Buletinul Ştiinţific al Universităţii "Politehnica" din Timişoara

Seria ELECTRONICĂ şi TELECOMUNICAŢII
TRANSACTIONS on ELECTRONICS and COMMUNICATIONS

Tom 49(63), Fascicola 2, 2004

Efficient Implementation of The Second Order Volterra Filter

Georgeta Budura, Corina Botoca[1]

Abstract – This paper investigates a new implementation of the second order isotropic filter using the Walsh Hadamard Transform. The new implementation is used in a second order Volterra filter. It's performances are evaluated in a typical nonlinear system identification application. For the second order adaptive Volterra filter an LMS adaptive algorithm with variable step size for the linear and the quadratic parts is proposed. Experimental results show that by using the WHT, the computational complexity of the adaptive second order filter is considerably reduced and the convergence rate of this filter is also significantly improved. Adaption is working well even for high levels of the input signal. The mean-squared error of the proposed filter is compared with those of a classic second order LMS adaptive filter.

Keywords: Adaptive nonlinear filter, Walsh Hadamard transform, LMS adaptation algorithm with variable step size.

I. INTRODUCTION

The second order Volterra filter has been increasing research interest in nonlinear filtering techniques. It has been extensively studied and has been employed in system identification, channel equalization, echo cancellation and image processing [1]. A second order Volterra filter consists mainly of a linear and a quadratic part described as follows:

$$y[n] = h_0 + H_1 X_n^T + X_n H_2 X_n^T \tag{1}$$

where h_0 is a constant required to make the output signal $y[n]$ an unbiased estimate, H_1 and H_2 are the linear and quadratic kernels respectively, and X_n is the input vector of the form: $X_n = \left[x[n], x[n-1], \ldots, x[n-N+1]\right]$, where N represents the filter length or filter memory.

The linear kernel is a *1xN* vector, and the quadratic, or second order, kernel is a *NxN* matrix:

$$H_1 = [h_1\ h_2\ h_3\ h_4] \tag{2}$$

$$H_2 = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{414} \end{bmatrix} \tag{3}$$

Many researches have been focused on the implementation of the quadratic filter, considered a prototype for the nonlinear filters. In the above representation we consider the same memory for the linear and the second order filter. The most general case would allow a different memory for each nonlinearity order.

A further simplification can be made to (3) by considering symmetric Volterra kernels. A second order Volterra kernel, having the elements $h_2(n_1, n_2)$, is symmetric if the indices n_1 and n_2 can be exchanged without affecting its value. Any asymmetric Volterra kernel can be easily symmetrised using the method given by [2]. So, most authors considered symmetric Volterra kernels which are in facts symmetric matrices. If the elements of a second order symmetric kernel also have the property $h_{i,j} = h_{N-i+1,N-j+1}$, $i, j = 1,2,\ldots N$, the kernel is called isotropic. This paper investigates a new implementation of the second order isotropic quadratic filter using the Walsh Hadamard Transform (WHT). This new implementation is used to construct an adaptive second order Volterra filter whose performances are evaluated to a typical nonlinear system identification application.

II. NEW IMPLEMENTATION FOR THE QUADRATIC KERNEL

To reduce complexity we consider a second order isotropic filter with the memory dimension N=4. The implementation of the second order kernel represents the major problem for this filter. The new implementation is based on the Walsh-Hadamard Transform.

The WHT is considered mainly for two reasons:

[1] Facultatea de Electronică şi Telecomunicaţii, Departamentul Comunicaţii Bd. V. Pârvan Nr. 2, 300223 Timişoara, e-mail getab@etc.utt.ro

(i) The use of orthogonal transform can improve the performance of LMS adaptive filters.

(ii) The WHT is a fast orthogonal transform which only involves the addition operation.

The WHT matrix is an *NxN* matrix ($N=2^k$, $k=1,2,3,\ldots$) usually defined recursively using a block-matrix decomposition as follows: $W_1 = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$ and $W_n = \frac{1}{\sqrt{2}}\begin{bmatrix} W_{n-1} & W_{n-1} \\ W_{n-1} & -W_{n-1} \end{bmatrix}$. The WHT matrix is denoted by W in the following discussion. An important property of the WHT matrix is given in eq.4:

$$W \cdot W^T = 1 \tag{4}$$

For the new implementation we consider the second order kernel given in eq.3 and the isotropic property:

$$H_2 = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{12} & h_{22} & h_{23} & h_{13} \\ h_{13} & h_{23} & h_{22} & h_{12} \\ h_{14} & h_{13} & h_{12} & h_{11} \end{bmatrix} \tag{5}$$

It can be easily seen that the matrix H_2 is symmetric according to its both diagonals. The nonlinear filter produces the output signal $y_2[n]$:

$$y_2[n] = X_n H_2 X_n^T = X_n W W^T H_2 W W^T X_n^T = X_{nW} H_{2W} X_{nW}^T \tag{6}$$

where: $X_{nW} = X_n W$ is the Walsh-Hadamard transform of the input vector and $H_{2W} = W^T H_2 W$ is the WHT of the second order Volterra kernel.

If we rearrange the input vector as: $X_n^{'} = \left[x[n], x[n-1], x[n-3], x[n-2]\right]$, then the corresponding isotropic kernel $H_2^{'}$ is:

$$H_2 = \begin{bmatrix} h_{11} & h_{12} & h_{14} & h_{13} \\ h_{12} & h_{22} & h_{13} & h_{23} \\ h_{14} & h_{13} & h_{11} & h_{12} \\ h_{13} & h_{23} & h_{12} & h_{22} \end{bmatrix} \tag{7}$$

We can easily demonstrate that:

$$X_n H_2 X_n^T = X_n^{'} H_2^{'} (X^{'})_n^T \tag{8}$$

This new second order kernel can actually be decomposed into four sub matrices of the form:

$$H_2^1 = \begin{bmatrix} H_{11} & H_{12} \\ H_{12} & H_{11} \end{bmatrix} \tag{9}$$

where the size of matrices H_{11} *and* H_{12} is half that of $H_2^{'}$. In that case the transformed kernel is a block diagonal matrix:

$$H_{2W}^1 = \begin{bmatrix} a\,b\,0\,0 \\ b\,c\,0\,0 \\ 0\,0\,d\,e \\ 0\,0\,e\,f \end{bmatrix} \tag{10}$$

where the variables a,b,c,d, e and f represent the independent elements. The output of the new second order filter now becomes:

$$y_2[n] = X_{nW} H_{2W} X_{nW}^T = X_{nW}^{'} H_{2W} (X_{nW}^{'})^T \tag{11}$$

This new implementation raises two problems:

-Is the reduction of the computational complexity accomplished? We have calculated the number of operations required for a direct implementation and those required for the new implementation. The results listed in Table 1 show that the new implementation requires less multiplication operations.

Tabel I

	Multiplications	Additions
Direct implementation	$N[(3/4)N+1]$	$(3N^2+2N-4)/4$
New implementation	[N(N/2)+1]	$N^2/2+N\log_2 N-1$

- Can the TWH of the input vector($X_{nW} = X_n W$) be substituted by the WHT of the rearranged input vector ($X_{nW}^{'} = X_n^{'} W, X_n^{'} = [x(n)\; x(n-1)\; x(n-3)\; x(n-2)]$)? We easily find the relationship by examining $X_{nW}^{'}$ and X_{nW}. For example, we have the following relationships between the two transformed vectors $X_{nW}^{'} = [X_{nW}^{'}(1)\; X_{nW}^{'}(2)\; X_{nW}^{'}(3)\; X_{nW}^{'}(4)]$ and $X_{nW} = [X_{nW}(1)\; X_{nW}(2)\; X_{nW}(3)\; X_{nW}(4)]$:

$$\begin{aligned} X_{nW}^{'}(1) &= X_n(1) \\ X_{nW}^{'}(2) &= X_n(4) \\ X_{nW}^{'}(3) &= X_n(3) \\ X_{nW}^{'}(4) &= X_n(2) \end{aligned} \tag{12}$$

Similar relationships can be established for the input vectors of size 8, 16, etc.

III. EXPERIMENTAL RESULTS

The new implementation was applied to a typical nonlinear system modeling problem, as shown in fig.1. An adaptive second-order Volterra filter is used to identify the nonlinear system having the input-output characteristic given in fig.2.

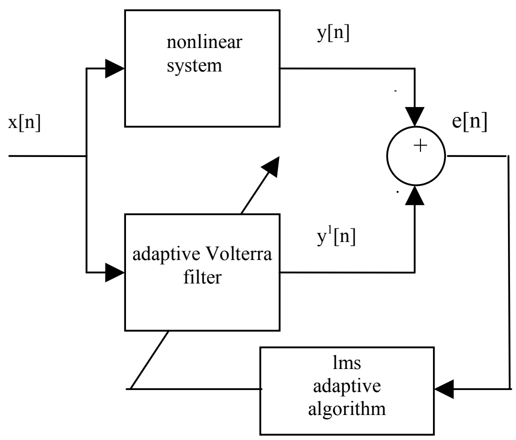

Fig1.Nonlinear system identification using adaptive quadratic filter

The nonlinear system output is given in Eq.13.

$$y[n] = A * X_n + X_n^T * B * X_n \tag{13}$$

where $A = [-0.78\ -1.48\ -1.39\ \ 0.04]$ and

$$B = \begin{bmatrix} 0.54 & 3.72 & 1.86 & -0.76 \\ 3.72 & -1.62 & 0.76 & 1.86 \\ 1.86 & 0.76 & -1.62 & 3.72 \\ -0.76 & 1.86 & 3.72 & 0.54 \end{bmatrix} \tag{14}$$

This system is a slight modification of that used in [1].
For a linear input signal *x[n]*, the resulted output signal is plotted in Fig.2.

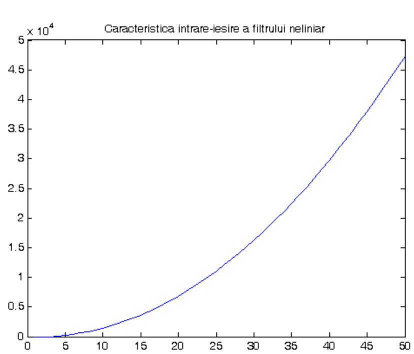

Fig. 2 Input-output characteristic of the nonlinear system

The adaptive filtering or system identification problem being considered is to try to adjust the set of filter weights so that the system output $y^1[n]$ tracks the desired signal $y[n]$. The error signal $e[n]$ is defined as the difference between the desired signal $y[n]$ and the predicted signal $y^1[n]$, as indicated in fig.1. The input process $x[n]$ is assumed to be a zero mean independent sequence with covariance, a positive definite matrix. This simplifying assumption is often made in literature [1,3].
The LMS type adaptive algorithm is a gradient search algorithm which computes a set of filter weights $H_1[k+1], H_2[k+1]$, at the time moment *k+1*, that seeks to minimize the error function, $E[(e[k])^2]$, cosidered at the time moment *k*.
The update equations for the Volterra adaptive filter weights are well known in the literature[3] and are given in eq. 15:

$$\begin{aligned} H_1[k+1] &= H_1[k] + 2\mu_1 e[k] X[k] \\ H_2[k+1] &= H_2[k] + \mu_2 e[k] X^T[k] * X[k] \end{aligned} \tag{15}$$

where μ_1 and μ_2 are in both cases two small positive constants (referred to as the step size) that determine the speed of convergence and also affect the final error of the filter output.
In our simulation we have used the efficient implementation for the quadratic kernel. In this case the update equation for that kernel is:

$$H_{2W}[k+1] = H_{2W}[k] + \mu_2 (X_W^{'}[k])^T * X_W^{'}[k] \tag{16}$$

where H_{2W} and $X_W^{''}$ are the WHT of the quadratic kernel respectively the WHT of the rearranged input vector.
Finally the output of the adaptive Volterra filter, $y^1[n]$, is:

$$y^1[n] = H_1 * X_n + (X_{nW}^{'})^T * H_{2W} * X_{nW}^T \tag{17}$$

The linear kernel is a 1x4 vector and the quadratic kernel is a 4x4 matrix. The input sequence is a random gaussian zero-mean sequence having 1500 values.
The majority of papers examine the LMS algorithm with a constant step size. The choice of the step size reflects a tradeoff between misadjustement and the speed of adaptation. The approximate expressions derived in [3] showed that a small step size causes small misadjustement, but also a longer convergence time constant.
For adaptive Volterra filters the problems seem to be much more complicated. In [3,5] the problems of step size for different order kernels are well discussed. The maximum step size bound is related to the maximum eigenvalue of the autocorrelation matrix of the input vector. Because we consider a second order Volterra filter without DC component included in the

estimation algorithm, the step size bounds for μ_1 and μ_2 are those given in [3]:

$$0 < \mu_1 < \frac{2}{3tr[R_{XX}]} ; 0 < \mu_2 < \frac{2}{3(tr[R_{XX}])^2} \tag{18}$$

In [6], a variable step size LMS algorithm for linear filter is proposed. The step size adjustment is controlled by the square of the prediction error. The motivation is that a large prediction error will cause the step size to increase to provide faster tracking while a small prediction error will result in a decrease in step size to yield smaller misadjustements.
In this paper we have used this variable step size algorithm for both, the linear and the quadratic filter.
The relation for adjusting the step size μ_1 is that given in [6]:

$$\mu_1[k+1] = \alpha\mu_1[k] + \gamma e^2[k] \tag{19}$$

with $0 < \alpha < 1$ and $\gamma > 0$
and

$$\mu_1[k+1] = \begin{cases} \mu_{1\max} & if \;\; \mu_1[k+1] > \mu_{1\max} \\ \mu_{1\min} & if \;\; \mu_1[k+1] < \mu_{1\min} \\ \mu_1[k+1] & otherwise \end{cases} \tag{20}$$

where $0 < \mu_{\min} < \mu_{\max}$. The initial step size $\mu_1[0]$ was chosen to be $\mu_{1\max}$ although the algorithm is not sensitive to this choice. As can be seen from eq.20, the step size μ_1 is always positive and is controlled by the size of the prediction error and the parameters α and γ. A large prediction error increases the step size to provide faster tracking. If the prediction error decreases, the step size will be decreased to reduce the misadjustements. The constant $\mu_{1\max}$ is chosen to ensure that the mean-square error(mse) of the algorithm remains bounded . A sufficient condition for $\mu_{1\max}$ to guarantee bounded mse is:

$$\mu_{1\max} < \frac{2}{3tr(R_{XX})} \tag{21}$$

Usually $\mu_{1\min}$ will be near the value that would be chosen for the fixed step size algorithm. In our simulation the value is $\mu_1 \min = 10^{-7}$. Parameter α must be chosen in the range $(0,1)$ to provide exponential forgetting. A typical value of α that was found to work well in simulations is $\alpha = 0.97$. The parameter γ is usually small ($4,8*10^{-4}$ was used in our simulations.)

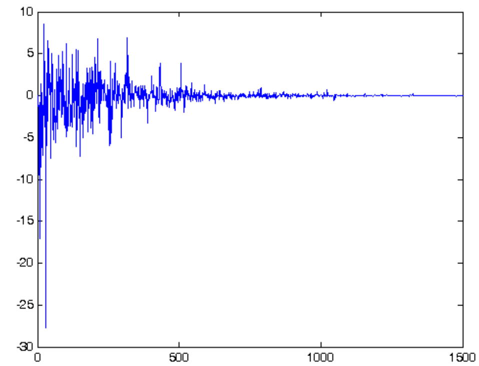

Fig.3 The adaptive filter error with white input signal

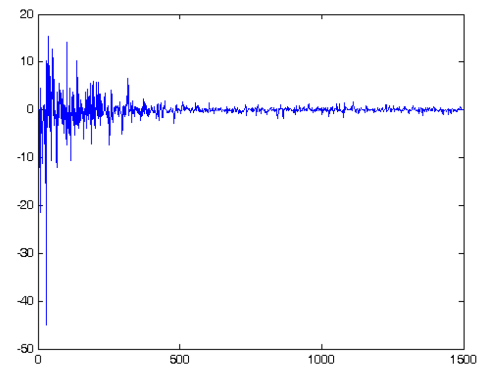

Fig.4 The adaptive variable step size filter error with white input signal

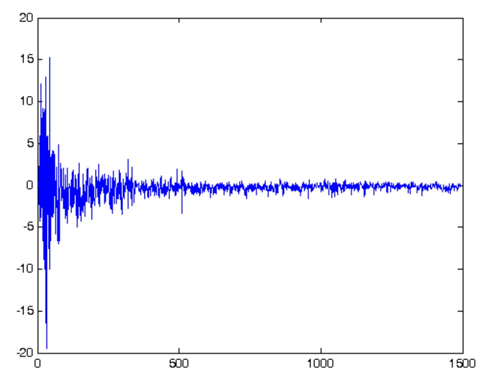

Fig.5 The adaptive variable step size filter error with colored input signal

The colored input signal was generated as specified in eq.22.

$$x[n] = 0.25 * r[n-1] + r[n-2] + 0.25 * r[n-3] \tag{22}$$

where *r[n]* is a random, normal distributed sequence.
In Fig.6 we have compared the mean-squared error of the proposed filter (represented with solid line) with those of a classic second order LMS adaptive filter(represented with doted line) . We also consider the case of the colored input signal for the new filter (represented with dashed line).

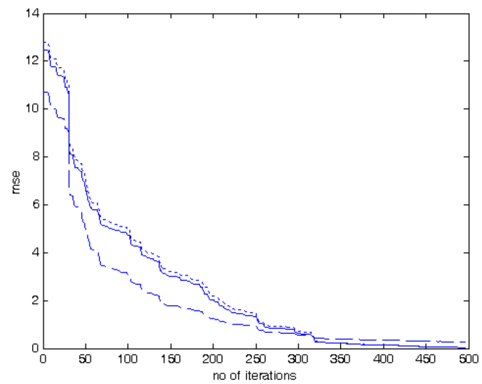

Fig.6 The mean-squared errors for the compared filters

For high level input signal the filter with variable step size still adapts, as can be seen in fig.7.

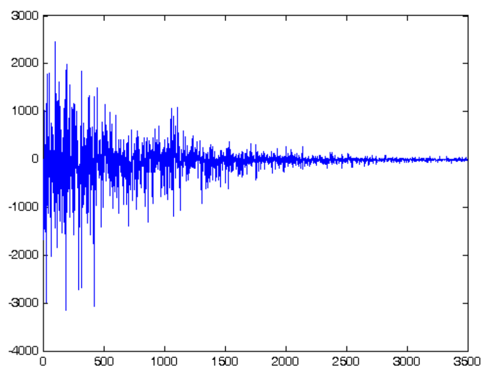

Fig.7. The adaptive variable step size filter error with high level input signal

IV. CONCLUSIONS

We have proposed a new implementation of the isotropic second order filter. This new implementation has two advantages: it requires less operations than the direct implementation and it has better performances in modelling a nonlinear system. We have also proposed a variable step size algorithm which improves the capabilities of the adaptive Volterra filter.

REFERENCES

[1] V.J. Mathews, "*Adaptive Polinomial Filters*", IEEE Signal Processing Magazine, No.7, pp. 10-23, July, 1991.

[2] M. Schetzen, *The Volterra and Wiener theories of nonlinear systems*", Wiley and Sons, New York, 1980.

[3] J. Tsimbinos, "*Identification And Compensation Of Nonlinear Distortion*", Thesis, http://www.unisa.edu.au/html

[4] G.Budura, "*Contributions to the nonlinear systems study using Volterra series*", Thesis, "Politehnica" University of Timisoara, 30 Sept.,1999.

[5] G.Budura, C. Botoca, "*Applications of the Volterra Models in Nonlinear Systems Identification*", Buletinul Universitatii „Politehnica", Seria Electrotehnica, Electronica si Telecomunicatii, Tom 47(61),Fascicola 1-2, pp.196-201, 2002.

[6] R.H. Kwong, E.W. Johnston, "*A Variable Step Size LMS Algorithm*", IEEE Transactions on Signal Processing, Vol.40, No.7, pp.1633-1642.