FACTA UNIVERSITATIS (NIŠ)
SER.: ELEC. ENERG. vol. 19, no. 1, April 2006, 133-141

Efficient Implementation of the Third Order RLS Adaptive Volterra Filter

Georgeta Budura and Corina Botoca

Abstract: Nonlinear adaptive filtering techniques are widely used for the nonlinearities identification in many applications. This paper proposes a new implementation of the third order RLS Volterra filter based on the decomposition of the input vector. Its performances are evaluated in a typical nonlinear system identification application. Different degrees of nonlinearity for the nonlinear system are considered. Comparations, based on the adaptive filter error, are made in all cases with a linear identifier. The experimental results show that the proposed nonlinear identifier has better performances than the linear one.

Keywords: Adaptive filter, Volterra filter, efficient implementation, nonlinear identifier performances.

1 Introduction

The dynamic development of nonlinear filtering is indicated by the amount of published research and the wide spread use of nonlinear digital filters. Specific applications that need nonlinear structures are encountered in many different areas, notably in: telecommunications, image processing, in geophysical and biomedical signal processing [1]. Detection, representation and identification of nonlinearities in different systems represent important tasks in many applications and had a major contribution to the development of the main nonlinear modelling techniques. The current trend in the telecommunication systems design is the identification and compensation of unwanted nonlinearities [2]. It was demonstrated that unwanted nonlinearities in the system will have a determinant effect on his performance.

Manuscript received June 2005

G. Budura and C. Botoca are with ”Politehnica” University of Timisoara, Faculty of Electronics and Telecommunications, Communications Department, V. Parvan 2, Timisoara, Romania (e-mails: `[georgeta.budura, corina.botoca]@etc.utt.ro`).

There are various ways of reducing the effects of undesired nonlinearities. The use of nonlinear models considered in this paper to characterize and compensate harmful nonlinearities offer a possible solution.

The Volterra series have been widely applied as nonlinear system modelling technique with considerable success. However, at present, none general method exists to calculate the Volterra kernels for nonlinear systems, although they can be calculated for systems whose order is known and finite [3]. When the nonlinear system order is unknown, adaptive methods and algorithms are widely used for the Volterra kernel estimation. The accuracy of the Volterra kernels will determine the accuracy of the system model and the accuracy of the inverse system used for compensation. The speed of kernel estimation process is also important. A fast kernel estimation method may allow the user to construct a higher order model that gives an even better system representation.

2 The Volterra Model

This section will discuss some important aspects of the third order Volterra model. For a discrete-time and causal nonlinear system with memory, with input x[n] and output y[n], the Volterra series expansion is given by:

$$\begin{aligned} y[n] = h_0 + & \sum_{k_1=0}^{M-1} h_1[k_1]\, x[n-k_1] \\ & + \sum_{k_1=0}^{M-1}\sum_{k_2=0}^{M-1} h_2[k_1,k_2]\, x[n-k_1]\, x[n-k_2] \\ & + \sum_{k_1=0}^{M-1}\sum_{k_2=0}^{M-1}\sum_{k_3=0}^{M-1} h_3[k_1,k_2,k_3]\, x[n-k_1]\, x[n-k_2]\, x[n-k_3] \end{aligned} \tag{1}$$

where, for simplicity we have considered a nonlinear model up to the third order.

In the above representation, the functions $h_i, i = \overline{1,3}$, represent the kernels associated to the nonlinear operators $H_i[x[n]]$. The input-output relation can also be written in terms of nonlinear operators as indicated in relation (2).

$$y[n] = H_o + H_1[x[n]] + H_2[x[n]] + H_3[x[n]] \tag{2}$$

The nonlinear model described by the relations (1) and (2) is called a third order Volterra model. Note that the above representation has the same memory for all 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 relation (1) by considering symmetric Volterra kernels. The kernel $h_i[k_1...k_i]$ is symmetric if the indices can be interchanged without affecting its value. If we consider symmetric kernels of memory M, the second order Volterra kernel requires the determination of $M(M+1)/2$ coefficients, while the third order kernel needs $M(M+1)(M+2)/6$ coefficients. The second order Volterra kernel is a symmetric $(M \times M)$ matrix. As presented in reference [4], the third order kernel is composed of M symmetric matrices having the dimension $(M \times M)$.

The kernel estimation accuracy becomes the major problem in the practical applications. It was shown that the Volterra operators are homogeneous and generally not orthogonal. As a consequence of this last characteristic the Volterra kernels can not be measured using the cross correlation techniques and the values of the Volterra kernels will depend on the order of the Volterra representation being used [2]. If the order of the Volterra model is changed the Volterra kernels will change and they must be recalculated. However, for an input having a symmetric amplitude density function, such as the Gaussian noise, the odd order Volterra functionals are orthogonal to the even order Volterra functionals. It follows that for this type of input, a 2nd order Volterra model, with zero DC component, is an orthogonal model. This leads to direct Volterra kernel measurement by the cross-correlation methods [5].

For higher order Volterra filters adaptive methods of kernels estimation are widely used. Due to the linearity of the input-output relation according to the kernels, or filter coefficients, the application of adaptive algorithms for the Volterra filters implementation is quite simple. The nonlinearity is reflected only by multiple products between the delayed versions of the input signal.

Next we will introduce the input vectors corresponding to different orders kernels. The first order input vector is defined as follows:

$$X_1 = \begin{bmatrix} x[n] & x[n-1] & x[n-2] \end{bmatrix} \tag{3}$$

”The second order input vector” can be expressed by:

$$X_2 = X_1' * X_1 \tag{4}$$

For ”the third order input vector” we propose to express the multiple input delayed signal products in relation (1) by matrices elements. These matrices can be generated by multiplying the ”second order input vector” defined according to relation (4) by the elements of the first order input vector. If we consider equal filters length memories, $M = 3$, it follows:

$$X_2 * x[n] = \begin{bmatrix} x^3[n] & x^2[n]x[n-1] & x^2[n]x[n-2] \\ \dots & x[n]x^2[n-1] & x[n]x[n-1]x[n-2] \\ \dots & \dots & x[n]x^2[n-2] \end{bmatrix} \tag{5}$$

$$X_2 * x[n-1] = \begin{bmatrix} \dots & \dots & \dots \\ \dots & x^3[n-1] & x^2[n-1]x[n] \\ \dots & \dots & x[n-1]x^2[n-2] \end{bmatrix} \tag{6}$$

$$X_2 * x[n-2] = \begin{bmatrix} \dots & \dots & \dots \\ \dots & \dots & \dots \\ \dots & \dots & x^3[n-2] \end{bmatrix} \tag{7}$$

Hence, ”the third order input vector” consists in fact of M matrices as indicated in relations (5)÷(7) and corresponds to a symmetric third order Volterra kernel.

This decomposition of ”the third order input vector” will be implemented for the RLS Volterra filter.

3 Volterra Kernel Estimation Using the RLS Adaptive Algorithm

A typical adaptive technique is shown in Fig.1.

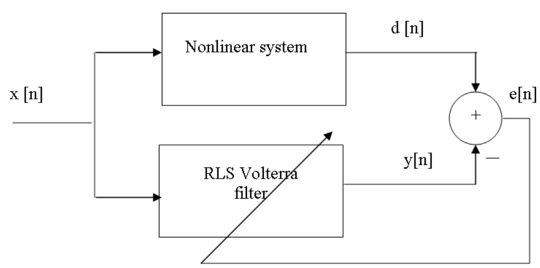

Fig. 1. The RLS Volterra Identifier

The Volterra filter of a fixed order and a fixed memory adapts to the unknown nonlinear system using one of the various adaptive algorithms. The use of adaptive techniques for Volterra kernel estimation has been well studied. Most of the previous research consider 2nd order Volterra filters and some consider the 3rd order case [6],[7]. A simple and commonly used algorithm is based on the LMS adaptation criterion. Adaptive Volterra filters based on the LMS adaptation algorithm are computational simple but suffer from slow and input signal dependant convergence behavior and hence are not useful in many applications.

The aim of this section is to discuss the efficient implementation of the RLS adaptive algorithm on a third order Volterra filter. For simplicity we have considered only odd order nonlinearities, up to the third order, in the system being identified. Due to the linearity of the input-output relation of the Volterra model with respect to filter coefficients, the implementation of the RLS algorithm can be realized as an extension of the RLS algorithm for linear filters.

Hence we define the extended input vector, for a third order Volterra filter which has only odd order kernels, as:

$$X = \left[\begin{array}{cccccccc} x[n] & \ldots & x[n-M+1] & x^3[n] & x^2[n]x[n-1] & \ldots & x^3[n-M+1] \end{array} \right] \quad (8)$$

and the extended filter coefficients vector as:

$$H = \left[\begin{array}{cccccccc} h_0 & \ldots & h_{M-1} & h_{000} & h_{001} & \ldots & h_{M-1M-1M-1} \end{array} \right] \quad (9)$$

The elements of the extended input vector can be easily actualized based on the first order and ”third order input vectors” using the proposed relations (5)÷(7).

As in the linear case the adaptive nonlinear system minimizes the following cost function at each time:

$$J(n) = \sum_{k=0}^{n} \lambda^{n-k} \left(d(k) - H(n) X'(k) \right)^2 \quad (10)$$

where $H(n)$ and $X(n)$ are the coefficients and the input signal vectors, respectively, as defined in (9) and (8), λ is a factor that controls the memory span of the adaptive filter and $d(k)$ represents the desired output. The solution of equation (10) can be obtained recursively using the RLS algorithm.

The RLS algorithm updates the filter coefficients according to the following steps:

I. Initialization:

$define\ the\ filter\ memory(lenght\ for\ H(n)\ and\ X(n))$

$H(0) = [0\ 0\ \ldots.0];$

$C_{XX}(0) = \delta * I$ where δ is a small positive constant;

II. Operations: for $n = \overline{1, nr.of\ iterations}$

1. Create the input vector:

$X(n)$

2. Compute the error:

$e(n/n-1) = d(n) - H(n-1) * X'(n);$

3. Compute the scalar:
   $\mu(n) = X(n) * C_{XX}(n-1) * X'(n);$
4. Compute the matrix:
   $G(n) = (C_{XX}(n-1) * H'(n-1)) / (\lambda + \mu);$
5. Updates the filter vector:
   $H(n) = H(n-1) + e(n/n-1) * G'(n);$
6. Updates the matrix C_{XX}:
   $C_{XX} = \lambda^{-1} * (C_{XX}(n-1) - G(n) * X(n) * C_{XX}(n-1));$

In the relations above C_{XX} denotes the inverse autocorrelation matrix of the extended input signal. Inversion was done according to the matrix inversion lemma [8].

4 Experiments and Results

The nonlinear system with memory being identified consists of a linear filter with impulse response given by:

$$h[n] = 2^{-(n+1)} \quad 0 \leq n \leq M_1 - 1 \tag{11}$$

followed by a nonlinear system without memory which input-output relation is:

$$y[n] = x[n] + b * x^3[n] \tag{12}$$

The linear filter memory in relation (11) is $M_1 = 10$. The coefficient b permits to change the nonlinearity degree.

The input signal $x[n]$ is a white Gaussian sequence $z[n]$, colored using the autoregresive filter described by:

$$x[n] = x[n-1] - 0,9x[n-2] + 0,5z[n] \tag{13}$$

Experiments have been done regarding the identification of a third order nonlinear system with different degrees of nonlinearity. The performance of the RLS adaptive Volterra filter was appreciated by comparing the error of the nonlinear identifier with the error of a linear identifier. The same memory $M = 10$ has been chosen for the linear and for the third order Volterra kernel according to the relation (9). For the linear identifier we fixed the same memory. The factor λ was chosen equal to $0,995$.

The simulations have been done in the MATLAB software.

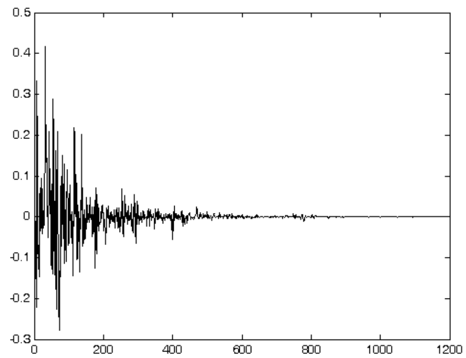

Fig. 2. The error of the RLS Volterra identifier: b=0.01

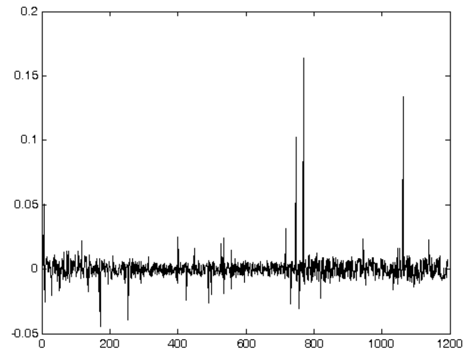

Fig. 3. The error of the RLS linear identifier: b=0.01

Nonlinear system identified: $b = 0,01$. The error using the RLS Volterra identifier is indicated in Fig.2.

Nonlinear system identified: $b = 0,1$. The error using the RLS Volterra identifier is indicated in Fig.4 and the error using the linear identifier is depicted in Fig.5.

Nonlinear system identified: $b = 1$. The error signal using the RLS Volterra identifier is indicated in Fig.6 and the error using the linear identifier is depicted in Fig.7.

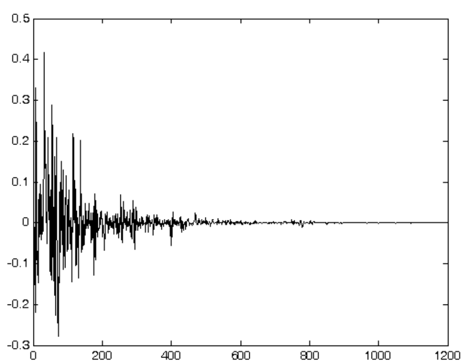

Fig. 4. The error of the RLS Volterra identifier: b=0.1

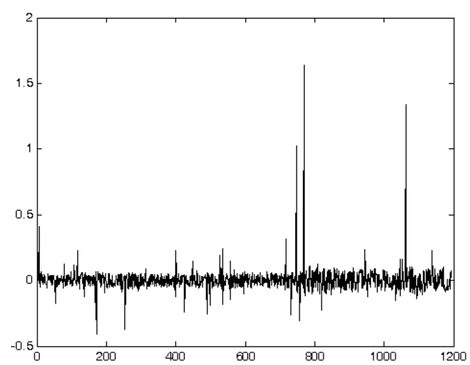

Fig. 5. The error of the RLS linear identifier: b=0.1

The mean and the dispersion of the error signals were calculated too, in order to characterize the performances of both identifiers. The corresponding values are indicated in Table 1. As it can be seen the nonlinear identifier performances (m_{eV},

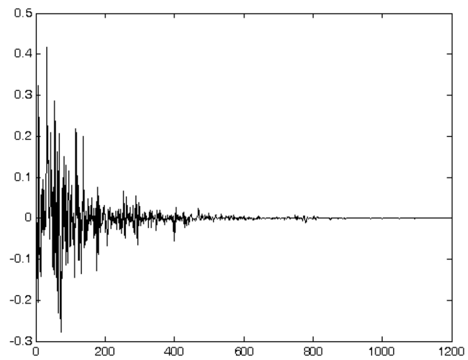

Fig. 6. The error of the RLS Volterra identifier: b=1

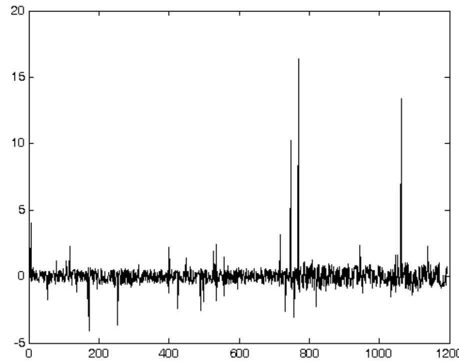

Fig. 7. The error of the RLS linear identifier: b=1

σ_{eV}) are very good for different nonlinearity degrees, while the linear identifier performances (m_{el}, σ_{el}) are unsatisfactory when the nonlinearity increases.

Table 1. The identifiers parameters

b	m_y	σ_y	m_{el}	σ_{el}	m_{eV}	σ_{eV}
0.01	$-76 \cdot 10^{-4}$	0.606	$2.38 \cdot 10^{-4}$	0.0083	$-2.32 \cdot 10^{-6}$	0.0367
0.1	$-85 \cdot 10^{-4}$	0.5972	0.0025	0.080	$-2.33 \cdot 10^{-6}$	0.0368
1	0.0123	$1,63$	0.025	$0,8$	$1.193 \cdot 10^{-5}$	0.0366

5 Conclusions

In this paper we have proposed a new implementation of the third order RLS Volterra filter based on the decomposition of the input signal and on symmetric kernels assumption. For that purpose we have introduced the concepts of ”second order input vector” and ”third order input vector”. For simplicity, for the RLS Volterra implementation there were considered only the first and the third order kernels, but also the second order kernel might be included.

The nonlinear adaptive filter performances were evaluated in a typical system identification application and compared with the performances of a linear identifier. The experimental results showed that the RLS Volterra filter performed better than the linear filter which performances whose inacceptable when the nonlinearity degree was increased.

The costs of these performances are paid by the computational complexity required by the nonlinear adaptive Volterra filter implementation .

References

[1] V. J. Mathews, "Adaptive polynomial filters," *IEEE, Signal Processing Magazine*, pp. 10–25, July 1991.

[2] J. Tsimbinos, "Identification and compensation of nonlinear distortion," Ph.D. dissertation, Institute for Telecommunications Research, University of South Australia, 1995. [Online]. Available: http://www.unisa.edu.au/html

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

[4] G. Budura and I. Nafornița, "Kernels measurement techniques for constructing nonlinear models," in *Proc. of the Symposium on Electronics and Telecomm, ETC 2002*, Timişoara, 2002, pp. 190–195.

[5] G. Budura and C. Botoca, "La construction d'un modele non lineaire a l'aide de series Volterra et Wiener," *Revue de l'Academie roumaine, Bucharest*, 2005, accepted to be published.

[6] ——, "Applications of the volterra models in nonlinear systems identification," in *Proc. of the Symposium on Electronics and Telecomm, ETC 2002*, vol. I, Timişoara, Sept. 2002, pp. 196–201.

[7] ——, "Efficient implementation of the second order Volterra filter," *Scientific Bulletin of the "Politehnica", University of Timişoara, Romania, Transactions on Electronics and Communications*, vol. 49, no. 63, pp. 91–95, 2004, fascicle 2.

[8] R. E. Morthensen, *Random Signals and Systems*. New York: John Wiley & Sons, 1987.