A RUNNING WALSH-HADAMARD TRANSFORM ALGORITHM AND ITS APPLICATION TO ISOTROPIC QUADRATIC FILTER IMPLEMENTATION

G. Deng and A. Ling

School of Electronic Engineering, La Trobe University
Bundoora, Victoria 3083 Australia
Phone: +61 3 479 2036; Fax: +61 3 471 0524
Email: d.deng@ee.latrobe.edu.au

ABSTRACT

Two problems associated with adaptive isotropic quadratic filters are the computational complexity and the speed of convergence. This paper presents a transform domain implementation scheme to solve these problems. A new implementation of the filter using the Walsh-Hadamard transform (WHT) is described. A running WHT (RWHT) algorithm is also proposed to reduce the computational cost. Theoretical analysis shows that the number of operations of the WHT implementation (using the RWHT) is considerably less than that of the direct implementation. The advantage of using the WHT implementation is illustrated by modelling a real nonlinear system. Results show that the WHT implementation converges significantly faster than the direct implementation.

1. INTRODUCTION

The second order Volterra filter, which is based on the input-output relations expressed in the form of a second order discrete Volterra series, has been extensively studied and has been employed in system identification, channel equalization, echo cancellation and image processing [1-3].

A second order Volterra filter mainly consists of a linear part and a quadratic part described as follows:

$$y(n) = h_0 + A\boldsymbol{x}^T + \boldsymbol{x}\ B\boldsymbol{x}^T\ , \qquad (1)$$

where $y(n)$ is the filter output, h_0 is the constant to make $y(n)$ an unbiased estimate, A and B are the linear and quadratic kernels, respectively, and $\boldsymbol{x}$ is the input vector given by:

$$\boldsymbol{x} = [x(n), x(n-1), \ldots, x(n-N+1)]\ . \qquad (2)$$

The linear kernel is a (1xN) vector, and the quadratic kernel is an (NxN) matrix, which is represented as:

$$B = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1N} \\ b_{21} & b_{22} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ b_{N1} & \cdots & \cdots & b_{NN} \end{bmatrix}. \qquad (3)$$

Owing to the complexity of the quadratic kernel, research has been focused on the design and implementation issues [2,3]. To reduce the complexity of the quadratic filter the matrix B is assumed to be a symmetric matrix, ie. $b_{j,i} = b_{i,j}$. The symmetric property permits fast implementation schemes using a number of matrix decomposition techniques [2], such as lower-upper triangular, singular value, Jordan and Walsh-Hadamard transform (WHT). Alternatively, other techniques such as multi-memory decomposition [4] and reduced-rank decomposition [5] have been recently proposed.

To further reduce the complexity of the quadratic filter, Rampoli [3] has recently introduced the concept of isotropic quadratic kernel, which is also a symmetric matrix. The elements of an isotropic kernel B have the relationship:

$$b_{i,j} = b_{N-i+1,N-j+1}\ ,\ i,j = 1,2,\ldots,N\ . \qquad (4)$$

As an example, the quadratic kernel of size (4x4) is given below:

$$B = \begin{bmatrix} \boldsymbol{b_{11}} & \boldsymbol{b_{12}} & \boldsymbol{b_{13}} & \boldsymbol{b_{14}} \\ b_{12} & \boldsymbol{b_{22}} & \boldsymbol{b_{23}} & b_{13} \\ b_{13} & b_{23} & b_{22} & b_{12} \\ b_{14} & b_{13} & b_{12} & b_{11} \end{bmatrix}. \qquad (5)$$

It is observed that the matrix is symmetric along the two diagonal lines, and the independent kernel elements are those in bold-face. The number of independent kernel elements for a matrix of size N is:

$$\left(N^2 + 2N\right)\big/4 \qquad N \text{ even}$$

$$\left((N+1)/2\right)^2 \qquad N \text{ odd.}$$

In general, the number of operations required by a direct implementation of a quadratic filter to process one data sample is listed in Table 1.

Although the computational complexity of a quadratic filter is reduced by using an isotropic quadratic kernel, the amount of computation is still very large for a real time adaptive system. Since the isotropic quadratic kernel has strong symmetric properties, a natural way to further reduce the computational costs is to perform linear transformation on the kernel such that the resultant kernel is a sparse matrix or contains a lot of zero elements.

In section 2, the implementations of the 1-D isotropic quadratic filter using the Walsh Hadamard transform (WHT) is briefly reviewed. In Section 3, a running WHT (RWHT) algorithm is proposed to reduced the computational cost for performing the WHT. Using the RWHT algorithm the transform coefficients are updated recursively when the input data is shifted. In section 4, experimental results on both simulated data and real data are presented. Finally, conclusions are made in section 5.

2. THE WHT IMPLEMENTATION [6]

The WHT matrix, a $(N \times N)$ matrix $(N = 2^k, k = 1,2,3\ldots)$, is usually defined recursively using a block-matrix decomposition as follows:

$$W_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \qquad (6a)$$

and

$$W_n = \frac{1}{\sqrt{2}}\begin{bmatrix} W_{n-1} & W_{n-1} \\ W_{n-1} & -W_{n-1} \end{bmatrix}. \qquad (6b)$$

In the following discussion, the WHT matrix is denoted by W for simplicity. It is easy to see that

$$y_q(n) = \boldsymbol{x}B\boldsymbol{x}^T = \boldsymbol{x}W^T WBW^T W\boldsymbol{x}^T = \boldsymbol{X}D_w\boldsymbol{X}^T \quad (7)$$

where $\boldsymbol{X} = \boldsymbol{x}W^T$ is the WHT of the input vector and $D_w = WBW^T$ is the WHT of the quadratic kernel. Without losing generality, a vector of four elements is used in the following presentation. If the input vector is rearranged as:

$$\boldsymbol{x}' = [x(n), x(n-1), x(n-3), x(n-2)], \qquad (8)$$

then the corresponding isotropic kernel B' is represented by:

$$B' = \begin{bmatrix} b_{11} & b_{12} & b_{14} & b_{13} \\ b_{12} & b_{22} & b_{13} & b_{23} \\ b_{14} & b_{13} & b_{11} & b_{12} \\ b_{13} & b_{23} & b_{12} & b_{22} \end{bmatrix} \qquad (9)$$

It can easily be shown that $\boldsymbol{x}B\boldsymbol{x}^T = \boldsymbol{x}'B'\boldsymbol{x}'^T$. This new quadratic kernel can actually be decomposed into four sub-matrices of the form:

$$B' = \begin{bmatrix} B_1 & B_2 \\ B_2 & B_1 \end{bmatrix}, \qquad (10)$$

where the size of the matrices B_1 and B_2 is half of that of B'. Although this property is derived from a (4x4) kernel, one can prove that it is generally true for any isotropic kernel of size $N=2^k$.

As a result, the transformed kernel is a block-diagonal matrix

$$D_w' = WB'W^T = \begin{bmatrix} \alpha & \tau & 0 & 0 \\ \tau & \beta & 0 & 0 \\ 0 & 0 & \theta & \lambda \\ 0 & 0 & \lambda & \delta \end{bmatrix} \qquad (11)$$

where the variables α, β, θ, δ, τ and λ represent the independent kernel elements. Now the new quadratic filter becomes:

$$y_q(n) = \boldsymbol{X}D_w\boldsymbol{X}^T = \boldsymbol{X}'D'_w\left(\boldsymbol{X}'\right)^T. \qquad (12)$$

An important issue is the computational complexity. The number of operations needed for the WHT implementation have been calculated. The formulae are listed in Table 1, which shows that the WHT implementation requires less multiplication operations than that required for the direct implementation. The extra $N\log_2 N$ addition operations due to the computation of the fast WHT (FWHT) can be reduced by using a running WHT algorithm described below.

3. A RUNNING WHT ALGORITHM

Consider the case where N=4, ie $\boldsymbol{x_2} = [x(n), x(n-1), x(n-2), x(n-3)]$. Here the subscript is used to indicate the number of elements in a vector. For example, notations $\boldsymbol{x_2}$ and $\boldsymbol{x_3}$ represent vectors of 4 and 8 elements, respectively. Since $W = W^T$, the superscript of WHT matrix "T" has also been omitted in the following discussion. The WHT of the input vector is represented by:

$$\boldsymbol{X_2} = \boldsymbol{x_2}W_2 = [X(0), X(1), X(2), X(3)] \qquad (13)$$

Let the WHT of the shifted vector $\tilde{\boldsymbol{x}}_2 = [x(n+1), x(n), x(n-1), x(n-2)]$ be represented by:

$$\tilde{\boldsymbol{X}}_2 = \tilde{\boldsymbol{x}}_2 W_2 = \left[\tilde{X}(0), \tilde{X}(1), \tilde{X}(2), \tilde{X}(3)\right]. \qquad (14)$$

It is easy to show that:

$$\tilde{\boldsymbol{X}}_2 = [\boldsymbol{X_2}]_R + \boldsymbol{U_2}, \qquad (15)$$

where

$$[\boldsymbol{X_2}]_{\boldsymbol{R}} = [X(0), -X(1), X(3), -X(2)] \tag{16}$$

is a re-ordered vector of $\boldsymbol{X_2}$, $\boldsymbol{U_2} = [\Delta, \Delta, \Delta, \Delta]$ is an updating vector and

$$\Delta = x(n+1) - x(n-N+1)\,. \tag{17}$$

In the following, the superscripts 1 and 2 are used to represent the first and second half of a vector respectively. For example, $\boldsymbol{X_3^1}$ and $\boldsymbol{X_3^2}$ represent the first and the second half of a vector $\boldsymbol{X_3}$.

Using the block matrix decomposition method, one can easily derive the WHT transform equations for vectors $\boldsymbol{x_3}$ and $\tilde{\boldsymbol{x}}_{\boldsymbol{3}}$ (k=3, N=8):

$$\begin{aligned} \boldsymbol{X}_3 &= \begin{bmatrix} \boldsymbol{X_3^1} & \boldsymbol{X_3^2} \end{bmatrix} \\ &= \boldsymbol{x_3} W_3 \\ &= \begin{bmatrix} \boldsymbol{x_3^1} & \boldsymbol{x_3^2} \end{bmatrix} \begin{bmatrix} W_2 & W_2 \\ W_2 & -W_2 \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{x_3^1} W_2 + \boldsymbol{x_3^2} W_2 & \boldsymbol{x_3^1} W_2 - \boldsymbol{x_3^2} W_2 \end{bmatrix} \end{aligned} \tag{18}$$

and

$$\begin{aligned} \tilde{\boldsymbol{X}}_3 &= \begin{bmatrix} \tilde{\boldsymbol{X}}_{\boldsymbol{3}}^{\boldsymbol{1}} & \tilde{\boldsymbol{X}}_{\boldsymbol{3}}^{\boldsymbol{2}} \end{bmatrix} \\ &= \tilde{\boldsymbol{x}}_{\boldsymbol{3}} W_3 \\ &= \begin{bmatrix} \tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{1}} & \tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{2}} \end{bmatrix} \begin{bmatrix} W_2 & W_2 \\ W_2 & -W_2 \end{bmatrix} \\ &= \begin{bmatrix} \tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{1}} W_2 + \tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{2}} W_2 & \tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{1}} W_2 - \tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{2}} W_2 \end{bmatrix} \end{aligned} \tag{19}$$

Using equation 15, the updating equation for the first half of the vector is given by:

$$\begin{aligned} \tilde{\boldsymbol{X}}_{\boldsymbol{3}}^{\boldsymbol{1}} &= \tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{1}} W_2 + \tilde{\boldsymbol{x}}_{\boldsymbol{3}}^{\boldsymbol{2}} W_2 \\ &= \left[\boldsymbol{x_3^1} W_2\right]_R + \boldsymbol{U_3^1} + \left[\boldsymbol{x_3^2} W_2\right]_R + \boldsymbol{U_3^2} \\ &= [\boldsymbol{X_3^1}]_R + \boldsymbol{U_3^1} + \boldsymbol{U_3^2} \end{aligned} \tag{20a}$$

where $[\boldsymbol{X_3^1}]_R$ is the re-ordered version of $\boldsymbol{X_3^1}$ (in the same way as equation 16), $\boldsymbol{U_3^1}$ and $\boldsymbol{U_3^2}$ are the updating vectors. Following the same method, the updating equation for the second half of the vector is given by:

$$\tilde{\boldsymbol{X}}_{\boldsymbol{3}}^{\boldsymbol{2}} = [\boldsymbol{X_3^2}]_R + \boldsymbol{U_3^1} - \boldsymbol{U_3^2} \tag{20b}$$

A combination of equations (20a) and (20b) yields:

$$\tilde{\boldsymbol{X}}_{\boldsymbol{3}} = [\boldsymbol{X_3}]_R + \boldsymbol{U_3}\,. \tag{21}$$

where

$$\boldsymbol{U_3} = [\Delta, \Delta, \Delta, \Delta, \Sigma, \Sigma, \Sigma, \Sigma] \tag{22}$$

and

$$\Sigma = x(n+1) + x(n-N+1) - 2 * x(n - N/2 + 1)\,. \tag{23}$$

Therefore, one can easily prove the following general updating equation:

$$\tilde{\boldsymbol{X}}_{\boldsymbol{k}} = [\boldsymbol{X_k}]_R + \boldsymbol{U_k}\,. \tag{24}$$

The re-ordered vector $[\boldsymbol{X_k}]_R$ and the updating vector $\boldsymbol{U_k}$ can be obtained by following the above method. Thus, when the input signal shifts one sample, it is not necessary to perform the WHT for the new vector. The WHT of the new vector can be obtained by adding the updating vector to the re-ordered version of the previous transformed vector. In this paper, this method will be called the *running* WHT (RWHT).

It has been proved that the number of addition operations T_k required by the RWHT is:

$$T_k = 2^{k-2} + 2T_{k-1}\,, \quad k \ge 3 \quad \textit{and} \quad T_2 = 5 \tag{25}$$

It has also been proved that the number of addition operations required by the RWHT is less than that required by FWHT, ie,

$$T_k < N \log_2 N = k2^k\,, \quad k \ge 2\,. \tag{26}$$

A more interesting result is that ,

$$k2^k - T_k > \frac{3}{4} 2^k \text{, when } k \ge 3\,. \tag{27}$$

This means that the RWHT requires at least $\frac{3}{4}2^k$ less operations than FWHT. The proofs of the above results are given in [7].

Table 1 summarises the number of operations required for the direct and the WHT implementations of an isotropic quadratic filter. $N = 2^k$ is the size of the data vector, p represents the extra addition operation to calculate the WHT, $p = N \log_2 N$ for a FWHT and $p = T_k$ for a RWHT.

Table 1 The number of operations required by the direct and the WHT implementations

	Multiplications	Additions
Direct implementation	$N(\frac{3}{4}N + 1)$	$\frac{3N^2 + 2N - 4}{4}$
WHT/RWHT implementation	$N(\frac{N}{2} + 1)$	$\frac{N^2}{2} + p - 1$

4. SIMULATION RESULTS

An adaptive second order Volterra filter (the size of the quadratic filter kernel being (4x4) and the linear kernel size being 4) is applied to a typical nonlinear system modelling problem, which is shown in Fig. 1. In [6], it has been shown using simulated data that the WHT implementation results in a significantly faster convergence than the direct implementation. We have also performed simulations using a real nonlinear system. The nonlinear system is simulated by a circuit whose block diagram is shown in Fig.2. The pseudo random (PN) code generator is used to simulate the baseband signal source. The output of this nonlinear system (distorted signal) together with the original signal is sampled and processed using the standard LMS algorithm [8]. Fig.3 shows that the distorted signal has been successfully recovered. This also confirms our early study [6] on the performance of adaptive quadratic filters. Because the nonlinear system shown in Fig. 3 is an approximation of a nonlinear channel, the proposed technique is a useful nonlinear channel Equalizer.

5. CONCLUSION

We have proposed a running WHT algorithm which requires less operations than the fast WHT. We have also applied the RWHT to implement the adaptive isotropic quadratic filter. Our new implementation has two advantages: (1) it requires less operations than the direct implementation, (2) it has better performance in modelling a nonlinear system. The RWHT can also be applied to adaptive linear filters.

6. REFERENCES

[1] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Magazine*, v. 8, No. 6, pp. 10-26, July, 1991.

[2] G. L. Sicuranza, "Quadratic filters for signal processing," *Proc. IEEE*, v. 80, No. 8, pp. 1263-1285, Aug. 1992.

[3] G. Ramponi, "Bi-impulse response design of isotropic quadratic filters," *Proc. IEEE*, v. 78, No. 4, pp. 665-677, Apr. 1992.

[4] W. A. Frank, "MMD-an efficient approximation to the 2nd order Volterra filter," *Proc. ICASSP'94*, v.3, pp. 517-520, 1994.

[5] S. M. Lau, S. H. Leung and B. L. Chan, "A reduced rank second-order adaptive Volterra filter," *Proc. Int. Symposium on Signal Processing and Its Applications*, pp. 561-564, Gold Coast, Australia, Aug. 1992.

[6] G. Deng, "Implementation of isotropic quadratic filters using Walsh-Hadamard transform," *Electronics Letters*, v. 31, pp. 1422-1423, Aug., 1995.

[7] G. Deng, "Implementation of isotropic quadratic filters using Walsh-Hadamard transform and the discrete cosine transform," Submitted to *IEE Proc. Vision, Image and Signal Processing*.

[8] B. Widrow and S. D. Stearns, *Adaptive signal processing.* Prentice-Hall, Inc. Englewoods Cliffs, N. J. 1985.

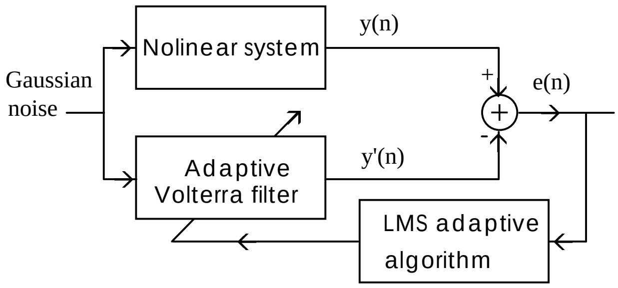

Fig. 1 Nonlinear system modelling using adaptive Volterra filter

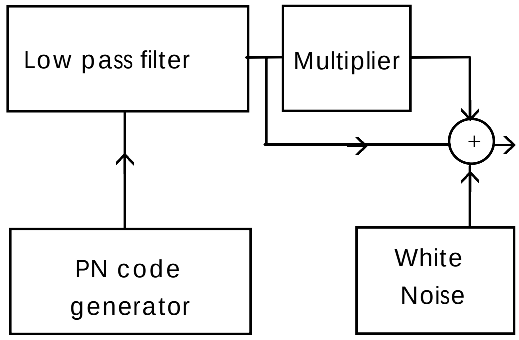

Fig.2 Block diagram of a circuit for nonlinear system simulation

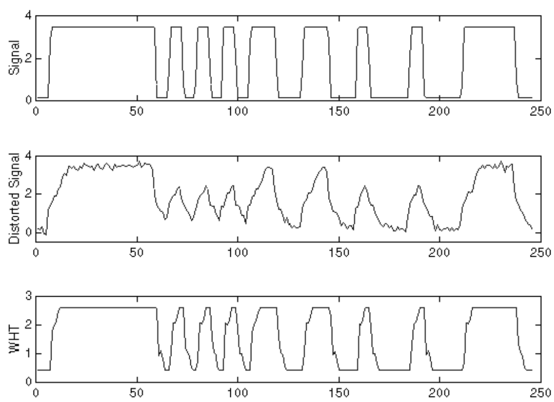

Fig. 3 Original signal (upper), distorted signal (middle) and recovered signal (lower)