi)^n} \left[ \int_{\Theta} |F_0(U)|^p \, dU \right]^{1/p} \left[ \int_{\Theta} |\Xi(U; X)|^q \, dU \right]^{1/q} \\
\leq \frac{A_{11}^{1/p}}{(2\pi)^n} \left[ \int_{\Theta} W_1(U)|\Xi(U; X)|^q \, dU \right]^{1/q} \\
+ \frac{A_{11}^{1/p}}{(2\pi)^n} \left[ \int_{\Theta} W_2(U)|\Xi(U; X)|^q \, dU \right]^{1/q} \\
(1/p + 1/q = 1 ; \; X \in \Theta) \quad (13)
\]

holds, where

\[
\Xi(U; \mathbf{X}) = e^{iU \cdot X} - \sum_{K \in \Lambda} \sum_{P \in K^p} \psi_{KP}(X) H_{KP} e^{iU \cdot KP} \quad (14)
\]

The conditions that the inequalities in ineq. (13) hold at \( X = Y \) are expressed as

\[
\frac{F_0(U; Y)}{W_0(U)} = \frac{\sigma_{0}(Y) W_0(Y)^{-1}}{\Xi(U; Y) \cdot \Xi(U; X)^{-1}} \quad (U \in \Theta ; \; Y \in \Theta ; \; \mu = 1, 2) \quad (15)
\]

\[
\int_{\Theta} \left| \frac{F_0(U; Y)}{W_0(U)} \right|^p \, dU = A_{\mu} \quad (Y \in \Theta ; \; \mu = 1, 2) \quad (16)
\]
where \( \alpha_r(Y) \) is the scaling function which independent of \( U \), and is chosen so that \( R_{\mu}(U; Y) \) satisfies eq. (16), that is
\[
\alpha_r(Y) = \frac{A_{1/\mu}^{1/\mu}}{\left[ \int_{\mathcal{J}_0} W_{\mu}(U) ||\mathbb{E}(U; Y)||^\mu dU \right]^{1/\mu}} \\
(\mu = 1, 2)
\] (17)

The Fourier spectrum of the optimal waveform \( f(X; Y) \) is given by \( R_{\mu}(U; Y) = F_{\mu}(U; Y) \) in eqs. (15) and (16). Hence, by a similar consideration to the case of one-dimensional waveforms [1], we obtain the following theorem.

**Theorem 1**

\[
\epsilon_{\text{max}}(X) = \frac{A_{1/\mu}^{1/\mu}}{(2\pi)^n} \left[ \int_{\mathcal{J}_0} \left| W_{\mu}(U) \right|^n ||\mathbb{E}(U; X)||^n dU \right]^{1/n}
\] (18)

We call eq. (18) the integral expression of \( \epsilon_{\text{max}}(X) \). Although the detail discussion is omitted, \( \epsilon_{\text{max}}(X) \) is convex for all the \( \psi_{K,R}(X) \). Hence, using an ordinary numerical optimization procedure, we easily obtain the "global" optimum \( \psi_{K,R}(X) \) at each \( X \).

5 Optimality of the proposed approximation

5.1 Interpolation functions minimizing \( \epsilon_{\text{max}}(X) \)

Letting \( Q \in \mathbb{Z}^n \), we get the following equation of arbitrary \( K \in \Lambda \) and \( P \in \mathbb{Z}^n \):

\[
X_{K,P+Q} - X_{K,R} = (Q + T)G^i
\] (19)

As shown in eq. (19), the difference between \( X_{K,P+Q} \) and \( X_{K,R} \) is independent of \( P \) when \( T \) and \( G \) are fixed. Therefore the pattern of arrangement of \( X_{K,P+Q} \) is the parallel translation of that of \( X_{K,R} \). By using the periodicity for the arrangement of the sample points, the following equation holds for arbitrary \( K \in \Lambda \) and \( P \) and \( Q \in \mathbb{Z}^n \):

\[
\epsilon^{ij} K_{P+Q} (Q + T)G^i = \epsilon^{ij} X_{K,R+Q}
\] (20)

Transform \( \epsilon_{\text{max}}(X) \) of eq. (18) by using eq. (20) and perform a similar discussion as the one-dimensional case [1]. Then, we can obtain the following theorem.

**Theorem 2** Let the interpolation functions minimizing the measure of error \( \epsilon_{\text{max}}(X) \) be \( \psi_{K,R}(X) \) \( (K \in \Lambda; P \in \mathbb{Z}^n) \), then

\[
\psi_{K,P+Q}(X) = \psi_{K,R}(X - (Q + T)G^i) \quad (K \in \Lambda; P \in \mathbb{Z}^n)
\] (21)

holds. Thus, putting

\[
\psi_{K,Q}(X) = \psi_{K,R}(X) = \psi_{K}(X - A_k G^i)
\] (22)

we get the following equation.

\[
\psi_{K,P}(X) = \psi_{K}(X - (P + T)G^i - A_k G^i)
\] (23)

where \( \theta \) is a zero vector.

(End of the theorem)

Theorem 2 shows that the interpolation functions minimizing \( \epsilon_{\text{max}}(X) \) are expressed as the finite number of parallel transformations of functions. Thus, the optimum interpolation functions can be realized as the impulse responses of the finite number of linear shift invariant filters.

Now we discuss the set of sample points actually contributing to the approximation. Firstly we consider the coordinate vectors corresponding vertices of a n-D unit cube.

\[
\begin{align*}
1_1 &= (0, 0, \ldots, 0) = \theta \\
1_2 &= (1, 0, \ldots, 0) \\
1_3 &= (0, 1, \ldots, 0) \\
\vdots \\
1_d &= (1, 1, \ldots, 1) 
\end{align*}
\] (24)

Moreover, letting \( P \) be an arbitrary vector in \( \mathbb{R}^n \), we define \( \tilde{P} = P + 1_i \) \( (i = 1 \sim d) \). Further, by fixing \( K = \theta \) and changing \( P \) with \( P_1, P_2, \ldots, P_d \) in \( X_{K,P} \), we get the lattice points \( X_{1}, X_{2}, \ldots, X_{d} \) \( (d = 2^n) \). A parallel polyhedron having these lattice points as its vertices is written as \( \Delta \). Using the periodicity of the arrangement of the sample points, we can prove that the space \( \mathbb{R}^n \) is filled with \( \Delta \), without the overlaps except their boundaries.

Now, we notice the above described regions \( B_{K,R} \) and write the boundaries of \( B_{K,R} \) as \( C_{K,R} \). Moreover, changing \( K \) and \( P \) within \( \Lambda \) and \( 2^n \), respectively, \( \mathbb{R}^n \) is divided into disjoint regions by \( C_{K,R} \) without overlaps except boundaries. We denote these regions by \( W_i \) \( (i = 1, 2, \ldots) \).

Eq. (19) shows that \( X_{K,P+Q} \) is a parallel translation of \( X_{K,R} \) by \( (Q + T)G^i \). Therefore, by using the argument of sec.3.1, \( B_{K,R+Q} \) is a parallel translation of \( B_{K,R} \) by \( X' \). Thus, apparently the boundary \( C_{K,R+Q} \) is a parallel translation of \( C_{K,R} \) by \( X' \).

This fact shows that a parallel translation of the region \( W_i \) by \( X' \) is also a congruent region (we call it \( W_j \)). Since \( Q \) in \( X' = (Q + T)G^i \) is an arbitrary vector belonging to \( \mathbb{Z}^n \), the set of \( W_i \) is classified into the groups where the regions are the parallel translations one another. If \( W_i \) is the parallel translation of \( W_j \), we write this fact as \( W_i \sim W_j \). Let \( W_p = \{ W_{1}, W_{2}, \ldots, W_{n} \} \) be the set of \( W_i \) having a common point with \( \Delta \), except boundaries (by changing the subscripts appropriately). Moreover, when \( W_p \) contains the regions such that \( W_i \sim W_j \) \( (i \neq j) \), we leave in \( W_p \) only one of them. The new \( W_p \) obtained after this operation is written by changing subscripts appropriately, \( W_p = \{ W_{1}, W_{2}, \ldots, W_{n} \} \).

Further, we can easily prove that \( \Delta_{W_p} \) is a parallel translation of \( \Delta \) by \( X' = (Q + T)G^i \) and the space \( \mathbb{R}^n \) is filled with disjoint \( \Delta_{W_p} \) except common boundaries.

Now, at a certain point \( X \), and the fixed \( K \), we consider the set of \( \{(K, P)\} \) expressed as

\[
\Xi_{\mathcal{K}} = \{(K, P) | X \in W_i, W_i \in \Omega_p, W_i \subset B_{K,R}\}
\] (25)

It should be noted, in this case, the sample values \( f_K(X_{K,R}) \) \( \{(K, P) \in \Xi_{\mathcal{K}}\} \) actually contribute to the approximation.

5.2 Property of the optimum waveforms

As the important property of the optimum waveforms described in sec.4.1 the following theorem holds.
Theorem 3

\[ f_X(X;K,Y) = 0 \quad \{ (K, P) \in \Xi_p, \ Y \in W_i \} \quad (26) \]

(Proof) By the condition of the optimum interpolation functions, expanding eq.(18) with respect to \( \phi_{X,K}(x) \) and \( \psi_{X,K}(x) \) and taking the partial derivative with respect to the real part and the imaginary part of \( \phi_{X,K}(x) \) and letting them be equal to zero and combining the results, then we can get

\[
\frac{\partial \mu_{\max}(x)}{\partial \phi_{X,K}(x)} = B \cdot \left[ \frac{A_{1p}^{(p)}}{(2\pi)^n} \int_{\Theta} W_1(U) \right.
\]

\[
\left. \left\{ \{e^{i\mu_{X,K}(x)}Y - \sum_{\lambda \in \Phi_X} \psi_{\lambda}(x)H(\Omega) e^{i\mu_{X,K}(x)} \} \cdot H_K(U) \right\} dU \right\}^{-1/2} \]

\[
+ \frac{A_{1p}^{(p)}}{(2\pi)^n} \int_{\Theta} W_2(U) \]

\[
\left\{ \{e^{i\mu_{X,K}(x)}Y - \sum_{\lambda \in \Phi_X} \psi_{\lambda}(x)H(\Omega) e^{i\mu_{X,K}(x)} \} \cdot H_K(U) \right\} dU \right\}^{-1/2} \]

\[ = 0 \quad (27) \]

where \( B \) is a nonzero real number. On the other hand, from eq.(9) we get

\[
f_X(X;K,Y) = \frac{1}{(2\pi)^n} \int_{\Theta} F(\Omega)H(\Omega)e^{i\mu_{X,K}(x)}dU \]

\[ + \frac{1}{(2\pi)^n} \int_{\Theta} F(\Omega)H(\Omega)e^{i\mu_{X,K}(x)}dU \quad (28) \]

Substituting eq.(15) into eq.(28) and by using eq.(17), we get

\[
f_X(X;K,Y) = \frac{1}{(2\pi)^n} \int_{\Theta} W_1(U) \]

\[
\left\{ \{e^{i\mu_{X,K}(x)}Y - \sum_{\lambda \in \Phi_X} \psi_{\lambda}(x)H(\Omega) e^{i\mu_{X,K}(x)} \} \cdot H_K(U) \right\} dU \right\}^{-1/2} \]

\[
+ \frac{A_{1p}^{(p)}}{(2\pi)^n} \int_{\Theta} W_2(U) \]

\[
\left\{ \{e^{i\mu_{X,K}(x)}Y - \sum_{\lambda \in \Phi_X} \psi_{\lambda}(x)H(\Omega) e^{i\mu_{X,K}(x)} \} \cdot H_K(U) \right\} dU \right\}^{-1/2} \quad (29) \]

Comparing eq.(27) with eq.(28), we get the following equation.

\[ f_X(X;K,Y) = 0 \quad \{ (K, P) \in \Xi_p, \ Y \in W_i \} \quad (30) \]

(Q.E.D.)

Let \( R(X) = R(\cdots, f_X(X,K), \cdots; X) \) be another approximation formula and let

\[ \epsilon_{\max}^{(1)}(X) = \sup_{f \in \mathcal{F}} [f(X) - R(X)] \]

We assume \( R(\cdots, 0, \cdots; X) \equiv 0 \). Further, we assume that \( \epsilon_{\max}^{(1)}(Y) < \epsilon_{\max}(Y) \) holds at a fixed \( Y \). Then, if \( f(X) \) is equal to the optimal function \( f(Y) \), we obtain \( [f(Y)] = [f(Y) - R(Y)] \leq \epsilon_{\max}(Y) < \epsilon_{\max}^{(1)}(Y) = \epsilon(Y) = \epsilon(Y) - g(Y) \) which makes a contradiction. Hence, at all \( Y \), \( \epsilon_{\max}(Y) \geq \epsilon_{\max}^{(1)}(Y) \) holds. Namely, when we adopt other arbitrary approximation formulas using the same sample values, the upper bounds of approximation error rather increase. Thus, as long as the measure of error and the set of waveforms are not changed, the proposed approximation formula is the optimum among all other approximation formulas. Extending the discussion, similar arguments are possible for the case that Fourier spectrums are divided into more than two bands. In this paper, we have discussed when the weighted \( L_p \) integrals are bounded. In particular, when \( p = 2 \) we can still more extend the discussion mathematically [2]. However, when \( p \neq 2 \) we cannot extend the discussion any further, because an bilinear form can solely be defined by an integral expression in this case.

It should be noted, from theorem 2, the interpolation functions minimizing \( \epsilon_{\max}(X) \) are expressed as the parallel transformations of the finite number of functions. Thus, the interpolation functions can be realized as the impulse responses of the finite number of linear shift invariant filters.

6 Conclusion

Presented linear approximation for multi dimensional waveforms is superior to all other approximations using the same set of waveforms, sample values and measure of error. This shows that the superiority of nonlinear signal processing using, for example, neural devices mainly depends on the choice of the set of waveforms.

Acknowledgement

We would like to thank Professor Kenneth Jenkins of Illinois University and Professor A. Petrakis of Ruhr-Universitat Bochum and Mr. Somasak Sa-ngamokutchakon of Tokyo Institute of Technology for their useful advice in writing this paper.

References


DETAIL-PRESERVING FILTERS WITH IMPROVED LOWPASS CHARACTERISTICS

Irek DEFFE, Riku Soininen
and Yrjö Neuvonen

Signal Processing Laboratory,
Department of Electrical Engineering,
Tampere University of Technology,
P.O. Box 553, SF-33101 Tampere, Finland.

A new scheme for high-quality, detail-preserving and aliasing-free filtering is proposed. The scheme uses combination of a nonlinear detail-preserving filter and linear lowpass filter. An algorithm is devised for switching between the nonlinear and linear filter in order to eliminate aliasing errors and preserve details whenever possible.

1. INTRODUCTION

Increasing resolution and faithfulness of images processed and displayed by electronic means leads to a growing demand for high-quality image filtering. Ideal filters should robustly attenuate noise and contaminations, without removing details or introducing blurring or artefacts. Classical linear filters are effective in attenuating Gaussian noise and have well-defined frequency response. On the other hand, they suffer from blurring of edges and details, which is unacceptable in high-quality applications. Also, linear filters cannot be applied in the case of impulsive noise contamination.

To alleviate problems with linear filters, many nonlinear filter schemes based on different concepts were proposed [1]. Probably the most thoroughly investigated and used among them are filters based on the median operation. Median and median-type filters have simple implementation and can be designed to preserve image details [2]. They operate robustly in impulsive noise and are moderately effective in Gaussian noise. All these desirable features made median filters strong contenders for high-quality image processing applications.

When considering the high-quality demands, however, one should realize that the impact of frequency response of nonlinear filters on the quality of processing has not been given sufficient attention. The most prominent illustration of this problem is that inadequate attenuation of higher signal frequencies may lead to the generation of severe aliasing artefacts. The aliasing is described by shifting of high frequency spectra to the baseband. One can talk about aliasing-blurring dichotomy when investigating properties of linear and nonlinear filters. Linear filters can be easily designed to eliminate aliasing components, but at the price of blurring. Nonlinear filters may not exhibit blurring but their frequency behavior can not be shaped so effectively and as a result aliasing artefacts may appear. It can be thus seen that a fresh approach is required to find algorithms resistant to all the quality impairments.

In this paper, a new scheme is proposed to solve the aliasing-blurring dichotomy. The scheme is based on a combination of linear and nonlinear filters and a decision structure. The decision structure is designed in such a way that it switches between the linear and nonlinear filters depending on the presence of a signal component which could give rise to serious aliasing artefacts. In this way, filtering with the elimination of both blurring and aliasing is achieved.

2. COMBINED FILTERING SCHEME

2.1. Blurring and Aliasing Problem

The aliasing-blurring dichotomy is illustrated in Fig. 1. Linear filter causes blurring of sharp edges (Fig. 1b), but is able to eliminate effectively high frequencies (Fig. 1d). Median filtering, on the other hand, does not introduce blurring (Fig. 1b), but high frequencies are not sufficiently attenuated (Fig. 1d). The effect of insufficient attenuation of high frequencies by the median filtering can be seen in Fig. 2 where the original "Young Couple" image has been filtered by a multilevel median [2] filter operating in a 3 x 3 window and decimated by two. A very annoying aliasing artefacts can be seen in the shirt area (Fig. 2b). Such artefacts would not be created by properly designed linear lowpass filter, but at the cost of blurring.

![Fig. 1. Step edge (a) and a high-frequency signal (c), filtered by linear and 3-point median filters (b, d).](image-url)

From these observations, we can conclude that both linear and nonlinear (median in this case) filters have desirable features, but also some drawbacks which can not be tolerated in critical applications like HDTV, printing industry and multimedia. The goal of this paper is to introduce a new filtering structure which would be able to preserve...
only positives of both filter types, without bringing up their drawbacks. This consideration leads us to an idea of a combined structure with both linear and median filters operating in a switched mode, depending on a local signal shape. Such structure would satisfy the requirements of the high quality applications.

![Image](a) ![Image](b)

Fig. 2. "Young Couple" (a), filtered with a $3 \times 3$ multilevel median filter, and decimated by two, (b).

2.2. New Filtering Structure

Blurring and aliasing could be eliminated if lowpass characteristics of median-type filters were improved. To achieve this, we developed a new structure which combines operation of linear lowpass and nonlinear median-type detail-preserving filters, [2]. This structure is shown in Fig. 3.

![Diagram](a) ![Diagram](b)

Fig. 3. New filtering structure.

The system shown in Fig. 3 should operate in such a way that it would switch to linear filter in case of the presence of local signal segment which could cause aliasing, and it would switch to a nonlinear filter in other cases, to avoid blurring. The main problem obviously is to find an algorithm for switching between the linear and nonlinear filter. In [3], a scheme has been proposed in which the switching is based on the output of a linear bandpass filter tuned to the frequency at which median filter has lowest attenuation. However, the use of the linear filter for the switching decision may lead to problems since such filter does not provide sufficient information for the discrimination between the highly-localized and high-frequency signal segments. It is felt that another algorithm is needed to detect only those signal parts which are responsible for aliasing.

3. SWITCHING ALGORITHM

3.1. Basic Algorithm

In developing of the switching algorithm we use the fact that monotonic (e.g., edges), or slowly varying signal segments do not contribute to the creation of aliasing errors. Such errors are caused only by the signal segments having sufficiently fast variation. The local signal variation can be described by the presence of local signal maxima and minima within certain specific distance. Assume for example that decimation by two is going to be performed. Then, aliasing can be generated from those signal segments where the distance between two local signal maxima or minima is three pixels or less, since they correspond to signals with frequencies higher than one fourth of the original sampling rate, which is maximum allowed in the decimated signal (Fig. 4a). However, this is not solving the problem completely since certain signal segments can be classified as "low frequency segments contaminated by high-frequency" (Fig. 4b). They should rather be preserved than eliminated, particularly if the high-frequency contamination is small.

![Diagram](a) ![Diagram](b)

Fig. 4. High-frequency signal segments.

For the decimation factor of two, the condition for the detection of aliasing-generating signal segment means that, at any given point, we have to check whether there are two local maxima or minima within the distance of three or less points from it. An algorithm searching for this has been devised as follows.

We take a length-5 window and check if a subset of the two first samples is not mixing with the subset of the three next samples, i.e., the first two samples are greater than, or smaller, than the rest. Then we check also if the three first samples are not mixing with the subset of the two last samples. We move next to the previous point and perform the same test and after that we perform this test also in the point next to the center point.

If any of those tests shows us that there is no mixing, we check the variation amplitude of the signal on both sides of the point (Fig. 4b). This is because we may have monotonic signals "contaminated" by noise. If the amplitude of the contamination is lower than the difference between the varying parts (Fig. 4b), we classify the signal as "contaminated" monotonic signal. Also, if all those tests shows that there is no mixing, we classify the signal as "contaminated" monotonic signal, even without the amplitude check. In Fig. 5, we see the signal divided into two non-mixing subsets of length two and three.

![Diagram](a)

Fig. 5. Example grouping in length-5 window.

Now we can see that for an edge like in Fig. 4b, we always have at least one case of non-mixing subsets (the higher part of the edge and the lower part of the edge), and if the
contamination is low enough, i.e. the amplitude is lower than the height (difference) of the edge, we have a "contaminated" monotonic signal. Then, if we have a pure high frequency signal the following example shows that we always detect it.

If we take point 4 of the high frequency signal segment in Fig. 4a and divide the 5-point set into the two subsets (2, 3) and (4, 5, 6) we see that there is clear mixing of samples between those subsets. Similarly, we also have mixing between subsets (2, 3, 4) and (5, 6). Next we move to point 3 and see that there is mixing between (1, 2) and (3, 4, 5) as well as between (1, 2, 3) and (4, 5). The same check is still done in the point next to the center point, i.e. the point 5. The result is again mixing between (3, 4) and (5, 6, 7) as well as between (3, 4, 5) and (5, 7).

What if point 2 would have been lower than point 3? It would have meant that between subsets (2, 3) and (3, 4, 5) there wouldn't have been mixing. Then we would have calculated the variation of the signal in (2, 3) and (4, 5, 6) and found out that the variation would have been higher comparing to the difference in variation between the two subsets. The result would again have been classification as high frequency signal segment by the above procedure.

In practice, we only have to calculate the mixing for the last of the three points, because the status of the two first points, which have already been calculated previously, can be stored in the memory. So, we actually have to check the mixing only for one point at each direction (ver./hor./diag.) and check memory for the two others, which makes this scheme very simple.

Our basic algorithm is now as follows: If the high frequency signal segment is detected, then linear filtering is applied. In the opposite case, median filter is used.

3.2. Detection of Outliers

In our basic switching algorithm, signals classified as "low frequency", can still have impulsive noise contamination (Fig. 6), which may require removal by median filter.

Fig. 6. Impulsive contamination in low frequency segment.

In order to remove impulsive noise contamination from the low frequency signal segments, we need some kind of a signal-dependent way of detecting it. The idea used is that we take a 3x3 mask and check if the value of the center point of the mask is higher than the output of a linear lowpass filter in the same point. If it is so and if at least 7 of the other points covered by the mask have lower values than the center point value, we classify the point as an outlier. This means that if the size of an outlier is less than 3 pixels, it will be filtered out. Otherwise, it will be classified as a significant structure in the image and it will be preserved.

If we find an outlier and therefore use median filtering, we must also use median in all the surrounding 8 points. Otherwise, possible linear filtering at the next point could still cause some annoying effects, because the impulse may still have noticeable effect if linear filter is used. Negative outliers are of course checked in the same way.

The final algorithm with outlier detection is illustrated in Fig. 7.

![Fig. 7. The final algorithm.](image)

4. FILTER IMPLEMENTATION

The filtering scheme from Fig. 3. has been implemented using carefully selected median and linear part. In the 2-D implementation, the ordering tests can be performed in two (vertical/horizontal) or four (vertical, horizontal and 2 diagonal) directions. In the implementation, a multilevel median filter, based on a 5-point cross- and diamond-shaped medians in a 3x3 mask has been used, because of its excellent detail preservation [2]. In this algorithm we can use of course any other non-linear filter, but for this specific implementation we found the multilevel median to be the best. As linear filters, three different filters were used: an optimal Burt filter in a 5x5 mask [4], an optimized 13x13 halfband filter [5], and a simple 3x3 averaging filter.

5. TESTS

5.1. Visual Tests

Visual tests were made to check if there is visible aliasing left in images after filtering. In Fig. 8a, we see the "Young Couple" filtered using our algorithm (with high frequency check in 4 directions) and decimated by two. In Fig. 8b, we show where the algorithm used median and linear filters. White areas indicate the use of median filter and black areas indicate the use of linear lowpass filter.

![Fig. 8. "Young couple" after filtering (a). Areas of operation of median and linear filter (b).](image)
Clear difference can be seen, when compared to median filtered "Young couple". After our algorithm there is no aliasing left and sharp edges are still preserved, whereas after median filtering we have quite annoying visible aliasing left in the image (Fig. 2b).

5.2. Numerical Tests

5.2.1. Entropy

A good measure showing statistical information content in images is entropy. Entropy can be considered for example as the minimum amount of bits/pixel needed in compressing the image. The basic equation for the entropy in a 256 gray scale value image is:

\[
E = \sum_{h=0}^{255} (-p_h \times \log p_h)
\]  

(1)

where \(p_h\) is probability of a gray scale value in the image. The difference image is calculated by subtracting the filtered image from the original image. The lower the entropy of the difference image is, the less information is lost in filtering. Entropies for the difference images after filtering of standard test images "Lena" and "Zone", are presented in Table 1.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Lena (256x256)</th>
<th>Zone (256x256)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multilevel Median</td>
<td>2.5341728</td>
<td>4.6622376</td>
</tr>
<tr>
<td>3x5 aver. filter</td>
<td>4.0353536</td>
<td>7.752631</td>
</tr>
<tr>
<td>Burt filter, 3x5</td>
<td>3.874144</td>
<td>7.477157</td>
</tr>
<tr>
<td>Opt. 13x13 halfband</td>
<td>4.694448</td>
<td>7.541777</td>
</tr>
<tr>
<td>Our alg. with 3x5</td>
<td>2.540062</td>
<td>6.324831</td>
</tr>
<tr>
<td>Our alg. with 5x5</td>
<td>2.461635</td>
<td>6.356431</td>
</tr>
<tr>
<td>Our alg. with 13x13</td>
<td>2.383714</td>
<td>6.338446</td>
</tr>
</tbody>
</table>

Table 1. Entropies.

5.2.2. Gaussian Noise Suppression

To evaluate the Gaussian type noise suppression capability of the algorithm, the ratio between the variance of the processed noise and the original noise is calculated. The Gaussian noise of mean 0 and standard deviation 10 is used in this test. The results for multilevel median, the three different linear filters and the basic algorithm using those three linear filters, are presented in Table 2. Linear filters have the best Gaussian noise suppression capabilities as expected and the median filter performs worst. Our algorithm gives results between them, which is predictable.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Ratio of variances</th>
</tr>
</thead>
<tbody>
<tr>
<td>Multilevel median</td>
<td>0.310</td>
</tr>
<tr>
<td>3x5 averaging filter</td>
<td>0.118</td>
</tr>
<tr>
<td>Burt filter, 3x5</td>
<td>0.453</td>
</tr>
<tr>
<td>Opt. 13x13 halfband</td>
<td>0.190</td>
</tr>
<tr>
<td>Our alg. with 3x5</td>
<td>0.332</td>
</tr>
<tr>
<td>Our alg. with 5x5</td>
<td>0.273</td>
</tr>
<tr>
<td>Our alg. with 13x13</td>
<td>0.163</td>
</tr>
</tbody>
</table>

Table 2. Gaussian noise suppression.

5.2.3. Impulsive Noise Suppression

For impulsive noise suppression, the mean absolute error (MAE) for noisy points is calculated. The impulsive noise used is "salt-and-pepper" type, i.e. randomly located black and white impulses. The results can be seen in Table 3.

When the noise level is high, our algorithm works slightly better than multilevel median filter because closely located impulses can be detected as high frequency segments, where pure multilevel median treats such impulse groups as parts of the image and preserves them. This is quite interesting result, because median is supposed normally to operate much better in impulsive noise than linear filter, but yet a combination of median and linear filter gives even better results.

6. CONCLUSION

Both linear and median filters have their drawbacks. Linear filters cause blurring on edges and details and median filters cause aliasing in high frequency areas. In this paper a new filtering scheme is proposed, which combines the features of the filters, leaving out the drawbacks of them. The new filter has proved to be efficient in the impulsive noise suppression and even Gaussian noise suppression capability is improved comparing to the median. Results of visual and numerical tests confirm for the expected behavior of the new filter.

REFERENCES

SYNTHESIS OF INHERENTLY STABLE AND LOW-SENSITIVE 2-D IIR FILTERS USING THE $l_1$-PASSIVITY CONCEPT

MAREK DOMAŃSKI

Politechnika Poznańska, Instytut Elektroniki i Telekomunikacji
ul. Piotrowo 3a, 60-965 Poznań, Poland

The concept of $l_1$-passivity has been obtained by extension of the well-known concept of passivity of digital systems. It has been proved that the stability properties implied by the $l_1$-passivity are similar but even little bit stronger as those implied by the classical passivity related to the $l_2$-norm. Both classical passivity and $l_1$-passivity are powerful tools to obtain low-sensitive filters. The paper presents some techniques to design $l_1$-passive recursive 2-D digital filters. Some low-sensitive filter structures are described.

1. $l_1$-PASSIVE 2-D DIGITAL FILTERS

The very good properties of the wave digital filters [1] are well established also in the two-dimensional (2-D) case. The passivity related to boundedness of the weighted $l_1$-norm of the state vector is the kernel of the theory of such filters. Recently it has been shown that the use of some more general functions of the state vector, of the input, and of the output also leads to reasonable definitions of the (generalized) passivity [2-4]. Among the great many possible measures that based on the $l_1$-norm is of the particular interest.

A 2-D digital filter (cf. Fig.1) is said to be $l_1$-passive iff

$$\sum_{i=1}^{N} |a_i(n)| - |b_i(n)| \geq 0 \quad (1)$$

for all $n = (n_1, n_2)$ where $n_1, n_2$ are integers, e.g., coordinates of a pixel in a digital image) and for all admissible $a(n) = |a_1(n)|, \ldots, |a_N(n)|$. Here $a(n)$ and $b(n)$ are the vectors of the signals which are incident to the shift-free network and reflected from it, respectively.

We could define also $l_1$-lossless linear digital systems by setting the equality in Eq. (1). Nevertheless, it has been proved that a linear non-degenerate $l_1$-lossless system with real signals does not exist [4.5]. The $l_1$-lossless linear digital systems have been considered hitherto in the case of nonnegative signals only [3,4].

![Fig. 1. A digital filter.](image)

Now, we briefly report some properties of the $l_1$-passive systems:

1) A linear system is $l_1$-passive iff there exist $g_{ij} > 0 \quad (i=1, \ldots, N)$ such that

$$\sum_{i=1}^{N} g_{ij} \leq g_{ij} \quad (2)$$

for each $j=1, \ldots, N$ and each $n$, where $S = [s_{ij}]_{N \times N}$ describes the shift-free network $b(n) = S \cdot a(n)$. (3) Let assume $g_{ii} = 1 \quad (i=1, \ldots, N)$. It can be

*The paper has been prepared in part on leave to Lehrstuhl für Nachrichtentechnik, Ruhr-Universität Bochum, Germany under support of the Alexander von Humboldt-Foundation.
shown that such an assumption does not influence the generality of any further considerations.

2) There is \( M(\omega_1, \omega_2) \leq 1 \),

where \( M(\omega_1, \omega_2) \) is the magnitude response of a \( \ell_1 \)-passive system. This property gives an opportunity to obtain low-sensitive structures. Some filters considered in this paper exhibit zero sensitivity of their transfer functions at certain characteristic frequencies. The proper design techniques place such characteristic frequencies in the passbands and reduce the magnitude sensitivity in the passbands.

3) The \( \ell_1 \)-passivity implies the BIBO-stability of 2-D linear shift invariant systems while the classical passivity (related to the weighted \( \ell_\infty \)-norm) does not. This phenomenon is related to the possibility of existence of the non-essential singularities of the second kind on the boundary of the unit bidisc which often lead to severe testing problems in the systems which are not passive. On the other hand, the non-essential singularities of the second kind on the boundary of the unit bidisc are rather isolated points, and therefore, in real conditions of the finite precision arithmetic, they are not crucial.

4) A 2-D \( \ell_1 \)-passive digital systems is output \( \ell_1 \)-stable.

The definition of the output \( \ell_1 \)-stability (3) is a little bit complicated and will be omitted here.

The above property implies that a \( \ell_1 \)-passive 2-D system remains stable also under finite precision arithmetic conditions so far as Condition (2) holds for a shift-variant system being equivalent to the real system with constant coefficients and finite word-length. That is, the zero-input parasitic oscillations are suppressed under such a condition.

The theory of \( \ell_1 \)-passivity has already become fruitful in the design of the low-sensitive 1-D and 2-D FIR filters (6,7). The aim of this paper is to propose some new inherently stable 2-D recursive filter structures and corresponding new design techniques which result in low-sensitive filters.

2. THE DIRECT SYNTHESIS

A given transfer function \( H(z) \) is to be implemented in the structure shown in Fig. 2. The transfer function

\[
H(z) = \frac{F(z)}{Q(z)} = \frac{Z_1^{T}PZ_2}{Z_1^{T}QZ_2},
\]

where \( Z_i = (1, z_i^{-1}, \ldots, z_i^{-M_i})^T \), \( i = 1, 2 \),

\[
z = (z_1, z_2), L_1, L_2, M_1, M_2 \in \mathbb{R}^+;
\]

\[
P = [p_{ij}] \in \mathbb{R}^{L_1 \times L_2}, Q = [q_{ij}] \in \mathbb{R}^{L_2 \times M_2}
\]

has to be rearranged as

\[
H(z) = \frac{(1-\beta)F_1(z)F_3(z)}{1-\beta F_1(z)F_2(z)}
\]

where \( 0 < \beta < 1 \), and \( F_1(z) \), \( F_3(z) \), \( F_4(z) \) can be implemented using the low-sensitive nonrecursive networks from Fig. 3.

---

Fig. 2. Simple structure for direct synthesis of \( \ell_1 \)-passive 2-D filters.

Fig. 3. The low-sensitive nonrecursive building block.

There is \( F_j(z) = \sum_{k} k_s \delta_{ij} f_i(z) \).

\[
j=1, 2, 3, a_{ij} > 0, a_{ij}^{+} a_{ij}^{-} = 1, k_s = 1, m_{ij} = 1, m_{ij}^{+} = 1, m_{ij}^{-} = 1, m_{ij}^{+} m_{ij}^{-} = 1.
\]

A transfer function \( H(z) \) can be rea-
lized in the considered structure with the accuracy to a constant factor if
\[ \sum_{i=1}^{M_1} \sum_{j=0}^{M_2} |q_{ij}| + \sum_{i=1}^{M_1} |q_{0j}| < 1. \]
(8)

The system remains \( l_1 \)-passive as long as the coefficients \( d \) (cf. Fig. 3) are in the range \( 0 < d < 1 \), i.e., the system exhibits the property of \( l_1 \)-passivity also for rounded coefficients. This observation proves stability for any coefficients rounded according to the common sense. Low sensitivity of the amplitude characteristic can be mentioned in several practical examples already examined.

**EXAMPLE 1.** Given are

\[
P = \begin{bmatrix}
0.0780 & 0.0386 & 0.0084 \\
0.0386 & 0.0260 & 0.0010 \\
0.0084 & 0.0010 & 0.0000 \\
1.0000 & 0.0000 & 0.0000 \\
-0.3120 & -0.1544 & -0.0336 \\
-0.1544 & -0.1040 & -0.0040 \\
-0.0336 & -0.0040 & 0.0000
\end{bmatrix},
\]

\[
Q = \begin{bmatrix}
0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000 \\
0.0000 & 0.0000 & 0.0000
\end{bmatrix}.
\]

**Fig. 4.** The amplitude response.

The corresponding low-sensitive network is given in Fig. 5, where \( d_1 = 0.390 \), \( d_2 = 0.6328 \), \( d_3 = 0.5804 \), \( d_4 = 0.8936 \), \( \beta = 0.8 \). The network exhibits an attenuation zero in \( \omega = \omega_0 = 0 \) where the transfer function is independent from the above coefficients.

**Fig. 5.** The filter from Example 1.

3. THE MULTILOOP TECHNIQUE

Another technique possible [8] consists in application of the multiloop structure. The filter obtained exhibits a characteristic frequency where the transfer function is independent from the values of the coefficients in the filter structure. Thus the amplitude sensitivity which is a continuous function of the frequency tends to be small in the neighborhood of the characteristic frequency which is possibly located in a passband.

Let consider synthesis of an all-pole filter \( 1/Q(z) \). We use the three groups of disjoint loops (cf. Fig. 6). There are \( M_{p1} \) loops \( \beta_{11} z^{-1} \) (group 1), \( M_{p2} \) loops \( \beta_{21} z^{-1} \) (group 2), and some loops (group 3) being disjoint with those above. There is \( M_{p1} M_{p1}, M_{p2} M_{p2} \).

**Fig. 6.** The multiloop structure.

If only the loops of the types 1 and 2 are used, we have \( 0 < \beta_{k1} < 1 \), \( q_{01} < 0 \), \( q_{01} < 0 \), \( \beta_{11} \ldots \beta_{1M_{p1}} = q_{10} \), \( \beta_{21} \ldots \beta_{2M_{p2}} = q_{21} \), and summing up over all the ordered pairs of respective \( \beta \)'s we get

\[
\sum_{i=1}^{M_{p1}} \beta_{1i} = q_{10}, \quad \sum_{i=1}^{M_{p2}} \beta_{2i} = q_{21}.
\]

The expressions for the other coefficients \( q_{ij} \) we get from the Mason theorem. It is useful to get

\[
\beta_{11} = \ldots = \beta_{1M_{p1}} = \beta_{11}, \quad \beta_{21} = \ldots = \beta_{2M_{p2}} = \beta_{22},
\]

where \( 0 < \beta_{k1} < 1 \) (\( k=1,2 \), \( i=1,\ldots,M_{pk} \)).

Then, using the loops of type 3 (with transfer functions \( t_{3i} z^{-i} \), \( t_{3i} z^{-2} \)) we get

\[
-\beta_{1M_{p1}} t_{30i} = q_{10}, \quad q_{10} < 0, \quad q_{01} < 0,
\]

\[
-\beta_{2M_{p2}} t_{30i} = q_{20}, \quad q_{20} < 0, \quad q_{01} < 0,
\]

\[
\frac{M_{p1}}{2} (M_{p1} - 1) \beta_{1} t_{320} = q_{30},
\]

\[
\frac{M_{p2}}{2} (M_{p2} - 1) \beta_{2} t_{330} = q_{30}.
\]
\[ M_{p1}^1M_{p2}^1\beta_1\beta_2^{-1}t_311 = q_{11} \]
\[ M_{p1}^1(M_{p1} - 1)M_{p2}^1\beta_1\beta_2^{-1}t_312 = q_{21} \]
\[ M_{p1}^1M_{p2}^1(M_{p1} - 1)\beta_1\beta_2^{-1}t_312 = q_{12} \] and so on. The design problem is to find out the numbers \( M_{p1}, M_{p2}, \) and \( \beta \)'s. Then, some other groups of loops can be added. The technique can be extended in order to implement simultaneously the numerator \( P(z) \).

**EXAMPLE 2.** Let
\[ A = \begin{bmatrix} 0.02500 & 0.01799 & 0.02500 \\ 0.01799 & 0.05837 & 0.01799 \\ 0.02500 & 0.01799 & 0.02500 \end{bmatrix} \]
\[ B = \begin{bmatrix} 1.00000 & -0.58701 & 0.08614 \\ -0.58701 & 0.33142 & -0.05056 \\ 0.08614 & -0.05056 & 0.00177 \end{bmatrix} \]

![Fig.7. The amplitude response.](image)

4. CASCADE STRUCTURE

We briefly report the synthesis in the cascade structure [9] which consists of the digital two-ports (cf. Fig.9) implementing the lines of infinite attenuation (cf. Fig. 10).

![Fig.9. A digital two-port.](image)

![Fig.10. Lines of infinite attenuation.](image)

**REFERENCES**

\[ s = W_s^5 \]

\[ y = W_s \]

\[ n = W_n^\omega \]

\[ z = v + W_n^\omega + W W_s^e \]

\[ s = (H z^{-\frac{1}{2}}) z = (H z^{-\frac{1}{2}})(v + W_n^\omega + W W_s^e) \]

\[ e \Delta s - s = W_s^5 - (H z^{-\frac{1}{2}})(v + W_n^\omega + W W_s^e) = I - H z^{-\frac{1}{2}} W W_s^5 - (H z^{-\frac{1}{2}})(v + W_n^\omega) \]

### 2.2 Summary of Assumptions

1. Noise sources are assumed to be zero mean.
2. The measurement noise source \( \{v(t)\} \) is statistically independent of the other noise sources and has a covariance matrix \( R \).
3. The covariance matrix for the white noise signal \( [\xi(t)^T, \omega(t)^T]^T \) is denoted by \( \begin{bmatrix} G & F \\ F^T & I \end{bmatrix} \).
4. The signal channel \( W \) and the coloured measurement noise model \( W_n \) can be assumed to be asymptotically stable.
5. There can be no unobserved hidden modes in the individual plant subsystems or in the cascade system \( W W_s^e \).

### 2.3 Polynomial Matrix Descriptions

The system models have the following polynomial matrix descriptions:

\[ \begin{bmatrix} W & W_n \end{bmatrix} = A^{-1} [C & C_m] \]

\[ W_s = A_s^{-1} C_s \]

where \( A, C, C_m, A_s, \) and \( C_s \) are polynomial matrices in \( z^{-1} \) of compatible dimensions. Introduce the left-coprime pair \( C_1 \) and \( A_1 \) which satisfy:

\[ C A_s^{-1} = A_1^{-1} C_1 \]
and assume, without loss of generality, \( A_1(0) = 1 \) and \( A_2(0) = 1 \). Note for later use from Assumption (2i) in §2.2 the model \( \mathbf{CA}_t^b \) can include no unstable hidden modes.

### 2.4 Spectral Factorization

The solution of the \( H_2 \) deconvolution filtering problem (Shaked, 1976[6]) requires the introduction of a spectral-factor \( Y_f \) which satisfies the following equation:

\[
Y_f = W_n W_s^* + W_{n0} W_n^* + R
+ W_s G W_n^* + W_{n0} G^T W_s^* W_n^*
\]  

Replacing the system models by their polynomial matrix counterparts, note that \( Y_f \in \mathcal{R}(z^{-1})^{n_T \times n_T} \) may be written as:

\[
Y_f = (A_1 A)^{-1} D_f
\]

(10)

where \( D_f \) is a polynomial matrix left spectral-factor defined using:

\[
D_f = C_1 C_n^* C_1^* + A_1 C_n C_n^* A_1^* + A_1 A R A^* A_1^* + C_1 G C_1^* A_1^* + A_1 C_n G^T C_1^* C_1^* + A_1 C_n G^T C_2^* C_2^* C_1^* + A_1 C_n G^T C_2^* C_1^*
\]

(11)

The Hurwitz spectral-factor \( D_f \) exists, if and only if,

\[
\text{rank} \begin{bmatrix} C_1 C_n^* & A_1 C_n \\ G^{T} & 1 \end{bmatrix} \geq 1/2
\]

and \( A_1 A R^{1/2} \) have no left common factors with zeros on the unit-circle in the z-plane.

**Assumption (2iv)**: The noise and signal source models are assumed to be such that \( D_f \) exists and is strictly Hurwitz.

### 3. The \( H_2 \) Cost Minimization Problem

In this section the problem of the minimization of the variance of the weighted estimation error is considered (Grimble and Johnson, 1988[7]). These results are needed before attention can turn to the \( H_2 \) optimal estimation problem.

The \( H_2 \) optimal deconvolution problem involves the minimization of the estimation error:

\[
e(t - t) = s(t) - \hat{s}(t | t - t)\]

(12)

where \( \hat{s}(t | t - t) \) denotes the optimal linear estimate of the signal \( s(t) \) at time \( t \), given observations \( z(t) \) up to time \( t - t \). The scalar \( f = 0 \) (filtering), \( f < 0 \), (fixed lag smoothing) and \( f > 0 \), (prediction). The usual estimation error cost-function which is minimized has the form:

\[
J = \text{trace} \left\{ E[e(t - t) e^H(t - t)] \right\}
\]

(13)

where \( E\{\} \) denotes the unconditional expectation generator.

Introduce the asymptotically stable \( n_x \) square weighting function matrix: \( W_p = A_2^2 B_{p1} \) where \( A_2 \) and \( B_{p1} \) denote square polynomial matrices. Then the \( H_2 \) cost-function to be minimized becomes:

\[
J = \text{trace} \left\{ E[t W_p e(t - t) (W_p e(t - t))^H] \right\}
\]

(14)

and this can be written in complex integral form:

\[
J = \text{trace} \left\{ \frac{1}{2 \pi} \int_{D} e(t) W_p e(t) e(t) d^2 \right\}
\]

(15)

where \( D \) denotes the unit-circle contour in the z-plane and the adjoint \( W_p(z^{-1}) = W_p^H(z) \).

The following theorem summarizes the main results.

**Theorem 3.1**: \( H_2 \) Optimal Deconvolution Estimator

Consider the signal processing system and assumptions described in §2 and illustrated in Fig. 1. The optimal deconvolution filter can be calculated.

**Spectral factor**:

\[
D_2 = C_1 C_n C_1^* + A_1 C_n C_n^* A_1^* + A_1 A R A^* A_1^* + C_1 G C_1^* A_1^* + A_1 C_n G^T C_2^* C_2^* + A_1 C_n G^T C_2^* C_1^*
\]

(16)

**Diophantine Equations**:

The polynomial matrix solution \( (F_0, G_0, S_0) \), with \( F_0 \) of smallest degree, is required:

\[
A_2 F_0 + G_0 D_2 S_0 = C_1 C_n C_1^* + G C_1^* A_1^* + C_1 G^T C_2^* C_1^*
\]

(17)

**Robustness weighting diophantine equation**: If the cost-weighting \( W_p \) is dynamic the diophantine equation solution \( (F_1, N) \), with \( F_1 \) of smallest degree, is required using:

\[
A_2 F_1 + N D_s S_1 = B_2 F_0
\]

(18)

**Optimal estimator**:

\[
H_2 = (A_2^{-1} C_2 + B_2^{-1} N D_2^{-1} A_1 A
\]

(19)

**Proof**: Presented in Grimble (1982[8]).

**Lemma 3.1**: Properties of the Deconvolution Filter, Predictor and Smoother

The properties of the optimal deconvolution filter, defined in Theorem 3.1, can be detailed as follows.

**Minimal cost**:

\[
J_{\text{min}} = \text{trace} \left\{ \frac{1}{2 \pi} \int_{D} e(t) F_1 D_1^{-1} F_1^H + W_p W_p^H d^2 \right\}
\]

(20)

**Implied equation determining stability**:

\[
C_1 C_2 S_1 + A_1 S_1 \frac{-L+L}{D_2} = D_f
\]

(21)
4. $H_\infty$ Multivariable Estimator

The $H_\infty$ multivariable estimation problem can be solved using the $H_2$ optimization results in the following lemma derived by Kwakernaak (1984 [9]). Note that $W_\sigma$ represents a weighting filter.

**Lemma 4.1: Auxiliary $H_2$ Minimization Problem**

Consider the auxiliary problem of minimizing the $H_2$ criterion:

$$ J_\sigma = \frac{1}{2 \pi} \int_{|z|=1} \text{trace} (W_\sigma (z^{-1}) X(z^{-1}) W_\sigma^* (z^{-1})) \, dz $$  \hfill (23)

Suppose that for some real-ratio matrix: $W_\sigma^* (z^{-1}) W_\sigma (z^{-1}) \geq 0$, the cost function $J_\sigma$ is minimized by a function $X(z^{-1}) = X^*(z^{-1})$, for which $X(z^{-1}) = \lambda^2 T_d, \lambda > 0$ (a real constant matrix on $|z|=1$). Then the function $X(z^{-1})$ also minimizes:

$$ J_w = \frac{1}{2} \| X(z^{-1}) \|^2 = \frac{1}{2} \sup_{|z|=1} \{ \sigma_{\text{max}} (X(z^{-1})) \} $$  \hfill (24)

where $\| X(z^{-1}) \|_2$ denotes the spectral norm. \hfill (*)

**Proof:** The proof is similar to that in Kwakernaak, 1984 [9].

A solution has been presented in §3 on an $H_2$ minimization problem which is similar to that referred to in the above lemma. To relate the two problems let the dynamic weighting function $W_p$ be written as the product of two terms:

$$ W_p = W_\sigma W_{p0}, \quad W_p \in \mathbb{R} (z^{-1})^{n_3 \times n_5} $$  \hfill (25)

Then the function $X$ above can be related to the weighted estimation error as:

$$ X = W_{p0}^* W_{p0}^* , \quad X \in \mathbb{R} (z^{-1})^{n_3 \times n_5} $$  \hfill (26)

The $W_\sigma$ represents the weighting term, still to be determined, which appears in Lemma 4.1. The $W_{p0}$ will be assumed to represent the dynamic weighting selected by the designer to frequency shape the estimation error spectrum, so that the cost-function to be minimized is given as:

$$ J_w = \frac{1}{2} \sup_{|z|=1} \{ \sigma_{\text{max}} (X(z^{-1})) \} = \frac{1}{2} \sup_{|z|=1} \{ \sigma_{\text{max}} (W_{p0}^* W_{p0}^*) \} $$

The problem is now to determine the function $W_\sigma$ which will ensure the conditions of the Lemma are satisfied.

4.1 Derivation of the weighting filter $W_\sigma$

Assume that the scalar $\lambda$ in the above lemma is for the moment known. To ensure the conditions of the above lemma are satisfied, the $W_\sigma$ must now be found which leads to an equalizing solution

$$ X(z^{-1}) = \Lambda \Lambda^T z^{-1} T_d $$

The matrix $\Lambda$ introduced in the above expression may be defined as $\Lambda = \lambda T_d$ at the optimum. However, once the maximum singular value has been minimized and a solution for the controller obtained, it is sometimes possible to improve upon the solution by defining a more general $\Lambda$. The improvement can, for example, be in terms of reducing the maximum values of the remaining singular values.

Comparison of (20) and (23), evaluated at the $H_2$ optimum, therefore gives:

$$ W_\sigma T_d = W_{p0} T_d W_{p0}^* = W_\sigma \Lambda \Lambda^T W_{p0}^*$$

or

$$ W_\sigma T_d = W_{p0} T_d W_{p0}^* = W_\sigma \Lambda \Lambda^T W_{p0}^* $$

where

$$ F_1 \in \mathbb{P} (z^{-1})^{n_3 \times n_5} $$

**Introduction:** The stable minimum-phase transfer function $S \in \mathbb{P} (z^{-1})^{n_3 \times n_5}$ which satisfies:

$$ SS^* = \Lambda \Lambda^T - W_{p0} T_d W_{p0}^* $$

and assume that $\Lambda$ is chosen so that the transfer $S$ is invertible. From (27) and (28) now obtain:

$$ F_1 D_{f_1}^* D_{f_1} = W_\sigma SS^* W_\sigma^* $$

Define the spectral-factor $D_{f_0}$ which satisfies:

$$ D_{f_0}^* D_{f_0} = D_{f_1}^* D_{f_1} $$

and let the left-coprime pair $(F, D)$ satisfy:

$$ D^{-1} F = F_1 D_{f_1}^* $$

Then, to ensure the equalizing solution is obtained from (29) and (31):

$$ D^{-1} F S^{-1} D^{-1} F = W_\sigma SS^* W_\sigma^* $$

Let $F_3$ denote a Hurwitz polynomial matrix which satisfies:

$$ F_3 S^{-1} F_3 = FR $$

then from (32):

$$ W_\sigma^* D_{f_0} = D_{f_1}^* F_3 $$

To ensure a suitable definition $F_3 \in \mathbb{P} (z^{-1})^{n_3 \times n_5}$ is obtained the following assumption is required:

**Assumption (4):** The number of signal channels $n_s \leq n_s \leq n_3$.

The linear equation which enables $F_3$ to be computed can now be found. Introduce the left-coprime polynomial matrices $(S, F_3)$ as:

$$ S^{-1} F_3 = F_3 S^{-1} W_{p0} $$

then from (34) and (35):

$$ W_\sigma W_{p0} = A_p^{-1} B_p = D^{-1} S^{-1} F_3 $$

Hence, identify the desired weighting polynomials which ensure an equalizing solution is obtained, as:

$$ A_p = \tilde{S} D \quad \text{and} \quad B_p = \tilde{F}_s $$

and if $W_\sigma$ is required this may be found from (36) as:

$$ W_\sigma = A_p^{-1} B_p W_{p0} = (\tilde{S} D)^{-1} \tilde{F}_s W_{p0}^{-1} $$

4.2 Robust weighting diophantine equation

To calculate $(N, F_3)$ a more appropriate form of equation (18) is required. Substituting for the weightings in (37) the robustness diophantine equation
(18) becomes:

$$
\mathbb{S}DF_z + ND_z^*F = \bar{F}_S F_0 
$$

but from (31) : $DF_z = F D_{F0}$ and hence obtain:

$$
\mathbb{S}FD_{F0} + ND_z^*F = \bar{F}_S F_0 
$$

(39)

The optimal estimator follows from (19) but note that $B_z^{-1}$ enters the expression and hence from (33), (35) and (37) the matrix $F$ must be of normal full rank. The reason for introducing Assumption (4.1) should now be apparent.

**Theorem 4.1:** $H_{\infty}$ Estimator for Multichannel Estimation Problems

Consider the system shown in Fig. 1 and assume that the cost-function:

$$
J_{\infty} = \mathbb{S} \text{ sup } \mathbb{Z} \{ \sigma_{\text{max}}(W_{po} \Phi_{cc} W_{po}^*) \}
$$

is to be minimized where $\Phi_{cc}$ denotes the spectrum of the estimation error $e(t) \cdot e(-t)$ where $t = 0, \ldots, t < 0$ and $t > 0$ for filtering, smoothing and prediction problems, respectively. The $H_{\infty}$ optimal estimator may be computed by first calculating $D_{F}$ from (15) and then obtaining $O_{\text{c}}, \bar{S}_m, S_m^* F_0^*$ from (16) and (17). Also define the transfer $T_{F0}$ from (21), then the stable minimum phase transfer $S$ is defined using:

$$
SS^* = \lambda A^2 - W_{po} T_{F0} W_{po} = \lambda^2 I - W_{po} T_{F0} W_{po}^*
$$

(40)

The left-coprime pair $(F, D)$ can be found using:

$$
D^* F = F D_{F0}^* \text{ where } D_{F0} \text{ satisfies } D_{F0}^* D_{F0} = D_{Dz} = I
$$

(41)

Let $F_{G}$ denote the Hurwitz spectral factor satisfying:

$$
F_{G} F_{G}^* = F F^* \text{ and introduce the left-coprime polynomial matrices } (\bar{S}, \bar{F}_S)
$$

using:

$$
\bar{S}^{-1} \bar{F}_S = F_{G} S^{-1} W_{po}
$$

(42)

The $H_{\infty}$ optimal linear estimator can then be found by first calculating the solution $(N, F)$, with $F$ of smallest degree, of the linear equation:

$$
\bar{S} FD_{F0} + ND_z^*F = \bar{F}_S F_0
$$

(43)

**Optimal estimator:**

$$
H_{F} = (A_{G}^{-1} Q_{G} + \bar{F}_{G}^{-1} N) D_{z0} A_{G}
$$

(44)

**Optimum function and minimum cost:**

$$
X_{\text{min}} = \lambda^2_0 I \text{ and } J_{\text{min}} = \lambda^2_0
$$

(45)

where $\lambda^2_0$ is the smallest scalar such that equations (40) to (43) are satisfied.

**Proof:** The theorem follows by collecting the previous results and by invoking the results of Theorem 3.1.

5. **Conclusions**

A solution to the optimal $H_{\infty}$ multichannel deconvolution estimation problem was presented which included coloured measurement noise which could be correlated with the signal to be estimated. The results apply to filtering, prediction and fixed-lag smoothing problems by appropriate specification of the integer $z$. The deconvolution problem has not previously been considered from a $H_{\infty}$ norm minimization viewpoint but it should offer potential advantages for uncertain systems.

**References**


Underdetermined Growing and Sliding Window Covariance Fast Transversal Filter RLS Algorithms

DIRK T.M. SLOCK
Institut EURECOM
2229 route des Crêtes
Sophia Antipolis
F-06560 Vallonnet, FRANCE

Abstract—We consider the covariance formulation, or hence a rectangular window for the least-squares (LS) criterion. We are specifically interested in the case when the window length is shorter than the filter length, leading to an underdetermined LS problem. We develop Fast Recursive LS (RLS) algorithms in Transversal Filter form (ETF) for the two cases of an underdetermined growing or sliding window (UGC/USWC). The USWC case allows for a fast initialization of GWC algorithms, obviating the need for a soft-constraint initialization. The USWC case obviously represents an adaptive algorithm, where the growing window case can easily be converted into one by constantly sweeping the window length from one up to some finite value. These two classes of underdetermined LS algorithms can be situated in between the Normalized Least-Mean-Square (NLMS) algorithm and the conventional RLS algorithms. An inherent prewhitening mechanism renders their convergence less sensitive to the coloring of the input signal spectrum than is the case for the NLMS algorithm. Their underdetermined LS character additionally endows them with relatively fast tracking characteristics. Relations to existing algorithms, modifications and simplifications will also be discussed.

1. Introduction

The tracking characteristics of the (N)LMS algorithms are inherent in the iterative nature of the algorithm; the rate of convergence depends on the eigenvalue spread of the input covariance matrix. For RLS algorithms, the tracking is independent of the eigenvalue spread and can be made arbitrarily fast by appropriate choice of the (effective) window length. Fast RLS algorithms and especially the ETF algorithms [1] take only $2N/2N$ (stabilized form) or $2N/15N$ flops per input (pruning weighting), or $2N/15N$ flops in the SWC formulation [7], [8]. This should be compared to the $2N$ flops for the LMS or Normalized LMS algorithms.

Here we propose a new class of algorithms which are intermediate between LMS and fast RLS in complexity, and in tracking performance (or sometimes the best). Before we get more specific, let us consider the following motivations.

1.1. Stochastic Considerations

A general stochastic gradient algorithm is characterized by the following update equations:

$$
\begin{align*}
\mathbf{w}_k &= \mathbf{w}_{k-1} + \lambda_k \mathbf{x}_k \mathbf{e}_k \\
\mathbf{e}_k &= \mathbf{x}_k - \mathbf{w}_{k-1}^T \mathbf{x}_k \\
\mathbf{w}_{k-1} &= \frac{\mathbf{w}_{k-1}}{\mathbf{w}_{k-1}^T \mathbf{w}_{k-1} + \mu} \\
\mathbf{R}^{-1} &= \mathbf{R}^{-1} + \mathbf{w}_{k-1} \mathbf{w}_{k-1}^T
\end{align*}
$$

(1)

where ideally $\mathbf{w}_k \approx \mathbf{w}_k \mathbf{x}_k$ and $\mathbf{e}_k$ is used to compensate for the non-white spectrum of the input signal $\mathbf{x}_k$. The NLMS algorithm omits this decorrelation $(\mathbf{R} = \mathbf{w}_k \mathbf{w}_k^T)$, while the RLS algorithm uses the sample covariance matrix $\mathbf{R}_k$ as an estimate for $\frac{1}{\mu} \mathbf{R}$. One should notice that in the RLS algorithm, the size of the decorrelation matrix is automatically put equal to the length of the (FIR) filter that is being adapted. In many applications however, this coupling may not be well-motivated. Consider for instance the problem of acoustics echo cancellation. Here, FIR filters of very long length are being used. On the other hand, the input signal is typically a speech signal for which in speech coding, prediction filters with an order of typically only 10 are used (to whiten the signal). So one could consider using a biased matrix for $\mathbf{R}^{-1}$ in (1) with a (one-sided) bandwidth of only 10. This has been proposed in [5], in which non-fast and fast recursive algorithms are then developed. The resulting algorithm could be considered to be of the symmetric Instrumental Variable type. Notice that this approach corresponds very closely to first sending the input signal through an adaptive prediction filter of order 10, and then using the whitened input signal as input to the (N)LMS algorithm. This last approach has been proposed in [6]. One may remark that in this last approach, when considering tracking non-stationarities, the lag of the (N)LMS algorithm has to be added to the lag in the prewhitening process. In the Fast Newton Transversal Filters (PNTF) of [3] on the other hand, some peculiarities have been observed in [5] when the input signal is non-stationary, due to the fact that the optimal low-order prediction filter may change considerably over the length of the adapted FIR filter.

1.2. Deterministic Considerations

Nevertheless, the algorithms described above yield a significant improvement in convergence speed (in particular a substantial reduction in sensitivity to eigenvalue disparity of $\mathbf{R}$) when the input signal is strongly colored. However, it is useful to also take a look at some deterministic aspects. Indeed, based on only the above stochastic considerations, it is not possible to explain why the RLS algorithm converges significantly faster than the (N)LMS algorithm (e.g. by a factor of 5) even in the white input signal case! This advantage of the RLS algorithm is due to the fact that the RLS algorithm solves a (overdetermined) set of equations exactly at each time step ($N$ equations are sufficient to get an unbiased estimate of the $N$ FIR filter coefficients). Although asymptotically for large window lengths, the rectangular window and the exponential window give equivalent performance for corresponding window lengths, there is reason to believe that the rectangular window leads to better tracking performance for short windows in strongly non-stationary environments.

The SWC RLS algorithm provides the least-squares solution to $L \geq N$ equations, which are obtained by writing out the error signal at $L$ consecutive time steps:

$$
\min_{\mathbf{w}_k} \left\{ \| \mathbf{d}_k - \mathbf{x}_N \mathbf{w}_k \mathbf{w}_k^T \| \right\}
$$

(2)
When $L < N$, the above problem is underdetermined. A unique solution can still be found through the minimum-norm solution. In fact, the NLMS algorithm (with $\beta = 1$) provides this minimum-norm solution for $L = 1$. We shall consider algorithms for the cost function (2) with $1 \leq L \leq N$, which hence cover a continuous range between the SWC RLS algorithm and the NLMS algorithm, and which have the deterministic projection interpretation of making L consecutive error samples exactly zero.

As far as stochastic considerations are concerned, it turns out that in the Underdetermined Growing and Sliding Window Covariance (UGWC/USWC) RLS algorithms considered here, there is also inherently a replacement of the input signal by its Lth order prediction residual. From this point of view, it is possible to view some recently proposed algorithms as approximations to the USWC RLS algorithms.

2. A Unified Least-Squares Criterion

We shall mostly adhere to the notation of [1], [7], [8]. The following criterion will prove to encompass quite a number of different adaptation techniques. Consider

$$\min_{W_{L,L}} \left\{ \| d_{L,L} - X_{N,L,L} W_{L,L} \|_2^2 + \| W_{L,L} - W_{L,L-M} \|_2^2 \right\}$$

(3)

where \(\| \cdot \|_2 = \sqrt{\text{tr}(\cdot)}\) and \(S_{L,L}, T_{L,L}\) are Hermitian positive definite matrices. Before discussing various applications of this formulation, let us consider the minimization of this criterion (3). Setting the gradient equal to zero, and applying the Matrix Inversion Lemma (MIL) to solve the resulting linear equation in \(W_{L,L}\), we get

$$W_{N,L} = W_{N,L-M} + \left( A_{N,L} - B_{N,L-M} X_{N,L,L} \right) X_{N,L,L} S_{N,L}^{-1} X_{N,L,L}^{*}$$

(4)

The Hessian of the quadratic criterion (3) is

$$H_{L,L} = T_{L,L}^{-1} + X_{N,L,L} S_{L,L}^{-1} X_{N,L,L}^{*}$$

(5)

to which we have applied the MIL to obtain (4). Usually, \(M = 1\), unless we consider block processing. In all applications we consider here, the idea is to equalize the nature of both terms that constitute \(H_{L,L}\). In a first class of algorithms, we take \(L = 1\) and \(S_{L,L} = I\), which renders the nature of the second term in (5) of a covariance type (be it of low rank). Hence, \(T_{L,L}^{-1}\) is chosen to be representative of the covariance matrix of \(X_{N,L}(b)\). In the stochastic Newton algorithms, we choose (apart from a scalar multiple) \(T_{L,L}^{-1} = R_{L,L} = E X_{N}(k) X_{N,L}^{*}(k)\). In the absence of second order statistics, the RLS algorithms with exponential weighting factor \(\lambda\) uses \(T_{L,L}^{-1} = \lambda R_{N,L-M}\), where \(R_{N,L} = \sum_{k} \lambda^{k-N} X_{N}(k) X_{N,L}^{*}(k)\). In the Fast Newton algorithm [3], \(R_{L,L}\) is replaced by the Maximum Entropy extension of a banded restriction of \(R_{N,L}\) (with banded inverse).

In a second class of algorithms, one takes \(T_{L,L} = I\), which has all eigenvalues equal to a constant. Hence, now \(S_{L,L}\) should be chosen such that the eigenvalues of the second term in (5) are either constant or zero (since the term is not of full rank). The way to achieve this exactly is to make the second term in (5) proportional to a projection matrix and hence to choose \(S_{L,L} = \mu_{L} X_{N,L,L}^{*} X_{N,L,L} \). With \(L < N\), we have an "underdetermined" problem (strictly speaking, we only get an underdetermined problem as \(\mu_{L} \to 0\)). The filter solution (4) can now be rewritten as

$$W_{N,L} = W_{N,L-M} + \frac{1}{\mu_{L}} \left( A_{N,L} - B_{N,L-M} X_{N,L,L} \right) X_{N,L,L}^{-1}$$

(6)

where \(R_{L,L,N,L}\) is the sample covariance matrix for a SWC problem with filter length \(L\) and window length \(N\) interchanged. One immediately recognizes the NLMS algorithm when \(M = L = 1\). (Use (6), one can establish the following relation between the \(a_{i}\) posteriori and the \(a_{i}\) priori error vectors

$$\mu_{L} d_{L,L} - W_{N,L} X_{N,L,L} = \frac{1}{1 + \mu_{L}} \left( \mu_{L} d_{L,L} - W_{N,L-M} X_{N,L,L-M} \right)$$

(7)

One may note that the convergence of (6) is governed by a product of matrices of the form \((1 - \frac{1}{1 + \mu_{L}})\), from which one can see a prewhitening effect of order \(L = 1\) transpose. In particular for \(\mu_{L} = 0\), this becomes the projection matrix \(X_{L-L} X_{L-L}^{*}\) leading to \(L\) a posteriori errors being zeroed.

3. Underdetermined Sliding Window Covariance FTF

Here we take \(M = 1\) and fixed \(L < N\). Using the relation (7) between a priori and a posteriori error vector, and the shift structure of the data vectors, one can establish the following recursion

$$d_{L,L} = X_{N,L,L} W_{L,L-M} + \frac{1}{\mu_{L}} \left( \mu_{L} d_{L,L} - W_{N,L-M} X_{N,L,L-M} \right) X_{N,L,L} X_{N,L,L}^{*}$$

(8)

where \(c_{N,L}(k) = d_{L,L} - W_{N,L} X_{N,L,L} \). This is one recursion that is part of the joint-process section, and it takes \(N\) operations, for the computation of \(c_{N,L}(k-1)\). On the other hand, we also have the well-known order update relations for \(R_{L,L}^{-1}\). We get by multiplying with \(X_{N,L,L} a_{L,L}\) and using the shift structure of this matrix,

$$R_{L,L}^{-1} X_{N,L,L}^{*} = \begin{bmatrix} R_{L,L}^{-1} X_{N,L,L}^{*} & 0 \\ R_{L,L-M}^{-1} X_{N,L,L-M} & A_{L,L}^{*} S_{L,L}^{-1} A_{L,L} \end{bmatrix}$$

(9)

where \(a_{L,L} = \frac{1}{\mu_{L}} (d_{L,L} - W_{N,L-M} X_{N,L,L-M}) X_{N,L,L}^{*} \). Let

$$g_{L} = \frac{1}{1 + \mu_{L}} \left( \mu_{L} g_{L-1} + A_{L,L}^{*} S_{L,L}^{-1} A_{L,L} \right)$$

(10)

Then combining (8) and (9), one can work out a recursion for \(g_{L}\) and the remaining part of the joint-process section can be written as

$$g_{L} = \mu_{L} g_{L-1} + \xi_{L} S_{L,L}^{-1} A_{L,L} \xi_{L}^{*}$$

(11)

$$W_{N,L} = W_{N,L-M} + g_{L}$$

(12)

where \(\xi_{L}\) and \(\xi_{L}^{*}\) are quantities that take \(O(L)\) operations to compute. Apart from the \(O(L)\) operations, the joint-process part takes strictly speaking \(4N\) multiplications and \(4N\) additions, but in fact one should count these as \(4N\) multiply-accumulates. A small simplification occurs when \(\mu_{L} \to 0\). In that case we have

$$g_{L} = c_{N,L}(k-1) A_{L,L}^{*} S_{L,L}^{-1} A_{L,L}$$

(13)

4. Underdetermined Growing Window Covariance FTF

In this case we let \(L\) and \(k\) grow. We shall consider two possibilities, corresponding to whether we take \(k = M\) to be some fixed time instant (the beginning of a sweep) or \(M \to N\). For the first possibility, which we shall refer to as UGW, let the sweep start at time instant \(k_{0} = k - M\) and we shall let \(\mu_{L}\) be constant within one sweep (of \(L = M\) increasing from 1 to some maximum value, and \(k = k_{0} + L\)). In this case, it turns out to be convenient to work with the two components
that make up the error vector separately. So we can rewrite (9) as

\[ W_{N,k} = W_{N,k_{0}} + \frac{1}{1+\rho_{a}^{2}} \frac{\sigma_{L_{1}+1}^{2}}{\sigma_{L_{1}}^{2}} N_{L_{1}+1} X_{N,k_{0}+1} \tag{14} \]

To find a recursion for \( W_{N,k} \), in this case, we shall exploit one more recursion, namely the one for the projection operators, viz.,

\[ P_{X_{N,k-1}} = P_{X_{N,k}} + P_{N_{k-1}} \quad P_{N_{k-1}} = P_{X_{N,k-1}} + P_{N_{k-1}} \tag{15} \]

of which we need the simultaneous order and time updates. This, together with (9) and the shift of \( d_{N,k} \), allows us to find the recursion

\[ \begin{align*}
W_{N,k+1} &= W_{N,k} + \\
\frac{1}{\rho_{a}^{2}} \left( d_{N,k+1}^{2} - W_{N,k} X_{N,k+1}^{2} \right) \sigma_{L_{1}}^{-1}(k+1) e_{N,k+1}^{2} \tag{16}
\end{align*} \]

So the joint-process part takes \( 2N \) operations.

In the second possibility, denoted as USWC2, we let \( M = 1 \). The derivation parallels more than the USWC algorithm, and in particular we can establish the following recursion, using the relation (7),

\[ d_{N,k}^{2} - W_{N,k} X_{N,k+1}^{2} = \left[ e_{N,k+1}^{2}(k+1) \right] \tag{17} \]

Using this recursion, which is again part of the joint-process part, and the recursion (9) (simultaneous order and time update part), we get

\[ g_{k+1} = \frac{\mu_{a}}{1+\mu_{a}^{2}} \frac{1}{1+\mu_{a}^{2}} g_{k} \tag{18} \]

\[ \times \left( d_{N,k+1}^{2} - W_{N,k} X_{N,k+1}^{2} \right) \sigma_{L_{1}}^{-1}(k+1) e_{N,k+1}^{2} \tag{19} \]

where \( g_{0} = 0 \). Here, the joint-process part takes \( 3N \) multiplications and \( 3N \) additions, or \( 4N \) multiply-accumulates.

5. Prediction Parts

The joint-process parts of the above algorithms require vectors of \( N \) prediction error vectors. To derive updates for these prediction error vectors, we rely on the birectangularization procedure explained in [9]. Basically, this procedure consists of a double application of the well-known update identity for projection operators. It leads to a formulation based on \( 2 \times 2 \) rotations \( \Phi \) (circular) and \( \Psi \) (hyperbolic). We shall not provide the details for these rotation matrices. However, the following general facts hold. These rotations can be organized so that they take \( 2N \) multiply-accumulates. And the entry of these rotations can be computed in \( O(L) \) operations.

5.1. Underdetermined Sliding Window Covariance (USWC) FTF

The prediction part needs to provide \( e_{L,k} \) and \( r_{L,k} \) for all \( k \) (when \( \mu_{a} = 0 \), change \( L \) by \( L-1 \) in the procedure below).

This can be accomplished with the following four rotations

\[ \begin{align*}
&\left[ \begin{array}{c}
\frac{d_{N,k-1}^{2}}{\delta e_{N,k-1}}
\end{array} \right] = \Phi_{1} \left[ \begin{array}{c}
\frac{d_{N,k}^{2}}{\delta e_{N,k}}
\end{array} \right] \tag{20}
\end{align*} \]

\[ \begin{align*}
&\left[ \begin{array}{c}
\frac{e_{N,k}^{2}}{\delta e_{N,k}}
\end{array} \right] = \Phi_{1} \left[ \begin{array}{c}
\frac{e_{N,k-1}^{2}}{\delta e_{N,k-1}}
\end{array} \right] \tag{21}
\end{align*} \]

\[ \begin{align*}
&\left[ \begin{array}{c}
\frac{r_{N,k-1}^{2}}{\delta e_{N,k-1}}
\end{array} \right] = \Phi_{1} \left[ \begin{array}{c}
\frac{r_{N,k}^{2}}{\delta e_{N,k}}
\end{array} \right] \tag{22}
\end{align*} \]

\[ \begin{align*}
&\left[ \begin{array}{c}
\frac{r_{N,k}^{2}}{\delta e_{N,k}}
\end{array} \right] = \Phi_{1} \left[ \begin{array}{c}
\frac{r_{N,k-1}^{2}}{\delta e_{N,k}}
\end{array} \right] \tag{23}
\end{align*} \]

So the prediction part takes \( 8N \) multiply-accumulates.

5.2. Underdetermined Growing Window Covariance (UGWC) FTF

The prediction part needs to provide \( e_{L_{1},k} \), \( e_{L_{1},k+1} \), \( L_{1}, k = 1,2,\ldots, L \). This can be accomplished with the following three rotations

\[ \left[ \begin{array}{c}
\frac{e_{L_{1},k-1}^{2}}{\delta e_{L_{1},k-1}}
\end{array} \right] = \Psi_{1} \left[ \begin{array}{c}
\frac{e_{L_{1},k}^{2}}{\delta e_{L_{1},k}}
\end{array} \right] \tag{24}
\]

\[ \left[ \begin{array}{c}
\frac{e_{L_{1},k}^{2}}{\delta e_{L_{1},k}}
\end{array} \right] = \Psi_{1} \left[ \begin{array}{c}
\frac{e_{L_{1},k+1}^{2}}{\delta e_{L_{1},k+1}}
\end{array} \right] \tag{25}
\]

\[ \left[ \begin{array}{c}
\frac{e_{L_{1},k}^{2}}{\delta e_{L_{1},k}}
\end{array} \right] = \Psi_{1} \left[ \begin{array}{c}
\frac{e_{L_{1},k}^{2}}{\delta e_{L_{1},k}}
\end{array} \right] \tag{26}
\]

So the prediction part takes \( 8N \) multiply-accumulates.

6. Prewindowing Simplification

It is possible to choose \( S_{k} \) such that

\[ X_{L_{1},k} X_{L_{1},k}^{T} + S_{k} = (1 + \rho_{a}^{2}) \tag{27} \]

where \( S_{k} \) is the covariance matrix of size \( L \) for the prewindowing problem. In order to have forgetting, we would like to introduce an exponential weighting. Because the order and time dimensions are interchanged here w.r.t. the usual situation, one actually has to resort to data sequence weighting [10]. The projection matrix \( P_{X_{L_{1},k}} \) we had before is now replaced by

\[ P_{X_{L_{1},k}} = \left[ \begin{array}{c}
\frac{e_{L_{1},k}^{2}}{\delta e_{L_{1},k}}
\end{array} \right] X_{L_{1},k} \tag{28} \]

which is a \( N \times N \) submatrix of a projection matrix onto a \( L \)-dimensional subspace of prewindowed data vectors. If, with the data sequence weighting introduced, \( L_{1} \sim 1 \), then this submatrix is almost a projection matrix.

With the choice of \( S_{k} \) as in (27), we don’t have the simple relation between a priori and a posteriori error vector as we had in (7). The way to proceed now is to consider both parts of the error vector separately. This is immediately applicable to the USWC FTF algorithm. In its prediction part, (24) can be dropped now. So the joint-process and prediction parts of the USWC algorithm altogether take \( 8N \) operations with the prewindowing choice.

7. Further Discussions

7.1. Relation to Previous Work

It turns out that the USWC RLS algorithm has already been proposed as early as 1984 [11], where it was called the Affine Projection Algorithm (APA). This algorithm seems to have been largely confined to a community of Japanese researchers [12],[13]. The APA has a computationally complexity of \( O(N^{2}) + O(L^{2}) \). Perhaps the limited popularity of the USWC RLS algorithm is due to this complexity, which becomes comparable to \( O(N^{2}L) \) (conventional RLS) when \( L \) becomes comparable to \( N \). In [12], the Block Orthogonal Projection Algorithm (BOPA) was proposed, which minimizes the USWC criterion (with \( \mu = 0 \)), but only every \( L \) samples. This can be considered equivalent to running the USWC algorithm constantly for \( L \) samples. Again, the algorithm proposed in [12] is a non-fast algorithm, and furthermore, our USWC approach provides useful filter estimates also at the intermediate time instants during a sweep. In [13], the same USWC criterion is optimized, but at a rate intermediate between the APA and the BOPA rates.

In [14], a new adaptive filtering algorithm is proposed, based on conjugate gradient techniques. It turns out that the resulting algorithm is exactly equivalent to the USWC algorithm, but is again not fast. In [15], a stochastic gradient
technique is applied to the USWC criterion. The algorithm is made fast by exploiting divide and conquer techniques.

In the USWC algorithm with $n = 0$, the gradient has the direction of $e(t-1)X_{n-1}$. If we approximate this vector by $[c_{l-1}(k) \cdot \ldots \cdot c_{l-1}(k-N+1)]$, and introduce a stepsize, then we get exactly a stochastic gradient version of an instrumental variable technique. If we furthermore use a second stochastic gradient algorithm to generate the prediction errors $e_{l-1}(k)$, then we get the algorithm proposed [16].

Finally, a non-fast algorithm for the UGWC problem has been provided in [17]. We have proposed a fast alternative.

7.2. Computational Complexity

We propose to exploit a certain shift invariance structure in the problem to arrive at the UGWC and USWC FFT RLS algorithms. These algorithms have a complexity of $O(N^2) + O(L)$, which is similar to the complexity of some other FFT algorithms, but never exceeds the complexity of the SWC FFT algorithm. The cheapest one appears to be the USGC algorithm (with the prewinding approximation). The average complexity can be further reduced by leaving gaps between successive sweeps (data unused). This may lead to an algorithm that has an average complexity that is lower than that of the [8] algorithm, but still converges faster in badly conditioned circumstances [18].

7.3. Tracking Performance

When one compares the RLS and (N)LM algorithms for a variety of tracking problems, then either one can be the better one, depending on the characteristics of the problem. So the USWC RLS algorithm enlarges the scope of possible algorithms for a better match to the nonstationary characteristics of the tracking problem.

When one demands very high tracking capacity, the prewinded FFT algorithm with exponential weighting is not satisfactory (numerical stability considerations impose a lower bound on the weighting factor). One has to resort to the SWC FFT algorithm, which is also limited in flexibility however, due to the rectangular window of constant length. The underdetermined algorithms offer more flexibility: $\mu$ can be taken time-varying in the USWC and UGWC2 algorithms (this is not as straightforward as using a stepsize factor in the SWC algorithm to boost the gain, which makes the algorithm lose its LS nature), $L$ can be varied in the UGWC algorithms and one can restart at any desired time. Note that our UGWC algorithms with $\mu > 0$ are different from the GWC algorithm with a soft constraint initialization due to $\mu_0$.

In the FNTF algorithm of [5], the convergence dynamics are determined by the spectral variation of the covariance matrix of an order $L$ prewinded version of the input signal. Hence, with an AR(L) process as input, the convergence is pretty much insensitive to the input spectrum. In the under-determined FFT algorithms, the deterministic counterpart to the above stochastic consideration holds. If the input signal is persistently exciting of order at most $L$ (sum of $L$ exponential), then the underdetermined algorithms converge exactly in $L$ steps in the noiseless (perfect modeling) case.

An alternative to an algorithm with $\mu > 0$ is to use the algorithm with $\mu = 0$ and use coefficient filtering [18]. The algorithms with $\mu = 0$ provide the fastest tracking characteristics, leading to a substantial estimation noise component in the excess MSE. This can be reduced with a possibly time-varying trade-off between tracking speed and noise averaging.

References


CLASSIFICATION OF TEXTURE BY AN ASSOCIATION BETWEEN A PERCEPTRON AND A SELF-ORGANIZING FEATURE MAP

Eric MAILLARD*, Benoît ZERR†, Jean MERCKLE*

* I.R. P., 34 rue Marc SEGUIN, 68067 MULHOUSE-Cedex, FRANCE
† G.E.S.M.A., 29200 BREST NAVAL, FRANCE

- ABSTRACT - An original pattern classifier is presented. This model introduces the collaboration between supervised and unsupervised neural networks. A Multi Layer Perceptron is trained to classify patterns. A Feature Map is trained on the same data set, then its cells are labellized. Two kinds of uncertainty regions of the parameter space are detected: confusion regions and empty regions. When a pattern from one of these regions is presented, the Feature Map restrains the classification results of the Perceptron. Sigma-pi units are used in order to "mix" the answer of each constituent model to produce the final output of the global model. We show the efficiency of this paradigm through classification of textures.

1. INTRODUCTION

The process of pattern recognition consists roughly in two stages: feature extraction and classification [1]. The first step extracts useful information from raw data by mapping them on a feature space. This process is application dependant. The feature vectors are then classified by assigning them a class. Two ways of improving the efficiency of the classification are, first to banish wrong classification when the previous knowledge allows a right one, second to detect unknown patterns to avoid the risk of assessment error due to a lack of knowledge.

Neural networks are efficient classifiers [2], they are divided in two categories according to the data they need during learning. Supervised neural nets can form arbitrary association between input and output vectors given a training set of associations (input/output). After training they exhibit generalization capacities, so any input pattern raises an output signal. During unsupervised learning, neural nets cluster the input vectors according to a similarity or distance measure. So they are able to detect novelty when presented a test pattern.

The specific capacities of each approach are added in a new paradigm under name of collaboration. The first aim of this association is to increase the reliability of pattern classification using supervised neural net. After a description of the two neural nets, the mechanism of collaboration is presented and some simulations on texture classification are reported.

2. DESCRIPTION OF THE MODELS INVOLVED

A Multi-Layer Perceptron (MLP) is trained with the classical algorithm known as the backpropagation of the gradient of the error with addition of a momentum to the updating of the weights [3] [4]. The weights are updated according to:

\[ \Delta w_{ji}(t+1) = \eta(\delta_i o_j) + \alpha \Delta w_{ji}(t) \]

Thanks to the nonlinear activity within each cell, the MLP can form arbitrarily complex decision regions (figure 1).

![Figure 1: decision regions formed by a MLP.](image)

These regions are formed while trying to correctly classify a set of input-output pairs of
vectors. The internal representation developed by the MLP during this phase allows the network to generalize. Thus the design of the training set and the scheduling of the training phase are crucial.

An adaptive scheduling is introduced. It is performed automatically by varying the time constant $g$ of the activation function. The global error decrease is monitored. If the error stands still at a high value, then the time constant of the activation function is increased. The activation function is given by:

$$o_{pi} = \frac{1}{1 + e^{-net / g}}$$

A high value of the constant $g$ brings the output of the neuron closer to 0.5 (figure 2), this allows the weight change to be maximum.

While the MPL is trained with input-output pairs, the SOFM learns without any teacher. The main feature of the map is that it self-organizes in a way that matches the probability density function of the input patterns.

During the training phase an input is presented, the neurons of the grid compete with each other in a winner-take-all manner. Then the winning neuron (i.e. the best matching cell) and the ones in its neighborhood update their weights to get closer to the input vector in the input space. The convergence of the network is forced via the decrease of the learning rate.

As the euclidian distance is used as a proximity measure, we introduce an activation function which allows direct competition between the cells during both training and testing:

$$o_k = \frac{1}{1 + \sqrt{\sum_i (x_i - w_{ki})^2}}$$

As the neuron gets closer to the input pattern, its output signal tends towards 1.0. So during competition, the most active neuron is the one closest to the input pattern. The updating law for the winner and its neighborhood is:

$$w_i(t+1) = w_i(t) + \alpha(t)[x(t) - w_i(t)]$$

At the end of the training phase, the grid has "vector quantized" the input space. We use a rectangular grid, and an exponential decrease for both the learning rate and the neighborhood. Once the organization phase is over, the training patterns with addition of noise are presented again to the network and the cells are labeled by majority voting: they choose the more represented category among the pattern for which they win. If two or more categories are strongly represented, the cell is discarded. The remaining neurons can be viewed as prototypes for their classes (figure 3).

Since the network updates its weights after each presentation of a pattern, an augmented change of the weights "shakes" the network. The time constant then slowly decreases toward its previous value, this ensures the global stability of the network.

When training is over, we can expect the MLP to be efficient in its classification task if the input pattern is strongly correlated with one of the training set. But when a new pattern is presented, the classification is deficient. In the better case, it is ambiguous: the activations of the output units are rather low. In the worst case the activation of one output unit is strong while it is low for the others: the pattern lies far from any decision boundary but also far from any known pattern. Whatever the classification, it would be useful to detect the novelty of the input pattern.

Kohonen's self-organizing feature map (SOFM) [8], [6] is a competitive paradigm. The map consists in a grid of interconnected neurons. Each neuron is connected to its eight closest neighbors, and to all the input neurons.

Figure 2: The activation function for various values of $g$.

Figure 3: The SOFM matches the probability density function of the training set.
3. COLLABORATION BETWEEN THE MODELS

The aim of this collaboration is to give the MLP classification a confidence rate. The key issue is the possible openness of the decision regions formed by the MLP. The only assessment a MLP neuron can think of is: this pattern is on that side of my hyperplane. As the SOFM matches the probability density function of the input pattern, its neurons cluster where most of the training patterns are. The distance between a pattern and the closest neuron is an hint to the familiarity of that pattern. The collaboration of the two neural networks is shown in figure 4.

The resulting classification has three different meanings depending on the activation of the sigma-pi units:
- one sigma-pi unit shows a strong activation since the networks agreed on the classification.
- two or more sigma-pi units are activated: the perceptron have classified the pattern while the SOFM decided that the pattern was in an boundary region so the classification is uncertain
- no sigma-pi unit is activate: the SOFM has detected that the pattern is absolutely new so no prior knowledge allows the classification.

4. TEXTURE CLASSIFICATION

There is no universal and precise definition for the word “texture”. In fact, on a sea bottom image, an observer detects the limit between sandy surfaces and sand ripple areas as a change of “texture”, even though both are genuine sand. The texture features used for classification must be translation invariant. So instead of working on pixels grey levels, a preprocessing task extracts basic patterns from homogeneous texture areas from four different texture categories: sand, sand ripples, rocks and pebbles. (see figure 5). The methods used in the preprocessing stage are cooccurrence matrix and sum and difference histograms. The cooccurrence matrix is a set of second order spatial statistics for a given distance. Gagulowicz [7] shows the importance of those parameters in human visual system. Each element of the matrix is the proportion of cooccurrence of pairs of luminance when their pixel locations differ by a given displacement vector. As it would be difficult to use all matrix elements as features, we compute some expressions that synthesize the cooccurrence matrix values configuration: contrast, correlation, variance...[8].

As the forementioned method is time and memory resources consuming, we use an alternative to compute this matrix: the sum and difference histogram proposed by UNSER [9]. Those histograms represent, in a picture, the proportion of occurrence of grey level sum and grey level difference for a given pixel displacement vector. If the pixels pair is the realization of two random and independent variables the cooccurrence matrix can be compute from the sum and difference histograms. The reality is not absolutely so, but the experiment shows the validity of the approximation.

The training set is composed of 645 feature vectors manually labelled in four categories. We extract randomly ten exemplars of each categories and then train the neural nets independently on those 40 patterns. When the networks have converged, a test is carried out
using the 645 patterns. The result is shown in figure 6. When a test pattern is presented to the Feature Map, the network activates the prototypes that lies in the vicinity of the patterns. As shown in figure 6, the size of this vicinity modifies the performance of the final classification. When the size is too small, no prototypes are activate when the test pattern is slightly different from known exemplars. On the opposite, when the size is too big many prototypes from different categories are activated, inducing a confusion in the classification. But when the size is optimum, the performance of the model is promising. The network is able to detect wrongly labelled pattern in the test set.

5. CONCLUSION AND DISCUSSION

We presented here the collaboration of two neural networks as a new model of neural classifier. This model takes advantage of both a supervised and an unsupervised learning algorithm. The results on texture classification shows the efficiency of this new paradigm. The feature space determination has a key role in the design of a pattern recognition system. Future developments include an adaptive decision-making process with the use of previous performance results as an hint to the feature space discrimination potential. The discarded neurons of the self-organizing map will be used to address the major issue of texture boundary. Since they got mixed up between different texture patterns, those neurons may be used as boundary detectors.

REFERENCES


![Figure 6: Results of the classification of 645 patterns.](image)

![Figure 5: different textures from left to right: sand, sand ripples, rocks.](image)
TWO-LAYER LEARNING VECTOR QUANTIZER FOR COLOR IMAGE QUANTIZATION

C. Kotropoulos  E. Augé  I. Pitas

Department of Electrical Engineering, University of Thessaloniki, Thessaloniki 540 06, GREECE

A novel two-layer LVQ architecture is proposed which incorporates second order statistics in its training phase and allows training parallelism by splitting patterns into groups. The learning and recall procedures for the LVQ networks in the first and second layer are described. The proposed algorithm is based on statistical tests on the mean vectors and the dispersion matrices. A simplification based on a proximity test is presented. The application of the two-layer LVQ to color image quantization is also discussed.

1. INTRODUCTION

Neural networks is a rapidly expanding research field which attracted the attention of engineers and scientists in the last decade. One of the most prominent neural networks in the literature is the Learning Vector Quantizer (LVQ) [1,2]. It is an autoassociative nearest-neighbor classifier which classifies arbitrary patterns into p-many classes using an error-correction encoding procedure related to the competitive learning. It has already found extensive application for phoneme recognition [3,4] as well as in image processing, control and combinatorial optimization [2]. The dynamic weighting of input signals and a definition of neighborhood in the LVQ by a minimal spanning tree are proposed in [5]. The modification of the LVQ by using a differential competitive learning algorithm is discussed in [6].

The main contribution of this paper is the design of a novel two-layer LVQ architecture which incorporates second order statistics in its training phase and allows training parallelism by splitting patterns into groups. The proposed two-layer LVQ architecture is shown in Figure 1. It is comprised of N LVQ networks working independently in the first layer and a single LVQ network in the second layer. The training patterns of the first layer LVQ’s are input patterns. The training patterns of the second layer LVQ are the weight vectors of the first layer after the convergence of the first layer LVQ’s. The second layer classifies the weight vectors provided by the N networks of the first layer. Let us suppose that the N LVQ’s of the first layer classify q-dimensional data into p-many classes, then the second layer LVQ has q input nodes and N x p output nodes at most. Some of them have been trained by patterns extracted from the same population, therefore they must be merged. Some others are reference vectors associated with different populations, therefore they must be preserved. The incorporation of homogeneity and proximity statistical tests based on second-order statistics in the second layer LVQ learning algorithm is proposed in order to group partial results provided by the first-layer LVQ’s and considered in the evaluation of the final winner vectors. Thus, the proposed learning algorithm takes into account the presence of outliers and provides more accurate reference vectors for the clusters presented in the input patterns. Furthermore, the proposed two-layer LVQ architecture is easily parallelized and consequently makes faster computationally intensive tasks such as color image quantization and segmentation.

2. TWO-LAYER LVQ ALGORITHM

In the following, the learning and recall procedures of the first and second layer LVQ’s are described. The learning procedure of each first layer LVQ is the typical one of a multiple winner LVQ [2,3]. The recall procedure of each LVQ network in the first layer is applied only to the patterns used for the training of this network. It provides the necessary information about the sample mean vector and the sample dispersion matrix of the classes produced at the output of the network. Let \( \mathbf{V}_i = (v_{1i}, \ldots, v_{qi})^T \) \( i = 1, \ldots, p \) be the weight vectors for a first layer LVQ. The recall procedure of a network in the first layer has the following steps:

1. Initialize the sample mean vector \( \mathbf{m}_j \), the sample dispersion matrix \( \mathbf{S}_j \) and the number of patterns \( n_j \) associated with each class.

\[
\mathbf{m}_j(0) = 0_{q \times 1} \quad \mathbf{S}_j(0) = 0_{q \times q} \quad n_j(0) = 0 \quad j = 1, \ldots, p
\]  

(1)
2. Determine the class $C_g$ represented by the weight vector $V_g$ to which the training pattern $X(k)$ is most closely associated with.

\[ X(k) \in C_g \text{ if } \|X(k) - V_g\| = \min_{j=1}^{\gamma} \{\|X(k) - V_j\|\} \tag{2} \]

3. Increment the number of patterns belonging to $C_g$ by one and update the sample mean vector and the sample dispersion matrix of this class.

\[ n_g(k) = n_g(k-1) + 1 \]
\[ d_g(k) = X(k) - m_g(k-1) \]
\[ m_g(k) = n_g(k-1) + \frac{1}{n_g(k)} d_g(k) \]
\[ Q_g(k) = Q_g(k-1) + \frac{n_g(k-1)}{n_g(k)} d_g(k) d_g(k)^T \]
\[ S_g(k) = \frac{1}{n_g(k)} Q_g(k) \]  

For the remaining classes $j = 1, \ldots, p$ where $j \neq g$, the number of patterns, the sample mean and the sample dispersion matrix are not altered:

\[ n_j(k) = n_j(k-1) \quad m_j(k) = m_j(k-1) \quad S_j(k) = S_j(k-1) \]  

The LVQ network of the second class is used to find the input vectors which are candidates for merging or not. The criterion of minimum Euclidean norm used in the LVQ is not sufficient for the above-described task because it does not take into account the presence of outliers. Consequently, additional tests must be implemented in order to test the similarity between the weight vector provided by the first layer LVQ's and the winner vector determined by the second layer LVQ.

The following learning algorithm for the second layer LVQ is proposed:

1. Initialize randomly all the weight vectors $W_l = (w_{l1}, \ldots, w_{lp})^T$  \( l = 1, \ldots, p' \) where $p \leq p' \leq N_p$.

2. For each weight vector provided by the first layer LVQ's $V(k) = (v_1(k), \ldots, v_q(k))^T$:

   a. Find the closest weight vector of the second layer LVQ, i.e., the final winner vector $W_g(k)$ by using:

   \[ \|V(k) - W_g(k)\| = \min_{j=1}^{\gamma'} \{\|V(k) - W_j\|\} \]  

   b. If there is an output node of the second layer LVQ which has not been activated, i.e., there exists a free class:

   (i) Test the similarity between $W_g(k)$ and $V(k)$

   (ii) If $W_g(k)$ and $V(k)$ are proved similar, then merge them. The final winner is updated as LVQ suggests:

   \[ W_g(k+1) = W_g(k) + a(t)(V(k) - W_g(k)) \]  

   Modify the sample mean vector and the sample dispersion matrix. Let $n_g(k), m_g(k), S_g(k)$ denote the number of patterns, the sample mean vector and the sample dispersion matrix associated with the class of $W_g$ respectively. Let also $n_V, m_V, S_V$ be the corresponding quantities associated with the class of $V(k)$. The above-mentioned modifications are given by:

\[ m_g(k) = n_g(k-1)m_g(k-1) + n_Vm_V \]
\[ S_g(k) = n_g(k-1)S_g(k-1) + n_VS_V \]  

If $V(k)$ were previously merged with another class, say $C_j$, $m_j(k), S_j(k)$ would also be modified by replacing addition with subtraction in both the numerator and the denominator of (7), (8).

(iii) Otherwise, assign $V(k)$ to a free class, say $C_f$. Update $W_f(k)$ by using (6). Set the sample mean vector $S_f(k)$ and the sample dispersion matrix $m_f(k)$ of the free class as follows:

\[ m_f(k) = m_V \quad S_f(k) = S_V \]  

c. If there is no free class, merge unconditionally $V(k)$ and $W_g(k)$. Update the winner vector and modify the sample mean vector and the sample dispersion matrix associated with the class $C_g$ by using (6), (8).

3. Repeat step 2 for $t = 1, 2, \ldots$ until convergence is attained.

The recall procedure of the second layer LVQ is used for the classification of input patterns which have either been taken from the training set or not. It is used to determine the class $C_g$ represented by $W_g$ to which the input pattern $X(k)$ is most closely associated with.

\[ X(k) \in C_g \text{ if } \|X(k) - W_g\| = \min_{j=1}^{\gamma'} \{\|X(k) - W_j\|\} \]  

The homogeneity of the winner vectors evaluated by the LVQ in the second layer and the input weight vectors provided by the LVQ’s of the first layer can be tested (step 2.b.(i)) by employing statistical tests on the mean vectors as well as on the dispersion matrices. Let $\mu_g(k), \Sigma_g(k)$ denote the statistical mean vector and the statistical dispersion matrix associated with the class of $W_g$ respectively. Let also $\mu_V, \Sigma_V$ be the corresponding quantities associated with the class of $V(k)$. The homogeneity of the dispersion matrices $\Sigma_g$ and $\Sigma_V$ is tested by using the statistic [8] :

\[ T_1 = n_g |S_g^{-1}S| + n_V |S_V^{-1}S| \]  

\[ T_1 = n_g \ln |S_g^{-1}S| + n_V \ln |S_V^{-1}S| \]
where

\[ S = \frac{1}{n_u + n_v} \left( n_u S_u + n_v S_v \right) \]

(12)

and \( | \) denotes the determinant of a matrix. The statistic \( T_1 \) is distributed as \( \chi^2_{(u,v)} \). The following two cases are considered in order to test the homogeneity of the mean vectors:

1. Inhomogeneous dispersion matrices: A test statistic for the hypothesis that \( \mu_u, \mu_v \) are homogeneous is given by [8]:

\[ T_2 = (m_u - m_v)^T \left( \frac{1}{n_u} S_u + \frac{1}{n_v} S_v \right)^{-1} (m_u - m_v) \leq k_0 \]

(13)

The threshold \( k_0 \) in (13) can be approximately evaluated by the procedure described in [8].

2. Homogeneous dispersion matrices: A test statistic for the hypothesis that \( \mu_u, \mu_v \) are homogeneous is the following [7]:

\[ T_3 = |I_{u,v} + S_u^{-1} B|^{-1} \]

(14)

where

\[ S_u = n_u S_u + n_v S_v \]

\[ B = \sum_{i \neq j \neq \nu} n_i (m_i - m) (m_i - m)^T \]

\[ m = \frac{n_u m_u + n_v m_v}{n_u + n_v} \]

(15)

The statistic \( T_3 \) is approximately distributed according to the Wilks distribution \( \Lambda(x, n_u + n_v - 2, 1) \).

In many practical cases the above-described rigorous procedure is computationally demanding since it requires matrix inversion (although matrix inversion lemma [9] can be invoked) and the calculation of the determinant of a matrix. In addition, the number of matrices to be handled may be extraordinarily large as is the case in color image quantization to be discussed in the section 3. These problems can be alleviated by testing if there is any intersection between the hyperellipsoids associated with the winner vector of the second layer LVQ and the weight vectors provided by the first layer LVQ's. A simple proximity test of the form:

\[ \frac{|w_{ik} - v_{ik}(k)|}{\sqrt{S_{ik}} + \sqrt{S_{ik}}} \leq 1 \]

(16)

where \( S_{ik}, S_{ik} \) are the \( i \)-th diagonal elements of the sample dispersion matrices of the corresponding classes can be used in step 2.b.(i). Inequality (16) implies that the hyperellipsoids are approximated by hyperparallelpipeds and simply tests if there is overlap along the \( i \)-th dimension. If such an overlap exists along any dimension, it is inferred that \( V(k) \) and \( W \) are similar.

3. EXPERIMENTAL RESULTS

The two-layer LVQ architecture has been applied successfully to color image quantization both in serial and parallel implementations. Figure 2 shows an RGB color image of dimensions 256 \( \times \) 256 with 24 bits per pixel. Color image quantization aims at encoding each color pixel with one byte instead of three, thus reducing the number of RGB triplets to 256. The major drawback of the typical LVQ algorithm is the excessively large duration of the training phase due to the number of input pixels, i.e., color triplets to be considered. Therefore, parallel LVQ implementation is considered for speed-up. A straightforward parallelization (e.g., a parallel computation of the Euclidean distances) results in an implementation which has heavy communication load. If communication operations are slow, the implementation is very slow as well. On the contrary, if training parallelism is applied by splitting pixels into groups and the above-described novel LVQ architecture is used, a faster training procedure is attained. A two-layer LVQ having 16 networks in the first layer has been used. Each LVQ network of the first layer has 3 input nodes and classifies 1000 randomly selected pixels into 256 classes. The second layer LVQ receives 4000 weight vectors determined after four iterations of the 16 first layer networks. Six iterations have been shown adequate for the training of the second layer LVQ. The result of quantization is shown in Figure 3. The performance of the proposed algorithm has been compared to the one of the following algorithms:

1. standard LVQ (single-layer) [2,3] having 3 input nodes and producing 256 output classes
2. a modified LBG vector quantization algorithm which produces a codebook of 256 color vectors in 8 iterations [4]
3. nonuniform sampling of each color histogram by first equalizing each color histogram separately, uniformly sampling each equalized histogram in a number of predetermined samples and inverse transforming. Seven samples have been chosen for red channel, six for green channel and six for blue one. In total, 252 RGB triplets have been used.

In the comparative study outlined above, we have used as figures of merit the mean-squared-error (MSE) and the signal-to-noise ratio (SNR) measured in dB, given by:

\[ MSE = \sum_{i=1}^{M} \sum_{j=1}^{M} ||X(i,j) - \bar{X}(i,j)||^2 \]

\[ SNR_{dB} = 10 \log \frac{M \sum_{i=1}^{M} \sum_{j=1}^{M} ||X(i,j)||^2}{\sum_{i=1}^{M} \sum_{j=1}^{M} ||X(i,j)||^2} \]

(17)

where \( X(i,j) = (X_G(i,j), X_G(i,j), X_G(i,j))^T \) represents the \( (i,j) \) pixel in the original color image, \( \bar{X}(i,j) \) is a \((3 \times 1)\) vector which represents the \( (i,j) \) pixel in the quantized image and \( M \) is the number of rows/columns.
The results of the comparison are summarized in Table 1.

Table 1: Figures of Merit for Color Image Quantization

<table>
<thead>
<tr>
<th>Method</th>
<th>MSE</th>
<th>SNR</th>
</tr>
</thead>
<tbody>
<tr>
<td>Two-layer LVQ</td>
<td>156</td>
<td>-25.6</td>
</tr>
<tr>
<td>Single-Layer LVQ</td>
<td>207</td>
<td>-24.3</td>
</tr>
<tr>
<td>Non-uniform sampling of each color histogram</td>
<td>297</td>
<td>-22.8</td>
</tr>
<tr>
<td>LBG</td>
<td>3777</td>
<td>-11.7</td>
</tr>
</tbody>
</table>

It is seen that the proposed two-layer LVQ is superior to any other method used.

A two-layer LVQ which incorporates the proximity test (16) has also been implemented using 16 transputers T800 working at 20 MHz under HELOS operating system in farm topology. A speed-up of 3 has been observed between the parallel two-layer LVQ and a single-layer LVQ running in one transputer using the same number of pixels both in the learning and recall phase. A much larger speed-up would have been obtained, if other languages (e.g., OCCAM) supporting much faster communication had been used.

ACKNOWLEDGMENT

The first author is supported by a scholarship from the State Scholarship Foundation of Greece and the Bodosaki Foundation.

References


SELECTION OF HIDDEN LAYER NODES IN NEURAL NETWORKS BY STATISTICAL TESTS

Özer Çiftçioglu† and Erding Türkcan‡

A statistical methodology for selection of the number of hidden layer nodes in feedforward neural networks is described. The method considers the network as an empirical model for the experimental data set subject to pattern classification so that the selection process becomes a model estimation through parameter identification. The solution is performed for an overdetermined estimation problem for identification using nonlinear least squares minimization technique. The number of the hidden layer nodes is determined as result of hypothesis testing. Accordingly the redundant network structure with respect to the number of parameters is avoided and the classification error being kept to a minimum.

1 Introduction

In the last decade a number of feedforward neural network applications in many diverse areas is reported in the literature. The fast increase of the number of applications and application areas is still on the way and conspicuous. The underlying reasons for this trend can be viewed as the ease in implementations and the rather successful ensuing outcomes. It seems that the particular application in hand does not need to be modeled very much and the solution can be obtained by training from empirical data with little or no prior information about the application. In other words, in order to determine a sufficiently large training sample and an appropriate network architecture, a training algorithm is used to adapt the network parameters for an acceptable solution for implementation. However, it is seldom clearly stated what constitutes a sufficiently large training sample and an appropriate network architecture. The appropriate network structure essentially concerns the number of hidden layer nodes in a feedforward neural network. Viewing the complexity measures in classification problems for pattern recognition in the way as one considers entropy in communication and coding theory, a criterion for optimal number of hidden layer nodes might be of value to study using Shannon entropy [1]. Based on Shannon's entropy some results are reported in literature [2, 3]. In the following, a method for optimal number of hidden layer nodes, based on hypothesis testing is described to form optimal feedforward neural network architecture for a particular application in hand.

2 Pattern classification and parameter identification

Pattern classification takes an important place in the study of neural networks. Let us consider a collection of data of the form

$$\{(x_i^d, f_i^d) \mid 1 \leq i \leq N \}$$

(1)

where $x_i^d$ is the input vector forming a pattern; $f_i^d$ is the output vector corresponding to $x_i^d$. As result of pattern classification it is expected that a functional relationship is established so that

$$f(x_i) = f_i^d.$$  (2)

Here, one can easily see that such a problem is similar to classical interpolation and estimation problems, data points being replaced by a data vector as a pattern in function $f(x)$. The classification is accomplished by means of so-called classifiers of a linear form given by

$$\sum_{j=1}^{N} w_j x_j \leq \theta$$  (3)

where $w_j$ and $\theta$ are constants to be determined.

On the other hand, a neural network is composed of perceptrons. A perceptron is a simple computational element which is provided with some number of real quantities as input and it performs an affine linear transformation of those inputs of the form

$$\sum_{j=1}^{N} w_j x_j + \theta$$  (4)

with a patent computational similarity, referring to inequality (3) and eventual sigmoidal non-linearity of the form

$$O(x) = \frac{1}{1 + e^{-\sum_{j=1}^{N} w_j x_j + \theta}}.$$  (5)

Hence, it is clear that a perceptron can act as a classifier in a system for pattern recognition. In a neural network system, perceptrons are arranged in the form of layers so that the output of one layer is taken as the input for the following. The outputs of the last layer are weighted and summed together to produce the output of the network. A network may have multiple outputs providing different weightings and summations at the last layer. In order to surmise the potential performance of a neural network, Cybenko's theorem [4] is essential.

This implies the universality of neural networks with only one hidden layer. However, it makes no commitment about the number of nodes needed. In other words, given an arbitrary collection of input/output data and by using a network with adequate number of hidden nodes in a single hidden
layer, the network parameters can be found to achieve any desired modelling error criterion.

Feedforward neural network techniques for classification and recognition applications may be viewed as non-model based. That is, a mathematical model relevant to the classification problem is not required for the neural network approach to the problem. What is needed is a sufficiently large training data and an appropriate network architecture so that network parameters are adapted in a suitable manner for solution. However, the same problem can be considered as a parameter identification problem. Parameter identification involves the estimation of parameters for a model from measured experimental data points. The model may be based on the physical principles underlying a process or it may be based on empirical relationships which is the very relationship between the neural network input/output data, for example. Hence, in this respect, identification of the variables in a network structure turns out to be a model parameter estimation problem where the neural network is a structural model of the problem. The exact structure, i.e. number of hidden layer perceptrons, remaining to be determined.

In model parameter estimation, the estimated parameter values minimize some error functional that measures the difference between the experimental data and the model predictions from these observations. There are several forms of the error functional and a variety of techniques for identifying the optimum parameter set to minimize the functional. The form adapted for use in this work is the weighted least squares error functional. The weighted sum of squares of the deviations provides a measure of the discrepancy between the model estimation and experimental observations.

Let us denote that the experimental data point or 'demand' is \( y_i \) and the neural network output is \( Y(z_i^j, w) \) where \( w \) is the weight parameter vector and \( z_i^j \) is the pattern vector of the pattern number \( j \). In the least squares method, the minimization scheme is set up so that the function of 'merit' or 'cost'

\[
I(y, w) = \sum_j [y_i - Y(z_i^j, w)]^2
\]

is minimized with respect to the parameter vector \( w \) where \( y \) is the vector of demands \( y_i \) and \( Y(z_i^j, w) \) is the function that estimates the corresponding demand. Assuming that the demands \( y_i \) are experimental yields from measurement with statistical uncertainty \( \sigma_i \) in \( y_i \) as standard deviation, then for independent measurements, the function

\[
\chi^2(y_i, w_m) = \sum_{j=1}^{N} \frac{[y_i - Y(z_i^j, w_m)]^2}{\sigma_i^2}
\]

is a random variable with \( \chi^2 \) statistics, where \( w_m \) is the weight parameter vector at the minimum; \( N \) is the number of patterns introduced.

This statistic reveals whether the solution is consistent with experimental accuracy of the data, that is whether \( Y(z_i^j, w_m) \) and \( y_i \) are equivalent. If all demands \( y_i \) are obtained from the same instrumentation system, one can assume that \( \sigma_i = \sigma \) constant and Eq. 7 takes the form

\[
\chi^2(y_i, w_m) = \frac{1}{\sigma^2} \sum_{j=1}^{N} [y_i - Y(z_i^j, w_m)]^2.
\]

The solution should be consistent within the experimental accuracy which implies that the \( \chi^2 \) statistics should be less than or equal to \( N \). The most likely value is \( \chi = N - p \), where \( p \) is the number of parameters being estimated [5]. For the number of parameters \( p_1 \) and \( p_2 \), the variances are defined as

\[
\sigma_{1,2}^2 = \frac{1}{N - p_{1,2}} \sum_{j=1}^{N} [y_i - Y(z_i^j, w_{m1/2})]^2.
\]

(9)

Using Eq. 8, we write

\[
\chi^2_{1,2}(y_i, w_m) = \frac{\sigma_{1,2}^2}{\sigma^2} \nu_{1,2}
\]

(10)

with the respective degrees of freedom \( \nu_1 = N - p_1 \) and \( \nu_2 = N - p_2 \).

Considering that \( (N - p_1) \sigma_1^2 = (N - p_2) \sigma_2^2 \) is independent of \( \sigma_2^2 \) as is the case in linear least-squares estimation, we form from Eq. 10

\[
\chi_2^2 = \chi_1^2 - \chi_2^2 = \frac{(N - p_1) \sigma_1^2 - (N - p_2) \sigma_2^2}{\sigma^2}
\]

(11)

By the partition theorem for the \( \chi^2 \)-distribution [6], \( \chi^2_2 \) is independent of the variable \( \chi^2_1 \) and has a \( \chi^2 \)-distribution with \( \nu_2 = p_2 - p_1 \) degrees of freedom. Hence

\[
F = \frac{\chi_2^2}{N - p_2} = \frac{\chi_1^2 - \chi_2^2}{\nu_2}
\]

(12)

is a \( F_{\nu_1, \nu_2} \)-distributed variable with \( \nu_1 = N - p_1 \) and \( \nu_2 = N - p_2 \) degrees of freedom.

For the minimization use is made of a rapidly convergent variable metric method [7, 8, 9] according to Davidson. A brief description is given elsewhere [10].

3 Hypothesis testing

If the function \( Y(z_i^j, w_m) \) with the number of parameters \( p_1 \) is an appropriate representation of the demand \( y_i \) then \( E[\sigma_{p_1}^2] = E[\sigma_{p_2}^2] = \sigma^2 \), where \( E \) is the expectation operator. Thus determination of the model order represents a hypothesis testing, namely

\[
H_0: \sigma_{p_1}^2 = \sigma_{p_2}^2
\]

(13)

against the hypothesis

\[
H_1: \sigma_{p_1}^2 > \sigma_{p_2}^2
\]

(14)

where \( H_0 \) is the null hypothesis in the statistical terminology. This test cannot be carried out by considering the statistic \( \chi_2^2 \) from Eq. 11 as it includes the true variance \( \sigma^2 \), which is unknown. Therefore, we have to consider the \( F \) statistic given by Eq. 12, which has a \( F_{\nu_1, \nu_2} \)-distribution.

For the application of \( F \)-test a decision level \( l \) must be defined in advance, corresponding to type I error probability in the hypothesis testing, such that the null hypothesis is accepted if \( F \leq l \) and rejected if \( F > l \). The decision level \( l \) corresponding to the type I error \( \alpha \) is computed from

\[
\int_l^{\infty} f(F) \, dF = \alpha
\]

(15)
where \( f(F) \) is the probability density function of the \( F \)-variable.

In feedforward neural network, the number of parameters \( p_1, p_2 \) are related to the number of the hidden layer nodes \( m \), so that hypothesis testing is performed using \( m \) and \( m + 1 \) number of nodes.

4 Experimental investigation

Experimental investigation is performed by means of the data from the Borssele nuclear power plant. The Borssele power plant is a two-loop pressurized water reactor with nominal electrical power output of 477 MW. The on-line signal analysis system designed for the multi-level mode operation is capable of monitoring the plant states by tracking 32 DC and 32 AC signals simultaneously since 1983 [11]. In this study two data sets (in wide a range of operation) were selected for the demonstration of universality of the methodology:

Two data sets [3] were collected, one during a 9\( \frac{1}{2} \) hours of shutdown and another during normal operation with temporary power reduction for a certain duration and sampled with 8 s/s for AC and 1 s/s for DC. Afterwards the DC data are formed with a sampling rate of 1 s/minute to be used in this study. For the first set the corresponding electricity power varied from 375 MWe to 91 MWe and for the second set the power was reduced from 475 MWe to 146 MWe and after a certain lapse of time from 146 MWe to 475 MWe back again. From the 32 DC signals the set of selected signals used in this study are shown in Fig. 1 and indicated in Table 1.

![Diagram](image)

**Figure 1: Schematic representation of the nuclear power plant and the sensors used for the measurements.**

The first data set was obtained during the stretch out followed by normal shutdown. From the 32 available signals the most relevant ones are selected for the test. Table 1 gives the data sets and the selected signals in the neural network application. In the data set 1, a total number of 137 patterns are used out of possible 569 patterns with constant skipping factor of 3 during the training. The remaining 21 patterns are considered for extrapolation.

The second data set was obtained from a normal operation including considerable reduction of power for a certain time interval. In data set 2, a total number of 100 pattern are used out of possible 300 patterns with constant skipping factor of 2 during the training.

Several cases (Table 1) have been used studying the two different data sets. For data set 1, having 7 different channels information with relatively little correlated signals is used as an input to predict generated electrical power. The \( F \)-test described in the preceding section resulted in the number of hidden layer nodes as \( M=4 \) for 1000 iterations for Training using data set 1. In the statistical terminology, type I and type II errors are two important concepts [5]. In this work, the type I error probability adopted for the \( F \)-test is 0.01. The signals involved are shown in Table 1 case 4 (also see Fig. 1) and the results obtained after the recalculating procedure are given in Fig. 2. Increasing the number of hidden layer nodes to 32 (case 5) did not improve the prediction error. Increasing the data set up to 10 different channels of information the number of hidden layer nodes is found to be 4 again, as in the previous cases. Since the information is more compared to the preceding case, the prediction error is slightly smaller.

During normal operation there is a power reduction of 70% for a duration of about 2 h as shown in Figs. 3 and 4. The number of hidden layer nodes determined by the \( F \)-test is found to be \( M=4 \) (Fig. 3). The same experiment is carried out for \( M=32 \). Also in this case the differences between the data for training and the predicted values are not improved (Fig. 4), even they are slightly worsened.

<table>
<thead>
<tr>
<th>Signal description</th>
<th>Case 1</th>
<th>Case 2</th>
<th>Case 3</th>
<th>Case 4</th>
<th>Case 5</th>
<th>Case 6</th>
</tr>
</thead>
<tbody>
<tr>
<td>input</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>output</td>
<td>GEP</td>
<td>GEP</td>
<td>GEP</td>
<td>GEP</td>
<td>GEP</td>
<td>GEP</td>
</tr>
<tr>
<td>Figures</td>
<td>2</td>
<td>3,4</td>
<td>4, 5</td>
<td>6</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

Table 1: Selected channels and calculated cases
5 Discussion and Conclusions

Indeed, the neural networks patently suitable for using as classifiers. Since the data applied to neural network are essentially for pattern recognition, the network is in a way an appropriate empirical model for the model parameters, i.e. weights subject to identification. In this respect hypothesis testing yields the adequate number of weights through the number of hidden layer nodes so that the true network dimensionality can be identified. The excessive number of weights beyond those required by the true network's modelling structure are simply redundant and they do not add to the network dimensionality at all. This conclusion corroborates with the experimental results which were obtained using the same data but different number of nodes at the hidden layer determined by the hypothesis testing procedure as well as conventional empirical ones, e.g., see [2]. Indeed, as a matter of fact in the overdetermined case, the network performance is slightly better as one should expect with reference to classical parameter estimation problem. In the underdetermined case the error of recognition for pattern recognition is relatively higher, the network performance becomes finally poor in this case.

Hence it is to conclude that the hypothesis testing is an important tool to identify the number of hidden layer nodes in a feedforward neural network structure. Additionally, as result of the optimization process, the errors estimates of the network parameters are obtained together with the correlations between the parameters through the inverse second derivative (Hessian) matrix. This additional information is an important aid for further statistical analyses on neural networks.

References

CONVERGENCE ANALYSIS OF THE HAMMING NETWORK

K. KOUTROUMBAS and N. KALOUPSIDIS
UNIVERSITY OF ATHENS, DEPT. OF INFORMATICS
PANEPISTIMIOPOLIS, TYPE BUILDINGS, ATHENS 157 71, GREECE.

ABSTRACT. In this paper results concerning the convergence of the parallel and asynchronous modes of the Hamming Network are presented. Also, an upper bound of the maximum number of steps that are required to reach a stable state is given.

I. Introduction

Neural Networks have drawn considerable attention in the recent years mainly because of their interesting learning abilities. Many Neural Networks architectures have been proposed during these years but all of them can be classified in two major categories: feedforward Neural Networks and recurrent Neural Networks. The latter category includes the Hamming Network which is the objective of the present work.

One interesting application of the Hamming Network is minimum distance decoding. Also, the Hamming Net may be employed as a lateral inhibition mechanism in competitive learning architectures such as the Kohonen’s Self-organizing map and the Counter Propagation Network (see for example [4], [5]).

A typical architecture for the Hamming Network is illustrated in figure 1 (see also [1]). The Hamming Net consists of two subnets. We assume that the patterns are presented as inputs in the lower subnet are bipolar vectors of length N with values -1 and +1. The lower Net has M nodes each one corresponding to one of M exemplar patterns.

The goal of the lower subnet is to compute M matching scores that are equal to N - d_i, i = 1, . . . , M, where d_i is the Hamming distance (Hamming distance of two patterns is the number of bits the patterns differ) between the input pattern and the i-th exemplar pattern. These scores are integers satisfying:

0 \leq N - d_i \leq N, i = 1, \ldots, M.

The scores obtained by the lower subnet are then forwarded to the upper Network. The goal of this Net is to pick up the maximum matching score, by forcing all matching scores corresponding to all nodes to zero, except the matching score that corresponds to the node with the maximum initial matching score. The output of the upper set of nodes can then be used for further processing (the complete algorithm can be found in [1]).

The Network will operate until a desired stable state is reached, that is, until one node possesses a positive value and the remaining nodes become zero. It is stated in [1] that the Hamming Net always finds a node with a maximum value when \( \varepsilon < \frac{1}{M} \). It turns out that this is not always the case. A more delicate situation arises for example, when there exist two initially equal and maximum matching scores. A complete convergence analysis of the Hamming Net is discussed in the following sections. More specifically, we introduce a deterministic extension of the operation of the Hamming Net which enables us to deal with both parallel and asynchronous modes. We give some results concerning the convergence properties of the generalized Hamming Net. An upper bound for the number of steps required in order to obtain a solution is derived.
II. Basic definitions and theorems

From now on we will refer only to the upper subnet of fig. 1. As can be seen from 1-b step of the Hamming algorithm, the updating at each step of the algorithm is fully parallel i.e., all nodes are updated simultaneously. We next define two new modes of operation, the partially parallel and the serial or asynchronous mode of operation. In the partially parallel mode of operation only a fraction of the set of nodes is updated simultaneously. In the serial mode of operation each node is updated asynchronously with the others. From now on we will refer to the partially parallel mode of operation and we will consider the other modes as special cases of the partially parallel mode of operation. Before we proceed we will define here the time step and the updating cycle. Time step is the time interval in which an updating of a specific group of nodes takes place and updating cycle is the number of time steps needed for the completion of one update of all nodes. In more specific terms let us consider fig. 2. The set of nodes is divided in \([M/\lambda]\) groups (where \(\lceil r \rceil\) denotes the integer part of \(r\)) each one containing \(\lambda\) nodes, \(1 \leq \lambda \leq M\). At each updating cycle we first update the first group, then the second group and so on. With \(t\) we shall denote the updating cycle and with \([k/\lambda]+1\) the time steps needed for the update of the \(k-th\) node in a given updating cycle. Thus, \(\mu_k(t, l)\) denotes the state of the \(k-th\) node at the time cycle \(t\) and at time step \(l\) in the \(t-th\) cycle. The term \([k/\lambda]+1\) denotes the time steps that must be spent in a specific time cycle in order the updating of the \(k-th\) node takes place.

**Example:**

Let \(\lambda = 5\) and \(M = 20\). Then \(\mu_2(2, 4)\) denotes the state of node 3 at the time cycle 2 and at time step 4 in the time cycle 2. \(\mu_9(2, [\frac{3}{\lambda}]+1)\) or \(\mu_2(3, 3)\) denotes the state of node 9 at time cycle 2 and at time step 3 in the time cycle 2. At this time step, node 9 will be updated for the time cycle 2. We set:

\[
c_k = \left[\frac{k}{\lambda}\right] + 1, \quad 0 \leq k \leq M - 1.
\]

In order to simplify notation we will write \(\mu_k(t)\) instead of \(\mu_k(t, c_k)\). Note that knowing \(k\) we can easily find that at the \([k/\lambda]+1\) time step of updating cycle \(t\) the node \(k\) will be updated. For example, if \(M = 20\) and \(\lambda = 5\), \(\mu_3(t)\) is equal to \(\mu_3(t, [\frac{3}{\lambda}]+1)\) or \(\mu_3(t, 2)\). Note that \(c_k\) is independent of the time cycle \(t\). To facilitate the analysis we shall make three simplifications which however can be removed. First, we divide the set of nodes in groups, each one containing the same number of nodes. Second, we assume that \(\lambda\) divides \(M\). Finally, the nodes are grouped in their natural ordering. Alternate groupings can be accommodated via permutations of the nodes without affecting subsequent analysis. On the basis of the above discussion, the updating equation of the upper Hamming subnet becomes:

\[
\mu_k(t + 1) = f[\mu_k(t) - \varepsilon \sum_{m=0}^{[k/\lambda]A-1} \mu_m(t + 1)
- \varepsilon \sum_{m=[k/\lambda]A, m \neq k}^{M-1} \mu_m(t)]
\]

with \(k = 1, 2, \ldots, M\) and \(t = 1, 2, \ldots\). \(f\) is defined as:

\[
f(x) = \begin{cases} 
0, & x < 0 \\
x, & 0 \leq x \leq N \\
N & x < N
\end{cases}
\]

Eq. (1) states that in order to update node \(k\) at time cycle \(t+1\) we use the updated values up to \([k/\lambda]\) nodes and all the values of the remaining nodes from the previous time cycle \(t\). Note that for the nodes that belong to the same group with \(k\), contribute with their value at time \(t\) in the above updating equation. In the sequel we shall assume that the network parameter \(\varepsilon\) is chosen to satisfy the following bound:

\[
\varepsilon < \rho(A, N, 1)
\]
where

\[ \rho(A, a, b) = \begin{cases} \frac{1}{(M-1)n}, & A = M \\ \frac{b}{A |A \setminus (M-1)a|}, & A < M \end{cases} \]

Let \( G \) denote the set of sets that initially have at least one node with maximum value. Let \( G \in \mathcal{G} \) denote the set that is updated last.

Let \( T_k \) be defined as

\[ T_k = \sup \{ t : \mu_k(t) \neq 0 \}, \quad k = 0, 1, \ldots, M - 1. \]

Proposition 1: If \( T_k \) is finite then \( \mu_k(t) = 0 \), for \( t > T_k \).

Let

\[ \Lambda_k = \lim_{t \to \infty} \mu_k(t). \]

Let \( i_1 < i_2 < \ldots < i_p \) such that

\[ \mu_{i_1}(0) = \ldots = \mu_{i_p}(0) = \max_{i} \mu_i(0, i) \].

Assuming that \( G \) contains \( i_p \) and no other maximum, the following hold:

Theorem 1: \( \Lambda_k = 0 \), for \( k = 0, \ldots, M - 1 \) and \( k \neq i_p \) and \( T_k \) is finite. Moreover, the limit \( \Lambda_{i_p} \) is attainable after a finite number of steps and satisfies \( \Lambda_{i_p} > 0 \).

The next Theorem shows that if there are two or more initial maxima in \( G \), then all nodes of the net stabilize to zero after an infinite number of steps.

Assume that \( \mu_{i_1}(t), \ldots, \mu_{i_p}(t) \) (with \( q < p \)) belong to \( G \) and initially have the maximum value.

Theorem 2: It holds

\[ \Lambda_{i_1} = \ldots = \Lambda_{i_p} = 0. \]

Moreover, these limits are attained in infinite number of steps.

Remark: The asynchronous case results when \( A = 1 \) whereas the parallel case occurs for \( A = M \). In the latter case the bound \( \frac{1}{A} \) for \( \varepsilon \) slightly improves the condition \( \varepsilon < \frac{1}{A} \) reported in [1].

Theorem 3: An upper bound for the number of steps required to obtain the solution, when only one maximum belongs to \( G \), is:

\[ t_3 = \max \left[ \frac{\ln(\frac{\mu_{i_p}(0)}{\Delta \mu_{i_p}(1)})}{\ln(1 + \varepsilon)} + 2, \right. \]

\[ \frac{\ln(\frac{\mu_{i_p}(0) - K}{\Delta \mu_{i_p}(1) - K})}{\ln(1 + \varepsilon)} + 2 \]

where

\[ K = \varepsilon^2 (M-1) N(\frac{M-1}{A}) + 1. \]

It is worth mentioning here that in the fully parallel mode of operation, the Hamming Net will converge after exactly

\[ \ln(\frac{\Delta \mu_{i_p}(t_3 - 1)}{\Delta \mu_{i_p}(1)}) \frac{1}{\ln(1 + \varepsilon)} + 2 \]

steps. An upper bound for the number of steps in that case is

\[ \ln(\frac{\mu_{i_p}(0)}{\Delta \mu_{i_p}(1)}) \frac{1}{\ln(1 + \varepsilon)} + 2. \]

III. Conclusions and remarks

Summarizing the above mentioned facts we have that if \( 0 < \varepsilon < \rho(A, N, 1) \) and the group of nodes \( G \) that is updated in one time step contains only one node with the maximum initial value and updated after all other nodes with maximum initial value belonging to other groups, the Hamming Net will converge in a finite number of steps to a nonzero solution. Furthermore the limiting values of all nodes will be zero except the limiting value of the node in \( G \) with the maximum initial value, which will be positive. An upper bound for the number of steps required has been obtained.

If the group \( G \) contains more than one maxima then the values for all nodes asymptotically approach zero.
Further insight into the above results is provided by the following geometric interpretation. We note first that the group of nodes that controls the solution is group $G$. The values of nodes in $G$ are represented by a point in $R^A$ space and follow a trajectory as time passes. If the initial point is not in one of the hyperplanes that bisect the angles between the reference hyperplanes $\{(x_1, \ldots, x_A) : x_1 = \ldots = x_{i-1} = x_{i+1} = \ldots = x_A = 0, i = 1, \ldots, A\}$ then the limit point will be on the axis that corresponds to the node $i_p$, i.e. $\{(x_1, \ldots, x_A) : x_1 = \ldots = x_{i_p-1} = x_{i_p+1} = \ldots = x_A = 0, x_{i_p} \neq 0\}$. In the opposite case, the limit vector will be the $(0, 0, \ldots, 0)$ point. If $A = 1$ we always get a solution since group $G$ will contain only one node with maximum initial value.

If $A = M$ and there exist two or more nodes with maximum initial value, then the limiting point is zero.

It follows from the above discussion that if we are interested in the speed of convergence, large $A$ are preferable. On the other hand, if we are interested in taking an accurate solution in most of the cases, a small $A$ is better off.

It is also important to note that the above analysis concerns only the upper subnet. The only condition that involves parameters of the lower subnet is eq. (3). Our results become independent of the lower subnet, if eq. (3) is replaced by where $b$ is the maximum possible initial value to the upper subnet and $\alpha$ is the minimum possible difference between the values of two nodes.

References


Modes of the Hamming Network", under review.


Fig 1: The Hamming Network Architecture

Fig 2. The grouping of the nodes.
A novel algorithm for training a bidirectional associative memory

Marina MEILA PREDOVICIU, Petre STOICA

Department of Automatic Control, Polytechnic Institute of Bucharest
Spl. Independentei 313, 77206 Bucharest, Romania

A training algorithm for discrete-time bidirectional associative memories (BAMs) is presented. It ensures the storage of the desired items at stable equilibrium points of the network. The algorithm is proved to work for a broad class of nonlinearities, provided the desired memories lie in the corners of the state-space. Some necessary conditions for the existence of a solution are outlined, and simulations illustrate the algorithm's performance.

1. INTRODUCTION

A key problem in designing neural associative memories is finding the connection weights matrix which ensures the storage of the desired items at stable equilibrium points of the network.

The outer product method [1,3] accomplishes this task in a direct and easy way. However, as it offers only a storage capacity of about 15 percent of the number of neurons in the net [1,6] and does not offer stability of the stored equilibrium points, it has been replaced by methods which allow the storage of a greater number of items [4,5] and the control of the domain of attraction of each stored state [2,7]. Except for [4], these works deal with Hopfield memories [1], an important special case of associative memory.

This paper focuses on bidirectional associative memories. A new algorithm is devised, which improves significantly the storage capacity and ensures the desired minimal radius for the domain of attraction of each stored item. Section 2 introduces the BAM model and the encoding problem. Section 3 presents the training algorithms and Section 4 extends it to networks with neurons having a bounded, nondecreasing activation function. In Section 5, some necessary conditions for the existence of a solution are deduced, while Section 6 shows the results of simulations.

2. THE BAM MODEL

The BAM model used in the following has been introduced by Kosko [3]. The memory is built up of two layers containing \( n_x \) and \( n_y \) neurons respectively. Each neuronal output can take only two values, let them be \((-1, +1)\). The outputs of all the neurons in one layer are connected to the inputs of each neuron in the other layer and vice versa.

If we denote by

\[
X(t) = [X_1(t), X_2(t), \ldots, X_{n_x}(t)]^T
\]

\[
Y(t) = [Y_1(t), Y_2(t), \ldots, Y_{n_y}(t)]^T
\]

the vectors of the neurons outputs for the two layers at time \( t = 0,1,2, \ldots \) then the evolution of the network is given by the following equations:

\[
X(0) = X_0 \quad \text{(1)}
\]

\[
Y(t-1) = \text{Sgn}(W^TX(t)) \quad \text{(2)}
\]

\[
X(t+2) = \text{Sgn}(WY(t+1)) \quad t = 0,1,2, \ldots
\]

where \( W \) is the \( n_x \times n_y \) connection weight matrix and \( \text{Sgn} \) is the signum function applied to each component of its vector argument.

\[
\text{Sgn}(z) = \begin{cases} 
  1 \text{ if } z > 0 \\
  -1 \text{ if } z < 0 
\end{cases} \quad \text{(3)}
\]

If a component of the arguments in (2) is equal to \( 0 \), then the corresponding neuron output is not modified. In this context \( \text{Sgn} \) is called activation function.

The pair \( (X(t), Y(t)) \) is called the BAM state at time \( t \).

It has been shown [3] that from any initial state the network (2) evolves in a finite time to an equilibrium state where no more changes in either \( X \) or \( Y \) occur. The set of all equilibrium states for a BAM is determined by its weight matrix \( W \).

Hence, the problem of encoding some
desired items in a BAM can be stated as follows:

Being given the set of pairs \((\rho^+, \eta^+), \mu_1, \ldots, \mu_p\), \(\rho^- \in (1, -1)^{n_x}, \eta^- \in (1, -1)^{n_y}\), find a matrix \(W = (w_{ij})\) such that the convergence condition \((CC)\) below holds:

\[ d_X + d_Y > 0 \quad \text{(CC)} \]

for each \(X\) and \(Y\) with

\[ d_X(X) \leq d_Y(X) \quad \text{and} \quad \mu = 1, \ldots, p. \]

In \((4)\), \(d\) denotes the Hamming distance.

\[ d(X, Y) = \frac{1}{2} \sum_{i=1}^{n_x} | X_i - Y_i | \quad \text{and} \quad |X| = \frac{1}{2} \sum_{j=1}^{n_y} | Y_j | \]

and \(f_x, f_y\) are two positive integers - the desired radii of the domains of attraction.

It should be noted that along with the pair \((\rho^+, \eta^+)\), the pair \((\rho^-, \eta^-)\) is stored as well. Note also that \(W\) in \((2)\) can be formed in a similar way without any restriction. Let us denote by \(W\) the set of \((n_x n_y)\) matrices \(W\) satisfying \((CC)\).

**Lemma 1:** Whenever the set \(W\) of matrices satisfying \((CC)\) is nonempty, there exists an element \(W^*\) of \(W\) such that \([W^*]_{ij}\) are integer-valued.

**Proof:** Obviously, \(W\) is an open set. Suppose it is not empty. Since \(W = (n_x n_y)^2\), there exists a \(W^*\) in \(W\). Note that \(W^*\) is in \(W\) for every \(W\) in \(W\) and for every \(\omega_\in\omega\). Then, by choosing \(W = N W\) where \(N\) is a common multiple of the denominators of \(W_{ij}\) for all \(i\) and \(j\), the lemma is proved. 

3. THE TRAINING ALGORITHM

The following training procedure is an extension of the training with noise procedure introduced by Gardner et al. [2] for Hopfield networks.

**Training Algorithm 1**

For \(k = 0 : W(0) = \sum_{\mu=1}^{\mu=1} \rho_{\mu}^+ \eta_{\mu}^- \quad \text{(6)}\)

For \(k = 1, 2, \ldots\) repeat

1. Choose at random \(X\) and \(Y\) satisfying \((4)\) for some \(\mu = 1, \ldots, p.

2. Compute

\[ \epsilon_i = \operatorname{diag} \{ \xi_i^+, i = 1, \ldots, n_x \} \]

\[ \epsilon_j = \operatorname{diag} \{ \xi_j^+, j = 1, \ldots, n_y \} \]

\[ \xi_i = \{ 0 \text{ if } (\rho_i^+ \eta_i^-) > 0 \}
\]

\[ 1 \text{ otherwise} \]

\[ \xi_j = \{ 0 \text{ if } (\rho_j^+ \eta_j^-) > 0 \}
\]

\[ 1 \text{ otherwise} \]

3. \(W(k) = W(k-1) + \Delta W(k) \)

\[ \Delta W(k) = c \epsilon_i \epsilon_j^T + X_k \epsilon_i \]

until \(W(k)\) satisfies \((CC)\).

**Theorem 1:** If \(W\) is nonempty, then the training algorithm converges, regardless of the value of \(W(0)\), to a \(W^*\) in \(W\) in a finite number of steps.

**Proof:** Suppose there exists some \(W^*\) to satisfy \((CC)\). We will show that all \(\epsilon\) and \(\epsilon^*\) can take nonzero values only for a finite number of steps. Let us define the scalar product and corresponding norms on the set of real \((n_x n_y)\) matrices:

\[ \langle W, V \rangle = \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} w_{ij} v_{ij} \]

\[ ||W|| = \langle W, W \rangle^{1/2} \]

Since there is a finite number of inequalities to be satisfied in \((CC)\), there exist \(\gamma_0\) and \(\delta > 0\) such that

\[ 0 < \gamma_0 \langle v_i, v_j \rangle < \delta ||W|| \]

\[ \forall k = 1, \ldots, n_x \]

\[ \forall j = 1, \ldots, n_y \]

Then, at each step \(k\)

\[ \langle W, v_i \rangle = \sum_{j=1}^{n_x} v_{ij} W_{ij} \]

\[ \langle W, v_i \rangle = \sum_{j=1}^{n_y} v_{ij} W_{ij} \]

\[ \langle W, v_i \rangle = 0 \quad \forall k = 1, \ldots, n_x \]

\[ \langle W, v_j \rangle = 0 \quad \forall j = 1, \ldots, n_y \]

In the above we omitted the superscript \(\mu\) and we made the notations

\[ e_i = \sum_{j=1}^{n_y} v_{ij} \]

\[ e_j = \sum_{i=1}^{n_x} v_{ij} \]

\[ e = \sum_{i=1}^{n_x} e_i \]

\[ e' = \sum_{j=1}^{n_y} e_j \]

\[ S(N) = \sum_{i=1}^{n_x} e_i (e_i - e'_i) \]

By summing \((7)\) and \((8)\) respectively for \(k = 1, \ldots, N\) we obtain

\[ \langle W, W(N) \rangle > \langle W, W(0) \rangle + \delta ||W|| S(N) \]

\[ ||W(N)||^2 < ||W(0)||^2 - 2 \max(n_x, n_y) S(N) \]

Then

\[ -||W||S(N) > \delta S(N) \quad \text{(const)} \]

The left term of \((9)\) is obviously less than \(1\); meanwhile, the right term tends to infinity as \(S(N) \to \infty\). Hence \(S(N)\) must be bounded, which implies that

\[ e_i = e'_i = 0 \quad \forall k \geq N, q.e.d. \]

Remark: Given the initialization in \((6)\), the above algorithm will only produce integer-valued \(W\) matrices. Thus, the algorithm will converge to an integer-valued \(W^*\) in \(W\) (according to Lemma 1, there always exists such a \(W^*\)).

The rule the algorithm uses to adapt the weights is a Hebb-type rule: to the connection weight between two neurons a positive quantity (equal to 1) is added if the desired states for the two neurons corresponding to the given input and output
patterns have the same sign (and thus the two neurons should mutually activate themselves); otherwise, a negative quantity is added.

4. GENERAL ACTIVATION FUNCTION

Let us replace now the Sgn function in (3) by a function \( \sigma \) satisfying

\[
\sigma(-\infty,0) = [-1,1] \\
\sigma(0,\infty) = [1,1]
\]

(10)

Then, equations (2) become

\[
Y(t+1) = \sigma(WX(t)) \tag{11}
\]

and the network state space will fill the hypercube \([-1,1]^N \times [-1,1]^N\).

Such a network is generally not globally stable. But we will show that if the pairs \((\mathbf{e}^i, \mathbf{n}^\mu)\) are corners of the state space, i.e.,

\[
|e_{j}^{i}| = |n_{j}^{\mu}| = 1 \quad \text{for all } i,j,\mu
\]

(12)

and the training set is given by

\[
T_X = \{ X_m = \{-1,1\}^N, \exists \mu \ d(X_m,\mathbf{e}^i) \leq f_X \}
\]

(13)

\[
T_Y = \{ Y_m = \{-1,1\}^N, \exists \mu \ d(Y_m,\mathbf{n}^\mu) \leq f_Y \}
\]

then the training algorithm will create stable equilibrium points with attraction domains defined by \(f_X\) and \(f_Y\) near the given pairs. In (13) and in the following, the distance \(d\) defined in (5) generalizes the Hamming distance on the two spaces.

Since \(\sigma\) may not reach its extreme values for finite input, we shall admit a tolerance \(\Delta \theta\) for the accuracy of the retrieval. From (13), there exists \(N_\theta\) such that

\[
|\sigma(x)| > 1 - \Delta \text{ for } |x| > N_\theta \tag{14}
\]

The convergence conditions (CC) should be rewritten

\[
\mathbf{e}^i \circ W \mathbf{y} > N \quad \text{(CC)}
\]

for \(X \in T_X, \ Y \in T_Y\).

The algorithm will be modified as follows:

**TRAINING ALGORITHM 2**

For \(k = 0 : W(0) = \sum_{\mu=1}^{p} \mathbf{e}^i \mathbf{n}^{\mu}\)

For \(k = 1,2,\ldots\) repeat

1. Choose at random \(X\) and \(Y\) satisfying \((13)\)

2. Compute

\[
\mathbf{e}' = \text{diag}(\mathbf{e}'_j, j = 1,\ldots,n_Y)
\]

\[
\mathbf{e}'' = \text{diag}(\mathbf{e}'_j, j = 1,\ldots,n_Y)
\]

\[
\mathbf{e}_i = \begin{cases} 
0 & \text{if } (\mathbf{e}^i \circ W \mathbf{y}) > N \\
1 & \text{otherwise}
\end{cases}
\]

\[
\mathbf{e}'_j = \begin{cases} 
0 & \text{if } (\mathbf{e}'' \circ W \mathbf{x}) > N \\
1 & \text{otherwise}
\end{cases}
\]

3. \(W(k) = W(k-1) + \Delta W(k)\)

\[
\Delta W(k) = \epsilon_X \mathbf{e}^i \mathbf{y}^T + \epsilon_Y \mathbf{e}'' \mathbf{e}''
\]

until \(W(k)\) satisfies \((CC)\).

**Theorem 2** If \(W\) is not empty and \(\sigma\) satisfies \((10)\), then the Training Algorithm 2 converges in a finite number of steps, for any value of \(W(0)\).

**Proof**

Take \(W(\infty)\), \(W(\infty)\), and \(W(\infty)\). Then \(W\) satisfies \((CC)\). Further the proof follows that of Theorem 1.

**Theorem 3** If \(W\) satisfies \((CC)\) for every \(X\) in \(T_X\) and every \(Y\) in \(T_Y\), then \(W\) satisfies \((CC)\) for every \(X\) and \(Y\) for which \(d(X,\mathbf{e}^i) \leq f_X\) and \(d(Y,\mathbf{n}^\mu) \leq f_Y\) for some \(\mu\).

**Proof**

\[
B(\mathbf{e}^i, f_X) = \{ X_m = \{-1,1\}^N, d(X_m,\mathbf{e}^i) \leq f_X \}
\]

\[
B(\mathbf{n}^\mu, f_Y) = \{ X_m = \{-1,1\}^N, d(X_m,\mathbf{e}^i) \leq f_Y \}
\]

\[
B(\mathbf{e}^i, f_X) \text{ is a simplex and } B(\mathbf{n}^\mu, f_Y) \text{ is the set of its extreme points. Since } B(\mathbf{e}^i, f_X) \text{ satisfies the linear inequalities } (CC) \text{, it follows from the Krein-Milman theorem that all } X \in B(\mathbf{e}^i, f_X) \text{ satisfy } (CC). \text{ A similar reasoning for } Y \text{ completes the proof. See } (8) \text{ for mathematical background and } (9) \text{ for the detailed proof.}
\]

5. BOUNDS ON THE ATTRACTION DOMAINS

The above algorithm converges to a solution of the associative encoding problem, provided such a solution exists. In the following, some direct necessary existence conditions are given.

A. From the fact that the domains of attraction of each two stored states should be disjoint, it follows that:

\[
2f_X \leq d(\mathbf{e}^i, \mathbf{e}^j) \leq n_\xi - 2f_X - 1 \tag{15}
\]

\[
2f_Y \leq d(\mathbf{n}^\mu, \mathbf{n}^\nu) \leq n_\eta - 2f_Y - 1
\]

B. For the compatibility of system \((CC)\), it is necessary that

\[
f_X < \frac{1}{2} \left( 1 - \sqrt{1 - d/n} \right) \tag{16}
\]

where \(f,n\) stay for either \(f_X,n_X\) or \(f_Y, n_Y\) and

\[
d = \min \{ d(\mathbf{e}^i, \mathbf{e}^j), d(\mathbf{e}^i, \mathbf{n}^\mu), d(\mathbf{n}^\mu, \mathbf{n}^\nu) \}
\]

or

\[
d = \min \{ d(\mathbf{e}^i, \mathbf{n}^\mu), d(\mathbf{n}^\mu, \mathbf{n}^\nu) \}
\]

respectively. For the demonstration of this condition see \((10)\).

![Figure 1](image1.png)

Figure 1 Upper limit of the attraction radius versus minimum distance between patterns (scaled with 1/number of units)
6. SIMULATIONS

Numerical simulations have been performed on a BAM having \( n=182 \) and \( n=64 \). For the outer product encoding method of (3), unreliable storing has occurred at rather small values of \( p \) (average \( p=5.5 \)), depending on the items stored.

For every training set, the following procedure was used: \( f_x \) and \( f_y \) were kept equal to 0 till all \( p \) equilibrium points were stored. Then both \( f_x \) and \( f_y \) were increased by 1 and training proceeded by choosing \( X \) and \( Y \) at distance \( f_x \) or \( f_y \) from the stored items. After the error frequency dropped under a given value, \( f_x \) and \( f_y \) were again increased by 1.

Note that the values of \( f_x \) and \( f_y \) determine the domains of initial states from which the BAM converges to \((0, \cdot, \cdot)\) in no more than two time steps. The real domains of attraction are larger and not necessarily regularly shaped. Fig. 2 shows a case of retrieval from a distance greater than \( f_x \) in more than 2 steps. Separate error frequency measurements for initial states randomly chosen around the stored items were performed using the \( W \) obtained as explained above. The results summarized in Fig. 3 show that although the training was stopped before convergence, the stored items are attractors even for initial states as far as 11 units.

7. CONCLUSIONS

The training algorithm introduced in this paper offers very good storage capabilities using simple computations in only fixed point arithmetic. But, as the number of restrictions grows, it might become computationally intensive.

![Figure 2](image1.png)  
**Figure 2** Average retrieval error frequency vs. Hamming distance for \( p=16 \) stored items, \( n=182 \), \( n=64 \) and 5025 training cycles with \( f_x=f_y=2 \).

![Figure 3](image2.png)  
**Figure 3**. Retrieval in the above memory from an initial state (upper left) at \( d=28 \) from the stored item (pair lower right).

It has been also shown that the algorithm can be used for BAMs with general activation functions \( \phi \). The closeness of the network state to the desired items in this case can be prescribed if \( \phi \) is known.

Some necessary conditions for the existence of a solution for a given set of data were outlined, but stronger convergence conditions are still to be found.

![Figure 4](image3.png)  
**Figure 4** Number of training cycles (1 cycle=2p steps, one for each \( g \) and \( p \)) for different \( p \) and \( f \) for a 182 x 64 BAM.
ACKNOWLEDGEMENT

The authors are thankful to the colleagues of Elisabeth Gardner with the Physics Laboratory of the University of Edinburgh for their friendly support at the beginning of this work.

REFERENCES

A BANK OF MULTILAYER PERCEPTRONS FOR THE
ON-LINE RECOGNITION OF HANDWRITTEN CHARACTERS

Eveline J. Bellegarda†, Jin H. Kim†, and Jerome R. Bellegarda†
†IBM T.J. Watson Research Center, Yorktown Heights, New York 10598, USA
†Center for Artificial Intelligence, KAIST University, Seoul, Korea

We describe a neural network approach to tackle the problem of automatic recognition of handwritten characters. We introduce a novel segment-based approach to handwriting, in which a segment may be viewed as a representative building block of handwriting. The segmentation scheme is used in conjunction with a bank of multilayer feedforward neural networks to perform character recognition. We have evaluated the approach on tasks involving (i) discrimination between similarly-shaped characters and (ii) recognition of discretely written upper-case characters.

I. INTRODUCTION

The automatic recognition of handwritten characters is of paramount importance in applications where handwriting is the desirable input channel, such as personal note taking. As it allows natural interactions between man and machine without the use of a keyboard, this mode of input is increasingly gaining acceptance. Pen-based user interfaces capture handwriting through a special pen operating on an electronic tablet; the handwriting data recorded on the tablet is then sent to a recognizer which outputs a string of typeset characters. Such operation requires the on-line recognition of the handwritten data, which is to be distinguished from optical character recognition (OCR).

So far, the traditional approach to on-line handwriting recognition has been through template-matching algorithms, where character recognition is performed by comparing the unknown character against a set of prototypes constructed during a training phase. Common strategies have focused on stroke-based character recognition, where a stroke is defined as the trajectory of the pen while it remains in contact with the tablet [1]. This approach suffers from inherent difficulties such as (i) a large set of strokes is needed to cover the entire handwriting alphabet, and (ii) stroke lengths and shapes vary greatly for different input modes (discrete, run-on, cursive, or unconstrained handwriting); cf. Fig. 1. Alternatively, a neural network-based strategy was proposed in [2], where a system was designed to perform digits and upper case letter recognition on a touch terminal. This solution is more appealing than template-matching because of its lesser sensitivity to noise; however, it tends to require more training data.

In this paper, we present a novel neural network approach to character recognition. The new algorithm employs a bank of multilayer feedforward neural networks for more efficient use of the available training data. Also, it relies on a preprocessing scheme significantly different from the ones proposed previously, which aims at a more robust description of the different variations (allographs) around each character. More specifically, it is our contention that each handwritten character can be described by a small number of representative segments (building blocks), where a segment corresponds to a finer subdivision of a stroke. These segments are used as inputs to the bank of neural networks.

We define a segment as the amount of information between two consecutive feature points, where a feature point is characterized by a local extremum in the change in the angle of the tangent to the curve at this point. In essence, all points within a segment share somewhat similar characteristics. Since each character is obtained as a collection of distinct segments, various segment-encoding schemes can be investigated to consistently represent each allograph. The encoding parameters are suitably selected to encompass the number of points in a segment and the location of these points within the segment. Once the segments have been isolated, we construct one multilayer neural network for each category of i-segment allographs resulting from this segmental encoding.

The paper is organized as follows. In the next section, we present the feature extraction front-end associated with the segment-based approach. In Section III, we describe the neural network architecture used in conjunction with each segment-based representation. In Section IV, we present experimental results on two different tasks: (i) discrimination between similarly-shaped characters, such as (Z, 2, s), and (ii) discretely-written upper-case character recognition.

II. FEATURE EXTRACTION

1. Pre-processing

Handwriting can be interpreted as a continuous signal \((x(t), y(t))\), where \((x, y)\) are the coordinates of the path followed by the pen and \(t\) is the time. The data captured by the tablet are represented as a sequence of points regularly sampled in time. The pre-processing procedure (cf. [3]) is
composed of four different steps. In the first step, we aim at reducing variability in the raw data, such as time and scale distortion, by normalizing the sequence of pair coordinates to make all the character sizes equal. In the second step, centerizing is performed, to have the characters fall within the same box. In the third step, an elastic filter is applied to remove some points for smoothing effects. In the fourth step, the variations in writing speed are compensated for by resampling the data to obtain regularly spaced points. Due to this re-sampling, the average number of pair coordinates decreases from approximately 50 to about 20.

2. Segmentation

Segments are obtained by chopping a stroke into appropriate sub-constituents. This is done according to the change in the angle of the tangent to the curve at successive points. For each point \( P_i \) of coordinates \((x_i, y_i)\), the angle of the tangent to the pen trajectory at \( P_i \) is defined by:

\[
\theta_i = \arctan \frac{\Delta y_i}{\Delta x_i},
\]

where the horizontal and vertical incremental changes are computed as: \( \Delta x_i = x_{i+1} - x_i \), \( \Delta y_i = y_{i+1} - y_i \). The difference in angle between two successive points is thus obtained as: \( \Delta \theta_i = \theta_{i+1} - \theta_i \). For each stroke, the chopping procedure is initialized by taking as feature points the starting and ending point. The direction of change \( \Delta \theta_i \) is then computed at the midpoint of the stroke, where the anchorg is half of the total pen trajectory. If \( \Delta \theta_i \) is a local extremum, the corresponding point \( P_i \) is selected as a possible feature point candidate, and the procedure is re-applied recursively. Among all feature point candidates, the points for which \( ||\Delta \theta|| > T_{\theta} \), where \( T_{\theta} \) is a pre-set threshold, are kept as final feature points.

Several issues arise regarding the choice of the threshold angle \( T_{\theta} \). If \( T_{\theta} \) is too small, this will result in a large number of segments, which may allow more detailed representation of the stroke but may be more sensitive to noise. On the other hand, too large a value of \( T_{\theta} \) results in a small number of segments, which may induce too much smoothing, and local features might get lost in the process. Thus, several values of the threshold angle have to be experimented with to find a reasonable compromise.

3. Segment Encoding

Among all points belonging to a given segment, we select the \( x \) and \( y \) coordinates of 7 points, 5 of them equispaced and the last 2 taken at the midpoint of the beginning and the end sections of the segment (see Fig. 2). We content that these extra two points are required to suitably encode the critical information present at the beginning and ending of the segment. In addition, information regarding the connection of one segment to another was taken into account by encoding the direction of change \( \Delta \theta \) at the beginning and end of the segment. In summary, each segment was represented by \( 7 \times 2 + 2 = 16 \) numbers.

We have empirically determined (see Section IV) that for the upper-case or lower-case Roman alphabet, each handwritten character can be adequately represented with at most 5 segments. This enables a segment-based recognition implemented using 6 different neural networks. Through the front-end described above, each character is directed toward the proper network according to the category to which it belongs.

III. NEURAL NETWORK ARCHITECTURE

As described above, the training data is partitioned into \( s \)-segment data \((s = 1, \ldots, 6)\), and one fixed-size neural network is constructed for each category of characters. The \( s \)-segment training data also defines the number \( K_1 \) of output classes for the letters observed. Hence, each neural network differs by its number of input and output units, due to segmental encoding and data-dependent labels, respectively.

For each neural network, a multilayer feed forward architecture is considered; as in [2], the layers successively perform higher level feature extraction and the final classification. We implement 6 feed forward perceptrons with one input layer, two hidden layers, and one output layer. The two hidden layers have 20 and 10 hidden units, respectively. Each neural network has the following number of connection weights: \( (4 \times 16) \times 20 + 20 \times 10 + (10 \times K_1) \), where \( s = 1, \ldots, 6 \).

Each neural network is trained by the standard backpropagation algorithm [4]. A training token is composed of an input pattern and its corresponding output label (one of the \( K_1 \) labels). In this context, the neural network can be viewed as a mapping from input patterns to target patterns; and the backpropagation algorithm is an iterative version of the mapping function computation. The results of training are the weights associated with all the connections. Before training, the weights are initialized with random values, uniformly distributed between 0 and 1. Subsequently, the weights are modified according to the backpropagation error.

The learning times of the different networks are reported in terms of epochs. An epoch, or iteration step, is one presentation of the entire set of \( L \) training patterns. Both local and global updates are investigated. In the case of global update, the weights are updated after the entire set of \( L \) training patterns has been seen, i.e., we have one update per epoch. In the case of local update, the weights are updated after seeing each input pattern, i.e., this results in \( L \) updates per epoch. In general, local update results in faster learning time but in more oscillations in the sum of the squared errors \( SSE \) for all the outputs. The learning progression is measured by plotting the curve of \( SSE \) vs. the number of epochs. When it falls below some fixed value or when the number of epochs for the current session reaches the maximum number of iterations \( M \), the learning is completed.

The backpropagation learning parameters of interest are: \( \mu \), the learning rate; \( \sigma \), the momentum factor; and \( r \), the range of the random initial weights. Selection of those learning parameters is something of a black art, and small differences in these parameters can yield large differences in learning times [4]. The learning parameters can be manipulated to trade off speed and accuracy. It was suggested in [4] that an \( r \) value of 1, a \( \mu \) value of 0.5 to 0.0, and an \( \sigma \) value of 0.9 to 0.5 should, in principle, yield faster learning. Of course, those "rules" have to be used as guidelines since the implementation of the network is strongly problem-dependent. A sigmoid prime shift function was considered (see [4]) to eliminate the "flat spot" and speed up the training. Notice that, contrary to [2], our weights were independent, making the present approach more general.

During recognition, each network uses the weights acquired during the training phase to classify the unknown in-
put character against the respective inventory of classes.

IV. EXPERIMENTAL RESULTS

1. Discrimination Task

One major source of errors in character recognition comes from the lack of discrimination between similarly-shaped characters such as, for example, (2, Z, z); (5, S, s); (U, V, u, v); (P, D); (t, +). Template matching algorithms, in particular, tend to discriminate poorly between such characters, which originally provided some motivation to develop the segment-based neural network. To illustrate the potential of the above approach to automatically extract features distinguishing confusable pairs, we consider in this section the problem of discriminating between 2, Z, and z, and compare the results with those obtained with a template-matching algorithm. The performance of the bank of neural networks was evaluated for both writer-dependent and writer-independent recognition tasks. In writer-dependent recognition, the templates or classes obtained during the training of a particular writer are used to recognize or classify handwriting of the same writer, while in writer-independent recognition the writers used for training are totally independent from the writers used for decoding.

The tablet used had a resolution of 0.1 mm and a sampling rate of 70 fps. We considered a collection of (2, Z, z) from 10 writers. On average, each of the writers provides 23 2's, 12 Z's, and 12 z's. For writer-dependent recognition tasks, the data from each writer was split into training (80%) and testing (20%) while for writer-independent, the first 8 writers provided the training data set and the 2 remaining writers the test data set.

For the neural network training, the following learning parameters were used: \( \mu = 0.5 \) for the learning rate, \( \alpha = 0.9 \) for the momentum rate, and \( \theta_e = 90^\circ \) for the threshold angle for the direction of change. The input layer to the network had \( 16 \times i \) for \( i = 2, 3, 4 \) input units and 3 output units, representing the three classes. The two hidden layers had 10 and 4 hidden units, respectively.

For writer-dependent decoding task, the recognition accuracy of the bank of neural networks was comparable to the template-matching algorithm. For writer-independent recognition task, the bank of neural networks achieved a recognition accuracy of 83.3% while the performance of the template matching algorithm was 79.6%. Similar results were obtained when discriminating between \( (t, +); (U, V); (P, D) \).

2. Upper-Case Character Recognition

We considered the vocabulary of 26 characters in the upper-case Roman alphabet. Training data was collected in a box with discrete mode from 7 writers over a period of several days. Among the writers, two were left-handed. The training data consisted of 2353 characters, or 330 characters per writer on the average. The classification performance of the networks was measured on data provided by 2 additional (test) writers, who each provided about 350 test characters.

To evaluate the relative contribution of each neural network in the bank, we computed for the training data, the frequency distribution of the number of segments as obtained through the segmental encoding described earlier. The distribution of i-segment upper-case characters \( (i = 1, \ldots, 6) \), from the 7 writers considered is displayed in Fig. 3. It can be observed that most of the upper case characters fall in the 3 segments category. Thus, the performance of the 3-segment neural network will be critical to the ultimate recognition accuracy. Also, it was found that the 3 and 4 segments neural networks generated the largest number of classes; thus, it is expected that classification with such networks will be more difficult than with the other networks.

For the training, the following learning parameters were used: \( \mu = 0.5 \) for the learning rate, \( \alpha = 0.9 \) for the momentum rate, \( \theta_e = 90^\circ \) for the threshold angle for the direction of change. Local update policy was selected to speed up the training. The parameter \( SWE \) was recorded every 10 epoch. When it fell below some fixed value of \( 10^{-2} \) or when the number of epochs for the current session reached the maximum number of iterations (by default set to 10000), training was deemed completed.

Fig. 4 shows the individual performance of (i) the 2-segment neural network, in the 85% recognition rate; (ii) the 3-segment neural network, in the upper 70% recognition rate; and (iii) the 4-segment neural network, also in the upper 70% recognition rate. The individual performance of the 1-segment neural network was in the 90% recognition rate and the 5-segment neural network in the upper 60%. The performance of the 6-segment neural network is not given, since no test character was chopped into 6 segments.

Fig. 5 shows the global recognition rate obtained when all the neural networks are combined, as a function of the number of writers used for training. The overall performance of the bank of neural networks is around 80%. Note that the recognition rate increases noticeably as more writers are added to the training data banks but levels off after a critical number of writers is reached. This may indicate that for this particular pool of writers (and all from the New York City area) three or four writers are sufficient to capture essentially all the necessary syllogisms.

Note that the simplest classification scheme was used, i.e., the decision of the network was the one corresponding to the output unit having the highest value. In future studies, it might be worthwhile to investigate a classification scheme which provides a way to automatically rejects ambiguous and/or meaningless patterns. This can be embedded into 2 rules based on the output values of the neural network. The first rule rejects the character when the activation level \( L_1 \) of the highest of the outputs does not exceed a threshold \( T_{max} \), i.e., \( L_1 < T_{max} \). The second rule defines the character as ambiguous if the difference between the highest activation level \( L_1 \) and the second highest activation level \( L_2 \) is less than a threshold \( T_{ Huffman } \), i.e., \( L_1 - L_2 < T_{ Huffman } \).

V. CONCLUSIONS

The segment-based approach described in this paper appears to be a viable route to the recognition of handwritten characters. Novel pre-processing and segment encoding schemes were developed, with which an acceptable recognition accuracy was reached. In this respect, we have not explored all possibilities; for example, increasing the number of points used in the encoding scheme may lead to a more accurate description, at a probable expense in computational cost.
Moreover, additional efforts are required to fine tune the different neural networks; slightly different topologies and learning parameters may yield better results. Nonetheless, the bank of neural networks was found to outperform a conventional template-matching approach, which is especially noteworthy for writer independent tasks. In the future, one may envision using such neural networks in the second stage of a template-matching system, for example to discriminate between confusible pairs.

REFERENCES

Fig. 1. Various types of handwriting.

Fig. 2. The 7-point encoding scheme.

Fig. 3. Distribution of i-segment characters.

Fig. 4. Recognition rate vs. number of training characters for individual neural networks.

Fig. 5. Global recognition rate vs. number of training characters (all the neural networks combined).
LEFT VENTRICULAR BOUNDARY DETECTION IN ECHOCARDIOGRAPHICAL IMAGES USING ARTIFICIAL NEURAL NETWORKS

Y. Xin  P. Boekaerts  M. Bister  Y. Taeymans  J. Cornelis

Vrije Universiteit Brussel
Department of Electronics / Research Group IRIS
Pleinlaan 2, 1050, Brussel, Belgium.
Tel. 0032/2/641.29.62
Fax. 0032/2/641.28.83

INTRODUCTION

Two-dimensional real-time echocardiography is a noninvasive technique widely used in clinical cardiology. It is especially attractive because of its low cost, the minimal discomfort it causes to the patient, the absence of ionising radiation, and its possible application for patient monitoring through real-time processing. It is however well-known that images produced by sonographic methods have poor noise characteristics and low spatial and grey value resolutions [1]. Quantitative analysis and boundary detection on such images is difficult, which explains the large amount of research in this area [2]-[5]. In [2], a temporal tracking algorithm is proposed; [3] describes a radial boundary search algorithm based on simulated annealing; [4] uses optical flow calculation; [5] is a 1-D matched filtering approach. [2], [3] and [4] are computationally heavy and require large amounts of memory. [2] requires important temporal averaging which leads to uncertainty in the location of the ventricular border; [3] has problems to find acceptable heart cavity borders when the valves are open; [5] suffers from the difficulty to define a good model of a "cardiac boundary".

The aim of the research behind this paper is to explore a new approach for reliable echocardiographic left ventricular boundary detection using artificial neural networks. The novelty resides in the heart cavity delineation — even when the valves are open — without having to define explicitly a model for the cardiac boundary. The importance of the work resides in the fact that the intrinsic parallel nature of artificial neural networks allows an implementation bringing the time performances in a range which is acceptable for physiologists.

METHODS AND EXPERIMENTAL RESULTS

For the detection of parts of the left ventricular boundary in local windows, one should face the problem that the orientation of the border of the cavity changes for each window position. To reduce this orientation problem, the Cartesian coordinate system of the echocapherical image is transformed to a polar coordinate system with an origin lying inside the cavity ([3],[5]). All subsequent processing is applied on polar images (radius, angle).

1. The NN Boundary Detector

1.1 Method

A Feedforward Neural Network (NN) is used as a local matched filter in order to classify the central pixel of the filter window based on the information of its environment. Let \( x=(x_1,x_2,\ldots,x_n) \) be the vector of grey values of \( n \) pixels in a local rectangular window. We define 2 classes of windows (A and B). In the case of left ventricular boundary detection, A is defined as the class of windows for which the central pixel belongs to the boundary and B is the class of windows for which the central pixel does not belong to the boundary. We want to design a Feedforward Neural Network \( N(x) \in [0,1] \), which performs the following mapping operation:

\[
N(x) \rightarrow 1, \quad \text{if } x \in A; \\
N(x) \rightarrow 0, \quad \text{if } x \in B.
\]

(1)

During training, several windows \( (i = 1, \ldots, M) \) are clamped to the input of the neural network and a cost function \( J = \sum J_i \) is minimized. \( J_i \) is given by

\[
J_i = (N(x_i) - 1)^2, \quad \text{if } x_i \in A; \\
J_i = N(x_i)^2, \quad \text{if } x_i \in B.
\]

(2)

The Gradient Backpropagation learning algorithm ([6],[7]) is used to force the neural network to generate a high (or low) activity level \( N(x) \) in the output neuron if the central pixel of the window belongs to the boundary (or not) in order to minimize the cost \( J \). The learning set used to train
the neural network to recognize the boundary is
extracted from the polar image of grey values in
which we selected the boundary by hand. It is
composed of a set of windows of which we know
whether the central pixel belongs to the left
ventricular boundary or not.

The trained neural network is applied sequentially
to all windows centred around the points lying on a
given horizontal line in the polar image. The central
pixel of the window with maximum output response
N(x) in the line is selected as the left ventricular
boundary point. This operation is repeated for each
horizontal line in the polar image.

1.2 Experiments and Results

The Gradient Backpropagation algorithm was used
to train the neural network on 200 samples (the
learning set). This learning set contains 100
patterns (i.e. x vectors) in windows of class A
and 100 patterns of class B.

Although most windows of class A and B are chosen
at random from a given polar image, some manual
effort has been done to eliminate the so called
"ambiguous" patterns. It can be easily understood
that, if the boundary lies too closely near the central
point of a class B window, it could be easily
misclassified by the neural network as belonging to
class A. Such patterns can cause local minima
during learning and decrease the convergence
speed. We have found experimental evidence that
even if these patterns are rejected from the learning
set, the neural network still classifies them properly
afterwards: the output response of the neural
network decreases when the boundary slightly shifts
from the centre of the window.

Furthermore, to make the neural network immune
to the typical speckle noise of echocardiographical
images, 10% of the windows of the learning set
were selected to contain important speckle noise.

As a last constraint in the choice of the learning set,
we took about 5% of the windows on the mitral
valves, when they are open. We found experimental
evidence that such choice makes the neural
network immune to the configuration of the mitral
valve opening during the application phase.

For the evaluation of the performance of the trained
neural network, we used two test images which
were not used in the learning set. A mean square
geometric distance between the manually selected
boundary points and the computer selected
boundary points is defined as an error measure.

$$\text{error} = \frac{1}{n} \sqrt{\left(\sum (d_{c,i} - d_{m,i})^2\right)}$$

where $d_c$ is the computer selected distance and $d_m$ is
the manually selected distance of the boundary
point relative to the origin of the polar image. Fig.1
shows the evolution of the error measure (3),

expressed in number of pixels, for a trained
Feedforward Neural Network during testing on two
images (error 1 for PICT 1 and error 2 for PICT 2) as
a function of the window size. From this experiment,
we can conclude that the error made by the network
for the detection of the left ventricular boundary
significantly drops when the window size becomes
larger than 9 x 9. This is especially true for image
PICT 1. This can easily be understood if one
realizes that image PICT 1 contains an open mitral
valve and also contains more speckle noise than
image PICT 2.

Fig. 1 The evolution of the error expressed in
number of pixels, in function of the
window size (3 neurons in hidden layer,
200 learning samples). Error 1 corresponds
to PICT 1. Error 2 corresponds to PICT 2.

The window size affects the convergence time
during training in a negative way however.
Therefore, one should find a good compromise
between the window size of the boundary detector
and the learning speed during training. Test and
application time (for 1500 window positions and a
neural network with 81 input neurons, 3 hidden
neurons and 1 output neuron) is typically of the
order of 30 seconds for simulations on a SPARC
station 1+.

Fig.2 shows the boundary detection accuracy
(expressed in number of pixels) of a given trained
neural network (200 learning samples, window size
9 x 9 pixels) for both test images as a function of the
number of neurons in the hidden layer.

From this kind of experiments, we concluded that
the extension of the hidden layer with more than
three neurons does not increase the accuracy
significantly.

The influence of the number of learning samples on
the quality of the boundary detection has been
investigated. As could be expected, the accuracy of
the network increases if one extends the number of
learning samples. Obviously, if we increase the
number of learning samples, the classification task
becomes more and more well-defined allowing the
network to learn a good internal representation of
Fig. 2 The evolution of the error expressed in number of pixels, in function of the number of hidden neurons (200 learning samples, window size 9 x 9 pixels).

2. The Hybrid boundary detector

2.1 Method

In some of the previous type of experiments, the neural network boundary detector tends to fail to classify the left ventricular boundary properly for echographical images in which the mitral valve is in the open configuration (see PICT1, the boundary is not detected correctly near the open mitral valve). One possible way to solve this problem is to combine the output of the neural network with a smoothness term.

$$s = \frac{(d_i - d_{i-1})^2 + (d_i - d_{i+1})^2}{2d_i^2}$$  \hspace{1cm} (4)

The smoothness term $s$ for the current pixel $i$ is equal to the normalized mean squared difference in radius (or distance from the origin) of the two neighbouring boundary points $i$-1 and $i$-2. This term is combined with the NN output $N(x)$ in a linear decision function:

$$D = \alpha N(x) + \beta s$$  \hspace{1cm} (5)

or

$$D = wy$$  \hspace{1cm} (6)

where $w = [\alpha, \beta], y = [N(x), s]^T$. The value of the weighting coefficient $w$ is obtained, during the learning phase, by the Fisher discriminant analysis [8].

$$w = S^{-1}(m_A - m_B)$$  \hspace{1cm} (7)

where:
- the vectors $m_A$ and $m_B$ are the mean of vectors $y$ respectively of class A and B.
- $S$ is the variance-covariance matrix.

We can make the decision:

$$x \in A \quad \text{if } D > \theta$$

$$x \in B \quad \text{if } D < \theta$$  \hspace{1cm} (8)

where $\theta$ is the threshold which decides about the attribution of a pixel to class A or B.

Fig. 3 Comparison of a neural network based boundary detector without smoothness term (c) to a hybrid system with smoothness term (b) for PICT 1 (a).

2.2 Experiments and Results

Experiments showed that the hybrid detector improves the accuracy of boundary detection significantly. Tests on PICT 1, show that the
boundary detection error given in (3) decreases from 2.4 pixel to 1.1 pixel after introducing the smoothness term (for a network with 3 hidden neurons and a window size of 9 x 9 pixels).

Figure 3 gives an idea of the difference in accuracy of boundary detection for a simple neural network boundary detector without smoothness term (b) compared to a hybrid system with smoothness term (c).

(a)  
(b)  

Fig. 4 Result of the hybrid boundary detector for PICT 2, (a) original image, (b) boundary detection result.

Figure 4 shows boundary detection with the hybrid detector. Experiments on sequences of images from different patients reveal that the accuracy of boundary detection of the hybrid system is equally good during most of the myocardial cycle, except when the heart cavity reaches its minimal size.

DISCUSSION AND CONCLUSION

As a general conclusion, we can state that the neural network based boundary detector, if trained properly with the Gradient Backpropagation learning algorithm, is able to detect the left ventricular boundary in images with a closed mitral valve. Problems related to the detection of the left ventricular boundary in images with an open mitral valve can be solved by combining the response of the neural network with a smoothness term, forming a so called hybrid detector.

In order to apply such boundary detector for clinical use in real time patient heart monitoring, the intrinsic parallel nature of neural networks should be exploited in the implementation to speed up the application time. Remaining problems, such as large contrast changes during the heart cycle and changes in orientation of the boundary within the window, will be investigated in the near future. Large scale testing should be performed before clinical application, mainly on the generalisation properties (from patient to patient) of the trained networks.

REFERENCES

A SPECIFIC PROCESSOR FOR THE COMPUTATION OF TDNN ALGORITHMS
WITH APPLICATIONS IN PHONETIC CODING

M. Pérez-Castellanos*, V. Rodellar†, V. Peinado‡, A. Díaz‡ and P. Gómez-Vilda‡

*Facultad de Informática, Urb. Montepriincipio, Rodilla del Monte, 28660 Madrid-Spain
†Escuela Universitaria de Informática, Km. 7 Carretera de Valencia, 28035 Madrid-Spain

This paper describes the structure of a Specific Processor capable of supporting
Time Delay Neural Network algorithms in real time, as well as other NN and digital
signal processing algorithms sharing an inner product computational structure. The
specific processor has been designed using serial arithmetics. It has the advantage of
presenting a very flexible data path, through the inclusion of a set of multiplexers
which allow independent access to the different functional units of the processing
elements. The structure described is being used for Phonetic Coding applications.

1. INTRODUCTION

TDNN algorithms have been widely used in different speech problems due to its main
characteristic of hybridizing the concepts of Digital Filter and Neural Network algorithms
[Ham.90], [Hum.91]. We have been using TDNN algorithms in the application of phonetic
coding adapted to the Phonetics of Spanish language [Rod.91b]. The particular topology of
the algorithm suitable for such a problem is shown in Fig.1, consisting in an input layer
containing six nodes, a hidden layer of eight nodes and an output layer of eight nodes also,
the digital filter part has order two, and the net is fully connected. The reasons for
choosing such a topology may be found in [Rod.91a].

The generic expression of a TDNN node may be written as:

\[ S_k(n) = \sum_{j=1}^{2} \sum_{d=0}^{2} F \left( \sum_{i=1}^{\text{nodes}} X_i(n-d) w_{ij} \right) \]

where "k" indicates the node number under computation inside the hidden or output
layers, \( 1 \leq k \leq 8 \), "n" is the discrete time index, \( F \) is the non-linear function, "j" is the
node number of the source layer, which contributes to the evaluation of the current
node, \( 1 \leq j \leq 8 \); \( F \) being the total number of nodes in the source layer, then \( 1 \leq j \leq 8 \) for the hidden
layer and \( 1 \leq j \leq 8 \) for the output layer, "d" is the delay time index, "x(k)" represents the
node inputs, "w" is the set of coefficients obtained during the training period and finally
"i" distinguishes between the hidden (i=1) and output layers (i=2).

Let's see the data dimensionality and computational structure of the algorithms. The
NN will be fed by a vector \( \text{IN}[r] \) containing three sets of samples of speech spectrum
parameters taken in three consecutive time instants, then \( r=6 \times 3=18 \). Similarly, the output
of the hidden layer will be a vector \( \text{HI}[s] \) containing three data sets corresponding to the
operations performed in three consecutive time instants on the nodes involved in the
computation of the current node, hence \( s=8 \times 3=24 \). The output of the NN will be a
vector \( \text{OU}[v] \), with \( v=8 \). The set of weights, obtained during the trained simulation phase,
associated to the hidden layer will be a matrix \( \text{WH}[v,r] \), being \( v=8 \) and \( r=18 \), giving a
total amount of 144 elements to be stored. The set of weights for the output layer will be
another matrix of 192 elements, \( \text{WO}[v,s] \) with \( v=8 \) and \( s=24 \). And finally the computation of
the non-linear function will be done by means of a look-up table of eight elements \( \text{LUT}[8] \).

The computation of the algorithm is as follows; first we will evaluate the nodes of the
hidden layer, this operation may be seen

Fig.1 TDNN algorithm topology.
as a matrix-vector product:

\[ \mathbf{WH}(\mathbf{v}, \mathbf{r}) = \mathbf{IN}(\mathbf{r}) = \mathbf{Q}_{n}(\mathbf{v}) \]  

(2)

where \( \mathbf{Q}_{n}(\mathbf{v}) \) is a vector of eight components corresponding to a node of the hidden layer. To complete the evaluation of this layer, the results of this vector must be passed through the look-up table in order to compute the non-linear function \( \mathbf{F}(\cdot) \):

\[ \mathbf{F}(\mathbf{Q}_{n}(\mathbf{v})) = \mathbf{H}(\mathbf{v}) \]  

(3)

The computation of the output layer may also be written as a matrix-vector product:

\[ \mathbf{W}_{O}(\mathbf{v}, \mathbf{s}) \mathbf{H}(\mathbf{s}) = \mathbf{P}_{n}(\mathbf{v}) \]  

(4)

and after the computation of the non-linear function the final results will be obtained:

\[ \mathbf{F}(\mathbf{P}_{n}(\mathbf{v})) = \mathbf{O}(\mathbf{v}) \]  

(5)

2. THE SPECIFIC PROCESSOR

The structure of the algorithm described above will be supported by a chip containing four processing units, each PU being locally connected to its neighbours, border processors having connection capability with the external world. The control of the chip is done by means of a PLA, which provides a flexible basis for the reprogrammability of the chip's control. The basic structure of one of these PUs is shown in Fig. 2, it consists in an Arithmetic Unit, a double port RAM memory, a ROM memory, a set of multiplexers, two input data lines and two output data lines. The internal data paths of the PUs are bit serial in order to save silicon area.

![Diagram of the PU's](image)

Fig. 2 Basic structure of the PU's

The kernel operations involved in the computation of this algorithm are matrix-vector products, then the Arithmetic Unit will consist basically in an Inner Product Unit structure, containing a multiplier of 16 bits, which uses fixed point arithmetics and extended sign for protection against overflows, a 16-bit accumulator and a 16-bit adder. The structure also incorporates two 3:1 multiplexers that allow to route selectively data from the RAM memory, the multiplier or the accumulator to the adder. The double port RAM memory will be devoted to store the weights, input data and auxiliary variables, its capability is 256x16, and the data transfers from and to memory are done in parallel through the interface registers I/00 and I/01 which provide serial/parallel and parallel/serial conversion. The ROM memory locates the look-up table values for the computation of the non-linear function, its size is 8x16. The structure of the PUs is completed with a set of multiplexers that provide an enormous flexibility to the internal data routing in the rigid schema of a bit serial approach. The multiplexers 3:1 introduce selectively data from the input EI, the result from the look-up table, the contents of the port I/OI or I/O0 respectively, the result of the accumulator or the input E0 into the RAM memory registers I/O0 or I/O1. And finally the 2:1 multiplexers allow routing the contents of both I/O RAM memory registers to the PU's outputs.

The next step is to show how to compute the proposed algorithm in a set of four PUs with a reduced architecture as the one just described. If we take a look to the data dimensionality of the problem, for example keeping our attention in the matrices \( \mathbf{WH}(\cdot) \) and \( \mathbf{W}_{O}(\cdot) \), we notice that none of them can be located in the RAM memory of the same PU, then a segmentation and mapping strategy for the algorithm must be adopted. Different segmentation and mapping strategies may be proposed using as performance parameters the load balance among PUs, memory occupation, and communication simplicity, avoiding waiting times among transfer data. Some possible strategies have been already proposed [Rod.91a].

Although the PU structure has been designed having in mind a given algorithm for a very specific application, this PU can also easily support other NN models and Digital Signal Processing algorithms with an inner product computational structure. This can be easily accomplished altering the programming of the PLA, for such a purpose a Transfer Register Language is being used.

3. TNNN ARITHMETICS

As it was mentioned before, the Inner
Product Arithmetic Unit of each PU was designed on a 16-bit fixed point data format, using Serial Arithmetic. At first glance, it seems that such a format will produce a very degraded behaviour in the arithmetics, due mainly to precision losses and overflows, when compared to wider floating point formats, but this is not necessarily so when some properties of Back Propagation Neural Networks are adequately exploited. For instance, when expression (1) is carefully examined, one can easily see that the basic operation to compute the output of a given node, is a 3xP order Inner Product, involving 3xP multiplications and the same amount of additions. Regularly, the inner product is carried out by simultaneously addressing each element of the corresponding vectors \( V_{1+n} \) and \( V_{21+n} \), multiplying them, and adding this product to the accumulated partial result \( A_{i} \), by the recursive expression:

\[
A_{i+1} = A_{i} + V_{1i+1} \times V_{2i+1} \quad (6)
\]

When the data involved in these operations are represented using a S+F bits normalized fixed point format, where S bits are used for sign extension, and F bits are used for fraction, it may easily be deduced that after each multiplication, a 2(S+F) bit number will be obtained. In it, the initial S bits will be disregarded, the following S+F bits will be used as the result, and the final F bits will also be rejected, giving a result again in S+F bits. When we assume that both vectors \( V_{1} \) and \( V_{2} \) have their elements normalized in the interval \([-1, +1]\), the result would also be a normalized S+F fixed point number. The multiplication would produce a loss of precision if \( V_{1i+1} \) or \( V_{2i+1} \) or both data are much lower than the unity, because, in that case, the significant bits of the result could appear in the rejected F final bits of the \( 2(S+F) \) product. On the contrary, the addition, could produce results higher than the unity, and in this case, an invasion of the S guard bits reserved for the sign extension would result. The worst case would take place if the first sign bit would be invaded, because the sign would be lost. This effect is more critical as the number of additions to be performed increases. As this number is given by 3xP, it would be worse for the output layer (P=8) than for the inner layer (P=6). To prevent these facts, the best solution is to adequately dimension S and F, keeping the constraint that S+F=16. For such, a set of simulations were conducted, using different S+F formats, and contrasting them with standard double precision floating point conventions. Randomly generated numbers were used as elements of the test vectors. A summary of the simulation results is presented in Table 1 for the more restrictive case of P=16.

<table>
<thead>
<tr>
<th>S</th>
<th>F</th>
<th>MNSI</th>
<th>MLB</th>
<th>MAPL</th>
<th>MQE</th>
</tr>
</thead>
<tbody>
<tr>
<td>#6</td>
<td>#5</td>
<td>#4</td>
<td>#3</td>
<td>#2</td>
<td>#1</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>-</td>
<td>-</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td>8</td>
<td>6</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>6</td>
<td>12</td>
<td>7</td>
<td>5</td>
<td>4</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1. Sign Invasions and Precision Losses

(MNSI) is plotted for each sign bit from the least significant (#6) to the most significant (#1). The Maximum Number of Lost Bits (MLB), Maximum Absolute Precision Loss (MAPL) and the Maximum Quantizing Error (MQE) are also plotted under the corresponding columns. It can be seen that with S=4 we could represent numbers in the interval [-8, +8], whereas with S=6 the interval would be [-32, +32]. The inner product in (6) would produce a maximum absolute value of 24 for P=8, or 48 for P=16 only when both vectors have their elements equal each to the other with a value of +1 or -1, but these cases are nonsense. Experimental estimations show that the maximum expected absolute value is under one third of those maximum figures, which means 8 for P=8 and 16 for P=16. This means that the format 4+12 is just in the acceptable limit, but the format 6+10 well inside limits. In fact, in this last case from Table 1, one can see that the first bit is never invaded for P=16, which covers the case of P=8. The MAPL keeps reasonably low, about 100 times above the MQE, due to its cumulative effect. In a practical case, with P=8 these effects would be almost neglectable. The results for a format 6+12 are given for a comparison, showing that the gain in precision does not justify the increase in Silicon area required (11%). On the other hand, maintaining the restriction S+F=16 an easy change among different formats could be done just "retailoring" the results after multiplications, as was commented above. Precision losses in Neural Networks are not as critical as they could be in other Signal Processing Applications, because of the nonlinear features of most Neural Networks. In fact, the second operation to be considered in (1) is the nonlinear mapping from the Inner Product Interval, for instance [-32, +32], to the Node State Interval [-1, 1]. This nonlinear mapping is mathematically expressed as a Sigmoid Function. In our case, this function is implemented by a Look-Up-Table. This structure may be seen as a Transcoding ROM which receives the S most significant bits of the result in (6), and uses them to address a 16 bit output out of a table of \( 2^{S} \) possible results. Having in mind this effect, a loss of precision in the multiplications does not seem critical, in the sense that it will seldom affect the bits used to address within the LUT. These facts have been checked using real data, and the proposed format, against a double precision floating point, and the performance degradation is rather acceptable, although the matter is still under study.
4. INTEGRATED CIRCUIT DESIGN

The whole structure is being prepared for its integration under a 1.5 μm CMOS technology, following the floorplan given in Fig. 3., although this floorplan is scaled for a 2μm process, under which different critical parts, as the multiplier, have been casted already.

Fig. 3. Floorplan of the supporting IC

The floorplan shows the physical disposition of the structures in Fig. 2 in a real scale. It may be seen that the most area consuming section is the Two Port Memory (34.6%), followed by the Arithmetics (20%), Data Paths (18%), Control (17.3%) and Clocks&Power (10%). These figures give a good justification to assume a fixed-point serial data format, because memory, arithmetics and data paths are very sensitive to data formats in the sense that more than 72% of the integrated circuit area is affected by those aspects. The floorplanning is roughly based in a "Manhattan Skyline" style, most of data and control routing being done using a central highway connecting memory to arithmetics. Following CMOS design restrictions, general tri-state busses have been substituted by the massive use of multiplexers. Having in mind that address buses DR00-7 and DR10-7 require parallel routing, the PLA generating these signals have been placed in the same waterfront side than the Two-Port Memory. The data routes are kept under reasonable limits in length. This can’t be said of Clock and Power, which must be routed all through the chip. These routes are not shown for the sake of clarity. Vss and Vdd are routed along the periphery of the chip, and the central corridor is used for Vdd and Vss. The design has been carried out using customized cells of regular structure in the multiplier, registers and multiplexers, and standard library cells for the PLA, LUT and Memory. The whole chip requires an area of 15 mm² in the 2μm process, although a reduction thus allowing the piling of up to 4 of these FUs in a medium size integrated circuit.

5. CONCLUSIONS

A customized design of the Arithmetic Unit has been casted and is currently under test. Other parts of the described structure are also under design, and a whole chip containing one FU will be available in brief. Communication schemes among FUs are being simulated on a four Transputer system. We have also studied different possible segmentation and mapping strategies of the TDNN algorithm in order to obtain the maximum computational efficiency and several solutions that fulfill the requirements for real time processing have been found. Actually we are working in the programming of the PLA in order to cast the structure proposed in a 1.5 micron CMOS technology. The applications of the described structure are found in the support of a joint auditory model for frequency extraction of the Speech trace and of a TDNN. The whole system will be devoted to real time phonetic transcription of Spanish in Computer Aided Speech Learning.

6. ACKNOWLEDGMENTS

This work is being supported by the grants: Acción concertada UPM A91-0020-02-78, Acciones Especiales de Microelectrónica MUC88-0398-E and MTC90-1219-E, and Plan de Investigación de la CAM C072/90.

7. REFERENCES


A SAMPLE PROTOCOL FOR THE HOPFIELD NEURAL NETWORK


University of Twente, Dep. of Electrical Engineering, P.O. Box 217, NL-7500 AE Enschede, The Netherlands.

A Hopfield neural network can be used as a pattern recognition system. In this paper a sample-controlled protocol for sequential pattern recognition with the Hopfield network is introduced.

I. INTRODUCTION

The paper by Hopfield of 1982 [1] explored the ability of a system of highly interconnected artificial neurons to have useful collective computational properties. He showed that such a artificial neural network can be used as a Content-Addressable Memory (CAM). An item that is stored in a CAM can be retrieved on basis of a disturbed version of that item. Therefore a CAM can be very well used for pattern recognition problems. In this paper we show that we can retrieve a pattern not only on the basis of a disturbed version of the pattern, but also on a sequence of local samples of the pattern to be recognized. Moreover the number of local samples can be reduced by controlling the sampling process on the basis of the previous responses of the neural network. Results on sample controlled recognition of hand-written numerals are given.

II. HOPFIELD NETWORK

In 1984 Hopfield introduced his time-continuous neural model [2]. An electronic scheme of such a neuron is shown in figure 1. This neuron contains a non-linear amplifier with an input $u_i$ and output $v_i$.

The transfer function $v_i = g(u_i)$ should be monotone and bounded [2]. We will consider the hard-limiter transfer function:

$$v_i = g(u_i) = +1 \text{ if } u_i \geq 0$$

$$v_i = -1 \text{ if } u_i < 0$$  \hspace{1cm} (1)

The neural network of Hopfield is totally connected by conductances $T_{ij}$. The conductance $T_{ij}$ and capacity $C_i$ establishes a time delay between a $v_i$ and state variable $u_i$. Each output $v_i$ of a

*Currently at University of Groningen, Department of Computing Science, P.O.Box 800, NL-9700 AV Groningen, the Netherlands. E-mail: klijn@cs.rug.nl.
neuron is an output component of the total system. The output vectors $v$ of the total system correspond with state vectors $u$.

Only a few output vectors $v$ of a Hopfield neural system with $N$ neurons are stable. An output vector is a stable output vector if and only if:

$$v_i = g(\sum_{j=1}^{N} T_{ij}v_j + I_i) \quad \forall i$$  \hspace{1cm} (2)

The initial state of the system corresponds with some selected output vector. For all output vectors which are not stable output vectors the dynamics of the system let these vectors converge to a near(est) stable vector (measured in Hanning distance).

If the network is used as a CAM for pattern recognition, then prototype patterns will be encoded as prototype vectors (PVs), with elements -1 or 1. These PVs will be stored in the neural net as stable states by means of adapting the connection matrix $T$ with the use of a learning rule. We used the geometric or outer-product rule [4] to establish the storage of stable states.

This learning rule creates a matrix $T$ which projects every vector $v$ on the sub-space spanned by the PVs. So all PVs are eigenvectors of the matrix $T$ with eigenvalue 1 and are stable according to (2) with $I_i = 0$ for all $i$. Our experiments revealed that with the geometric rule a good CAM can be constructed, i.e. disturbed patterns (= unstable output vectors) converge to their nearest prototype pattern (output vector), under the conditions:

A. The distortion of the pattern is within certain limits;

B. The number of stored prototype patterns $M$ is not too high ($M_{\text{max}} = 0.3N$) and

C. The correlation between prototype patterns is not too high [3].

If the diagonal elements of the connection matrix $T$ are set to zero, it can be proved that the PVs remain stable and better results with the CAM can be obtained. [3].

III. SAMPLE-CONTROLLED ALGORITHM

If a pattern $w$ is offered to a Hopfield network to be recognized the whole pattern $w$ can be used to realize the corresponding initial state of the system. The initial state can be imposed by using the inputs $I_i$ of each neuron. The complete pattern $w$ to be recognized is then encoded as a input (current) vector $I = \mu w$. After initialization of the system the input vector $I$ is set to the zero vector and the system will converge to the nearest stable output pattern.

In our approach the initial state of the system corresponds with some randomly chosen prototype vector $v$ and an initial input vector $I(w)$ composed of zero’s and one entry $\mu w_k$, corresponding with a local pixel sample $w_k$ is presented to the net. When a stable state is reached a second input vector $I(w)$ composed of zero’s and two entries $\mu w_k$ and $\mu w_l$ corresponding with pixel samples $w_k$ and $w_l$ is presented to the net. And so on.

It turns out that in this case only about 40 % of the pattern needs to be sampled for recognition. The second method has thus the advantage of using less information of a pattern than the first, especially if the sampling process can somehow be controlled on basis of previous responses. A good example of a system which has an efficient sampling process for scanning pictures is the Human Visual System (HVS): The eyes of a human being very rapidly jump ( not observable for the human itself) from one "interesting" part of a picture to another. Interesting parts are mostly parts with a sharp luminance difference. The HVS also only needs a small part of a total picture for recognition [5].

We developed a similar method to control the sampling process of patterns with the Hopfield network using information of the internal state during successive stages of the converging dynamical process of the network. The state of the network with $N$ neurons can be described by the $N$-dimensional vector $u$. An initial state will converge to a stable state for which [6]:

$$u_i = R_i \sum_{j=1}^{N} T_{ij}v_j \quad \forall i$$  \hspace{1cm} (3)

$$with \quad \frac{1}{R_i} = \sum_{j=1}^{N} |T_{ij}|$$

It can be shown that the absolute value of a component $u_k$ of a stable state vector $u$ is small the more the corresponding components $v_j$ of all the stable PVs are different from each other. These component $v_j$ correspond to pattern samples which are different for different prototype patterns at the same position in the pattern. One might expect that sampling a pattern mainly on the parts for which it differs from other patterns would yield a better performance. However selecting continually a sample position corresponding with the minimal absolute value of a state variable did not reduce significantly the percentage of necessary samples for recognizing a pattern in comparison to random sampling a pattern.

We can however use also the minimal absolute value of components of unstable states which we encounter during the preceding process of conver-
gence. It turns out that a better performance is obtained by sampling the pattern at those positions (i.e. \( w_i \)) where in a preceding dynamical convergence process the absolute value of the state component \( u_i \) becomes minimal.

One can consider the initial (randomly chosen) state, corresponding with some prototype vector \( v_1 \) as an initial hypothesis about the pattern \( w \) to be recognized. The initial state together with the initial input vector \( I_1(w,u) \) will cause the system to converge to the nearest stable output vector \( v_2 \), i.e. the next hypothesis about the pattern \( w \). During the first convergence process unstable vectors \( u \) will be passed. Small values of components \( u_1 \) of these unstable vectors \( u \) indicate that \( w \) and \( v_i \) differ at position \( i \). Thus it will be profitable to sample the pattern \( w \) at that position. This, together with \( I_1(w,u) \), gives us the second input vector \( I_2(w,u) \) for the second output vector \( v_2 \), etc. until \( v \) represents the same pattern as \( w \).

IV Experiments

A network with \( N \) neurons was learned with \( M \) prototype vectors/patterns.

| \( N=30 \) \( M=5 \) \( m\text{Hd}=5 \) | \( 21.6\% \) | \( 45.2\% \) |
| \( N=30 \) \( M=5 \) \( m\text{Hd}=10 \) | \( 32.7\% \) | \( 40.6\% \) |
| \( N=30 \) \( M=10 \) \( m\text{Hd}=10 \) | \( 37.9\% \) | \( 38.0\% \) |
| \( N=100 \) \( M=15 \) \( m\text{Hd}=15 \) | \( 23.5\% \) | \( 40.6\% \) |
| \( N=100 \) \( M=30 \) \( m\text{Hd}=15 \) | \( 29.2\% \) | \( 34.2\% \) |

Table 1

First experiment: For every PV as an initial state and for every PV, different from the initial state, as an pattern to be recognized an experiment was performed. We used the sample protocol and compare it with random sampling. The mutual minimum Hamming distance (mHd) between PVs could be adjusted, this mHd is a measure for the correlation between prototype patterns. The results show the average percentage of the input vector which has to be sampled before the network converges successfully.

The experiments show that the sample-controlled protocol operates better than the random sample protocol (table 1). Especially if the number of prototypes is low in comparison to the number of neurons, and the minimum Hamming distance between the PVs is low, i.e. the correlation between the PVs must be high (see \( N=100 \) \( M=15 \) \( m\text{Hd}=15 \) and \( N=30 \) \( M=5 \) \( m\text{Hd}=6 \)). In the other cases the difference is small or negligible.

We note that the performance of a Hopfield network as pattern recognition device in the classical way, thus without the sample controlled protocol, is low in case that the correlation between PVs is high.

Second experiment: In this experiment pictures of numerals (0-9) where used as patterns. 10 prototype numerals have been encoded to PVs by dividing the picture in 100 pieces and let each piece corresponded with -1/+1 if less/more than a quarter of the pixels of the piece were grey. A network with 100 neurons has been learned with these PVs. Initially the network is in a stable state, representing a prototype numeral. Another prototype numeral is offered for recognition (figure 2 and 3).

<table>
<thead>
<tr>
<th>( \text{random samples} )</th>
<th>( \text{sample controlled} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average</td>
<td>22.4%</td>
</tr>
<tr>
<td>Maximum</td>
<td>36.0%</td>
</tr>
</tbody>
</table>

Table 2

The results show that the sample-controlled protocol operates better than the random sample protocol (table 2).

fig. 2: Controlled sampled "8" with initial network state representing a "9". 12% need to be sampled.
fig. 3: Random sampled "8" with initial network state representing a "9". 30% need to be sampled.

V CONCLUSIONS

Using a Hopfield network for pattern recognition it is not necessary to present the whole pattern. Random sampling the pattern can reduce the required information from the pattern to be recognized with about 60%.

A sample-controlled sequential input protocol is developed, so that especially networks with a low number of highly correlated prototype vectors will need less samples (2 times less is possible) than a random sample-protocol. This controlled sample-protocol does not use prototype information.

REFERENCES


A COMPARISON OF ERROR DIFFUSION AND THE HOPFIELD NEURAL NETWORK AS METHODS TO QUANTIZE HOLOGRAM DISTRIBUTIONS

P. VAN DEN BULCK

EE-Department, Eindhoven University of Technology, Eindhoven
P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands

In computer-generated holography, the distribution of a hologram that is used to generate a given reconstruction is calculated by means of a digital computer. Since we use in the production step of such a hologram an output device that is able to generate a binary output only, the calculated distribution has to be quantized to a binary signal. This will lead to quantization noise in the reconstruction plane. With the so-called error diffusion halftoning algorithm it is possible to calculate a binary hologram with low noise power within a chosen region of the reconstruction plane. Using a linear model for the error diffusion process, optimal diffusion coefficients of a first-order causal feedback filter are calculated. The error diffusion concept with causal error feedback filters can be generalized to error diffusion with noncausal feedback filters. The algorithm then describes the dynamics of a discrete-time Hopfield neural network with serial updating. This iterative calculation of the binary hologram is computationally more expensive, but a lower noise power is achieved.

1. INTRODUCTION

Holograms are used in optical systems to realize certain signal processing functions. In computer-generated holography the hologram transmittance is calculated by means of a digital computer and drawn by e.g. a laser-printer or an e-beam writer. When a hologram with transmittance \( \phi(x, y) \) is illuminated with a coherent light source, we measure at some distance behind the hologram the wavefront \( \Psi(u, v) \). In the case of Fourier holograms, the reconstruction \( \Phi(u, v) \) of the hologram and the hologram distribution \( \phi(x, y) \) are Fourier transform pairs.

Suppose that we want to generate a certain object \( \Psi(u, v) \) in the reconstruction plane of the hologram. With the inverse Fourier transform we calculate the hologram distribution \( \psi(x, y) \). In general this will be a complex signal. Since the transmittance of an amplitude hologram is real and limited in dynamic range, we take the real part of \( \psi(x, y) \) and perform a scaling \( \phi(x, y) = |\Re(\psi(x, y))| / C \), with \( C = \max(|\Re(\psi(x, y))|) \). For symmetry-reasons we will use a bipolar signal \(-1 \leq \phi(x, y) \leq 1\), without considering the extra mapping to a positive transfer function that is necessary for an amplitude hologram. Due to these operations, the hologram will reconstruct \( \Phi(u, v) = |\Psi(u, v) + \Psi^*(-u, -v)| / (2C) \), with \( \Psi^*(-u, -v) \) called the twin image of the original object. Special care must be taken to avoid overlap of the original and its twin image in the reconstruction plane.

Therefore we define a region \( \mathcal{R} \) where \( \Psi(u, v) \) can be nonzero only [1].

In practice we sample the reconstruction plane and use the inverse discrete Fourier transform to calculate the samples \( \phi[n_1, n_2] = \phi(n_1 \Delta x, n_2 \Delta y) \) of the hologram distribution. In the digital-to-analog conversion (performed by printing the hologram), the calculated hologram samples are interpolated and rectangular cells with a binary transmittance (black or white) are formed. This means we must quantize our calculated hologram samples to binary samples. This operation will always lead to quantization noise in the reconstruction plane. Neglecting interpolation effects due to the finite size of the cells, we will consider two methods to separate the quantization noise from the object in the reconstruction plane.

The error diffusion algorithm [2] is often used as a halftoning-technique for pictures. It shapes the spectral content of the quantization noise to higher spatial frequencies and thus makes the quantization noise more or less invisible to our 'low-pass' eyes. In the reconstruction plane of a Fourier hologram we measure the spectrum of the quantization noise. Thus, when using error diffusion to quantize the hologram distribution [3], we are able to shape the quantization noise in the reconstruction plane and minimize the noise power in the region \( \mathcal{R} \).

The error diffusion algorithm processes the two-dimensional hologram distribution in a one-dimensional spatial ordering. To ensure recursive computability, er-
errors are diffused to certain directions only. In a generalization of this concept called the Hopfield neural network, errors are diffused in all directions. The binary hologram distribution is now calculated in an iterative way. This procedure is computationally more expensive than the error diffusion algorithm, but a lower noise power can be expected.

2. ERROR DIFFUSION

Calculating the binary signal \( b[n_1, n_2] \) for a discrete signal \( \phi[n_1, n_2] \) with the error diffusion algorithm, we use the following equations:

\[
\begin{align*}
\phi[n_1, n_2] &= \text{sat} \{ \phi[n_1, n_2] \} \\
b[n_1, n_2] &= Q \phi[n_1, n_2] \\
q[n_1, n_2] &= b[n_1, n_2] - s[n_1, n_2].
\end{align*}
\]  

(1)

In (1) \( \ttimes \) denotes two-dimensional convolution and \( Q(\phi \geq 0) = 1, Q(\phi < 0) = 0 \). Error diffusion can be regarded as processing \( \phi[n_1, n_2] \) with a nonlinear two-dimensional filter, as shown in Fig. 1. This filter consists of a

\[ \phi[n_1, n_2] \quad + \quad e[n_1, n_2] \quad \rightarrow \quad b[n_1, n_2] \quad \rightarrow \quad q[n_1, n_2] \]

Figure 1: Two-dimensional nonlinear filter.

nonlinear operation, i.e., a quantizer and a filter with a finite impulse response \( h[n_1, n_2] \) in the feedback loop. The quantizer errors \( q[n_1, n_2] \) that have been made in the quantization of 'past' pixels, are weighted and diffused to the present pixel in order to influence the decision of the quantizer. To ensure recursive computability of this algorithm, the impulse response of the error feedback filter \( h[n_1, n_2] \) must be causal, and therefore has asymmetrical half-plane support \( \mathcal{W} : \{1 \leq n_2 \leq M, 0 \leq n_1 \leq 2M \} \cup \{-M \leq n_1 \leq 0, 1 \leq n_2 \leq M\} \), as is shown in Fig. 2. \( M \) denotes the filter-order.

\[ h[n_1, n_2] \]

\[ \begin{array}{ccc}
\vdots & \vdots & \vdots \\
M & & M \\
\vdots & \vdots & \vdots \\
\end{array} \]

Figure 2: Error-feedback filter with asymmetrical half-plane support.

With the \( 2(M^2 + M) \) filter coefficients \( h[n_1, n_2] \) we are able to shape the spectrum of the quantization noise \( e[n_1, n_2] = \theta[n_1, n_2] - \phi[n_1, n_2] \). Since we measure this spectrum in the reconstruction plane of the Fourier hologram, we try to find the filter coefficients that cause the smallest noise power in the region \( \mathcal{R} \). To this end we introduce a linear model for error diffusion. In this model the quantizer error \( q[n_1, n_2] \) is regarded as an independent noise source. The set of equations (1) then simplifies to \( s = \phi - h \ttimes q \) and \( b = s + q \). The input-output relation of the linear model is

\[ B(z_1, z_2) = \Psi(z_1, z_2) + [1 - H(z_1, z_2)]Q(z_1, z_2), \]

(2)

with the two-dimensional Z-transform of a signal \( x[n_1, n_2] \) defined as

\[ X(z_1, z_2) = \sum_{n_2} \sum_{n_1} x[n_1, n_2]z_1^{-n_1}z_2^{-n_2}. \]

(3)

Since \( E(z_1, z_2) = B(z_1, z_2) - \Phi(z_1, z_2) \) is the two-dimensional Z-transform of the quantization error \( e[n_1, n_2] \) we can write

\[ E(z_1, z_2) = [1 - H(z_1, z_2)]Q(z_1, z_2). \]

(4)

Apparently the quantizer error is filtered with a noise shaping filter \( 1 - H(z_1, z_2) \) to form the quantization error. For \( z_1 = e^{j\Omega_1} \) and \( z_2 = e^{j\Omega_2} \) the Z-transform passes into the Fourier transform and (4) becomes an expression for the noise in the reconstruction plane. If we assume the quantizer noise to be white, with noise power \( \mathcal{E} Q(e^{j\Omega_1}, e^{j\Omega_2})^2 = \sigma_q^2 \), the total amount of noise in the region \( \mathcal{R} \) of the reconstruction plane will be

\[ P = \sigma_q^2 \int_{\mathcal{R}} |1 - H(e^{j\Omega_1}, e^{j\Omega_2})|^2d\Omega_1d\Omega_2. \]

(5)

When we construct a coefficient-vector \( h \) containing the filter-coefficients, we can rewrite (5) in a general quadratic form

\[ P = \alpha + \alpha^T h + \frac{1}{2} h^T A h. \]

(6)

Due to the feedback loop, instability can occur for some sets of filter coefficients. This means that the internal signal \( s[n_1, n_2] \) can grow without bound for a bounded input \( \phi[n_1, n_2] \). Thus, we call the algorithm stable, if for every bounded input \( |\phi[n_1, n_2]| \leq 1 \) the internal signal \( s[n_1, n_2] \) is bounded (BIBO stability). A sufficient condition to guarantee stability is [4,5]

\[ \sum_{(n_1, n_2) \subset \mathcal{W}} |h(n_1, n_2)| \leq 1. \]

(7)

Finding the optimal filter-coefficients is then a constrained quadratic optimization problem: minimize \( P \) under constraint (7).

For the region \( \mathcal{R} : 0 \leq \Omega_1 \leq \frac{\pi}{2}, 0 \leq \Omega_2 \leq \frac{\pi}{2} \) we found as optimal diffusion coefficients for a filter of order \( M = 1 \) : \( h[1, 0] = 0.3574, h[1, 1] = -0.1922, h[0, 1] = 0.0000 \) and \( h[-1, 1] = 0.5404 \). Calculation of higher-order filters did not result in a significantly lower
noise power. With these error diffusion coefficients, a hologram distribution (since $N^2 = 128^2$) that generates the EUT logo in the given region has been quantized. A comparison of the theoretical noise distribution and the simulation results (Fig. 3) shows that the noise distribution is well predicted. To measure the noise power we used an approximation

$$P \approx \frac{1}{N^2} \sum_{k_1,k_2} |B(e^{j\theta_{k_1} k}, e^{j\theta_{k_2} k}) - \Phi(e^{j\theta_{k_1} k}, e^{j\theta_{k_2} k})|^2,$$

where the summation extends over samples in the region of interest. A numerical result of about 55.7 was found.

The causal filter $h[n_1,n_2]$ takes only errors in account that have been made in the quantization of previous samples. This makes the algorithm recursive computable, provided that we process the samples in the right spatial ordering. This concept of error diffusion can be generalized, however. In that case, the decision of the quantizer for a given sample is influenced by the quantization errors of all other samples. Of course, it is not possible anymore to calculate the binary output recursively.

To derive an expression for noncausal error diffusion we rewrite (1) into

$$a[n_1,n_2] = \phi[n_1,n_2] - g[n_1,n_2] * \delta[n_1,n_2]$$
$$b[n_1,n_2] = \Omega a[n_1,n_2]$$
$$e[n_1,n_2] = b[n_1,n_2] - \phi[n_1,n_2],$$

which leads to

$$b[n_1,n_2] = \Omega \{-g[n_1,n_2] * \delta[n_1,n_2] + g[n_1,n_2] * \phi[n_1,n_2] + \phi[n_1,n_2]\}.$$  

The coefficients $g[n_1,n_2]$ form the impulse response of a filter with transfer function $G(z_1,z_2)$. This filter is related to the original feedback filter $H(z_1,z_2)$ according to

$$G(z_1,z_2) = \frac{H(z_1,z_2)}{1 - H(z_1,z_2)}.$$  

A filter $G(z_1,z_2)$ associated with an error feedback filter $H(z_1,z_2)$ with asymmetrical halfplane support, is also causal and has wedge support. Not restricting ourselves to the causality condition, we can also choose sequences $g[n_1,n_2]$ with four quadrant support. As we will see, (10) then describes the serial updating rule for a discrete-time Hopfield neural network.

3. THE HOPFIELD NEURAL NETWORK

We consider a Hopfield network with $N$ equal processor elements or neurons. With the interconnection strength between neuron $n$ and $m$ denoted by $w_{nm}$, the matrix $W$ describes the overall interconnection of the network. We assume that the interconnection is symmetrical and that neurons are connected to other neurons only. Therefore $W$ is a symmetrical matrix with zeros on its main diagonal. Each neuron sums its incoming input signals. If the sum is greater than its threshold, the neuron will generate 1 as output signal, −1 otherwise. The thresholds of all neurons form the threshold vector $\tau$. In [6] Hopfield has shown that such a network is described with a set of nonlinear differential equations

$$\frac{dy}{dt} = WQ(y) - \tau - y,$$  

with $y$ the neuron-state vector and $\tau$ a time constant. The hard-clipping operator $Q$ operates on each component of $y$. When the network has converged, it is in a stationary state

$$y = Wf(y) - \tau.$$  

Since we are interested in the binary output of the network, we introduce the binary output vector $b = Q(y)$. Thus, in equilibrium the binary output vector satisfies

$$b = Q(Wb - \tau).$$

In order to simulate the network on a digital computer, we divide the time axis into intervals $\Delta t$. As a consequence we must consider which neurons are updated in each interval. We will consider two extreme cases, namely serial and parallel updating. With serial updating, we select (according to some rule) only one neuron in every interval. To calculate the new binary value of the selected neuron we use

$$b_m = Q \left[ \sum_{n=0}^{N-1} w_{mn} b^{i\Delta t} - b_m \right],$$

with $b_m = b_m(i\Delta t)$. For this updating rule it is known [6] that the "energy"

$$H(b) = -\frac{1}{2} b^T Wb + b^T \tau b$$

can not increase during updating. Since the energy $H$ has an absolute minimum value, the network will always converge to a stable state.

In order to use a Hopfield network in finding binary hologram distributions with minimal noise power in the region $\mathcal{R}$, we must relate the quantization problem to
the energy function $H$. We are then able to calculate the weight matrix $W$ and the threshold vector $\xi$ and let the Hopfield network search for a solution. In order to maintain a simple notation we will first consider one-dimensional signals $\phi[n]$ and $b[n]$, represented by the signal-vectors $\Phi$ and $B$, respectively. The spectra of these signals are approximated with the discrete Fourier transforms $\Phi = F\phi$ and $B = Fb$, with $F$ the Fourier transform matrix. We now try to find a binary signal $F$ that minimizes the distance

$$D(F, \phi) = ||KF(F - \phi)||^2,$$

with $K$ a real diagonal frequency weighting matrix. This matrix determines the frequencies for which the noise power is minimized. It is possible to show [5] that minimizing the above equation is equivalent to minimizing

$$\frac{1}{2}F^T A F - \xi F^T A F,$$

with $A = F^H K H K F = F^{-1} K^2 F$ a real circulant matrix. Without loss of generality we can scale the main diagonal elements of $A$ to one, and write $A = G + I$, with $I$ the identity matrix. When we compare the above equation with the energy of a Hopfield network, we can conclude that a Hopfield network with $W = -G$ and $\xi = -A \xi$ will solve the quantization problem. For the serial updating rule this gives (in vector notation)

$$\dot{\Phi}_m = Q \left[-G\Phi^{m-1} + (G + I)\xi\right].$$

Since the matrix $G$ is also circulant, we can rewrite this equation in the form of a (periodic) convolution (denoted by $\otimes$)

$$b[n] = Q \{-g[n] \otimes \xi[n] + g[n] \otimes \phi[n] + \phi[n]\}.$$  

When we compare the two-dimensional extension of this equation with (10), we can conclude that the Hopfield network is a generalization of the causal error diffusion concept.

In order to save calculation time we will use parallel updating. With this updating rule, all neurons are updated in each interval. To derive this rule we approximate the derivative in (12) with a first-order backward difference and obtain

$$\frac{u'[t] - u'[t-1]}{\Delta t} = W Q(u'[t-1] - \xi - u'[t-1]),$$

with $u'[t] = u(t\Delta t)$. When we define $\alpha = \Delta t/\tau$ the parallel updating rule passes into

$$u'[t] = \alpha \left[W Q(u'[t-1] - \xi) + (1 - \alpha)u'[t-1]\right].$$

The interval $\Delta t$ must be smaller than the time-constant $\tau$ and therefore $\alpha \ll 1$. Although convergence is not guaranteed for parallel updating, we found that for a small $\alpha$ the Hopfield network always converged to a stable state with (local) minimal energy. Substitution of the calculated interconnection matrix and threshold vector into (22) gives

$$u'[t] = \alpha \left[-F^{-1}K^2FQ(u'[t-1]) + Q(u'[t-1]) + F^{-1}K^2F\xi + (1 - \alpha)u'[t-1]\right].$$

This updating rule describes an iteration between space and frequency domain [7]. In the space domain we perform quantization, in the frequency domain filtering. The extension to two-dimensional signals is straightforward. When using the fast Fourier transform, this algorithm is of complexity $O(N^2 \log N)$.

![Figure 4: Simulated reconstruction.](image)

We quantized the hologram distribution that generates the TUE logo with the Hopfield network. In Fig. 4 the reconstruction of the binary hologram after 30 iterations is shown. The Hopfield network indeed tries to minimize the quantization noise in the given region. Since the noise shaping is better matched to the region $\mathcal{R}$, we may expect a lower noise power. Numerical calculation with (8) resulted in a noise power of 40.2, which is indeed smaller than that obtained with the error diffusion algorithm.

**REFERENCES**

NEURAL PREDICTION IN IMAGE VECTOR QUANTIZATION

F.G.B.DeNatale, G.S.Desoli, S.Fioravanti, and D.D.Giusto

Dept. of Biophysical and Electronic Engineering, University of Genoa, Italy

A prediction scheme, based on a Perceptron neural network, is proposed, which allows one to exploit the residual inter-block correlation after VQ, without loss in the quality of reconstructed images. Main advantages lie in its efficiency (even for images not included in the training set), and in reduced memory requirements.

1. INTRODUCTION

Vector quantization [1] is one of the most widely used coding approaches, thanks to its high performances, in spite of its simplicity. However, it is worth noting that, after the encoding process, a notable amount of redundancy remains between neighbouring vectors, which are highly correlated. This fact is illustrated in Figure 1, where the height of each entry \((i,j)\) of the matrix is proportional to the probability for finding the vector \(i\), given the vector \(j\) in the previous position. Considerable efforts have been devoted to exploiting this inter-block correlation.

In a recent paper [2], a prediction scheme was proposed that allows one to obtain good results in terms of an increasing in the compression factor, and that does not cause any further loss in the SNR. But, such a scheme exhibits some drawbacks that make it not so attractive for software and hardware implementation. First of all, it requires the storage of large matrices, called transition matrices, which contain the conditional probabilities of finding the codevector \(i\) given the codevector \(j\) in the previous position along the principal directions \((0, 45, 90, \text{ and } 135\) degrees). Another problem is the computation of the score functions for all the codevectors; this task requires one to access the transition matrices too many times, and is therefore very time-consuming. Moreover, the scheme presents the disadvantages of the storage and re-ordering of the address codebook.

The approach proposed in this paper exhibits the advantage (over the above-mentioned scheme), of an easy hardware realization; furthermore, no additional storage (except for the synaptic weights of the neural network) is required. The approach involves the implementation of a Perceptron neural network [3], trained in such a way as to reconstruct an image block on the basis of the neighbouring vectors.

The network is made up of an input layer (whose dimensions depend on the vectors' size and on the predictor's memory), and of an output layer, of the same dimensions as the codevectors. No hidden layers are utilized, thus making it possible to speed up the prediction and to reduce storage requirements (see Figure 2). For the learning phase, the classical back-propagation algorithm [4] has been adopted, which aims to minimize the distance between the vector generated by the network on the output layer and the actual block present in the training set, when the input layer is presented a vector made up of the neighbouring blocks. During the coding phase, every time a vector has to be transmitted, its neighbouring blocks are given as input to the trained network, and the prediction pattern is generated by the output layer. The obtained set of patterns is then used to organize a codebook of smaller dimensions than the original
one, thus requiring that a smaller number of bits be addressed. In the case of a wrong prediction, a recovery is provided.

In the following Sections, the architecture of the proposed network is described, the learning and encoding-decoding phases are detailed, and some experimental results are reported.

2. NETWORK ARCHITECTURE AND TRAINING

Figure 2 shows the scheme of the network architecture. The input layer consists of a number of neurons that is equal to the number of elements of the four vectors $\mathbf{V}_{i-1,j}$, $\mathbf{V}_{i-1,j-1}$, $\mathbf{V}_{i,j-1}$, and $\mathbf{V}_{i+1,j-1}$ adjacent to the vector $\mathbf{V}_{i,j}$ to be predicted. The output layer has the dimensions of the vector $\mathbf{V}_{i,j}$.

No hidden layers are present, and the network is fully connected (e.g., when using 4x4-sized vectors in the quantizer, the number of synaptic connections is 1K).

The learning algorithm adopted is the classical back-propagation one. At each iteration, a set of configurations of blocks (surrounding the one to be predicted and preceding it in a left-to-right, top-down scan) are presented to the network input. The following updating rule is applied to the synaptic weights to minimize the mean square error (MSE) between the vector generated by the network (using the input and the weights computed during the previous iteration) and the target (i.e., the actual vector in the training set):

$$\Delta \mathbf{W}_{i,j}(1,j) = \eta \delta_j \mathbf{a}_j + \alpha \Delta \mathbf{W}_i(i,j)$$

where $\eta$ is the gain, $\alpha$ is the momentum, and $\delta$ is given for each neuron $\mathbf{a}_j$ of the output layer by the following equation:

$$\delta_j = (t_j - \mathbf{a}_j)(1 - \mathbf{a}_j)$$

For the present specific implementation of the algorithm, the initial weights are set to small random values, while the values of $\eta$ and $\alpha$, which strongly affect the speed of convergence, are dynamically updated according to the variations in the MSE.

3. THE ENCODING-DECODING PROCESS

The values of the synaptic weights computed during the learning phase are stored and used by the encoder and the decoder to achieve the prediction. During the encoding process, each block $\mathbf{V}_{ij}$ is first vector-quantized to obtain the actual associated codevector $\mathbf{C}_k (k = 1, \ldots, 2^n)$, where $n$ stands for the number of bits required to address the overall codebook. Then, the estimated pattern $\hat{\mathbf{V}}_{ij}$, generated by the neural network on the basis of the previously coded vectors $\mathbf{V}_{i-1,j}$, $\mathbf{V}_{i-1,j-1}$, $\mathbf{V}_{i,j-1}$, and $\mathbf{V}_{i+1,j-1}$, is used to organize a codebook of smaller dimensions, in which the codevectors $\mathbf{C}_h (h = 0, \ldots, 2^n-1, b<n)$, at a minimum distance from $\hat{\mathbf{V}}_{ij}$ are contained. Finally, if the actual codevector $\mathbf{C}_k$ is one of the $\mathbf{C}_h$, only $b$ bits are required to transmit the address $h$; otherwise, the $b$ bit configuration left free in the reduced codebook is transmitted as a fault configuration, followed by the $k$ bit address of the codevector $\mathbf{C}_k$ in the overall codebook. In the case of a correct prediction, compression is therefore increased by a factor of $b/n$, while, if the prediction is wrong, it is reduced by a factor of $b+n/n$.

Denoting by $P_{hit}$ the probability of a correct prediction, the factor $F$ that multiplies the compression ratio can be expressed as:

$$F = \frac{1}{P_{hit} \left( \frac{b}{n} \right) + \left( 1 - P_{hit} \right) \frac{b+n}{n}}$$

In order to obtain an increase in the compression factor, the threshold value $P_{hit}^T$ for the probability of a correct prediction can be computed by imposing $F$ equal to 1 and solving as a function of $P_{hit}$:

$$P_{hit}^T = \frac{b}{n}$$

The decoder must be synchronous with the encoder, in the sense that it has to accomplish the same sequence of operations, except for quantization, which is replaced by a simple look-up table. For each block, the prediction is calculated in the same way as in the encoder (the decoder
contains an exact copy of the neural network), thus generating the reduced codebook. If the b-bit configuration received is a valid address, the vector is selected from the reduced codebook; otherwise (fault condition), the next n bits are used to address the overall codebook directly. The global architecture of the encoding-decoding system is displayed in Figure 3.

4. RESULTS AND CONCLUSIONS

A new algorithm for inter-block redundancy reduction in VQ has been presented, which is based on the use of a neural network as a block predictor. The advantages offered by this approach lie in the reduced memory requirements (both static and dynamic) and in a lower computational complexity (also for a hardware implementation). The proposed scheme seems to be very efficient, thanks to the predictor structure and to the coding scheme. Several tests were performed on learned and unlearned images, using codebooks of dimensions ranging from 128 to 1024 vectors, and reduced codebooks of dimensions from 7 to 31 vectors. The diagram in Figure 4 shows the results of the tests carried out on the classical image of "Tiffany". The block dimensions were 4×4, and the image was not used for the construction of the codebook, nor for the learning of the predictor. The best results were obtained by using a 256-codevector VQ (n=8) and a reduced codebook of dimensions 7 (b=3). For such parameters, the value of $P_{\text{hit}}^T$ that ensured a reduction in the bit rate was about 37%, and the $P_{\text{hit}}$ obtained ranged from 60 to 75% (corresponding to a factor $F$ of 1.3 to 1.6). In the encoding phase, the synaptic weights can be further updated to take into account the specific characteristics of the image to be coded. Another possible alternative that was not considered in this paper is to assign variable-length configurations of bits to the vectors in the reduced codebook on the basis of the distances of such vectors from the prediction.

ACKNOWLEDGMENTS

This work has been supported by the National Research Council of Italy (CNR: Progetto Finalizzato Trasporti) and was carried out within the framework of the European Eureka project Prometheus (ProCom sub-program).

REFERENCES


Figure 1. 3D view of a transition probability matrix
Figure 2. Predictor structure vs neural network architecture

Figure 3. Architecture of the coding system

Figure 4. $F$ as a function of $b$ and $n$ for a classical test image (Tiffany).
NEW APPROACH FOR MULTI-PERCEPTRON ARCHITECTURE DESIGN APPLIED TO THE EDGE DETECTION PROBLEM

Ch. Molina, D. Bastard, P. Baylou and M. Najim

Equipe Signal et Image and GDR 134 - CNRS
ENSERB, Université de Bordeaux I
351 cours de la Libération, 33405 Talence cedex, FRANCE

This paper deals with a new method to determine the architecture for a one-hidden-layer perceptron restricted to the convex problems solution in pattern recognition. The main contribution of this method is to train and design the neural network in a polynomial time in opposition to the unpredictability of classical methods. This method is derived from the gift-wrapping algorithm which is mainly used in computational geometry for the construction of convex hulls in a d-dimensional euclidean space. To demonstrate its efficiency we apply this approach to the edge detection in image processing. We show that the edge detection can be solved by a neural network with only one hidden layer as an alternative to classical methods which employ the local maxima module of the gradient.

INTRODUCTION

The multiperceptron network is, at the present time, one of the most popular among the neuromimetic models. Its design is usually made by choosing a priori the neural network architecture. Pattern recognition is one of its privileged fields of application. When the learning phase ends, the neural net demarcates at best the decision regions delivering an exact answer for the learning set. The used algorithm for the learning phase is usually based on the back propagation rule [1].

Although this learning algorithm has proved its efficiency in solving numerous problems, it presents several disadvantages. The most important is the absence of rules allowing to fix the neural net architecture. The consequences can be redhibitory, because of the dependence between the learning capacity of the neural network and its architecture [2]. An under-dimensioned neural network with respect to a posed problem is revealed unable to perform a correct demarcation of the decision regions, leading to a miss-classification of the learning set. Likewise, an over-dimensioned neural network will tend to "memorize" or, in others words, to associate each point to a decision region and will not be efficient to generalize. On the other hand, it has been proved recently that learning a neural network by back propagation is a NP-complete problem [3].

In section one, we present the basis of a new method to design the one hidden layer perceptron architecture restricted to the convex problem solution in pattern recognition. The first idea of this approach is inspired by the paper of Lipmann [4] in which he brought emphasis to the relationship between the geometrical and topological properties of the decision regions of a pattern recognition problem and the neural network structure. The method is derived from the gift-wrapping algorithm [5] which is mainly used in computational geometry for the construction of convex hulls in a d-dimensional euclidean space. For a better understanding we only present the two dimensional version: the Jarvis's march algorithm [5].

In section two we show that edge detection can be solved by a neural network with only one hidden layer as an alternative to classical methods like Sobel and Prewitt [6] approaches which employ the local maxima module of the gradient.

Finally, in section three, we provide some numerical results using the geometrical method described above. The performances are very similar to those obtained by a Prewitt mask since the learning sample set employed to train the neural network was an image filtered by this edge operator.
1. USING A GEOMETRICAL ALGORITHM AS LEARNING RULE FOR NEURAL NETS

1.1. Definitions and notations
We present beforehand some geometrical definitions concerning computational geometry.

linear variety
Given $k$ distinct points $p_1, \ldots, p_k$ in $E^d (k \leq d)$, the linear combination
\[ a_1p_1 + a_2p_2 + \ldots + a_{k-1}p_{k-1} + (1 - a_1 - \ldots - a_{k-1})p_k \]
($a_j \in \mathbb{R}, j = 1,\ldots,k-1$) is a linear variety of dimension $(k-1)$ in $E^d$.

Convex set
A domain $D$ in $E^d$ is convex if, for any two points $p_1$ and $p_2$ in $D$, the segment $[p_1p_2]$ is entirely contained in $D$.

Convex hull and polytope
The convex hull of a set of points $S$ in $E^d$ is the boundary of the smallest convex domain in $E^d$ containing $S$.

The convex hull of a finite set of points in $E^d$ is a convex polytope and vice versa. A two-dimensional polytope is called a polygon.

A polyhedral set in $E^d$
A polyhedral set in $E^d$ is the intersection of a finite set of closed half-spaces (a half-space is the portion of $E^d$ lying on one side of a hyperplane).

A polyhedral set is convex, since a half-space is convex and the intersection of convex sets is also convex.

A convex polyhedral set partitions the space $E^d$ in two disjoint regions, the interior bounded and the exterior unbounded that are separated by the polyhedral surface. The term polyhedral denotes the union of the boundary and of the interior.

1.2. The geometrical algorithm

The Jarvis's March algorithm determines the convex polygon edges containing a set of points in a two-dimensional euclidean space. To initialize the algorithm, the knowledge of at least one of the convex polygon vertices is required. We choose the point having the lowest abscissa. This point called $p_1$ is certainly a hull vertex. Our purpose is to find the next consecutive vertex $p_2$ on the convex hull. The point $p_2$ is the one having the least polar angle or equal to zero with respect to $p_1$. Likewise, the next point $p_3$ has the least polar angle superior or equal to zero with respect to $p_2$ taken as origin, and so forth. The algorithm stops when the last detected vertex $p_n$ coincides with vertex $p_1$. If $h$ is the actual number of vertices of the convex hull, the Jarvis's March algorithm runs in $O(hN)$ time, where $N$ is the number of points belonging to the convex set.

In the $d$-dimensional case, the basic idea of the algorithm remains unchanged and the gift-wrapping algorithm may be used to find the polytope of a finite convex hull. In this case the algorithm complexity becomes $O((N^2)^{1/d})+O((N^2)^{1/d})O(N^2)$.

We give afterwards the steps for the construction of one hidden layer neural network applied to pattern recognition restricted to convex problems. We want to discriminate between two disjoint pattern classes, $C_1$ and $C_2$, characterized by $d$ real parameters $\{a_1, a_2, \ldots, a_d\}$. The patterns are represented by points in a $E^d$ space. The point coordinates correspond to the pattern classification parameters $a_j$. The gift-wrapping algorithm gives us the minimal convex polytopes containing each convex set of points. If it exists at least a convex polytope, as defined previously, which contains all the points belonging to only one class, then the two classes can be effectively separated by the convex polytope provided by the algorithm.

In this case it is straightforward to construct a one hidden layer neural network from the facets of the polytope. The input to the neurons of the hidden layer is the pattern. Each polytope facet forms a linear separation $a_1p_1 + a_2p_2 + \ldots + a_{k-1}p_{k-1} + (1 - a_1 - \ldots - a_{k-1})p_k$ ($a_j \in \mathbb{R}, j = 1,\ldots,k-1$) in $E^d$ of the delimited class. The linear separation coefficients $a_j$ are computed from the k-vertices $p_j$ which define the polytope facet. Each linear separation corresponds to a hidden layer neuron [4] and the coefficients $a_j$ correspond to the neuron coefficients. The neurons of the hidden layer are thresholded by a sign function. The single neuron in the output layer is the AND logical function of the binary output from the hidden layer neurons. Thus we obtain a neural net where the architecture and coefficients are fully determined (see figure 1).
Although the two pattern classes discrimination is made without error, in certain cases the problem can be solved using a polytope with less facets. To overcome this drawback, we suggest a method which eliminates the useless facets of the polytope. A polytope facet (or neuron) is useless if after its elimination the neural net output remains the same for all the learning set patterns. This operation is performed in $O(nN)$ time.

2. EDGE DETECTION USING A NEURAL NETWORK

An image can be considered like a continuous function $f(m,n)$ with a gray-level value for each pixel position $(m,n)$. Its gradient denotes the gray-level variation and will be maximum in the direction of the edge. Pixels with a high gray-level variation correspond conventionally to edge points in the image. Let $H$ be a $(p \times p)$ mask. We define, for an arbitrary image $U$, their correlation at location $(m,n)$ as

$$<U,H>_{m,n} = \sum_{i} \sum_{j} h(i,j) u(i+m,j+n)$$

$$= U(m,n) \otimes h(-m,-n) \quad (1)$$

For gradient operators a set of masks $\{H_1, H_2, ..., H_k \}$ is used to measure the gradient of the image $u(m,n)$ in $k$ directions. Defining the $i$-directional gradient

$$g_i(m,n) = <U, H_i>_{m,n} \quad (2)$$

the gradient vector magnitude is often calculated as

$$g(m,n) = \max_{i=1}^{k} ||g_i(m,n)|| \quad (3)$$

This function agrees with the membership union function $\mu_{\tilde{A}}(m,n)$ of the fuzzy logic defined as

$$\mu_{\tilde{A}}(m,n) = \max_{i=1}^{k} \{ \mu_{\tilde{A}_i}(m,n) \} \quad (4)$$

where $\mu_{\tilde{A}_i}(m,n)$ is the membership function of a fuzzy set $\tilde{A}_i$ which corresponds to the function $g_i(m,n)$ and $\tilde{A}$ is the fuzzy set of all the image points $(m,n)$.

The membership union function $m(m,n)$ represents the degree of membership of couple $(m,n)$ relative to the class $\tilde{A}$. We suppose that the fuzzy set $\tilde{A}$ is normal, i.e., the range of the values of the membership function is a subset of the nonnegative real numbers with the unity as the least upper bound.

The pixel location $(m,n)$ is declared an edge location if $\mu_{\tilde{A}}(m,n)$ exceeds some threshold $t$. The locations of edge points constitute an edge map $e(m,n)$, which is defined as

$$e(m,n) = \begin{cases} 1, & (m,n) \in I_g \\ 0, & \text{otherwise} \end{cases}$$

where

$$I_g = \{(m,n) : \mu_{\tilde{A}}(m,n) > t \}$$

The function $\mu_{\tilde{A}}(m,n)$ can be also defined using the membership intersection function and the membership function of the complement as follows

$$\mu_{\tilde{A}}(m,n) = 1 - \min_{i=1}^{k} \{ 1 - \mu_{\tilde{A}_i}(m,n) \} \quad (5)$$

Let us redefine the edge location function $e(m,n)$ as

$$e(m,n) = \begin{cases} 1, & (m,n) \in I_g \\ 0, & \text{otherwise} \end{cases}$$

where

$$I_g = \{(m,n) : \min_{i=1}^{k} \{ 1 - \mu_{\tilde{A}_i}(m,n) \} > t-1 \} \quad (6)$$

Expressed in this form, it is clear that the edge location function is a convex function due to the convexity of the fuzzy intersection function [7].

Let us consider $\mu_{\tilde{A}}(m,n)$ like a couple of hyperplanes ( $|g_i(m,n)|$ denotes two hyperplanes with the same coefficients but opposite signs) in the $E^d$ euclidean space. Therefore, the edge location function denotes the intersection of a finite half-spaces set resulting in a polyhedral convex set [5]. This half-space set forms an $d$-dimensional polytope which can be transformed to a one-hidden layer neural network as depicted in the precedent section. Points outside of the polyhedral sub-space form the edge pixel class, while others are positioned inside the polyhedral. The polytope constitutes the frontier between the two classes. The frontier position depends on the choice of the threshold $t$. A high value of $t$ increases the number of points inside the polyhedral sub-space and so reduce the number of edge points. Nevertheless, the polyhedral form remains the same independently of the threshold value.
3. EXPERIMENTAL RESULTS

We propose to find a new edge operator by a learning procedure. To this end, we use an image and its corresponding edge (see images 1-b and 1-b resp.), resulted from a Prewitt mask, as learning sample set. The images and the Prewitt edge operator have been obtained from the SIMPA library (Signal and Images softwares P'ackages - GDR 134 CNRS) [8]. The sample set for the learning phase contains one hundred samples of edge pixels and the same quantity of points which do not belong to the image edge. The input dimension of the neural network is four, which correspond to the orthogonal neighbors of the pixel to be evaluated.

At the end of the learning phase (which includes the elimination of useless neurons), we obtain a neural network derived from the polyhedral frontier between the two classes of pixels. The number of neurons for the hidden-layer is approximately ten (this number depends on the distribution of the learning samples but remains practically the same after each learning phase). In image 1-c, we illustrate the edge image given from the neural network for the learning image. As one can observe, the results are very similar to those obtained by the Prewitt mask.

4. CONCLUSION

An alternative method to neural network design and train, restricted to the convex problems solution for pattern recognition, has been presented. It has been applied to the edge detection and similar results have been obtained than classical edge operators. We summarize afterwards the inner advantages of the method:

1) The method differentiates the complexity of a problem in three categories: linear problems, convex problems and complex problems.

2) If the problem is convex, the method provides its neural network architecture (number of neurons in the hidden-layer) and the coefficient values.

3) The method insures a bounded learning time, which is polynomial in time and depends on problem dimension.

The main advantages of the method when applied to the edge detection are:

1) The edge detection can be thought to a neural network by sample in a polynomial time. We have chosen a Prewitt mask as a model for the learning phase and have obtained similar results but a more appropriate edge operator or a synthetical edge may be used to this goal.

2) The resulting neural network architecture matches well with the classical perceptron model and can be easily transformed to a classical edge operator structure.

ACKNOWLEDGMENTS

The authors want to thank Dr. J.M. Chassery for his valuable comments on an early unpublished version of this paper. We would also like to thank Mr. Dai Mo and Mr. A. Zoighadri for their scientific contributions.

REFERENCES