HOS-BASED VARIABLE STEP VOLTERRA FILTER

KAMAL MEGHRICHE[1], NACER K. M'SIRDI[2], MOURAD HADDADI[3]
[1,2] Versailles Laboratory of Robotics, 10-12 Avenue de l'Europe, 78140 Vélizy, FRANCE
[3] ENP, Département d'Electronique, Laboratoire de Traitement du Signal, 10, Avenue des Frères Ouadek, B.P. 182, 16200 Algiers, ALGERIA.

Abstract – In this paper, we are to develop a second-order LMS-type Volterra filter to reduce distortions of data transmission over analog telephone channels due the channel impulse response and inter-symbol interference (ISI).

A novel approach for updating the linear and quadratic coefficients vectors of a second-order Volterra filter is presented. The innovative features of this algorithm provide distinct convergence treatment of the linear and quadratic Volterra filter kernels, and adaptive formulation for the convergence factors, hence providing a unique platform for both linear and nonlinear channel adaptive handling.

Simulations are carried in an equalization set-up to compare the performance of this algorithm with conventional and variable step LMS algorithms. The obtained results show that this algorithm brings substantial increase in the convergence speed while keeping, to some extent, the simplicity of the conventional LMS algorithm. Considering the performance of the formulated algorithm, it could be a useful tool for adaptive equalization applications.

Key-Words: - Adaptive Filtering, Equalization, HOS-based LMS, Nonlinear Volterra filter.

1 Introduction

It is well known that linear filtering came across an unbeatable adversary when it comes to filter out impulsive noise. In fact, linear filters are not able to provide satisfactory performance when dealing with nonlinear noise sources. In data communication over a non-ideal communication channel, it has been shown that nonlinear adaptive equalization has to be used to come over the most encountered types of noise, dispersion and intersymbol interference (ISI) [1],[2]. HOS are becoming a very attractive tool to deal with nonlinearities. A good survey on HOS is given in [3] and [4]. Nonlinear adaptive filters and in particular Volterra filters have met lot of attention and found wide application in various fields. This 'success' is mainly due the convergence behavior and the parameter linearity of such filters. In [5], a comparative study of steady-state performance of adaptive algorithm bypassing nonlinearities in their update equations is given. [6] proposed a confined variable step-size normalized LMS, and [7] presented a segmentary update of NLMS-like algorithm, intrinsically reducing the algorithm convergence dependence on the adaptation step.

A variety of nonlinear Volterra filters have been reported in literature [8-14] to name but a few, where convergence treatment is performed on the input signal regardless being linear or nonlinear. In fact, a single convergence step expression is used for the adaptive algorithm.

In this paper, we propose a novel approach to update recursively the convergence factor of a least-mean-square (LMS) algorithm used to minimize the mean square error. This approach is distinctively applied to both the linear and quadratic part of a second order Volterra filter.

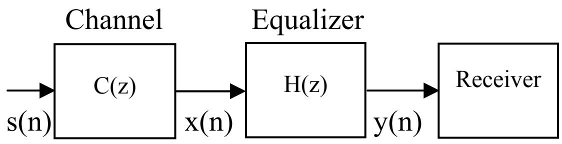

Fig. 1 Communication system block diagram

This paper is organized as follows. In section 2, we formulate the problem and give the necessary background and analysis. Section 3 presents the adaptive equalization of a nonlinear communication channel using a variable step adaptive algorithm and states the convergence conditions. In section 4, we discuss the obtained results and we draw conclusions in section 5.

2 Problem Formulation and Analysis

A general discrete-time system may be represented using a nonlinear model via Volterra series [15].

The input and output of a discrete-time linear system are related via

$$y(k) = \sum_{-\infty}^{\infty} h(i)x(k-i) \qquad (1)$$

where *x(k)* is the input, *y(k)* is the output and *h(i)* is the impulse response of the system.

The linear expression (1) is simple and widely used to represent practical systems with limited duration of *h(i)*. But in many cases where we use (1), we only get an approximation to the real world systems. So if more accuracy of the representation is needed, then a more complex model may be used by introducing the concept of Volterra series.

A general discrete-time system may be represented using a nonlinear model via Volterra series

$$y(k) = \sum_{i=-\infty}^{+\infty} h_1(i)x(k-i) + \sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty} h_2(i,j)x(k-i)x(k-j) + \sum_{i=-\infty}^{+\infty}\sum_{j=-\infty}^{+\infty}\sum_{l=-\infty}^{+\infty} h_3(i,j,l)x(k-i)x(k-j)x(k-l) + ... \qquad (2)$$

For practical reasons, this representation is truncated at the cubic term with finite kernels, designed in (2) by $h_1(i)$, $h_2(i,j)$ and $h_3(i,j,l)$, also called Volterra Kernels.

The problem here is to find the impulse response of Wiener Filter that minimizes the functional [1],[16]

$$J = E\left[e^2(k)\right] = E[(d(k) - y(k))^2] \qquad (3)$$

Where *d(k)* is the desired output.

For the linear case, the classical Wiener filter minimizes *J* when

$$y(k) = \sum_{i=0}^{N} h(i)x(k-i) \qquad (4)$$

Considering a quadratic (nonlinear) filter given by

$$y(k) = \sum_{i=0}^{N} h_1(i)x(k-i) + \sum_{i=0}^{M}\sum_{j=0}^{M} h_2(i,j)x(k-i)x(k-j) \qquad (5)$$

The problem is to minimize (3) when the output of Wiener filter is given by (5). We proceed as in the classical derivation of Wiener filter. First, form the data vector *X* and the coefficients vector *H*. From (6), one can see that the dimensions of *X* and *H* are $(N+1) + (M+1)^2$.

One of the properties of the Volterra kernels is that the quadratic kernel $h_2(i,j)$ is symmetric, which means that we do not have to consider all the elements of h_2 and consequently the dimensions of *X* and *H* become $(N+1) + (M+1)(M+2)/2$.

$$X = \begin{pmatrix} x(k) \\ x(k-1) \\ \\ x(k-N) \\ x^2(k) \\ x(k)x(k-1) \\ x(k)x(k-2) \\ \\ x^2(k-1) \\ \\ x^2(k-M) \end{pmatrix}, \quad H = \begin{pmatrix} h_1(0) \\ h_1(1) \\ \\ h_1(N) \\ h_2(0,0) \\ h_2(0,1) \\ \\ h_2(1,0) \\ h_2(1,1) \\ \\ h_2(M,M) \end{pmatrix} \qquad (6)$$

Expression (5) becomes

$$y(k) = X^T H = H^T X \qquad (7)$$

and the functional *J* of (3) takes the form

$$J_Q = E\left[\left(d(k) - X^T H\right)^2\right] \qquad (8)$$

Developing (8) leads to

$$J_Q = E\left[d^2(k)\right] - 2E\left[d(k)X^T\right]H + H^T E\left[XX^T\right]H \qquad (9)$$

Differentiating J_Q with respect to the coefficients of *H* and setting the resulting gradient to zero gives the following normal set of equations:

$$\Re H = P \qquad (10)$$

where $\Re = E(XX^T)$ (11a)

$P = E(d(k)X)$ (11b)

In this case, $\Re$ is no longer an autocorrelation matrix as in the classical case. For this quadratic filter, $\Re$

includes second order statistics, third order statistics and fourth order statistics.

To allow distinct processing of linear and nonlinear parts of the input signal sequence, X may be partitioned as $X = \begin{pmatrix} X_1 \\ X_2 \end{pmatrix}$, where X_1 is the linear part and X_2 is the quadratic part that involves $x(k-i)x(k-j)$.

Then

$$E(XX^T) = E\left\{ \begin{pmatrix} X_1 \\ X_2 \end{pmatrix} \begin{pmatrix} X_1^T & X_2^T \end{pmatrix} \right\} = \begin{pmatrix} E\left(X_1X_1^T\right) & E\left(X_1X_2^T\right) \\ E\left(X_2X_1^T\right) & E\left(X_2X_2^T\right) \end{pmatrix} = \Re$$

Given the input data $x(k)$ and the reference signal $d(k)$, the system (10) can be solved for H.

The use of Auto Regressive (AR) modeling can be extended to nonlinear case via Volterra series, where the use of HOS becomes necessary. Recall that an AR linear model is given by

$$y(k) = b_0 x(k) - \sum_{i=1}^{n} a_i y(k-i) \tag{12}$$

If one considers linear models, then a simple extension is possible using Volterra series. Most of the time, adding a quadratic term to (12) is sufficient; otherwise, the model becomes very cumbersome. A typical quadratic AR model can be written as

$$y(k) = b_0 x(k) - \sum_{i=1}^{n} a_i y(k-i) - \sum_{i=0}^{s}\sum_{j=0}^{s} c_{ij}\, y(k-i)\, y(k-j) \tag{13}$$

with the assumption that $x(k)$ and $y(k)$ are uncorrelated.

Notice that for all those nonlinear models, the output is a nonlinear function of the data (input and/or previous output) but it is linear in the coefficients. So, the computation of the latter can be performed using the classical well known methods of parametric modeling and optimal filtering.

3 Proposed Algorithm and Convergence Analysis

It this section, we describe the proposed novel approach to update the linear and quadratic kernels coefficients of a nonlinear adaptive Volterra filter and state the convergence conditions.

If we consider again the quadratic Wiener filtering problem given previously, the adaptive version using LMS will have the form

$$H_k = H_{k-1} + \mu e_k X_k \tag{14}$$

The index k is added here to indicate the iteration number.

e_k is the error given by $e_k = d_k - y_k$
and X_k is given by

$$X_k = \begin{pmatrix} x(k) & x(k-1) & & x(k-N) & x^2(k) & x(k)x(k-1) & & x^2(k-M) \end{pmatrix}^T \tag{15}$$

It is clear that (14) uses (implicitly) HOS to update the coefficients vector H_k.

In this paper, the step size μ is not the same for the linear and nonlinear parts of X_k. In this case, (14) becomes

$$H_k = H_{k-1} + e_k \begin{pmatrix} \mu_1 X_{k1} \\ \mu_2 X_{k2} \end{pmatrix} \tag{16a}$$

where

$$X_{k1} = \begin{pmatrix} x(k) \\ \\ x(k-N) \end{pmatrix}, \quad X_{k2} = \begin{pmatrix} x^2(k) \\ x(k)x(k-1) \\ \\ x^2(k-M) \end{pmatrix} \tag{16b}$$

and μ_1 and μ_2 are the adaptation steps for the linear and nonlinear parts respectively.

Using a similar approach as in [17], [18], the linear and quadratic parts adaptation steps are given by

$$\mu_{i_k} = \mu_{i_{k-1}} + \frac{1}{\sum_{j=1}^{N+1,(M+1)^2} X^2(k-j)} \quad i = 1,2 \tag{17}$$

where $(N+1)$ and $(M+1)^2$ are the orders of the linear and nonlinear filter kernels orders respectively.

After the first $(N+1)$ and $(M+1)^2$ first initializing iterations, μ_1 and μ_2 can be reformulated as

$$\mu_{i_k} = \frac{\mu_{i_{k-1}}}{1 + \Delta_i \mu_{i_{k-1}}} \quad i = 1,2$$

where $\Delta_1 = X^2(k) - X^2(k-N)$
and $\Delta_2 = X^2(k) - X^2(k-M^2)$
or again as,

$$\mu_{i_k} = \mu_{i_{k-1}} - \frac{\Delta_i \mu_{i_k}^2}{1 + \Delta_i \mu_{i_k}} \quad i = 1,2 \tag{18}$$

Finally, replacing μ by $1/\mu'$, equation (18) becomes

$$\mu'_{i_k} = \mu'_{i_{k-1}} + \Delta_i \quad i = 1,2 \tag{19}$$

thus reducing the number of extra mathematical operations required to update the coefficients vectors of (16a). This gives

$$H_{k+1} = H_k + e_k \begin{pmatrix} X_{1_k} / \mu'_{1_k} \\ X_{2_k} / \mu'_{2_k} \end{pmatrix} \tag{20}$$

From (17), it can be show that, as $k \rightarrow \infty$, $\mu_{i_k} \rightarrow \mu_{i_{k-1}}$ leading to similar convergence conditions on the adaptation step as in the classical LMS algorithm using a constant convergence factor, namely for the linear kernel adaptation step. Next, we state the convergence conditions on the adaptation step μ'_2 (or μ_2) for the quadratic filter part.

From (20), we get

$$H_{k+1} - H_k = e_k \begin{pmatrix} X_{1_k} / \mu'_{1_k} \\ X_{2_k} / \mu'_{2_k} \end{pmatrix}$$

or

$$Q_k^T (H_{k+1} - H_k) = e_k Q_k^T Q_k \tag{21}$$

with $Q_k = \begin{pmatrix} X_{1_k} / \mu'_{1_k} \\ X_{2_k} / \mu'_{2_k} \end{pmatrix}$ and T being the transpose operator.

Letting $\alpha = Q_k^T Q_k \quad and \quad \beta = Q_{k+1}^T Q_{k+1}$

(21) can be rewritten as

$$e_k = \frac{Q_k^T (H_k - H_{k-1})}{\alpha} \tag{22}$$

Similarly

$$e_{k+1} = \frac{Q_{k+1}^T (H_{k+1} - H_k)}{\beta} \tag{23}$$

The pseudo-random binary sequence (PRBS), used to drive the algorithm (input signal X_k), being bounded, applying the Milosavljević convergence condition [19]

$$(S_{k+1} - S_k) S_k < 0$$

and letting $S_k = e_k^T e_k \quad and \quad S_{k+1} = e_{k+1}^T e_{k+1}$, gives

$$\begin{aligned} &\{ \frac{1}{\beta^2} \left[(H_{k+1} - H_k)^T Q_{k+1} Q_{k+1}^T (H_{k+1} - H_k) \right] \\ &- \frac{1}{\alpha^2} \left[(H_k - H_{k-1})^T Q_k Q_k^T (H_k - H_{k-1}) \right] \} \\ &\{ \frac{1}{\alpha^2} \left[(H_k - H_{k-1})^T Q_k Q_k^T (H_k - H_{k-1}) \right] \} < 0 \end{aligned} \tag{24}$$

or $\left(\frac{\Gamma_1}{\beta^2} - \frac{\Gamma_2}{\alpha^2} \right) \frac{\Gamma_2}{\alpha^2} < 0$

with

$$\Gamma_1 = (H_{k+1} - H_k)^T Q_{k+1} Q_{k+1}^T (H_{k+1} - H_k)$$
$$\Gamma_2 = (H_k - H_{k-1})^T Q_k Q_k^T (H_k - H_{k-1})$$

Ultimately, to ensure convergence,

$$\Gamma_1 < \frac{\beta^2}{\alpha^2} \Gamma_2 \tag{25}$$

leading to $\mu'_2 > \mu'_1$, or $\mu_2 < \mu_1$.

4 Simulation Results and Discussion

The fist step in evaluating the performance of the variable step Volterra-based adaptive algorithm was to build a simulation model which was programmed in Matlab. Figure 2 shows the basic block diagram of the simulation model.

The simulation initial adaptation steps are μ_1=0.2 and μ_2=0.001. The linear filter part order is M=8 and the quadratic filter part order is N=4. Simulation is carried out for 1000 signal samples and results are averaged over 200 runs.

A PRBS (sequence) of period 14 is generated to feed the simulation block with a bipolar input signal.

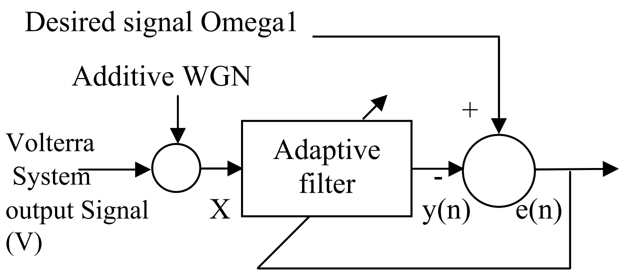

Fig. 2 Adaptive Volterra filter simulation setup

For all the remaining figures, except Fig. 8, only the first 100 samples out of 1000 are shown for figure clarity purpose.

Figure 3 shows the primary signal while Fig. 4 represents the Volterra system output signal when fed by a PRBS. Figure 5 shows the highly noise corrupted signal fed to the variable step nonlinear equalizer with a signal-to-noise ratio (SNR) of 10 dB.

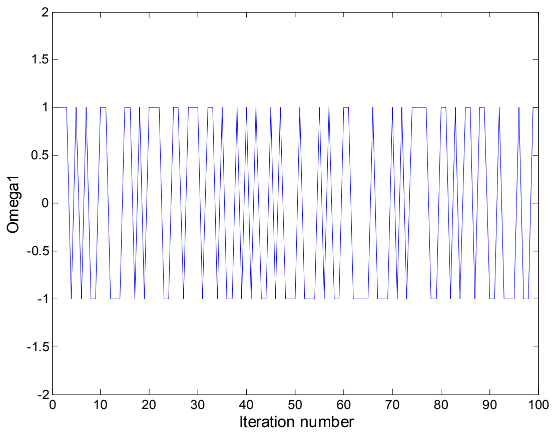

Fig. 3 Desired signal

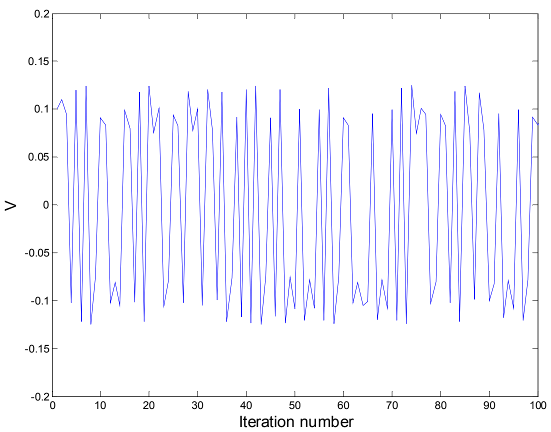

Fig. 4 Volterra system output

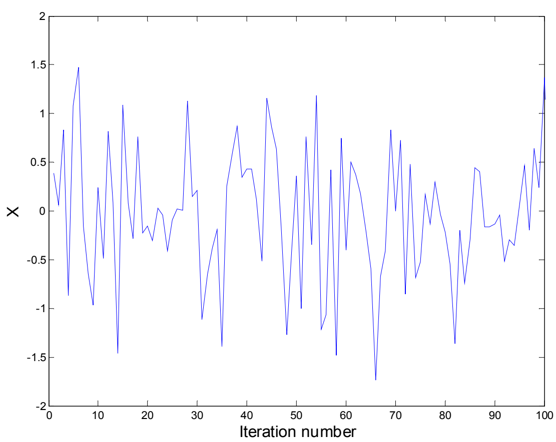

Fig. 5 Noise corrupted signal (SNR=10dB)

In Fig. 6, the variable step nonlinear filter starts the adaptation process to converge towards the desired signal. Convergence is obtained after 50 iterations.

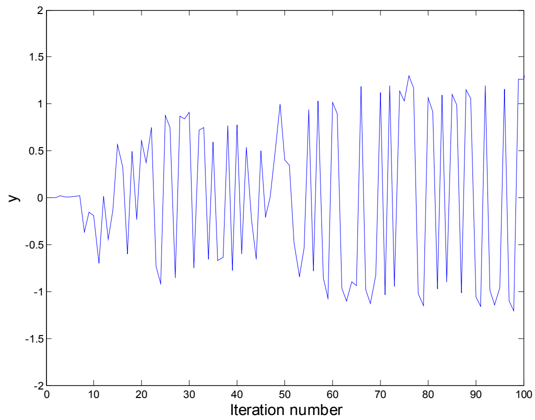

Fig. 6 Adaptive equalizer output

Figure 7 shows coefficients variations of the linear and quadratic filter kernels.

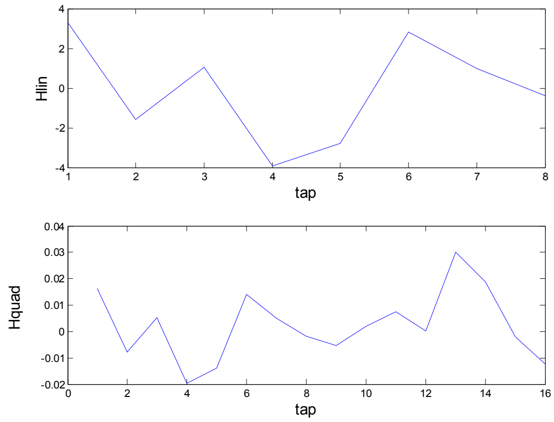

Fig. 7 Coefficients of linear (Hlin) and quadratic (Hquad) filter kernels

The mean-square error (MSE), see figure 8, which is a performance measure of the algorithm, converges faster compared to conventional LMS algorithm within similar simulation conditions.

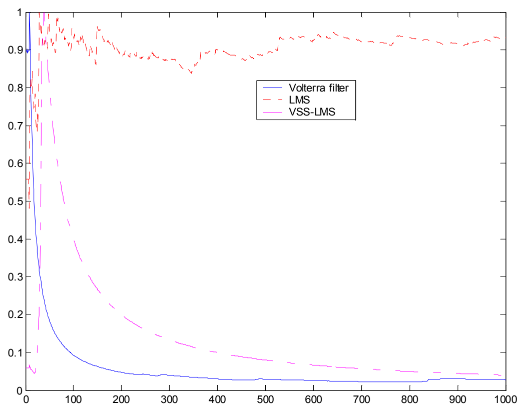

Fig. 8 Comparison of learning curves, for the same nonlinear input signal X, of standard LMS, VSS-LMS and variable step Volterra filter

5 Conclusion

This paper has described a new formulation of a variable adaptation factor for a nonlinear Volterra-based adaptive equalizer. Compared to the constant step-size algorithm, this expression allows faster convergence and better tracking performance.

As such, this variable step algorithm formulation could be used as a cost effective method of implementing the adaptive equalization of digital data transmission over nonlinear communication channels.

Using this type of nonlinear adaptive filtering, the performance has been significantly improved because extra information about the data is used. Application could be extended to digital data communication where echo canceling and other filtering operations are needed.

References:

[1] S. Haykin, *Adaptive Filter Theory,* Englewood Cliffs, NJ: Prentice-Hall, 1991.

[2] J. Mendel, "Tutorial on Higher Order Statistics (Spectra) in Signal Processing and Systems Theory: Theoretical Results and some Applications", *Proc. IEEE*, Vol. 79, 1991, pp. 278-305.

[3] C. L. Nikias, and J. M. Mendel, "Signal Processing with Higher-Order Spectra", *IEEE Signal Processing Magazine*, 1993, pp. 10-37.

[4] G. B. Giannakis, and E. Serpedin, "A Bibliography on Nonlinear System Identification", *Signal Processing, Elsevier Science B.V.*, Vol. 81, No. 3, 2001, pp. 533-580.

[5] N. R. Yousef, and Ali H. Sayed, "A Unified Approach to the Steady-State and Tracking Analyses of Adaptive Filters", *IEEE Trans. on Signal Proc.*, Vol. 49, No. 2, pp. 314-324, 2001.

[6] A. I. Sulyman, and A. Zerguine, "Convergence and Steady-State Analysis of a Variable Step-Size NLMS Algorithm", *Signal Processing, Elsevier Science B.V.*, Vol. 83, No. 6, 2003, pp. 1255-1273.

[7] Fotis E. Andritsopoulos & al., "Time Distributed Update of the NLMS Algorithm Coefficients for Acoustic Echo Cancellers", *Proc. WSEAS MATH 2004 Symposium*, paper N° 484-291, Miami, Florida, April 21-24, 2004.

[8] G. L. Scuranza, "Quadratic Filters for Signal Processing", *Proc. IEEE*, Vol. 80, No. 8, 1992, pp. 1263-1285.

[9] S. Suna, R. Tetzlaff, D. Wolf, "Identification of Nonlinear Systems by Volterra Systems", *Proc. European Conference on Circuit Theory and Design* (ECCTD'97), 1997, Budapest.

[10] X. Li, W. K. Jenkins, "A Computationally Efficient Algorithm for Adaptive Quadratic Volterra Filters", *Proc. IEEE International Symposium on Circuits and Systems*, Hong Kong, 1997, pp. 2184-2187.

[11] D. W. Griffith, J. R. Arce, "Partially Decoupled Volterra Filters: Formulation and LMS Adaptation", *IEEE Trans. on Signal Processing*, Vol. 45, No. 6, 1997, pp. 1485-1494.

[12] A. Stenger, L. Trautmann, R. Rabenstein, "Nonlinear Acoustic Echo Cancellation with 2nd Order Adaptive Volterra Filters", *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Phoenix, USA, 1999, pp. 877-880.

[13] E. Mumolo, A. Carini, "On the Stability of Discrete Time Volterra Filters", *IEEE Signal Processing Letters*, Vol. 6, No. 9, 1999, pp. 230-232.

[14] Sunil Gupta, "Performance Evaluation of an Adaptive Volterra/Hybrid Equalizer in a Nonlinear Magneto-Optic Data Storage Channel", *WSEAS Transactions on Circuits and Systems*, Issue 4, Vol. 2 , 2003, pp. 832-835.

[15] Andrzej Borys, *Nonlinear Aspects of Telecommunications: Discrete Volterra Series and Nonlinear Echo Cancellation*, CRC Press, 2000.

[16] B. Widrow, and S. D. Stearns, A*daptive Signal Processing,* Englewood Cliffs, NJ: Prentice-Hall, 1985.

[17] K. Meghriche, & al, "Adaptive Nonlinear Filter for Digital Data Communication", *Proc. IEEE-ISSPA-2001, 6th International Symposium on Signal Processing and its Applications*, Kuala Lumpur, Malaysia, 2001.

[18] K. Meghriche, and B. Derras, "Tracking Behavior of the LMS Algorithm in a Nonstationary Environment", *Proceedings COMAEI'96*, Tlemcen, Algeria, 1996.

[19] Č. Milosavljević, "General conditions for the existence of a quasi-sliding mode on the switching hyperplane in discrete variable structure systems", *Automation & Remote Control*, N°. 46, 1985, pp. 307-314.