Adaptive Volterra Filters Using Orthogonal Structures

V. John Mathews, *Senior Member, IEEE*

***Abstract*—This paper presents an adaptive Volterra filter that empolys a recently developed orthogonalization procedure of Gaussian signals for Volterra system identification. The algorithm is capable of handling arbitrary orders of nonlinearity P as well as arbitrary lengths of memory N for the system model. The adaptive filter consists of a linear lattice predictor of order N, a set of Gram–Schmidt orthogonalizers for N vectors of size $P+1$ elements each, and a joint process estimator in which each coefficient is adapted individually. The complexity of implementing this adaptive filter is comparable to the complexity of the system model when N is much larger than P, a condition that is true in many practical situations. Experimental results demonstrating the capabilities of the algorithm are also presented in the paper.**

I. Introduction

TRUNCATED Volterra series models have become very popular in adaptive nonlinear filtering applications [3]. Several stochastic gradient (SG) and recursive least-squares (RLS) adaptive Volterra filters have been developed in the last fifteen years or so [2]–[4], [6]. The SG algorithms are, in general, easy to derive and implement. However, they show slow and input-signal-dependent convergence characteristics. The RLS algorithms, on the other hand, exhibit fast convergence characteristics that are more or less independent of the input signal statistics. However, unlike their linear counterparts, even the most efficient RLS Volterra filters have significantly larger computational complexity than the SG Volterra filters.

One approach to improving the convergence characteristics of the SG adaptive filters is to employ structures that orthogonalize the input signal. Unfortunately, the lattice realizations of Volterra systems for arbitrary inputs are over-parameterized [4]. For example, the lattice realization of a second-order Volterra system with N-sample memory requires $O(N^3)$ parameters, even though the system model itself has only $O(N^2)$ parameters. Consequently, SG adaptive filters employing such structures have computational complexity that is comparable to the RLS algorithms. This paper presents an approach for developing adaptive lattice Volterra filters with computational complexity comparable to that of the system model when the input signal is Gaussian distributed. The derivations utilize a recently developed method for orthogonalizing Gaussian input signals for Volterra system identification tasks [5].

Manuscript received March 15, 1996. This work was supported in part by an IBM Departmental Grant. A version of this paper was presented at the IEEE International Conference on Acoustics, Speech, and Signal Processing, Detroit, MI, May 8-12, 1995. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. J. J. Shynk.

The author is with the Department of Electrical Engineering, University of Utah, Salt Lake City, UT 84112 USA (e-mail: mathews@ee.utah.edu).

Publisher Item Identifier S 1070-9908(96)09031-1.

II. Orthogonalization of Guassian Signals for Volterra System Identification

Consider a finite-memory and finite order Volterra system represented by the input–output relationship

$$y(n) = h_0 + \sum_{p=1}^{P} \overline{h}_p[x(n)] \tag{1}$$

where $x(n)$ is the input signal to the system, $y(n)$ is the output of the system, and

$$\overline{h}_p[x(n)] = \sum_{m_1=0}^{N-1} \sum_{m_2=m_1}^{N-1} \cdots \sum_{m_p=m_{p-1}}^{N-1} h_p(m_1, m_2, \cdots, m_p) \times x(n-m_1)x(n-m_2)\cdots x(n-m_p). \tag{2}$$

The above model incorporates the kernel symmetry without any loss of generality. The coefficients of the expression in (1) can be uniquely estimated under some mild conditions on the input signal.

All the products of input signal samples employed in (2) belong to the set

$$\{x^{m_1}(n)x^{m_2}(n-1)\cdots x^{m_N}(n-N+1)| \; m_1 + m_2 + \cdots m_N \le P\}. \tag{3}$$

The problem considered in this section is the orthogonalization of the elements of the input signal set in (3). The orthogonality is in the minimum mean-square error sense. We assume that the input signal is stationary and Gaussian with zero-mean value. The assumption that the input signal has zero mean value is not restrictive, since the mean value can be removed from any signal and the bias term h_0 in (1) can account for any contribution from the nonzero mean value of the input signal.

Consider the input vector

$$\boldsymbol{X}_L(n) = [x(n), x(n-1), \cdots, x(n-N+1)]^T \tag{4}$$

which consists only of the linear components in the input signal set in (3). We can find an orthonormal basis set for the elements of $\boldsymbol{X}_L(n)$ using a normalized lattice predictor [1]. Let $u_i(n); i = 1, 2, \cdots, N$ represent the orthogonal basis signals generated by the linear lattice predictor. Then

$$E\{u_i(n)u_j(n)\} = \delta(i-j) \tag{5}$$

where $\delta(n)$ represents the Dirac delta function.

Now, let us define a vector $\boldsymbol{U}_{P,i}(n)$ as

$$\boldsymbol{U}_{P,i}(n) = [1, u_i(n), u_i^2(n), \cdots, u_i^P(n)]^T. \tag{6}$$

Let $\boldsymbol{Q}_P$ be a lower triangular, $(P+1)\times(P+1)$ element matrix that orthogonalize $\boldsymbol{U}_{P,i}(n)$. Since all $u_i(n)'$s have identical distributions, the same $\boldsymbol{Q}_P$ will orthogonalize $\boldsymbol{U}_{P,i}(n)$ for all values of i. Let $\boldsymbol{V}_{P,i}$ be an orthogonalized vector obtained as

$$\boldsymbol{V}_{P,i} = \boldsymbol{Q}_P \boldsymbol{U}_{P,i}. \tag{7}$$

Let $v_{P,i,j}$ denote the jth element of $V_{P,i}$.

Theorem 1:

$$\{v_{P,1,m_1}(n)v_{P,2,m_2}(n)\cdots v_{P,N,m_N}(n)| \\ m_1+m_2+\cdots+m_N \leq P\}$$

is an orthogonal basis set for the signal set in (3). Note that $v_{P,i,0}(n) = 1$ for all i and that each m_i takes values from $0 \leq m_i \leq P$.

A proof for this theorem may be found in [5]. The lattice structure for second-order Volterra filters that was presented in [2] is a special case of the above procedure.

III. An Efficient Adaptive Lattice Filter for Gaussian Signals

Let $x(n)$ and $d(n)$ represent the input and desired response signals, respectively, of an adaptive filter. The objective of the adaptive Volterra filter is to model the relationship between $x(n)$ and $d(n)$ adaptively using the truncated Volterra series representation of (1) and the orthogonal structure described in the previous section.

The adaptive lattice Volterra filter consists of three stages. The first stage is an $(N-1)$-stage adaptive linear lattice predictor for the input signal $x(n)$. A normalized LMS lattice linear predictor can be realized using the following equations:

$$f_i(n) = f_{i-1}(n) - \rho_i(n)b_{i-1}(n-1) \tag{8}$$

$$b_i(n) = b_{i-1}(n-1) - \rho_i(n)f_{i-1}(n) \tag{9}$$

$$\rho_i(n+1) = \rho_i(n) + \frac{\mu}{\hat{\sigma}^2_{i-1}(n)}\{f_i(n)b_{i-1}(n-1) + b_i(n)f_{i-1}(n)\} \tag{10}$$

and

$$\hat{\sigma}^2_i(n) = \beta\hat{\sigma}^2_i(n-1) + (1-\beta)\{f^2_i(n) + b^2_i(n-1)\}. \tag{11}$$

In the above equations, $f_i(n)$ and $b_i(n)$ represent the ith-order forward prediction error and backward prediction error values, respectively, at time $n, \rho_i(n)$ is the ith reflection coefficient at time n, and μ is a small positive constant that controls the rate of convergence of the various stages of the lattice predictor. The parameter β is bounded above and below by 1 and 0, respectively, and controls the behavior of the adaptive power estimators. Usually, β is chosen as $(1-\mu)$. The prediction error signals $f_i(n)$ and $b_i(n)$ do not, in general, have unit variance. The iterations in (8) and (9) are initialized using $f_0(n) = b_0(n) = x(n)$. The reflection coefficients are initialized using some arbitrary values bounded by one. The prediction error power estimates $\hat{\sigma}^2_i(n)$ are initialized to some small, positive quantities.

The second stage of the adaptive lattice Volterra filter creates N vectors of $P+1$ elements each as

$$\boldsymbol{B}_{P,i}(n) = [1, b_i(n), b^2_i(n), \cdots, b^P_i(n)]^T \\ i = 0, 1, \cdots, N-1. \tag{12}$$

As discussed in the previous section, it is possible to design a Gram–Schmidt orthogonalizer for $\boldsymbol{B}_{P,i}(n)$ that is independent of the signal statistics when the input signals are Gaussian. However, to account for potential variations from the Gaussian distribution of the elements of $\boldsymbol{B}_{P,i}(n)$, we employ adaptive Gram–Schmidt orthogonalizers for each $\boldsymbol{B}_{P,i}(n)$. Let $u_{i,j,0}(n)$ denote the jth element of $\boldsymbol{B}_{P,i}(n)$, i.e.,

$$u_{i,j,0}(n) = b^j_i(n). \tag{13}$$

Then, the equations that describe the Gram–Schmidt orthogonalizers that employ a normalized least mean-square (LMS) adaptation algorithm are as follows:

$$u_{i,l,m}(n) = u_{i,l,m-1}(n) - \alpha_{i,l,m-1}(n)u_{i,m-1,m-1}(n) \\ l = m+1, \cdots, P \tag{14}$$

$$\hat{\gamma}^2_{i,m}(n) = \beta\hat{\gamma}^2_{i,m}(n-1) + (1-\beta)u^2_{i,m}(n) \tag{15}$$

and

$$\alpha_{i,l,m}(n+1) = \alpha_{i,l,m}(n) + \frac{\mu}{\hat{\gamma}^2_{i,m}(n)}u_{i,l,m}(n)u_{i,m,m}(n). \tag{16}$$

The third stage of the adaptive filter is the joint process estimator. The signal set that is used for joint process estimation is obtained by nonlinearly combining the various $v_{i,m}(n)$ as

$$s_{i_1,i_2,\cdots,i_N}(n) = v_{1,i_1}(n)v_{2,i_2}(n)\cdots v_{N,i_N}(n); \\ i_1+i_2+\cdots+i_N \leq P. \tag{17}$$

According to Theorem 1, the elements of the set described by the above equation will be orthogonal, or at least close to orthogonal, when the adaptive filter has converged to nearly optimal values and the input signal is Gaussian. Therefore, it is reasonable to develop the adaptive filter by individually adapting the coefficients of $s_{i_1,i_2,\cdots,i_N}(n)$. Let $\{z_k(n); k = 1, 2, \cdots, M\}$ represent an ordered arrangement of all signals $s_{i_1,i_2,\cdots,i_N}(n)$ involved in the joint process estimation. Here, M represents the total number of coefficients in the joint process estimator. The following equations represent a normalized LMS joint process estimator for the adaptive lattice Volterra filter.

$$e_k(n) = d(n) - \sum_{i=1}^{k} w_i(n)z_i(n) \\ = e_{k-1}(n) - w_k(n)z_k(n) \tag{18}$$

$$\hat{\kappa}^2_k(n) = \beta\hat{\kappa}^2_k(n-1) + (1-\beta)z^2_k(n) \tag{19}$$

and

$$w_k(n+1) = w_k(n) + \frac{\mu}{\hat{\kappa}^2_k(n)}e_k(n)z_k(n). \tag{20}$$

The recursive calculation of the error signal $e_k(n)$ in (18) is initialized using $e_0(n) = d(n)$.

IV. Experimental Results

The results presented in this section are ensemble averages over 50 independent simulations of a system identification problem. The unknown system was a second-order Volterra filter described by the following input–output relationship:

$$\begin{aligned} y(n) = &-0.78x(n) - 1.48x(n-1) + 1.39x(n-2) \\ &+ 0.04x(n-3) + 0.54x^2(n) + 3.72x(n)x(n-2) \\ &+ 1.86x(n)x(n-2) - 0.76x(n)x(n-3) \\ &- 1.62x^2(n-1) + 0.76x(n-1)x(n-2) \\ &- 0.12x(n-1)x(n-3) + 1.41x^2(n-2) \\ &- 1.52x(n-2)x(n-3) - 0.13x^2(n-3). \end{aligned} \tag{21}$$

Four different types of input signals were used in the simulations. Each signal set was generated as the output of a linear system with input–output relationship

$$x(n) = bx(n-1) + \sqrt{1-b^2}\xi(n) \tag{22}$$

where $\xi(n)$ was zero-mean and white Gaussian noise with unit variance and b was a parameter between 0 and 1 that determined the level of correlation between adjacent samples of the process $x(n)$. Experiments were conducted with b set to 0.00, 0.50, 0.90, and 0.99. When $b = 0$, the input signal is white. As the parameter b approaches 1, the signal characteristics become highly lowpass in nature. The desired response signals were generated by passing the input signals described above through the unknown system and corrupting the output signals with additive zero-mean and Gaussian noise with variance 0.1. The measurement noise sequence and the input signal $x(n)$ were mutually uncorrelated. In all the experiments, μ and β were chosen to be 0.001 and 0.999, respectively. Fig. 1 displays overlaid plots of the squared estimation error signal averaged over the 50 runs. These error curves were further smoothed by time averaging over 10 consecutive samples. It can be seen from the figure that the rate of convergence of the adaptive filter is reasonably close to each other in all cases, in spite of the fairly large disparity in the spectra of the signals employed.

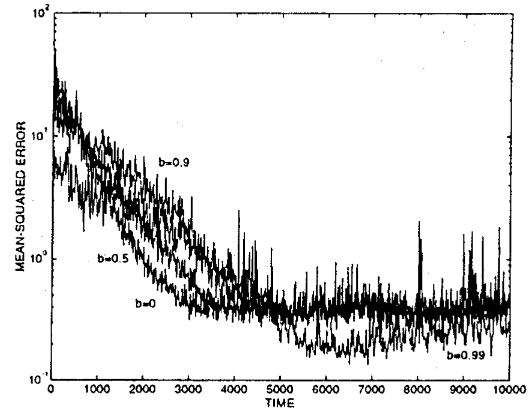

Fig. 1. Mean-squared estimation error of the adaptive lattice Volterra filter for four different input signals.

V. Concluding Remarks

This paper presented an adaptive lattice Volterra filter. The filter is based on a recent result for orthogonalizing Gaussian signals for Volterra system identification problems. The computational complexity of the adaptive filter is comparable to that of the system model when the system memory is much larger than the order of nonlinearity. The lattice filter is also appropriate for independent, identically distributed non-Gaussian input signals. The linear lattice predictor of the first stage is not required in such cases. The results of a limited number of experiments presented indicate that the filter has good convergence characteristics. Further performance evaluations are necessary to understand the properties of the adaptive filter when higher order system models are employed and also when the input signals are not Gaussian distributed.

References

[1] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Englewood Cliffs, NJ: Prentice-Hall, 1996.

[2] T. Koh and E. J. Powers, "An adaptive nonlinear filter with lattice orthogonalization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Boston, MA, Apr. 1983.

[3] ——, "Second-order Volterra filtering and its application to nonlinear system identification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, no. 6, pp. 1445–1455, Dec. 1985.

[4] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Mag.*, vol. 8, no. 3, pp. 10–26, July 1991.

[5] ——, "Orthogonalization of correlated Gaussian signals for Volterra system identification," *IEEE Signal Processing Lett.*, vol. 2, no. 10, pp. 188–190, Oct. 1995.

[6] G. L. Sicuranza and G. Ramponi, "Adaptive nonlinear digital filters using distributed arithmetics," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-34, no. 3, pp. 518–526, June 1986.