Efficient Implementation and Performance Evaluation 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
georgeta.budura@etc.upt.ro http://www.etc.upt.ro

Abstract: - Considered as a prototype of nonlinear system models the second order Volterra filter (FV_2) is characterized by an increased complexity in comparison with a linear filter. This complexity is given by the large number of filter coefficients, as well as by the filter operations and it requires an increased computational power in technical applications. The filter based on the multi memory decomposition (MMD) structure represents a good approximation of the FV_2 and significantly reduces the number of filter operations. In this paper we propose an efficient implementation of the MMD filter studied in a typical nonlinear identification problem. The simulations show the very good performance of our proposed MMD structure in comparison with the results obtained by using a second order LMS Volterra filter. We have evaluated the performance of our method based on the system response to different input signals.

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

1 Introduction

Detection, representation and identification of different system nonlinearities represent important tasks in many applications and are a major contribution to the development of the main nonlinear modeling techniques.

The Volterra series has been successfully and widely applied as a nonlinear system modeling technique and is able to represent a wide class of nonlinear systems [1].

This generalized form of the Taylor series expansion can be used to represent a nonlinear system with memory [2].

The series is a sum of generalized convolutions which can be considered as an extension of the linear case. The resulting nonlinear filter is fully specified by a set of functions, called Volterra kernels, which can be estimated from the input-output measurements performed on the system under study. The system kernels characterize the system input-output relationship and allow the prediction of the physical system response to an arbitrary input signal.

For the discrete time Volterra filter of a given order the kernels represent the so-called filter coefficients.

At present there exists no general method to determine the Volterra kernels for a nonlinear system model. This kernels can be calculated only for systems with a known and finite order. The values of the Volterra kernels depend on the order of the Volterra series representation used [1]. If the order of the Volterra model changes, the Volterra kernels values must be recalculated.

Given an appropriate Volterra model of order N and memory M for a particular system representation, the accurate kernel estimation becomes a major problem.

A significant advantage of the Volterra models, if compared with other nonlinear models, is that their input-output relation is linear with respect to the filter coefficients. The nonlinearity is reflected only by the multiple products between the delayed versions of the input signal.

Due to this fact, many methods from the linear filter theory can be applied to the Volterra filter.

For example, adaptive methods and algorithms are widely used in applications dealing with kernels estimation. Various Least Mean Square (LMS) and recursive least square (RLS) algorithms have been applied to the problem of Volterra kernel estimation [3], [4], [5], [6], [7], [8]. Some authors have used the frequency domain vector representations of a Volterra system and have computed the unknown frequency domain Volterra kernels using a MMSE criterion [9]. Recently more orthogonal search algorithms have been developed both in the time domain [10], [11] as well as in the frequency domain.

The major drawback of the Volterra filter is the large number of filter coefficients. Hence, the Volterra model implementation needs an increased computational power that represents a major drawback in real time applications [1]. Therefore, only low order nonlinearities can be modeled in an efficient way.

It is well known that the second order Volterra filter [3], [12] is extremely popular and widely used in applications concerned with unwanted nonlinearities identification.

In technical applications [4]-[9], [13], [14] accurate kernel estimation is very important because it directly influences the accuracy of the system model. The speed of the kernel estimation process is also important, since a fast kernel estimation method will result into a better system representation.

This paper presents an efficient implementation of the second order Volterra filter based on the MMD approach proposed by [15].

The proposed implementation of the FV_2, based on the MMD structure is studied in a typical nonlinear system identification problem. The results show the very good performance of our proposed implementation.

Although the MMD structure requires only one fourth of the filter operations of the general Volterra filter it performs much better than the second order LMS Volterra filter. This is mainly due to the larger overall filter memory that can be achieved with the MMD model.

The second order kernel of the MMD structure has been compared with that obtained using the LMS adaptive algorithm.

The implementation of the LMS algorithm in the case of the Volterra filter was constructed as an extension of the LMS algorithm for linear filters due to the linearity of the input-output relation of the Volterra model.

In order to evaluate the model performances, the responses of the MMD filter to different input signals were calculated and compared with those of the second order adaptive LMS Volterra filter.

2 The second order Volterra filter

For a discrete-time causal nonlinear system with memory, with input $x[n]$ and output $y_V[n]$, as indicated in Fig.1, the Volerra series expansion is given by relation (1):

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

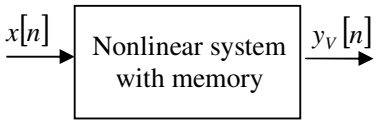

Fig.1 The Volterra model

where N represents the model nonlinearity degree.

In the above representation, the functions $h_{iV}[k_1,\cdots,k_i]$, $i=\overline{0,N}$ represent the Volterra kernels associated to the nonlinear operators from relation (2).

The input-output relation can also be written in terms of nonlinear operators, $\mathcal{H}_i[\ ]$, as shown in equation (2).

$$y_V[n]=\mathcal{H}_0[x[n]]+\mathcal{H}_1[x[n]]+\cdots+\mathcal{H}_N[x[n]] \quad (2)$$

The next step is to truncate the kernels memory to the length M, which generates the finite memory representation of (3):

$$y_V[n]=\sum_{i=1}^{N}\sum_{k_1=0}^{M-1}\cdots\sum_{k_i=0}^{M-1}h_{iV}[k_1,\cdots,k_i]x[n-k_1]\cdots x[n-k_i] \quad (3)$$

Please note that the above representations have the same memory for all nonlinearity orders. In the most general case, the relation (3) uses a different memory for each nonlinearity order. Relation (3) can be further simplified by considering symmetric Volterra kernels.

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

By choosing $N=2$, in relation (4) we express the input-output relationship of the FV_2 as follows:

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

In terms of nonlinear operators we have:

$$y_V[n]=\mathcal{H}_0[x[n]]+\mathcal{H}_1[x[n]]+\mathcal{H}_2[x[n]] \quad (5)$$

In relation (5) $h_{0V}=\mathcal{H}_0$ denotes the DC term, $h_{1V}[k_1]$ represents the first order Volterra kernel and

$h_{2V}[k_1,k_2]$ stands for the second order Volterra kernel of the nonlinear system.
The nonlinear model described by the relations (4) and (5) is referred to as the second order Volterra model.
Next we consider a second order Volterra filter having the DC term and the first order kernel equal to zero.
If the considered filter has a symmetric second order Volterra kernel, then it can be expressed as a $M(M+1)$ quadratic matrix.
In order to avoid redundancy in terms, the second order Volterra model with symmetric kernel can be written with the following summation limits:

$$y_{2V}[n]=\sum_{k_1=0}^{M-1}\sum_{k_2=k_1}^{M-1}h_{2V}[k_1,k_2]x[n-k_1]x[n-k_2] \qquad (6)$$

If we consider symmetric kernels, as indicated above, the second order Volterra kernel $h_{2V}[k_1,k_2]$ requires the computation of $\frac{M(M+1)}{2}$ coefficients [3], [12].
As it can be seen the number of filter coefficients increases with the square of the filter memory size *M*. For higher order Volterra kernels this dependency becomes more complicated.
For example, the third order Volterra kernel needs the computation of $\frac{M(M+1)(M+2)}{6}$ coefficients.
Many authors have aimed at finding Volterra filter implementations that reduce the complexity and the number of filter operations.
One solution is to reduce the number of filter coefficients [1], [16], [17].
Another possible solution is to find an efficient way of representing the "input signal" of the Volterra filter, which consists of multiple products and models the nonlinearity [4].
In this paper we propose an efficient implementation of the MMD filter that represents a good approximation of the second order Volterra kernel. This implementation reduces the number of filter coefficients and the number of filter operations. The simulations and the comparisons to the LMS approach show the good performance of our MMD approximation.

3 The MMD approach

The MMD approach represents an efficient approximation of the second order Volterra filter described by equation (6). The resulted filter, called the MMD filter, consists of 3 linear FIR filters and one multiplier, as indicated in Fig.2.

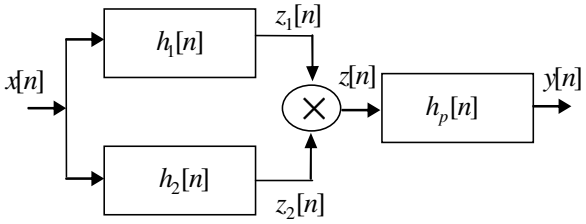

Fig.2 The structure of the MMD filter

The input - output relation of the MMD filter is:

$$y[n]=\sum_{k=0}^{M_p-1}h_p[k]\sum_{i=0}^{M_a-1}h_1[i]x[n-i-k]\sum_{j=0}^{M_a-1}h_2[j]x[n-j-k] \qquad (7)$$

Where M_a represents the memory length of the prefilters $h_1[n]$ and $h_2[n]$. M_p denotes 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.
Please note that, 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 second order Volterra filter implementation.
The second order kernel of the MMD structure can be obtained by relations (6) and (7).
The result is given in relation (8).

$$h_2[i,j]=\sum_{k=0}^{M_p-1}h_p[k]h_1[i-k]h_2[j-k] \qquad (8)$$

The multiplication of the prefilters $h_1[n]$ and $h_2[n]$ outputs has as result a quadratic filter with separable second 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 for different memory length M_p to generate the effective MMD kernel. Consequently,

the filter has a multi memory decomposition (MMD) structure.
The MMD kernel is non-symmetric and identically zero for displacements larger than $M_a - 1$.
As indicated in [1], it is easy to obtain a symmetric MMD kernel which generates the same filter output by applying relation (9).

$$h_{2s}[i,j] = \frac{1}{2}\left(h_2[i,j] + h_2[j,i]\right) \tag{9}$$

3.1 Computing the filter coefficients

Two different approaches can be used to determine the optimal filter coefficients for $h_1[n]$, $h_2[n]$ and $h_p[n]$.
The first one is based on the approximation of the effective MMD kernel with a given reference kernel.
The second approach employs an adaptive algorithm which uses the input and output measurements of the unknown nonlinear system.
The well known LMS algorithm is used to update the filter coefficients of the MMD structure.
Let $e[n]$ be the difference between the desired filter output and the current adaptive filter output:

$$e[n] = d[n] - y[n] \tag{10}$$

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

$$h_{1/2}[n+1] = h_{1/2}[n] + 2\mu e[n] Z_{1/2}[n] h_p[n] \tag{11}$$

$$h_p[n+1] = h_p[n] + 2\mu e[n] z[n] \tag{12}$$

where:

$$z[n] = \left[z(n)\, z(n-1) \cdots z(n - M_p + 1)\right]^T \tag{13}$$

The expression of $Z_{1/2}[n]$ is indicated in equation (14):

$$Z_{1/2}[n] = \left[z_{1/2}(n)\, x(n)\, z_{1/2}(n-1)\, x(n-1) \cdots \right. \\ \left. \cdots z_{1/2}(n - M_p + 1)\, x(n - M_p + 1)\right] \tag{14}$$

where $x(n)$ is a $M_a \times 1$ vector of input signal values similar to the one shown in relation (13).

3.2 Implementing the MMD structure

To describe the proposed implementation we introduce the matrices $H_1[n]$, $H_2[n]$ and $H_p[n]$ which contain the coefficients of the transversal filters from the MMD structure:

$$H_1[n] = \left[h_1[n]\, h_1[n-1] \ldots h_1[n - M_a + 1]\right] \tag{15}$$

$$H_2[n] = \left[h_2[n]\, h_2[n-1] \ldots h_2[n - M_a + 1]\right] \tag{16}$$

$$H_p[n] = \left[h_p[n]\, h_p[n-1] \ldots h_p[n - M_p + 1]\right] \tag{17}$$

The matrices are updated according to the adaptive algorithm at each time stamp.
The following instant matrix is attached to the input signal:

$$X[n] = \left[x[n]\, x[n-1] \ldots\ x[n - Ma + 1]\right] \tag{18}$$

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

$$z_1[n] = X[n] H_1^T[n] \tag{19}$$

and

$$z_2[n] = X[n] H_2^T[n] \tag{20}$$

The most important part of the proposed implementation is represented by the calculus of matrices $Z_{1/2}$ and Z in the update equation (11), respectively (12).
At each iteration step the columns of the matrices $Z_{1/2}$, having a $(M_a \times M_p)$ dimension are calculated as follows :

$$Z_{1/2}[:, c+1] = z_{2/1}[n] X^T[n - c + 1] \tag{21}$$

In relation (21) the parameter *c* denotes the number of columns in the $Z_{1/2}$ matrices and takes M_p values: $\overline{0, M_p - 1}$. $X[n]$ is the input matrix that "runs" on the input signals values.
The results are reported in the relations (22) and (23), presented at the end of the paper.
The elements of matrix Z are computed as follows:

$$z[n]=z_1[n]z_2[n] \tag{24}$$

with $z_1[n]$ and $z_2[n]$ given by the relation (19), respectively by (20).
The adaptation error is calculated according to:

$$e[n]=d[n]-Z[n]H_p^T[n] \tag{25}$$

Finally, the MMD structure linear filters update equations are:

$$H_1[n+1]=H_1[n]+ 2\mu e[n-M_a-M_p+1]H_p[n]Z_1^T[n] \tag{26}$$

$$H_2[n+1]=H_2[n]+ 2\mu e[n-M_a-M_p+1]H_p[n]Z_2^T[n] \tag{27}$$

$$H_p[n+1]=H_p[n]+ 2\mu e[n-M_a-M_p+1]Z[n] \tag{28}$$

where μ is the step size used in the adaptive algorithm.

4 Experiments and performance evaluation

We studied our implementation of the MMD filter (MMDF) in the context of a typical nonlinear system identification application represented in Fig.3. We have determined the second order MMD kernel that models the nonlinear system with memory.
The results have been compared with those obtained by using a LMS second order Volterra estimator as presented in [4] for the second order nonlinear system with memory.
The implementation of the second order LMS Volterra filter is also presented.
The second order kernels determined by using the two approaches have been represented and the results compared.
The nonlinear system with memory used in our application consists of a linear filter, with impulse response $h[n]$, connected in cascade with a nonlinear system without memory as shown in Fig.3.
The obtained system is a second order nonlinear system with memory that can be modeled using the Volterra series.

The impulse response of the linear filter was implemented by sampling the continuous response $h(t)$ from relation (29) as shown in relation (30).

$$h(t)=\frac{1256}{0.98}\exp(-251.2t)\sin(1231t)\sigma(t) \tag{29}$$

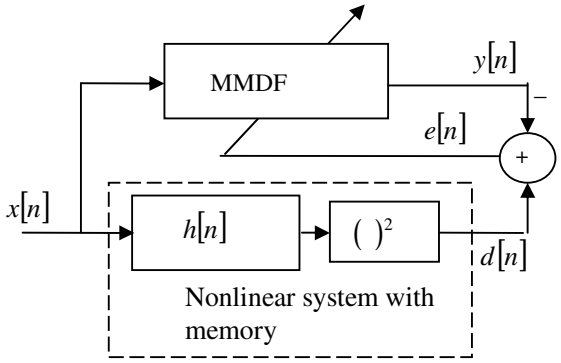

Fig.3 Nonlinear system identification using the MMD filter

$$h[n]=\frac{1}{T_e}h(nT_e) \tag{30}$$

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:

$$x[n]=2u[n]-2u[n-1]+1.8u[n-2] \tag{31}$$

The matrices $H_1[n]$, $H_2[n]$ and $H_p[n]$ containing the filter coefficients were initialized with small values randomly distributed in the interval (–0.01 0.01).
The important contribution of the adaptive implementation is represented by the update of the matrices $Z_{1/2}$, and Z used by equations $26 \div 28$.
The proposed formulas (21), (22) and (23) significantly reduce the computational effort.
Next we present the results of our simulations which implemented in Matlab.

Fig.4 shows the evolution of the adaptation error corresponding to the experiment described in Fig.3. The values chosen for the length of the filter were: $M_a = 30$ and $M_p = 25$.

The figure shows that significant adaptation is achieved after a certain delay. After this initial period we observe a fast adaptation as 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 in that case.

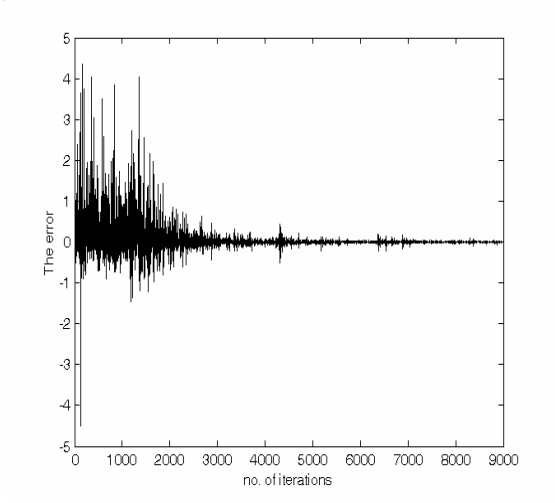

Fig.4 The error corresponding to the adaptive determination of the MMD filter coefficients

In order to evaluate the performance of our method we repeated the experiment described in Fig.3 and replaced the MMD filter by a second order LMS Volterra filter as shown in Fig.5.

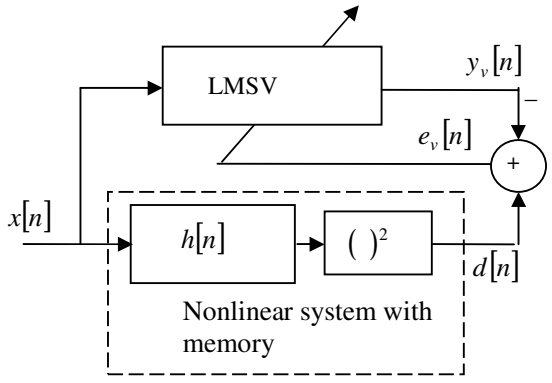

Fig.5 Nonlinear system identification using the LMS Volterra filter

The Volterra filter used in the nonlinear system identification described in Fig.5 is a second order filter having the dimension $(M \times M)$.

The well known LMS update equation for the linear filter is:

$$H_{k+1}^{(1)} = H_k^{(1)} + \mu e_k X_k^{(1)} \tag{32}$$

where μ is a small positive constant (referred to as the step size) that determines the speed of convergence and also affects the final error of the filter output. Relation (32) can be extended to the second order Volterra filter if we make some changes and introduce vector notations as follows.

First, the vector of the linear filter $H_k^{(1)}$ having the length M:

$$H_k^{(1)T} = \left[h_k^{(1)}[0] \quad h_k^{(1)}[1] \quad \ldots \quad h_k^{(1)}[M-1]\right] \tag{33}$$

becomes the vector of the Volterra kernel coefficients.

Second, the input vector $X_k^{(1)}$:

$$X_k^{(1)T} = \left[x_k \quad x_{k-1} \quad \cdots \quad x_{k-M+1}\right] \tag{34}$$

which for the linear case contains only the input signal values becomes more complicated.

In the equations (33) and (34) the filter order is marked by a superscript index. This notation will be kept consistent throughout the paper.

The Volterra representation with symmetric kernels, consists of two parts: (1) the Volterra kernels and (2) the products of the delayed input signal values.

If we express the Volterra kernels and the input signal products in a vector form, then we can formulate the adaptive Volterra filter output using the vector notation form which is very similar to the one used in the linear case.

The vector corresponding to the second order Volterra kernel is given by:

$$H_k^{(2)T} = \left[h_k^{(2)}[0,0]\, h_k^{(2)}[0,1] \cdots h_k^{(2)}[0,M-1] \cdots \right.$$
$$\left. \cdots\, h_k^{(2)}[M-1,M-1]\right] \tag{35}$$

As an input signal vector we consider the following vector:

$$X_k^{(2)T} = \left[x_k^2 \;\; x_k x_{k-1} \cdots x_k x_{k-M} \cdots x_{k-M}^2\right] \tag{36}$$

which contains all the products between the input signal values.
Hence, the adaptive filter output is given by:

$$y_{Vk} = H_k^{(2)T} X_k^{(2)} \tag{37}$$

At the *k*-th iteration, the desired output is d_k and the Volterra output filter is y_{Vk} .
According to the LMS algorithm the following equation has to be minimized:

$$E\left[e_{Vk}^2\right] = E\left[\left[d_k - H_k^{(2)T} X_k^{(2)}\right]^2\right] \tag{38}$$

The vector $H^{(2)*}$ that minimizes equation (37) can be expressed as a solution of the normal equation:

$$H^{(2)*} = R_2^{-1} g \tag{39}$$

Where: $R = E\left[X_k^{(2)} X_k^{(2T)}\right]$ is the input correlation matrix containing the 4-th order moments of the input signal and $g = E\left[X_k^{(2)} d_k\right]$ is the cross-correlation vector between the input and the desired output.
The update equation for the LMS Volterra filter can be written as:

$$H_{k+1}^{(2)T} = H_k^{(2)T} + e_{Vk}\mu_2 X_k^{(2)T} \tag{40}$$

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 shown in Fig.6.

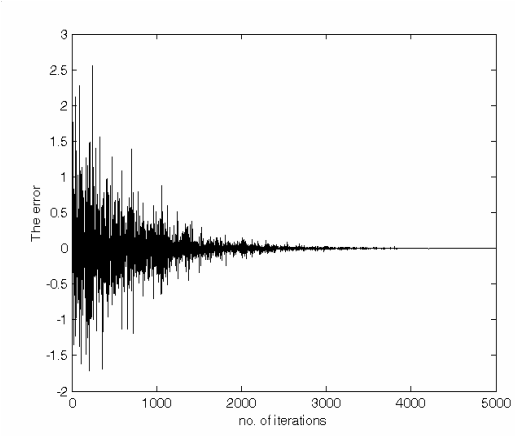

Fig.6 The adaptation error for the second order LMS Volterra filter

We computed the second order kernel of the MMD structure, based on the determined coefficient values of the filters $h_1[n]$, $h_2[n]$ and $h_p[n]$, using the relations (8) and (9).
The result is shown in Fig.7.

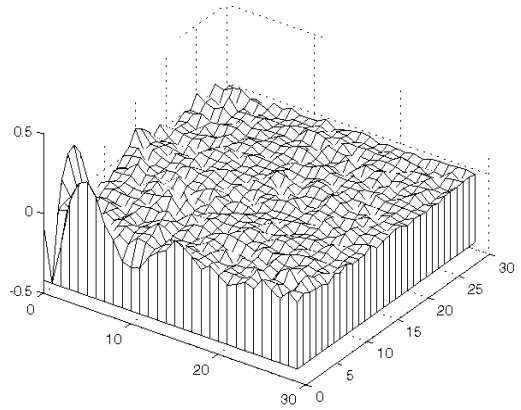

Fig.7 The second order kernel corresponding to the MMD structure

The second order Volterra kernel determined using the adaptive LMS approach presented above is represented in Fig.8.

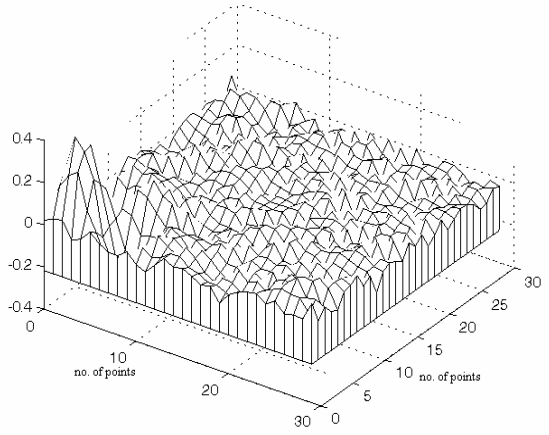

Fig.8 The second order kernel computed using the LMS algorithm

It can be seen that the MMD kernel represents a good approximation of the second order Volterra kernel.
Next we have calculated the frequency response of the MMD structure second order kernel as presented in [12]:

$$H_2(\Omega_1,\Omega_2)=\sum_{n_1=0}^{N-1}\sum_{n_2=0}^{N-1}h_2[n_1,n_2]e^{-j\Omega_1 n_1}e^{-j\Omega_2 n_2} \qquad (41)$$

where,

$$\Omega_1,\Omega_2=\left\{-\pi,\left(-\pi+\frac{2\pi}{M}\right),\cdots\left(-\pi+(M-1)\frac{2\pi}{M}\right)\right\}$$

represent the frequency domain used for the representation and $M=30$.
We have also computed the frequency response of the second order Volterra kernel as follows:

$$H_{2V}(\Omega_1,\Omega_2)=\sum_{n_1=0}^{M-1}\sum_{n_2=0}^{M-1}h_{2V}[n_1,n_2]e^{-j\Omega_1 n_1}e^{-j\Omega_2 n_2} \qquad (42)$$

using the same frequency domain.
Figures (9) and (10) represent the frequency response of the MMD structure second order kernel respectively the frequency response of the second order Volterra kernel.

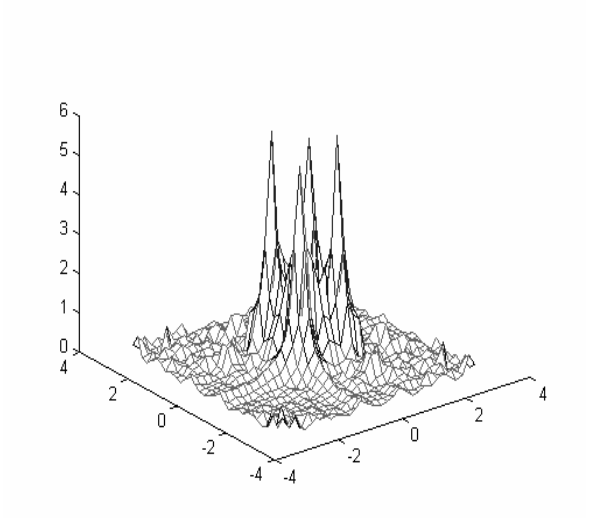

Fig.9 The frequency response of the MMD structure second order kernel

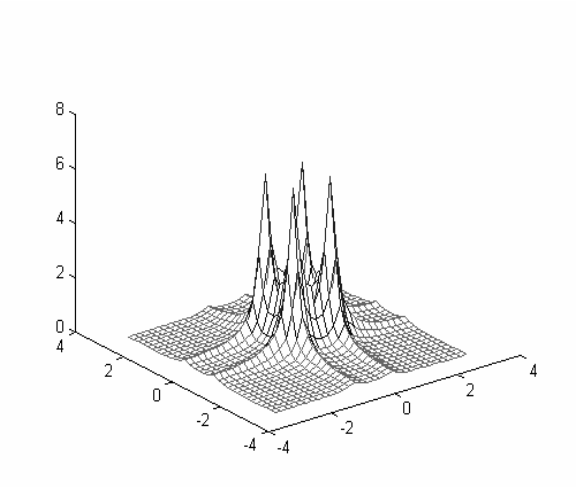

Fig.10 The frequency response of the second order Volterra kernel

Again, we observe that the two representations are similar.
In order to evaluate the model performances based on the estimated kernels we have computed the nonlinear system responses to some deterministic input signals, as well as to a white Gaussian input signal. The deterministic input signals are sinusoidal signals of different frequencies.
The output signal of the MMD structure (MMD) was calculated according to relation (7) based on previously computed coefficient values of the filters $h_1[n]$, $h_2[n]$ and $h_p[n]$.
The output signal of the Volterra filter (LMSV) is given by the relation (6), where $h_{2V}[k_1,k_2]$ represents again the values obtained by applying the adaptive LMS algorithm.
We repeated this experiment for different frequencies of the input signal.
In all cases, Fig.11 - Fig.14 the two graphs are highly similar, and only small differences can be detected.
The accuracy of the proposed MMD method for the kernel calculus was evaluated with respect to its error according to relation (43) [10]:

$$e_c=\frac{\sum_{n=1}^{N}(y_{2V}[n]-y[n])^2}{\sum_{n=1}^{N}y_{2V}^2[n]} \qquad (43)$$

where *N* denotes the length of a period in the output signal.

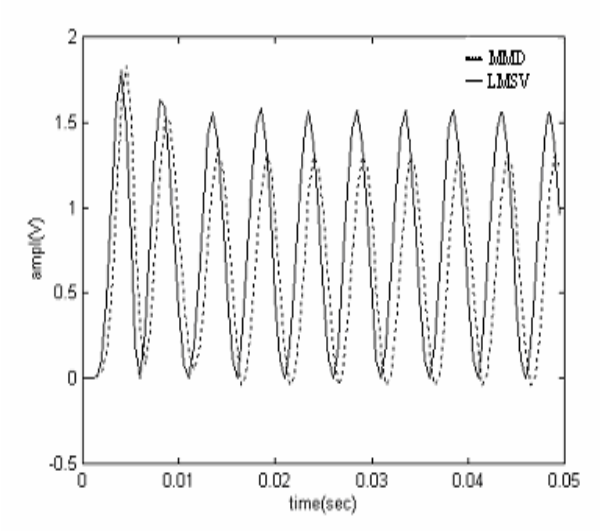

Fig.11 The second order systems responses to a sinusoidal signal (f=100Hz)

The error according to the relation (43) in the case of a sinusoidal input signal with the frequency of $f = 100Hz$ is: $e_c = 0{,}0512$.

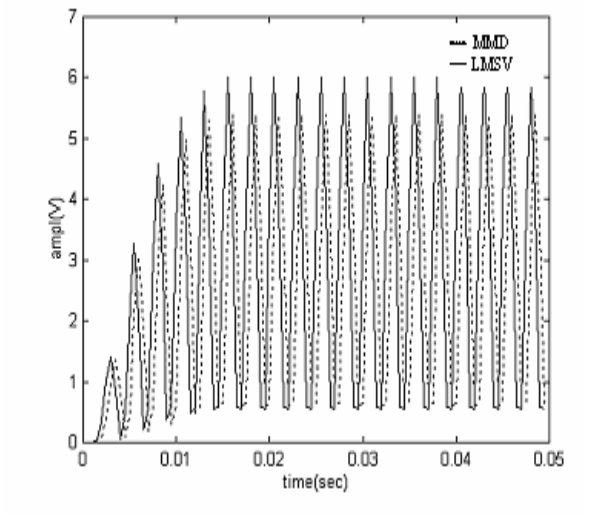

Fig.12 The second order systems responses to a sinusoidal signal (f=200Hz)

The error according to the relation (39) in case of a sinusoidal input signal with the frequency of $f = 200Hz$ is: $e_c = 0{,}009$.

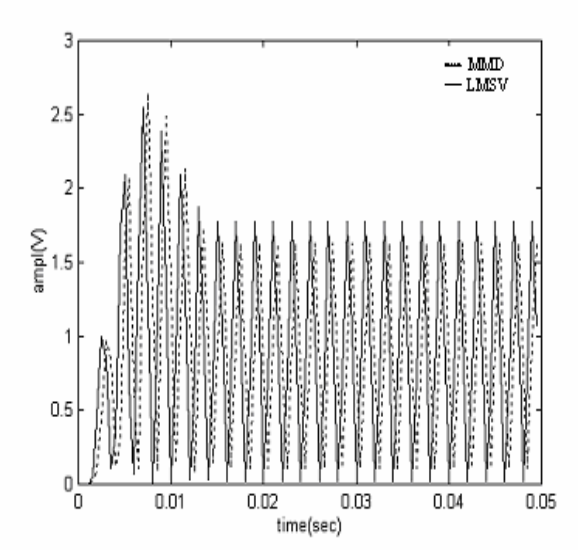

Fig.13 The second order systems responses to a sinusoidal signal (f=250Hz)

The error according to the relation (43) in the case of a sinusoidal input signal with the frequency of $f = 250Hz$ is: $e_c = 0{,}0092$.

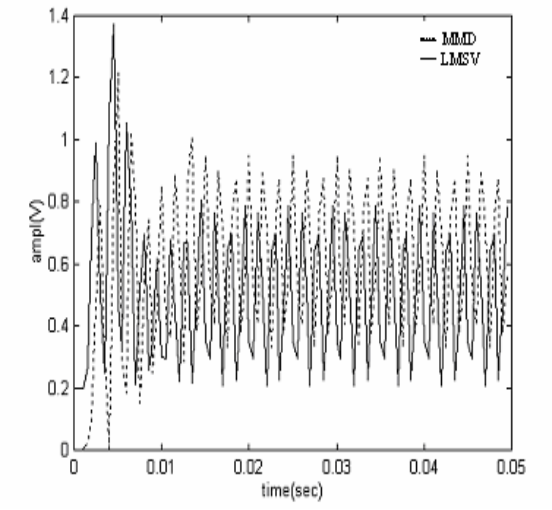

Fig.14 The second order systems responses to a sinusoidal signal (f=300Hz)

The error according to the relation (43) in the case of a sinusoidal input signal with the frequency of $f = 300Hz$ is: $e_c = 0{,}071$.

A more useful and hard test of the relative accuracy of the kernel implementation technique is to calculate the responses of the second order kernels to a white Gaussian noise.

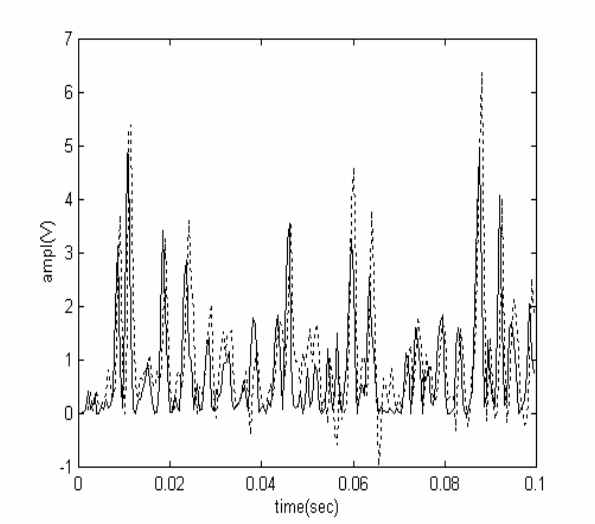

Fig.15 The second order systems responses to a white Gaussian noise input signal

The error according to the relation (43) in the case of a white Gaussian noise input signal is: $e_c = 0{,}09$.

In all the above presented experiments our method outperforms the LMS approach.

5 Conclusions

In this paper we present an efficient implementation of the second order discrete time Volterra filter widely used in technical applications. The proposed implementation is based on the MMD approximation of the Volterra filter.

The major advantages of our approach are: (1) the reduced number of filter operations and (2) the proposed update equations during the adaptive algorithm for the MMD kernel calculus. The MMD filter requires only one fourth of the operations needed for the general Volterra filter and it displays a similar performance.

The proposed implementation was evaluated in a typical nonlinear system identification application.

In order to measure the performance of the model we used two nonlinear structures: the MMD filter and the LMSV filter.

Based on the estimated kernel of the MMD structure we have computed and represented the equivalent second order kernel of the MMD filter and compared it to the LMSV second order kernel.

The frequency responses of the two second order kernels were also computed and represented.

We conclude that the MMD structure represents a very good approximation of the second order Volterra filter.

Further we have calculated and compared the nonlinear systems responses to some deterministic input signals and to the white Gaussian noise. The input signals were sinusoidal signals of different frequencies.

In all cases the two compared output signals are highly similar and only insignificant differences can be detected. To evaluate these differences we have computed the error e_c and concluded that the proposed implementation of the MMD structure represents a good approximation of the second order Volterra filter.

Our approach is very useful in applications that need a large number of filter coefficients and where the accuracy of the estimated second order kernel is very important.

As a future direction of this work, we plan to implement the MMD filter on a floating point digital signal processor. This implementation will be useful for real time applications as those of identifying unwanted nonlinearities.

References:

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

[2] S. A. Kanarachos, K. Spentzas, A method for the synthesis of nonlinear controllers using a Taylor series expansion of the control law, *WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS*, Issue 4, Vol.2 Oct. 2003, pp. 499-504.

[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] G. Sunil, Performance Evaluation of an Adaptive Volterra/Hybrid Equalizer in a Nonlinear Magneto-Optic Data Storage Channel, *WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS*, Issue 5, Vol.3 July. 2004, pp. 368-373. pp. 832-835.
[9] Y. S. Cho, E. J. Powers, Estimation of Nonlinear Distortions Using Digital Higherorder Spectra and Volterra Series, *Proc. IEEE Int. Symp. Circuits and Systems, (ISCAS'92)*, San Diego, USA, May 1992, pp. 2781-2784.
[10] M. J. Korenberg, L. D. Paarmann, Orthogonal Approches to Time-series Analysis and System Identification, *IEEE Signal Processing Magazine*, Vol. 8, No.3, pp. 29-43, July 1991.
[11] M. J. Korenberg, Parallel Cascade Identification and Kernel Estimation for Nonlinear Systems, *Annals of Biomedical Eng.*, 1991, Vol. 19, pp. 429-455.
[12] T. Koh, E.J. Powers, Second – Order Volterra Filtering and It's Application to Nonlinear System Identification, *IEEE Trans. On Acoust., Speech and Signal Processing*, Vol. 33, No. 6, Dec., 1985, pp. 1445 – 1455.
[13] W. Frank, Low complexity 3rd Order Nonlinear Filtering, *IEEE Workshop on Nonlinear Signal and Image Processing*, Neos Marmaras, Greece, 1995, pp. 384-387.
[14] 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.
[15] W.A. Frank, MMD - An Efficient Approximation to the 2nd Order Volterra Filter, *Proc. ICASSP'94*, 1994, pp. 517-520.
[16] Scott, B. Mulgrew, "Nonlinear System Identification and Prediction Using Orthonormal Functions", *IEEE Transactions on Signal Processing*, vol. 45, July, 1997, pp. 1842-1853.
[17] R.H. Kwong, E.W. Johnston, "*A Variable Step Size LMS Algorithm*", *IEEE Transactions on Signal Processing*, Vol. 40, No.7, pp. 1633-1642.

Update equations (22), (23):

$$Z_1[n]=\begin{bmatrix} z_2[n]x[n] & z_2[n-1]x[n-1] & \cdots & z_2[n-M_p+1]x[n-M_p+1] \\ z_2[n]x[n-1] & z_2[n-1]x[n-2] & \cdots & z_2[n-M_p+1]x[n-M_p] \\ \vdots & & & \\ z_2[n]x[n-M_a+1] & z_2[n-1]x[n-M_a]\cdots & & z_2[n-M_p+1]x[n-M_p-M_a+2] \end{bmatrix} \quad (22)$$

$$Z_2[n]=\begin{bmatrix} z_1[n]x[n] & z_1[n-1]x[n-1] & \cdots & z_1[n-M_p+1]x[n-M_p+1] \\ z_1[n]x[n-1] & z_1[n-1]x[n-2] & \cdots & z_1[n-M_p+1]x[n-M_p] \\ \vdots & & & \\ z_1[n]x[n-M_a+1] & z_1[n-1]x[n-M_a]\cdots & & z_1[n-M_p+1]x[n-M_p-M_a+2] \end{bmatrix} \quad (23)$$