Computational Efficient Implementation of the Second Order Volterra Filter Based on the MMD Approximation

GEORGETA BUDURA, CORINA BOTOCA
Communications Department
Faculty of Electronics and Telecommunications
Timisoara, Bd. V. Parvan, No.2
ROMANIA

Abstract: - The Volterra series have been successfully and widely applied as a nonlinear system modeling technique. Considered as a prototype, the second order Volterra filter (FV_2) has an increased complexity in comparison with a linear filter. The filter based on the multi memory decomposition (MMD) structure represents a good approximation of the FV_2 and significantly reduces the number of the filter operations. This paper proposes a computational efficient implementation of the MMD filter. The proposed implementation for the MMD filter is studied in a typical nonlinear system identification problem. The simulations show the very good performance of the proposed MMD structure in comparison with the results obtained by using a second order LMS Volterra filter.

Key-Words: - second order Volterra filter, multi memory decomposition, computational efficient implementation

1 Introduction

A Volterra filter (series) represents a wide class of nonlinear systems. The series is a sum of generalized convolutions and can be thought as an extension of the linear case. The major drawback of the Volterra filter is the large number of the filter coefficients. Hence, the Volterra model implementation needs an increased computational power that represents the major drawback in real time applications. Therefore, only low orders nonlinearities can be modeled in an efficient way.

For a discrete-time and causal nonlinear system with memory, with an input $x[n]$ and an output $y_V[n]$, the Volerra series expansion is given by [1]:

$$y_V[n]=\sum_{i=1}^{N}\sum_{k_1=0}^{M-1}\cdots\sum_{k_i=0}^{M-1}h_{iV}[k_1,\cdots,k_i]x[n-k_1]\cdots x[n-k_i] \quad (1)$$

where N represents the model nonlinearity degree. Choosing $N=2$, the input-output relationship of the FV_2 can be expressed:

$$y_V[n]=h_{0V}+\sum_{k_1=0}^{M-1}h_{1V}[k_1]x[n-k_1]+ \sum_{k_1=0}^{M-1}\sum_{k_2=0}^{M-1}h_{2V}[k_1,k_2]x[n-k_1]x[n-k_2] \quad (2)$$

The input-output relation can also be written in terms of nonlinear operators, $\mathcal{H}_i[\ ]$, as indicated in the following relation:

$$y_V[n]=\mathcal{H}_0[x[n]]+\mathcal{H}_1[x[n]]+\mathcal{H}_2[x[n]] \quad (3)$$

In the above representations, the functions h_{iV}, $i=\overline{0,N}$ represent the Volterra kernels associated to the nonlinear operators.

The nonlinear model described by the relations (2) and (3) is called a second order Volterra model. Note that the above representations have the same memory for all the nonlinearity orders. In the most general case, the relation (1) may use different memory for each nonlinearity order. A further simplification can be made to the relation (1) by considering symmetric Volterra kernels.

The kernel $h_{iV}[k_1,\cdots,k_i]$ is symmetric if the indices can be interchanged without affecting its value.

The second order Volterra kernel is a symmetric $M(M+1)$ quadratic matrix. In these conditions the system output can be expressed as indicated in the relation:

$$y_{2V}[n]=\sum_{k_1=0}^{M-1}\sum_{k_2=k_1}^{M-1}h_{2V}[k_1,k_2]x[n-k_1]x[n-k_2] \quad (4)$$

If we consider symmetric kernels, as indicated above, the second order Volterra kernel $h_{2V}[k_1, k_2]$ requires the determination of $M(M+1)/2$ coefficients [2], [3].

In the technical applications the kernel estimation accuracy becomes the major problem [4] - [9].

In situations that imply systems having time varying parameters, the adaptive methods for kernels estimation are widely used.

A big advantage of the Volterra models, if compared with other models, is that the system response is linear with the filter coefficients. The nonlinearity is reflected only by multiple products between the delayed versions of the input signal.

Due to this fact, many methods from linear adaptive filter theory can be adopted for the Volterra filter modeling. For example, the nonlinear adaptive filtering technique, based on the Volterra model, is used for the nonlinearities identification problem.

In [4] we have proposed a new implementation of the second order LMS Volterra filter, that can be extended to higher order Volterra filters.

This paper presents an efficient implementation of the second order Volterra filter based on the MMD approach proposed in [10].

The performances of the MMD filter are compared with those of a second order LMS Volterra filter.

2 The MMD approach

The MMD filter represents an efficient approximation of the second order Volterra filter. The filter structure is composed of 3 linear FIR filters and one multiplier, as indicated in Fig. 1.

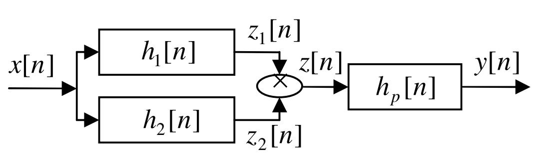

Fig.1 The structure of the MMD filter

The input - output relation of the MMD filter is:

$$y[n] = \sum_{k=0}^{M_p-1} h_p[k] \sum_{i=0}^{M_a-1} h_1[i] x[n-i-k] \sum_{j=0}^{M_a-1} h_2[j] x[n-j-k] \tag{5}$$

where M_a represents the prefilters $h_1[n]$ and $h_2[n]$ memory length and M_p is the memory length of the postfilter $h_p[n]$. The total memory length of the MMD filter is $M_a + M_p - 1$. The MMD filter requires $2M_a + M_p$ filter operations and one additional multiplication per time instance. As it can be seen, the number of the filter operations increases linearly with the memory length. This produces a significant reduction of the computational effort compared to the general Volterra filter implementation.

The second order kernel of the MMD structure can be obtained by comparing relations (4) and (5):

$$h_2[i,j] = \sum_{k=0}^{M_p-1} h_p[k] h_1[i-k] h_2[j-k] \tag{6}$$

The multiplication of the outputs of the prefilters $h_1[n]$ and $h_2[n]$ gives a quadratic filter with separable 2nd order kernel $h_1[n]h_2[n]$. This kernel is weighted by the coefficients of the post filter $h_p[n]$ and summed up at different memory length to give the effective MMD kernel. Consequently, the filter has a multi memory decomposition (MMD) structure.

The MMD kernel is identically zero for displacements larger than $M_a - 1$ and it is not symmetric. As indicated in [1], it is easy to obtain a symmetrized MMD kernel which gives the same filter output according to:

$$h_{2s}[i,j] = \frac{1}{2}\left(h_2[i,j] + h_2[j,i]\right) \tag{7}$$

2.1 Filter coefficients determination

Two different approaches can be used to determine the optimal filter weights for $h_1[n]$, $h_2[n]$ and $h_p[n]$. The first one is based on the approximation of the effective MMD kernel to a given reference kernel. The second approach is an adaptive algorithm and uses the input and output measurements of the unknown nonlinear system.

The well known LMS algorithm is used to update the filter coefficients.

Let it be $e[n]$ the difference between the desired filter output and the current adaptive filter output:

$$e[n] = d[n] - y[n] \tag{8}$$

The implementation proposed in this paper is based on the successive update of the linear filters in the MMD structure.
The update equations are:

$$h_{1/2}[n+1] = h_{1/2}[n] + 2\mu e[n] Z_{1/2}[n] h_p[n] \quad (9)$$

$$h_p[n+1] = h_p[n] + 2\mu e[n] z[n] \quad (10)$$

2.2 The proposed implementation

To describe the proposed implementation let's introduce the matrices notations $H_1[n]$, $H_2[n]$ and $H_p[n]$ for the coefficients of the transversal filters from the MMD structure:

$$H_1[n] = [h_1[n] h_1[n-1] \ldots h_1[n - M_a + 1]] \quad (11)$$

$$H_2[n] = [h_2[n] h_2[n-1] \ldots h_2[n - M_a + 1]] \quad (12)$$

$$H_p[n] = [h_p[n] h_p[n-1] \ldots h_p[n - M_p + 1]] \quad (13)$$

With the adaptive algorithms the matrices are updated at each moment.
To the input signal we attach the instant matrix:

$$X[n] = [x[n] x[n-1] \ldots x[n - Ma + 1]] \quad (14)$$

According to the above notations, the instant outputs of the linear filters $h_1[n]$ and $h_2[n]$ are:

$$z_1[n] = X[n] H_1^T[n] \quad (15)$$

and

$$z_2[n] = X[n] H_2^T[n] \quad (16)$$

The important part of the proposed implementation is represented by the calculus of matrices $Z_{1/2}$ and Z in the update equations (9), respectively (10).
At each iteration step the columns of the matrices $Z_{1/2}$, having a $(M_a \times M_p)$ dimension are calculated according to:

$$Z_{1/2}[:, c+1] = z_{2/1}[n] X^T[n - c + 1] \quad (17)$$

In (17) c represents the number of columns in $Z_{1/2}$ matrices and takes N_p values: $\overline{0, N_p - 1}$. $X[n]$ is the input matrix that "runs" on the input signals values. The results are indicated by the relations (18) and (19).
The matrix Z elements can be calculated according to:

$$z[n] = z_1[n] z_2[n] \quad (20)$$

with $z_1[n]$ and $z_2[n]$ given by the relation (15) respectively (16).
The adaptation error is calculated according to:

$$e[n] = d[n] - Z[n] H_p^T[n] \quad (21)$$

Finally, the filters update equations are:

$$H_1[n+1] = H_1[n] + 2\mu e[n - M_a - M_p + 1] H_p[n] Z_1^T[n] \quad (22)$$

$$H_2[n+1] = H_2[n] + 2\mu e[n - M_a - M_p + 1] H_p[n] Z_2^T[n] \quad (23)$$

$$H_p[n+1] = H_p[n] + 2\mu e[n - M_a - M_p + 1] Z[n] \quad (24)$$

where μ is the step size.

3 Experiments and results

The proposed implementation of the MMD filter (MMDF) was studied in a typical nonlinear system identification application, presented in Fig 2.
The results have been compared with those obtained by using a LMS second order Volterra estimator [4] for the second order nonlinear system with memory.
We have represented the second order kernels determined by using the two approaches and we have compared the results.
The nonlinear system with memory consists of a linear filter, with impulse response $h[n]$ in cascade interconnected with a nonlinear system without memory as shown in Fig.2. The obtained system is a second order nonlinear system with memory that can be modeled using the Volterra series.

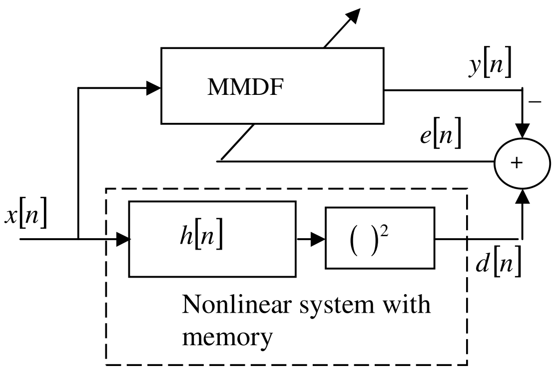

Fig.2 Nonlinear system identification using the MMD filter

The impulse response of the linear filter was implemented by sampling the continuous response $h(t)$ from relation (25) as indicated by relation (26).

$$h(t) = \frac{1256}{0.98}\exp(-251.2t)\sin(1231t)\sigma(t) \qquad (25)$$

$$h[n] = \frac{1}{T_e}h(nT_e) \qquad (26)$$

In the above relation T_e denotes the sampling period.
The input signal $x[n]$ was generated by coloring a white Gaussian sequence $u[n]$ with the filter described by the input – output relation:

$$x[n] = 2u[n] - 2u[n-1] + 1.8u[n-2] \qquad (27)$$

The filter coefficients matrices $H_1[n]$, $H_2[n]$ and $H_p[n]$ were initialized with small numbers randomly distributed between –0.01 and 0.01. The important part of the adaptive implementation is represented by the update of the matrices $Z_{1/2}$ and Z_p as indicated by the equations $(22) \div (24)$. The proposed formulas (17), (18) and (19) significantly reduce the computational effort.
Simulations have been done using the MATLAB software. Figure 3 shows the adaptation error evolution corresponding to the experiment described in Fig.2. The values chosen for the filters length were: $M_a = 30$ and $M_p = 25$.

As it can be seen, significant adaptation arises only after a certain delay. After this period there is a fast adaptation and the error becomes very small.
The number of iterations can be reduced only if one linear filter is updated per iteration, but the adaptation process becomes very slow.

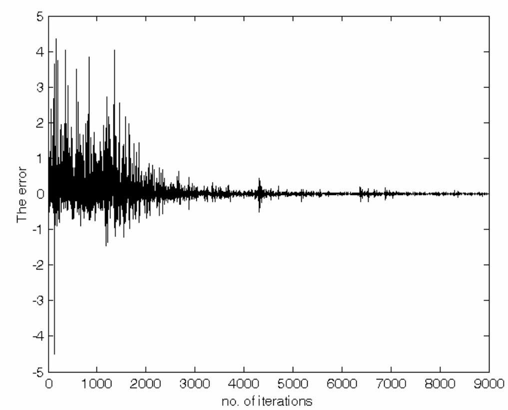

Fig. 3 The error corresponding to the adaptive determination of the MMD filter coefficients

The experiment described by Fig.2 was repeated using a second order LMS Volterra filter instead of the MMD filter. The implementation was made as in [4]. For the $(M \times M)$ dimension of the second order Volterra kernel, h_{2V}, we have chosen the value $M = 30$. The evolution of the adaptation error is indicated in Fig.4.

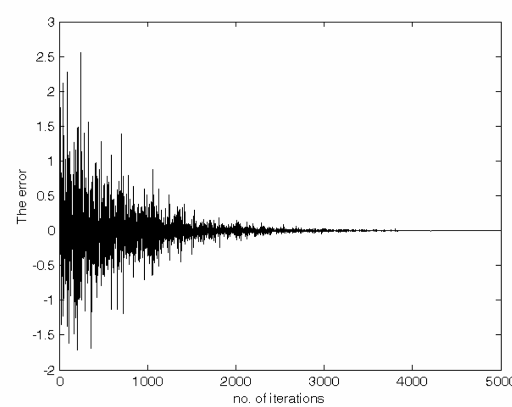

Fig.4 The adaptation error for the second order LMS Volterra estimator

The second order kernel of the MMD structure was calculated, based on determined coefficients values of the filters $h_1[n]$, $h_2[n]$ and $h_p[n]$, using the relations (6) and (7). It is depicted in Fig. 5 and represents a good approximation of the second order Volterra kernel.

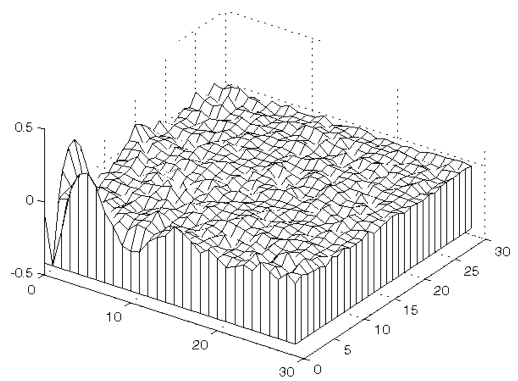

Fig.5 The second order kernel corresponding to the MMDF

The second order Volterra kernel determined using the adaptive LMS approach proposed in [4] is represented in Fig. 6.

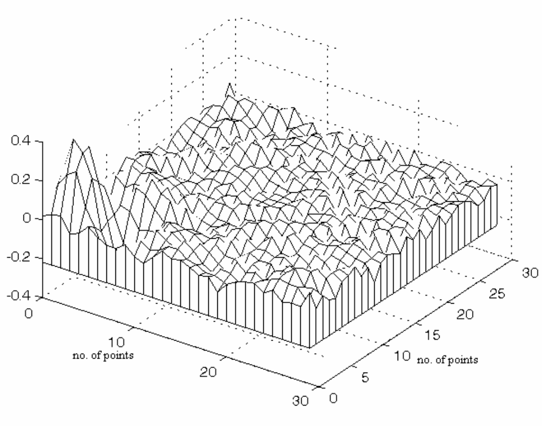

Fig.6 The second order kernel determined using the LMS algorithm

4 Conclusions

In this paper a computational efficient implementation of the second order Volterra filter based on the MMD approximation was presented. The reduced number of the filter operations for the second order kernel calculus and the proposed update equations represent the major advantages of this implementation.

The MMD filter requires only one fourth of the operations of the general Volterra filter and it a similar performance. This property recommends the MMD filter in applications where the accuracy of the estimated second order kernel is very important.

References:

[1] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, John Wiley and Sons, New York, 1980

[2] T. Koh, E.J. Powers, Second – Order Volterra Filtering and Its Application to Nonlinear System Identification, *IEEE Trans. On Acoust., Speech and Signal Processing*, Vol. 33, No. 6, Dec., 1985, pp. 1445 – 1455

[3] V. J. Mathews, Adaptive Polynomial Filters, *Signal Processing Magazine*, Vol. 8, No. 3, 1991, pp. 10-26

[4] G. Budura, C. Botoca, Modeling and Identification of Nonlinear Systems Using the LMS Volterra Filter, *WSEAS Transactions on Signal Processing*, no.2, February, 2006, pp.190-197

[5] G. Budura, C. Botoca, Efficient Implementation of the Third Order RLS Adaptive Volterra Filter, *Facta Universitatis (Nis)*, Ser.: Elec.Energ. vol.19, nr.1, April, 2006, pp. 133-141

[6] G. Budura, C. Botoca, Practical considerations regarding the identification of nonlinear systems, *Revue roumaine des sciences techniques, serie Electrotechnique et energetique*, Tome 51, 1, 2006, pp.79-90

[7] A. Stenger, W. Kellermann, RLS-Adapted Polynomial Filter for Nonlinear Acoustic Echo Cancelling, *Proceedings of the X EUSIPCO* Tampere, Finland, 2000, pp.1867-1870

[8] **W. Frank, Low complexity 3rd Order Nonlinear Filtering, IEEE Workshop on Nonlinear Signal and Image Processing, Neos Marmaras, Greece, 1995, pp. 384-387**

[9] P. Singerl, H. Koeppl, Volterra Kernel Interpolation for System Modeling and Predistortion, *Proceedings of the ISSCS 2005*, Volume 1, Issue, 14-15 July, 2005, pp 251 – 254

[10] W.A. Frank, MMD - An Efficient Approximation to the 2nd Order Volterra Filter, *Proc. ICASSP'94*, 1994, pp.517-520

$$Z_1[n]=\begin{bmatrix} z_2[n]x[n] & z_2[n-1]x[n-1] & \cdots & z_2[n-M_p+1]x[n-M_p+1] \\ z_2[n]x[n-1] & z_2[n-1]x[n-2] & \cdots & z_2[n-M_p+1]x[n-M_p] \\ \vdots & & & \\ z_2[n]x[n-M_a+1] & z_2[n-1]x[n-M_a] & \cdots & z_2[n-M_p+1]x[n-M_p-M_a+2] \end{bmatrix} \quad (18)$$

$$Z_2[n]=\begin{bmatrix} z_1[n]x[n] & z_1[n-1]x[n-1] & \cdots & z_1[n-M_p+1]x[n-M_p+1] \\ z_1[n]x[n-1] & z_1[n-1]x[n-2] & \cdots & z_1[n-M_p+1]x[n-M_p] \\ \vdots & & & \\ z_1[n]x[n-M_a+1] & z_1[n-1]x[n-M_a] & \cdots & z_1[n-M_p+1]x[n-M_p-M_a+2] \end{bmatrix} \quad (19)$$