Adaptive Polynomial Filters

V. JOHN MATHEWS

While linear filters are useful in a large number of applications and relatively simple from conceptual and implementational view points, there are many practical situations that require nonlinear processing of the signals involved. This article explains adaptive nonlinear filters equipped with polynomial models of nonlinearity. The polynomial systems considered are those nonlinear systems whose output signals can be related to the input signals through a truncated Volterra series expansion, or a recursive nonlinear difference equation. The Volterra series expansion can model a large class of nonlinear systems and is attractive in adaptive filtering applications because the expansion is a linear combination of nonlinear functions of the input signal. The basic ideas behind the development of gradient and recursive least-squares adaptive Volterra filters are first discussed, followed by adaptive algorithms using system models involving recursive nonlinear difference equations. Such systems are attractive because they may be able to approximate many nonlinear systems with great parsimony in the use of coefficients. Also discussed are current research trends and new results and problem areas associated with these nonlinear filters. A lattice structure for polynomial models is also described.

Linear filters have played a very crucial role in the development of various signal processing techniques. The obvious advantage of linear filters is their inherent simplicity. Design, analysis, and implementation of such filters are relatively straightforward tasks in many applications. However, there are several situations in which the performance of linear filters is unacceptable. A simple but highly pervasive type of nonlinearity is the saturation-type nonlinearity. Trying to identify these types of systems using linear models can often give misleading results. Another situation where nonlinear models will do well when linear models will fail miserably is that of trying to relate two signals with nonoverlapping spectral components.

When confronted with a nonlinear systems problem, many engineers shy away from the situation (in the words of Rugh [Ru81], "hoping that the problem will go away") mainly because the solutions are often difficult from an analytical and/or computational point of view. Moreover, the rich variety of highly developed tools available for solving linear systems engineering problems are just not there when it comes to most nonlinear systems problems. The difficulties mentioned above are much more magnified in the case of adaptive nonlinear systems. The purpose of this paper is to give the reader an introduction to adaptive non-

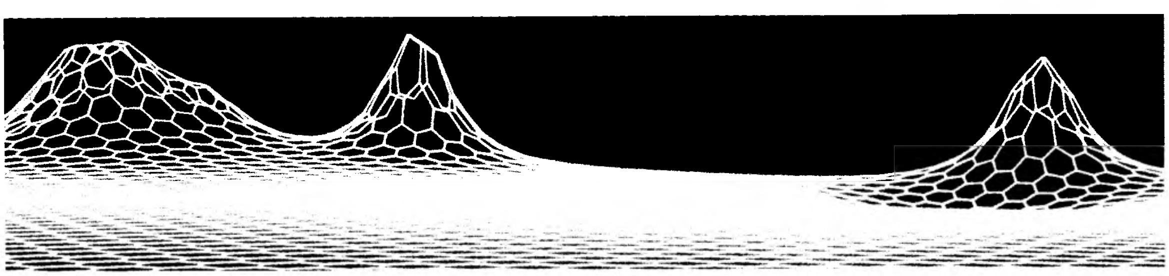

Photo credit: FPG

1053-5888/91/0700-0010$1.00©1991 IEEE

linear systems. Without going into great mathematical detail, this paper will discuss two common models of nonlinearity employed in adaptive filtering applications and some adaptive filter structures that evolve from the use of these models.

System analysis using nonlinear structures has several applications. High-speed communications channels often need nonlinear equalizers for acceptable performance. Although channel equalization using linear, tap delay line structures is adequate in many applications, there are several other situations when they will not work at all. For example, Lucky [Lu75] has conjectured that error probability performance of data transmission systems operating at rates better than 4800 bits/s is due almost entirely to nonlinear distortion.

In telephone transmission, nonlinearities arise principally from inaccuracies in signal companding. In digital satellite links, the satellite amplifiers are usually driven to near the saturation point and they exhibit highly nonlinear characteristics. Several researchers have used Volterra series representation [Sa83a, Sa83b, Sc80, Sc81, Ru81] of nonlinear systems to implement nonlinear channel equalizers [Be83, Be85, Be87, Bi84a, Fa78]. Other applications of nonlinear models and filtering in communication problems include echo cancellation [Ag82, Ca85, Si84, Sm88, Th71], performance analysis of data transmission systems [Be76, Be79, Ki83, Ma85], adaptive noise cancellation [Co80, St85], and detection of nonlinear functions of Gaussian processes [Ke85]. Nonlinear filters are very useful in modeling biological phenomena [Hu86, Ko86, Ma78], myoelectric signal processing [Ja84], characterization of semiconductor devices [Ja77, Na67, Na70, Pr75, Re84], image processing [Ra87, Ts88], modeling drift oscillations in random seas [Ko83b], and several other areas.

Unlike the case of linear systems which are completely characterized by the system's unit impulse response function, it is impossible to find a unified framework for describing arbitrary nonlinear systems. Consequently, the researchers working on nonlinear filters are forced to restrict themselves to certain nonlinear system models that are less general. Nonlinear filters developed using such models include order statistics filters [Bo83b, Le85, No82], homomorphic filters [Op68], morphological filters [Ma87a, Ma87b], and filters based on Volterra and other polynomial descriptions of the nonlinearities involved. Order statistics filters are attractive because of their robustness and computational simplicity. As the name suggests, they are based on the order statistics (i.e., the location of any given data sample in a rearrangement of the samples under consideration in the ascending or descending order of magnitude) of the input signal to the filter. A very widely used order statistic filter is the median filter. Such filters have good edge preserving properties and are very useful in removing additive impulse noise (in general, noise belonging to long-tailed distributions) from the input signals, and have found applications especially in image processing. Homomorphic filters are among the oldest types of nonlinear filters and have applications in image enhancement, seismic signal processing, and removal of multiplicative noise from input signals. Models of human visual systems based on homomorphic filters have been extensively used in image coding applications [St72]. Morphological filters utilize geometric features of the input signals and are employed in applications involving shape recognition, edge detection, and others. A good description of time-invariant nonlinear filters belonging to all of the above classes may be found in [Pi90a].

In this paper, we will concentrate on polynomial models of nonlinearity. Such models are more general than most of the other models that were discussed above. Two specific cases will be considered in some detail — adaptive filters employing truncated Volterra series representation of nonlinear systems and those using recursive nonlinear difference equations to relate the input and output signals of the system. Even though it is possible to treat the truncated Volterra series representation as a special case of the recursive nonlinear system representation and consider a unified framework for polynomial system representations, we will discuss the two cases separately. The Volterra system model is extremely popular in adaptive nonlinear filtering and has developed an identity of its own in the last few years. The theory of adaptive nonlinear filters employing nonlinear feedback models, on the other hand, is very much in its infancy; and while such systems are very attractive from an implementational point of view, there are several problems for which effective solutions have not yet been found. Discussing the two cases separately will enable us to treat such problems in a better manner. Adaptive order statistic filters are available [Pa88, Pi90b], but we will not discuss them here.

VOLTERRA SERIES EXPANSION FOR NONLINEAR SYSTEMS

Let x[n] and y[n] represent the input and output signals, respectively, of a discrete-time and causal nonlinear system. The Volterra series expansion for y[n] using x[n] is given by [Sa83a, Sa83b, Sc80, Sc81, Ru81]

$$y[n] = h_0 + \sum_{m_1=0}^{\infty} h_1[m_1]\, x[n-m_1] \qquad (1)$$

$$+\sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty} h_2[m_1, m_2]\, x[n-m_1]\, x[n-m_2] + \ldots$$

$$+\sum_{m_1=0}^{\infty}\sum_{m_2=0}^{\infty}\cdots\sum_{m_p=0}^{\infty} h_p[m_1, m_2, \ldots, m_p] x[n-m_1] x[n-m_2] \ldots x[n-m_p]$$

$$+ \ldots)$$

In (1), $h_p[m_1, m_2, \ldots, m_p]$ is known as the p-th order Volterra kernel of the system. Without any loss of generality, one can assume that the Volterra kernels are symmetric, i.e., $h_p[m_1, m_2, \ldots, m_p]$ is left unchanged for any of the possible p! permutations of the indices m_1, m_2, ..., m_p. We will not delve deeply into the questions

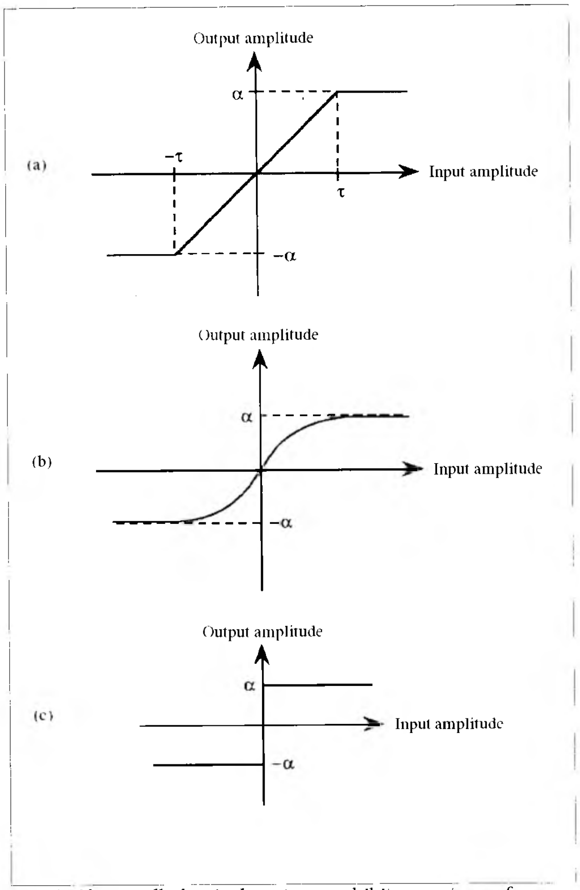

Fig. 1. Almost all physical systems exhibit some type of saturation effects. The figure shows three types of saturation nonlinearities for memoryless systems. A convergent Taylor series expansion exists for all of the real axis only for the nonlinearity depicted in (b). Fortunately, the saturation effects in a wide class of memoryless physical systems can be modeled adequately using the input-output relationship as shown in (b). The limitations and advantages associated with modeling dynamic nonlinear systems using Volterra series expansions in the input signals is similar to those of Taylor series expansions for memoryless nonlinearities.

of convergence and uniqueness of Volterra series expansions of nonlinear systems. The interested reader may refer to [Bo85, Br76, Le78, Sa83c].

One can think of the Volterra series expansion as a Taylor series expansion with memory. The limitations of the Volterra series expansion are similar to those of the Taylor series expansion — both expansions do not do well when there are discontinuities in the system description. As an example, consider a memoryless nonlinear system described by (see Fig. 1)

$$y[n] = A \text{ sign } x[n] \quad (2)$$

where sign{•} denotes the signum function of {•}. There is no convergent Taylor series expansion for the system in (2) about x[n] = 0, and it is straightforward to infer that no convergent Volterra series expansion exists for systems involving this type of nonlinearity. Even though clearly not applicable in all situations, Volterra system models have been successfully employed in a wide variety of applications, and such models continue to be popular with researchers in this area.

Among the early works on nonlinear system analysis is a very important contribution by Wiener [Wi58]. His analysis technique involved white Gaussian input signals and used "G-functionals" to characterize nonlinear system behavior. Following his work, several researchers have employed Volterra series expansion and related representations for estimation and time-invariant nonlinear system identification [Ba63, Ba64, Bo83a, Br70, Ew80, Ey63, Fa80, Ko84a, Ko84b, La81, Th84]. Two recent books [Ru81, Sc80] describe the theory of nonlinear system representation and parameter estimation using Volterra series expansions. The review articles [Bi80, Bi84c, Sc81] also detail some of the work done in (nonadaptive) estimation of nonlinear system parameters using Volterra series representation.

Since an infinite series expansion like (1) is not useful in filtering applications, one must work with truncated Volterra series expansions of the form (see Fig. 2)

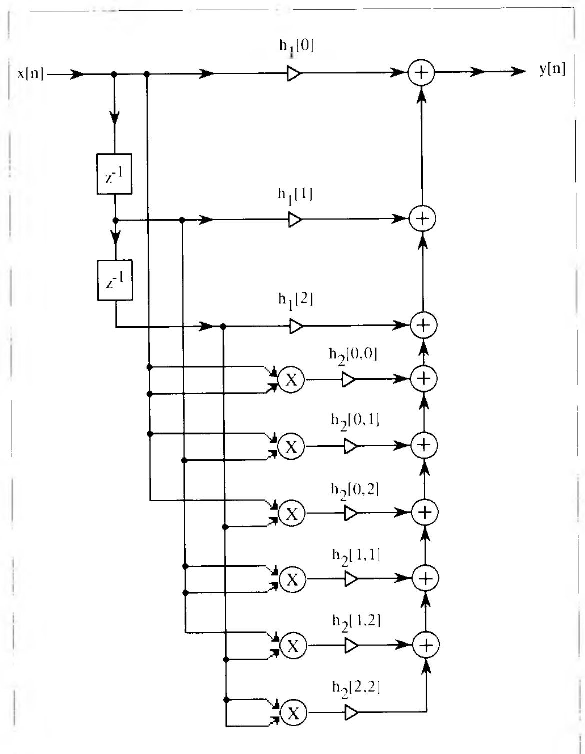

Fig. 2. A truncated Volterra system of order P = 2 and N - 1 = 2 delay elements. Note that this system is linear in the input signal to each coefficient. This fact highly simplifies the design problems involving Volterra series representations. On the other hand, even for moderately large values of N and P, the number of coefficients becomes very large. Consequently, the truncated Volterra series representation is most useful in applications where the values of N and P are relatively small.

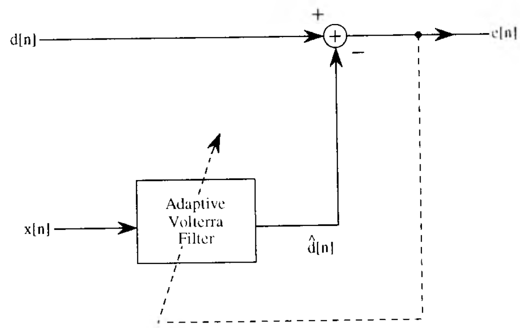

Fig. 3. A block diagram of the adaptive Volterra filter. d[n] is the desired response signal and x[n] is the input to the adaptive filter. $\hat{d}[n]$ is an estimate of d[n] and is computed as a truncated Volterra series expansion in x[n]. The objective, as in most adaptive filtering problems, is to choose the coefficients of the adaptive filter so that an appropriate convex function of the error signal (e[n]) is minimized. The adaptation algorithm depends on the choice of the above cost function. Among the most commonly used algorithms are the least mean square (LMS) algorithm and its variations. Recursive least squares (RLS) algorithms for adaptive Volterra filtering are at least an order of magnitude more complex than LMS-type algorithms.

$$y[n] = \sum_{m_1=0}^{N-1} h_1[m_1]\, x[n-m_1] + \sum_{m_2=0}^{N-1}\sum_{m_1=0}^{N-1} h_2[m_1,m_2]\, x[n-m_1]\, x[n-m_2] + \ldots + \sum_{m_p=0}^{N-1}\ldots\sum_{m_1=0}^{N-1} h_p[m_1, m_2, \ldots, m_p]\, x[n-m_1]\ldots x[n-m_p] \quad (3)$$

(h_0 can often be estimated outside the basic adaptive filter structure. Therefore, we will, without loss of generality, assume that $h_0 = 0$.) Note that there are $O(N^P)$ coefficients in this polynomial expansion (i.e., the number of coefficients is proportional to N^P). One big disadvantage for the Volterra system model as in (3) is that the complexity of implementing filters using this model can be very large even for moderately large values of N and P. Consequently, most of the practical applications of systems employing Volterra series expansions involve low-order models. Later on, we will consider a model that is more parsimonious in the number of coefficients.

ADAPTIVE FILTERS USING TRUNCATED VOLTERRA SERIES EXPANSIONS

Figure 3 shows the block diagram of an adaptive Volterra filter. For simplicity, let us consider a second order (P = 2) Volterra series expansion. The adaptive filter in this case would try to estimate the desired response signal d[n] using a second-order truncated Volterra series expansion in the input signal x[n] as

$$\hat{d}[n] = \sum_{m_1=0}^{N-1} \hat{h}_1[m_1; n]\, x[n-m_1] + \sum_{m_1=0}^{N-1}\sum_{m_2=m_1}^{N-1} \hat{h}_2[m_1, m_2; n]\, x[n-m_1]\, x[n-m_2] \quad (4)$$

$\hat{h}_1[m_1; n]$ and $\hat{h}_2[m_1, m_2; n]$ in (4) are the adaptive filter coefficients that are iteratively updated at each time so as to minimize some convex function of the error signal defined as

$$e[n] = d[n] - \hat{d}[n] \quad (5)$$

What makes the derivation of adaptive Volterra filters relatively straightforward is the fact that the error signal can be written as a linear combination of the input signal to each filter coefficient. (In the case of the second-order Volterra filter, the relevant signals are $x[n], x[n-1], \ldots, x[n-N+1], x^2[n], x[n]x[n-1], \ldots, x[n]x[n-N+1], \ldots, x^2[n-N+1]$.) This fact also makes the theoretical performance analysis of such filters a relatively straightforward extension of the linear filtering case. The LMS adaptive filter [Ha86] updates the coefficients at each time using a steepest descent algorithm that tries to minimize $e^2[n]$ at each time. The update equations for the second order Volterra filter can be easily shown to be [Co80, Ko85]

$$h_1[m_1; n+1] = h_1[m_1; n] - \frac{\mu_1}{2}\frac{\partial e^2[n]}{\partial h_1[m_1; n]} = h_1[m_1; n] + \mu_1 e[n]\, x[n-m_1] \quad (6)$$

and

$$h_2[m_1, m_2; n+1] = h_2[m_1, m_2; n] - \frac{\mu_2}{2}\frac{\partial e^2[n]}{\partial h_2[m_1, m_2; n]} = h_2[m_1, m_2; n] + \mu_2 e[n]\, x[n-m_1]\, x[n-m_2] \quad (7)$$

where μ_1 and μ_2 are small positive constants that control the speed of convergence and the steady-state/tracking properties of the filter. For more general cases, similar update equations can be easily derived. Several variations of the LMS algorithm are also available. Adaptive Volterra filers with time-varying convergence parameters are presented in [Si87]. The adaptation algorithm employed in these filters is a variation of the "sign algorithm" [Cl81, Ma87c] which is simpler to implement than the LMS algorithm. Adaptive Volterra filters based on distributed arithmetic implementation are presented in [Si86, Sm88]. A gradient adaptive quadratic filtering (only second-order coefficients are used here) algorithm employing an LU decomposition of the quadratic coefficient matrix is discussed in [Lo88]. This paper also discusses VLSI implementations of adaptive Volterra filters.

For notational simplicity as well as ease of performance analysis, it is usual to rewrite the adaptive

TABLE I
THE LMS SECOND-ORDER VOLTERRA FILTER

Coefficient Vector $H[n]=\left[h_1[0;n],h_1[1;n],\ldots,h_1[N-1;n],h_2[0,0;n],h_2[0,1;n],\ldots,h_2[0,N-1;n],h_2[1,1;n],\ldots,h_2[N-1,N-1;n]\right]^T$ Input Vector $X[n]=\left[x[n],x[n-1],\ldots,x[n-N+1],x^2[n],x[n],x[n-1],\ldots,x[n],x[n-N+1],x^2[n-1],\ldots,x^2[n-N+1]\right]^T$ Initialization $H[0]$ can be arbitrarily chosen. Algorithm $e[n]=d[n]-H^T[n]\,X[n]$ $H[n+1]=H[n]+\mu X[n]\,e[n]$ Note: $(\,)^T$ denotes matrix transpose. μ is a diagonal matrix with μ_1 appearing in the first N diagonal entries and μ_2 appearing in the rest of the diagonal entries.

filtering algorithm using vector notations. The relevant equations are shown in Table I. Note that the structure of the adaptive filter is different from that of the linear case only in the way in which the vectors are defined. It is relatively straightforward under some simplifying assumptions to show that the mean values of the coefficients converge (for stationary environments) to their optimal values if the convergence constant is chosen such that $0<\mu_1,\mu_2<2/\lambda_{max}$, where λ_{max} is the maximum eigenvalue of the autocorrelation matrix of the input vector X[n]. The problem, as is for the linear case, is that the eigenvalues of the autocorrelation matrix control the speed of convergence. In general, the larger the eigenvalue spread (the ratio of the maximum and minimum eigenvalues), the slower is the convergence speed. This is particularly troublesome in the nonlinear filtering case, since the eigenvalue spreads are in general very large. Even when the input signal is white, the presence of the nonlinear entries in the input vector will cause the eigenvalue spread to be more than one. Consequently, it is important to seek alternate algorithms and structures that have convergence behaviors that are independent of or less dependent on the statistics of the input signal. One approach is to use recursive least squares (RLS) algorithms in place of the LMS adaptive filter. Another alternative is to use lattice (or other orthogonalized) structures to implement the nonlinear filters. We will very briefly discuss the ideas behind RLS adaptive Volterra filters next.

The LMS adaptive filter can be considered as an approximate solution to the statistical optimization problem that tries to minimize the mean squared value of the estimation error at each time. RLS adaptive filters, on the other hand, yield the exact solution to an optimization problem formulated in a deterministic fashion. One such formulation gives rise to the exponentially weighted RLS adaptive filter and in the case of the second-order Volterra filter, such adaptive systems minimize the following cost function at each time

$$J[n]=\sum_{k=0}^{n}\lambda^{n-k}\left(d[k]-H^T[n]\,X[k]\right)^2 \tag{8}$$

where H[n] and X[n] are the coefficient and input signal vectors, respectively, as defined in Table I and λ $(0<<\lambda\leq 1)$ is a factor that controls the memory span of the adaptive filter. The solution to this problem at each time can be easily found by differentiating J[n] with respect to H[n], setting the derivative to zero, and solving for H[n]. The optimal solution at time n is given by

$$H[n]=C^{-1}[n]\,P[n] \tag{9}$$

where

$$C[n]=\sum_{k=0}^{n}\lambda^{n-k}X[k]\,X^T[k] \tag{10}$$

and

$$P[n]=\sum_{k=0}^{n}\lambda^{n-k}d[k]\,X[k] \tag{11}$$

H[n] can be recursively updated by realizing that

$$C[n]=\lambda\,C[n-1]+X[n]\,X^T[n] \tag{12}$$

and

TABLE II
THE RLS ADAPTIVE SECOND-ORDER VOLTERRA FILTER

Coefficient Vector $H[n]=\left[h_1[0;n],h_1[1;n],\ldots,h_1[N-1;n],h_2[0,0;n],h_2[0,1;n],\ldots,h_2[0,N-1;n],h_2[1,1;n],\ldots,h_2[N-1,N-1;n]\right]^T$ Input Vector $X[n]=\left[x[n],x[n-1],\ldots,x[n-N+1],x^2[n],x[n],x[n-1],\ldots,x[n],x[n-N+1],x^2[n-1],\ldots,x^2[n-N+1]\right]^T$ Initialization $H[0]=[0,0,\ldots,0]^T$ $C^{-1}[0]=\delta^{-1}I$ δ = a small positive constant Algorithm $k[n]=\dfrac{\lambda^{-1}C^{-1}[n-1]\,X[n]}{1+\lambda^{-1}X^T[n]\,C^{-1}[n-1]\,X[n]}$ $\varepsilon[n]=d[n]-H^T[n-1]\,X[n]$ $H[n]=H[n-1]+\mu k[n]\,\varepsilon[n]$ $C^{-1}[n]=\lambda^{-1}C^{-1}[n-1]-\lambda^{-1}k[n]\,X^T[n]\,C^{-1}[n-1]$ $e[n]=d[n]-H^T[n]\,X[n]$

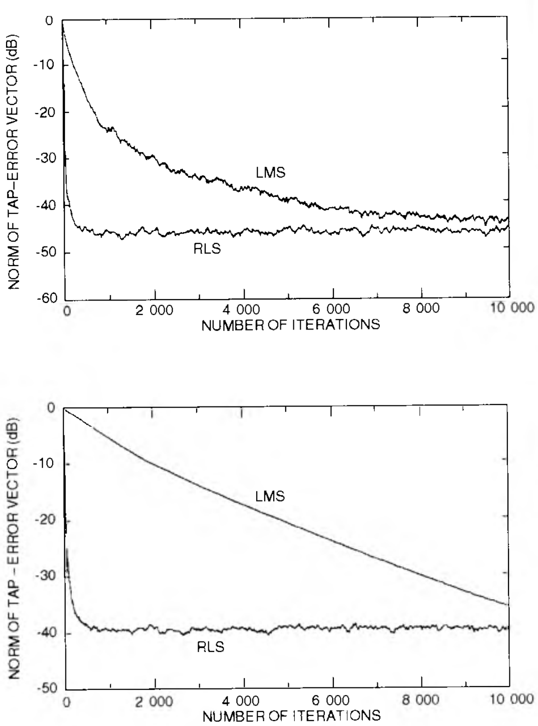

Fig. 4. The two curves in each figure compare the speed of convergence of the RLS and LMS adaptive Volterra filters. The performance measure is defined in the text. The plots on the top compare the performance of the linear coefficients and those at the bottom compare the performance of the quadratic coefficients of the adaptive filters. The parameters μ and λ of the filters were selected such that the curves eventually meet (i.e., the steady-state performances of the two systems are similar). The superior convergence behavior of the RLS algorithm in this example is obvious. However, this improved performance comes at the cost of a substantial increase in the computational complexity.

$$P[n] = \lambda P[n-1] + d[n] X[n] \qquad (13)$$

One can simplify the computational complexity a little bit by making use of the matrix inversion lemma for inverting C[n]. This will result in the algorithm given in Table II. The derivation is similar to that for the RLS linear adaptive filter given in [Ha86, Chapter 8].

We now present the results of an experiment that compares the performance of the two algorithms. In the experiment, both LMS and RLS algorithms were used to identify an unknown, time-invariant second-order Volterra system from measurements of its input signal and a noisy version of the output. The memory span of the unknown system was four samples long (i.e., N = 4) and the coefficients were given by

$$\{h_1[0], h_1[1], h_1[2], h_1[3]\} = [-0.78, -1.48, -1.39, 0.04]$$

and

$$\{h_2[0,0], h_2[0,1], h_2[0,2], h_2[0,3], h_2[1,1], h_2[1,2], h_2[1,3], h_2[2,2], h_2[2,3], h_2[3,3]\} = [0.54, 3.72, 1.86, -0.76, -1.62, 0.76, -0.12, 1.41, -1.52, -0.13] \qquad (14)$$

respectively. The input signal x[n] was obtained by processing a zero-mean and Gaussian signal with a linear filter with impulse response sequence given by

$$h_n = \begin{cases} 0.25; & n = 0 \\ 1.0; & n = 1 \\ 0.25; & n = 2 \\ 0.0; & otherwise \end{cases} \qquad (15)$$

The input signal variance was selected so that the power of the corresponding output of the unknown Volterra system was about 1. The desired response signal d[n] was obtained by adding a zero-mean and white Gaussian sequence (that was uncorrelated with x[n]) to the output of the unknown system. The output signal to measurement noise ratio was chosen to be approximately 30 dB. The forgetting factor λ was chosen to be 0.995. The step sizes μ_1 and μ_2 of the LMS filter were chosen so that the steady-state excess mean-squared estimation error of the LMS and RLS algorithms were about the same. Fifty different experiments were conducted using 20,000 data samples each. The data used in each of these experiments were uncorrelated with those used in the other 49 experiments. The results presented in Fig. 4 have been averaged over the 50 independent experiments.

Figure 4 displays a measure of the mean-squared deviations of the adaptive filter coefficients from the coefficients of the unknown system for the first 10,000 time samples. The linear and quadratic coefficients are considered separately. The measures displayed in the figure are defined as

$$\| V_L[n] \| = 10 \log \frac{\sum_{i=0}^{N-1} \left(\hat{h}_1[i; n] - h_1[i]\right)^2}{\sum_{i=0}^{N-1} \left(h_1[i]\right)^2} \qquad (16)$$

and

$$\| V_Q[n] \| = 10 \log \frac{\sum_{i=0}^{N-1} \sum_{j=i}^{N-1} \left(\hat{h}_2[i, j; n] - h_2[i, j]\right)^2}{\sum_{i=0}^{N-1} \sum_{j=i}^{N-1} \left(h_2[i, j]\right)^2} \qquad (17)$$

respectively.

These results demonstrate that the RLS algorithm clearly outperforms the LMS adaptive filter in terms of speed of convergence. The experiments were repeated for white Gaussian input signal as well as white and colored signals generated from a uniformly distributed random process. The results were similar to those shown in Fig. 4. The results of the performance com-

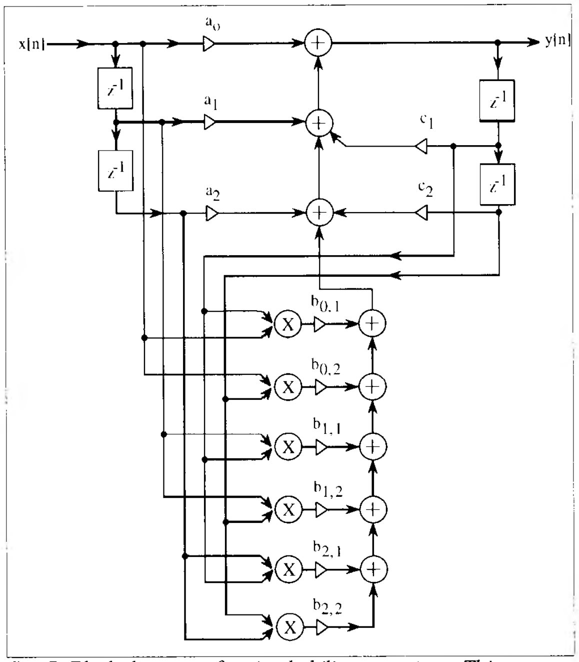

Fig. 5. Block diagram of a simple bilinear system. This system is representative of more general nonlinear systems that are described using recursive nonlinear difference equations. The key advantage of such systems is that it is possible to represent many nonlinear systems with relatively few coefficients when compared with Volterra system representations. The obvious disadvantage of these representations is that we must continuously monitor the adaptive systems using these models for stability. Another disadvantage that is not shared by recursive linear systems is that any noise in the input signals to the adaptive filter will appear in the system model in a multiplicative fashion and this will affect the performance of the adaptive systems.

parison of this example are typical of the behavior of the two systems. This statement is especially true when the signal-to-measurement noise ratio of the input signals is large.

An operations count will show that the LMS algorithm has a computational complexity that is proportional to N^2 ($O(N^2)$) multiplications per time instant, whereas the complexity of the RLS algorithm is $O(N^4)$ multiplications per time instant. The price paid for the better performance in terms of the increased computational complexity is exorbitant in many applications.

Fast algorithms that simplify the computational complexity by a considerable amount can be derived by making use of the fact that most of the elements of the data vectors X[n] and X[n - 1] are the same [Ci84, Lj78]. A particularly easy-to-understand exposition of the ideas involved in the derivation of the fast algorithms for the linear filtering case is given in [Al86]. Such an algorithm requiring $O(N^3)$ multiplications per time instant for second-order Volterra filtering has been developed in [Le91, Ma88]. Since this method is a more efficient realization of the algorithm in Table II, it exhibits better convergence and tracking properties than the LMS Volterra filters. It also seems to be more robust to the statistical variations of the input signals. However, note that the computational complexity is considerably more than the $O(N^2)$ complexity of LMS adaptive filters. A computationally simpler approximate RLS adaptive solution has been developed in [Da87]. However, the approximations assume that the input signal to the adaptive filter is Gaussian, and the system performance breaks down when the input sequence belongs to nonGaussian distributions. A significant problem with the methods in [Da87, Ma88] is the very poor numerical properties exhibited by the "fast" RLS algorithms.

ADAPTIVE FILTERS USING RECURSIVE NONLINEAR DIFFERENCE EQUATIONS

The major problem associated with Volterra series representation of nonlinear systems is that a very large number of coefficients are required to characterize many nonlinear processes. Consequently it is important to search for alternate representations that may be more parsimonious in their use of coefficients. One such model is that in which the input-output relationship is governed by a recursive nonlinear difference equation of the type

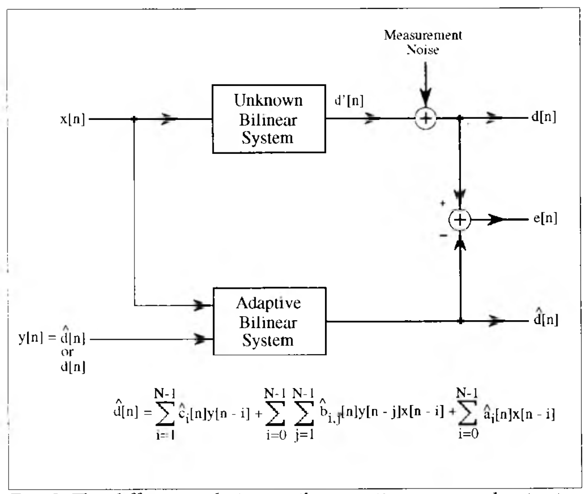

Fig. 6. The differences between the equation-error and output-error approaches of adaptive bilinear filtering is explained in the context of a system identification problem here. Equation-error algorithms use d[n] and x[n] as the inputs to the adaptive system to get the system output $\hat{d}[n]$. Since the statistics of d[n] are in general different from those of the "true" output of the unknown system d '[n], the estimates of the unknown coefficients will be biased in general. The output-error algorithms use past samples of $\hat{d}[n]$ to obtain $\hat{d}[n]$. Since $\hat{d}[n]$ is an estimate of d '[n], the statistics of $\hat{d}[n]$ will hopefully be close to those of d '[n] (at least after adaptation has taken place) and therefore we can expect to get unbiased (or at least close to unbiased) estimates of the coefficients. The relative merits of the two approaches are briefly discussed in the text. More details can be found in [Sh89].

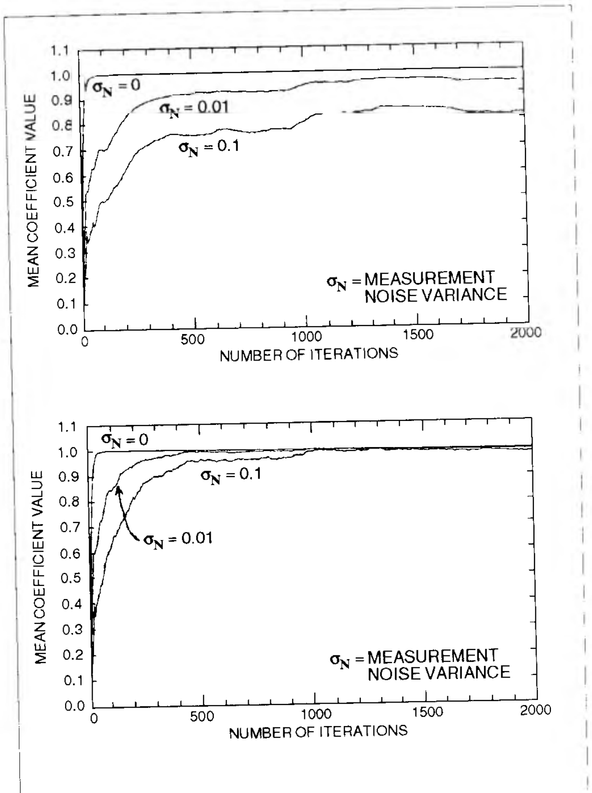

Fig. 7. One disadvantage the equation-error adaptive nonlinear filters share with their linear counterparts is that they will produce biased estimates in the presence of measurement noise (compared to the output-error adaptive algorithms). The plots on the top correspond to the average behavior of one of the coefficients of the adaptive filter under three different noise conditions. Note that they converge to wrong values (the correct value of the coefficient of the unknown system is 1). The plots at the bottom were obtained using an output-error algorithm, and the curves do converge to the correct values in this example. However, it is possible that the error surface of this system has local minima and differently initialized systems can converge to wrong coefficient values.

$$y[n] = \sum_{i=1}^{M} P_i\,(\,y[n-1],\, y[n-2],\, \ldots,\, y[n-N+1],\, x[n],\, x[n-1],\, \ldots,\, x[n-N+1]) \tag{18}$$

where $P_i(\bullet, \bullet, \ldots, \bullet)$ is an i-th order polynomial in the quantities within the parentheses. Just as linear IIR filters can represent many systems with far fewer coefficients than their FIR counterparts, system representations using recursive nonlinear difference equations can model many nonlinear systems with much more parsimony than Volterra series representations [Bi84b, Di88].

Perhaps the simplest of the nonlinear systems in this category is the bilinear system whose input-output relationship is given by

$$y[n] = \sum_{i=1}^{N-1} c_i\, y[n-i] + \sum_{i=0}^{N-1} \sum_{j=1}^{N-1} b_{i,j}\, y[n-j]\, x[n-i] + \sum_{i=0}^{N-1} a_i x[n-i] \tag{19}$$

In spite of the simplicity, this is an important nonlinear model since it can be shown under relatively mild conditions that a large class of nonlinear systems including Volterra systems can be approximated with arbitrary precision using bilinear system models with finite number of coefficients (see [Br76, Mo80] and the references in these papers for details). Furthermore, most of the ideas discussed here on bilinear systems can be easily extended to the more general recursive nonlinear system models. The block diagram of a bilinear system for the case when N = 3 is shown in Fig. 5. Several properties of bilinear time series are discussed in [Su81]. A survey of the applications, control, and identification of bilinear systems can be found in [Mo80]. Another work that extensively discusses the properties of bilinear systems is [Br74].

As for the case of the linear IIR adaptive filters, there are two different approaches to solving adaptive filtering problems using recursive nonlinear system models — equation-error and output-error approaches. The basic ideas behind these two approaches are depicted in Fig. 6. The interested reader may refer to the tutorial article [Sh89] by Shynk on adaptive IIR filters for more details regarding these two approaches.

Equation error algorithms are straightforward to develop, and the mean-squared estimation error surface has a unique minimum. However, this minimum may not be at the correct solution to the problem if there is noise present in the desired response signal. Furthermore, there is no guarantee that the adaptive filter solutions will be stable at all times (including at convergence). The basic idea is to simply use samples of the input signal x[n] and the desired response signal d[n] to obtain the adaptive filtering estimate as

$$\hat{d}\,[n] = \sum_{i=1}^{N-1} \hat{c}_i[n]\, d[n-i] + \sum_{i=o}^{N-1} \sum_{j=1}^{N-1} \hat{b}_{i,j}[n]\, d[n-j]\, x[n-i] + \sum_{i=0}^{N-1} \hat{a}_i[n]\, x[n-i] \tag{20}$$

where $\hat{c}_i[n]$, $\hat{b}_{i,j}[n]$, and $\hat{a}_i[n]$ are the adaptive filter coefficients at time n. The adaptive filter coefficients can be updated using a gradient algorithm or an RLS solution or some other appropriate technique. The gradient update equations (which can be derived as in (6) and (7)) for minimizing the mean-squared estimation error $E\{(d[n] - \hat{d}[n])^2\}$ are

$$\hat{c}_i[n+1] = \hat{c}_i[n] + \mu_c\, d\,[n-i]\; e\,[n] \tag{21}$$

$$\hat{b}_{i,j}[n+1] + \hat{b}_{i,j}[n] + \mu_b\, d\,[n-j]\; x[n-i]\; e\,[n] \tag{22}$$

and

$$\hat{a}_i[n+1] = \hat{a}_i[n] + \mu_a\, x\,[n-i]\; e\,[n] \tag{23}$$

where

$$e[n] = d[n] - \hat{d}[n] \tag{24}$$

is the estimation error at time n and μ_a, μ_b, and μ_c are constants that control the rate at which the adaptive filter converges. Note that if d[n] contains noise, the statistics of the input signal to the adaptive filter will be biased from the statistics of the "ideal" desired response signal and this will result in biased estimates. Actually, the presence of additive measurement noise in the input signals to the adaptive filter considerably complicates the problem when compared with the linear IIR filtering problem. Because of the existence of the product terms x[n - i]d[n - j] in the computation of the output of the adaptive bilinear system (see equation (20)), there will be multiplicative noise components present at the output. This situation is quite different from that of the linear IIR filtering problem. In spite of this, the above approach is attractive because of its simplicity and is very useful in low noise environments.

As explained in Fig. 6, the output-error methods feed back the output of the adaptive system to estimate the current sample of the desired response signal of the adaptive filter. Many of the gradient adaptive output-error algorithms described in [Sh89] can be extended to the nonlinear estimation problem. While active research is currently going on to understand the properties of such systems using empirical and theoretical analyses, no published results are available on these types of adaptive nonlinear filters. Perhaps the simplest among the various methods available in the literature is the suboptimal least-squares method (this method has also been referred to as the extended least-squares algorithm in [Fn87]) presented by Billings and Voon [Bi84b]. Moore [Mo82] has established convergence results for this algorithm when applied to linear estimation problems and this analysis seems to carry over to the nonlinear case also. The easiest approach to explaining the suboptimal least squares algorithm may be to use vector notation. Let

$$H[n] = \Big(\hat{c}_1[n], \hat{c}_2[n], \dots, \hat{c}_{N-1}[n], \hat{b}_{0,1}[n], \dots, \hat{b}_{N-1,N-1}[n], \hat{a}_0[n], \dots, \hat{a}_{N-1}[n]\Big)^T \tag{25}$$

and

$$X[n] = \Big(\hat{d}_{n-1}[n-1], \hat{d}_{n-2}[n-2], \dots, \hat{d}_{n-N+1}[n-N+1], x[n]\,\hat{d}_{n-1}[n-1], \dots, x[n-N+1]\,\hat{d}_{n-N+1}[n-N+1], x[n], \dots, x[n-N+1]\Big)^T \tag{26}$$

denote the adaptive filter coefficient vector and input vector to the adaptive filter, respectively. Here $\hat{d}_k[l]$ denotes the estimate of the desired response signal at time l made using the adaptive filter coefficients at time k. Then, the adaptive filter coefficient at time n is obtained as the solution that minimizes

$$J[n] = \sum_{k=0}^{n} \lambda^{n-k}\Big(d[k] - H^T[n]\,X[k]\Big) \tag{27}$$

The adaptive filter output at time n is given by

$$\hat{d}_n[n] = H^T[n]\,X[n] \tag{28}$$

and H[n] is estimated as

$$H[n] = C^{-1}[n]\,P[n] \tag{29}$$

where C[n] and P[n] are as defined in equations (10) and (11), and X[n] is as defined in equation (26). As discussed before, one can make use of the matrix inversion lemma to obtain a more computationally efficient solution to the problem.

Even though the formulation and the above solution of our problem look very similar to the RLS Volterra filtering problem we discussed earlier, equations (28) and (29) do not represent an exact least-squares solution in the following sense. The exact least-squares minimization problem in equation (8) requires that the cost function J[n] is defined using estimation error values

$$e_n[k] = d[k] - H^T[n]\,X[k] \tag{30}$$

computed at time k using the solution H[n] to be obtained at the current time. Thus the problem is formulated as if we were finding an entirely different solution at each time (even though it is possible to update the coefficients on the basis of the previous solutions). In the problem described in equations (25)-(29), $\hat{d}_{k-1}[k-1]$, $\hat{d}_{k-2}[k-2]$, ..., $\hat{d}_{k-N+1}[k-N+1]$ that appears in the input vector X[k] are computed at times k - 1, k - 2, ..., k - N + 1, respectively. Consequently, the coefficient solution at time n does depend on the previous solutions (at least implicitly), and the solution is not an exact least-squares solution. Even though clearly suboptimal in the above sense, experimental results presented in [Bi84b, Fn87] seem to indicate that this method performs very well. [Bi84b] also discusses two other somewhat more complicated algorithms for output-error adaptive filters for nonlinear systems described by recursive nonlinear difference equations. Several variations of the ideas discussed above have been presented in [Da89, Fn87, Ga89].

The advantage of output-error algorithms over equation-error algorithms is obvious — the former may be less sensitive to additive noise components present in the desired response signal than the latter. However, the error surface may have local minima (see [Sh89] for illustrations of this idea) and the adaptive filter may not converge to the global minimum, unless the system is initialized properly. Research aimed at getting a better understanding of the properties of and designing better and more efficient output-error adaptive nonlinear filters is currently going on.

We will now present a simulation example demonstrating some of the ideas discussed above. Once again, we will consider a problem involving identification of an unknown system. The system to be identified is bilinear and has input-output relationship given by

$$y[n] = ay[n-1] + by[n-2]\,x[n-1] + cx[n-1] \qquad (31)$$

where a = 1, b = -0.7, and c = 0.5. This system is the same as the one used in [Fn87]. The input signal to the adaptive filter x[n] was a white and zero-mean Gaussian sequence with variance 0.05. (The reason for selecting

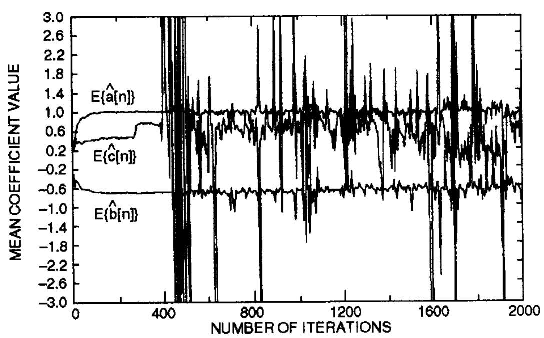

Fig. 8. The problems with the stability of recursive nonlinear systems are substantially larger than those associated with recursive linear systems. Often, the notion of stability of such systems is input signal dependent. It is possible to drive many such systems to instability by simply magnifying their input signal. (This can never happen with linear systems.) One such example is illustrated in this figure. The plots correspond to the behavior of the output-error filter when the input signal is an amplified version of that used to obtain Fig. 7b. One explanation for the erratic behavior of the coefficients is that the underlying unknown system is unstable for the present input and therefore the dynamics of the output signal is extremely large. Consequently the adaptive filter has a very hard time tracking the coefficients properly.

this low number for the input signal variance will become clear a little later.) The desired response signal was obtained by corrupting the output of the unknown system with additive white noise that is uncorrelated with the input signal. The adaptive filter was run using the same model as described by equation (31) (i.e., the only unknown quantities are a, b, and c). The results presented are ensemble averages over fifty independent experiments.

Figures 7a and b display the average behavior of the adaptive filter coefficient corresponding to the unknown parameter 'a' in equation (31) obtained using the equation-error and output-error methods, respectively, for different measurement noise levels. Both algorithms used $\lambda = 0.995$. All the coefficients were initialized to zero. Notice that the equation-error algorithm converges to the wrong solution for high noise level in accordance with our earlier discussion. The output-error algorithm, on the other hand, converges to the correct solution for all noise levels in this experiment. While it may be possible to artificially create situations when the output-error filter will converge to some local minimum, the algorithm has exhibited very good behavior in all our experiments. However, much work needs to be done before we can claim a complete understanding of the properties of such filters.

Just as adaptive IIR (linear) filters have many problems that are not shared by their FIR counterparts, adaptive nonlinear systems using recursive nonlinear difference equations also have many problems that are not shared by adaptive Volterra filters. The most important among them is the fact that such algorithms should either be guaranteed to be stable all the time or they should be monitored at all times for stability, and if found to be unstable at any time, steps must be taken to modify the coefficients such that the resulting filter is stable.

The problems associated with stability are much larger in the case of nonlinear systems than for linear systems. To see this, consider the bilinear system used in the experiments. Suppose for the time being that b = 0. Then, the system

$$y[n] = y[n-1] + 0.5\,x[n-1] \qquad (32)$$

is only marginally stable. In fact, when b is nonzero, one would expect that there would be a very large class of input signals that would make the system unstable. This statement is true in general of nonlinear feedback systems. One can almost always find bounded input signals that would drive the system to instability. (The notion of input-dependent stability may offend many purists. Even though most feedback nonlinear systems are unstable in the general sense, we can often define classes of input signals for which such systems will provide useful outputs and/or model signals and other real-world systems with good accuracy. Consequently, it is not at all unusual to talk about input-dependent stability in the context of recursive nonlinear systems.)

The above problem causes great difficulty in the design and analysis of adaptive feedback nonlinear systems. In order to illustrate this problem further, consider the experimental set up described earlier, with the difference that the input signal variance is 1.0 instead of 0.05. Figure 8 displays the average behavior of the three coefficients of the adaptive filter. Note that the coefficient behavior has become very erratic and this is caused at least in part by the fact that the underlying unknown system as well as the system that the adaptive filter has identified is unstable for the given input signal.

Most of the techniques that are currently available cannot adequately handle this problem without human intervention. One exception is the work by Fnaiech and Ljung [Fn87] which discusses several variations of the ideas presented in this section. In their work, they stabilize the filter by means of a time-varying Kalman filter. With the help of a theorem in [Ja70], they have argued that such a system will always result in a stable nonlinear system. They have also demonstrated the validity of the claim by means of simulation examples. However, the details are beyond the scope of this paper.

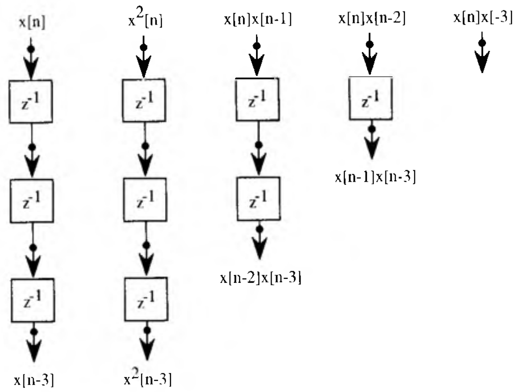

Fig. 9. The adaptive Volterra filtering problem can be easily translated into an adaptive, multichannel, linear filtering problem. In the example shown here with P = 2 and N = 4, one can visualize having five channels as shown. The signals are tapped from the input points as well as from the outputs of the delay elements and linearly combined to form the estimate of the desired response signal. What makes this "multichannel" problem somewhat different and perhaps a little difficult when compared with traditional multichannel adaptive filters is the fact that the number of delay elements in each channel is different from those in others. However, this problem can be overcome, and this structure is the basis for fast RLS Volterra filters and certain lattice realizations of the adaptive Volterra filter.

ADAPTIVE LATTICE POLYNOMIAL FILTERS

Adaptive lattice filters try to orthogonalize the input signals to the filter and then estimate the desired response signal as a linear combination of the transformed signals that are hopefully orthogonal to each other. The advantages of lattice filters in adaptive filtering applications are several. Lattice filters equipped with LMS-type adaptation algorithms tend to show faster and less input signal-dependent convergence behavior than their direct form counterparts. They also tend to have better numerical properties than direct form adaptive filters. It turns out that adaptation of the filter parameters in each lattice stage can be done independently of the rest of the stages. Also, the structure is fairly modular, and therefore adaptive lattice filters are very suitable for VLSI implementation.

In this section, we will develop a lattice structure for a second-order truncated Volterra system. The ideas developed will be equally applicable to other types of polynomial systems also. In order to develop the lattice parameterization of Volterra filters, it is convenient to visualize the nonlinear filtering problem as a linear multichannel filtering problem. This characterization is depicted in Fig. 9 for the second-order Volterra filter. The multichannel characterization is somewhat different from traditional multichannel adaptive filtering problems in the sense that each of the different channels uses a different number of delay elements (and coefficients) when compared with the rest of the channels. To overcome this difficulty, many lattice realizations of Volterra filters [Le86] use additional coefficients and delay elements in each "channel" to make the number of coefficients the same for every "channel." (This actually corresponds to special shapes for the region of support of the Volterra kernels.) Adaptive lattice Volterra filters that are designed specifically for Gaussian input signals and work well only for such signals have been presented in [Ko83a]. However, there are lattice structures (that are designed independently of the statistics of the input signals) available for truncated Volterra systems as given in equation (3). We will now discuss one such structure that is based on a multichannel lattice filter developed by Ling and Proakis [Li84] and a nonlinear lattice predictor developed by Zarzycki [Za85].

For simplicity, we will consider the case when N = 3 and P = 2. A block diagram of the nonlinear lattice predictor is shown in Fig. 10. Let us group the signals involved in the estimation at time n into three columns as shown below.

$$\begin{pmatrix} x[n] & x[n-1] & x[n-2] \\ x^2[n] & x^2[n-1] & x^2[n-2] \\ & x[n]\,x[n-1] & x[n-1]\,x[n-2] \\ & & x[n]\,x[n-2] \end{pmatrix} \tag{33}$$

$$\begin{matrix} \Uparrow & \Uparrow & \Uparrow \\ \textit{Column } 0 & \textit{Column } 1 & \textit{Column } 2 \\ =\underline{x}_0^b[n] & =\underline{x}_1^b[n] & =\underline{x}_2^b[n] \end{matrix}$$

The basic idea employed in the derivation of the lattice Volterra filter is to obtain a Gram-Schmidt orthogonal decomposition of $\underline{x}_0^b[n], \underline{x}_1^b[n]$, and $\underline{x}_2^b[n]$. (All lattice filters try to obtain Gram-Schmidt orthogonalization of appropriate input vectors.) Let $\underline{b}_0[n]$, $\underline{b}_1[n]$, and $\underline{b}_2[n]$ represent an orthogonal basis set for $\underline{x}_0^b[n], \underline{x}_1^b[n]$, and $\underline{x}_2^b[n]$. Then, any linear combination of the elements of $\underline{x}_0^b[n], \underline{x}_1^b[n]$, and $\underline{x}_2^b[n]$ can be equivalently written as another linear combination of the elements of $\underline{b}_0[n]$, $\underline{b}_1[n]$, and $\underline{b}_2[n]$, and vice versa. (In other words, the linear spans of the elements of both the sets of vectors are exactly the same.) What this means is that instead of estimating the desired response signal d(n) as a linear combination of the elements of $\underline{x}_0^b[n], \underline{x}_1^b[n]$, and $\underline{x}_2^b[n]$, we can compute the estimate as a linear combination of the elements of $\underline{b}_0[n]$, $\underline{b}_1[n]$, and $\underline{b}_2[n]$. Let

$$\hat{d}[n] = \left(\underline{k}_0^d\right)^T \underline{b}_0[n] + \left(\underline{k}_1^d\right)^T \underline{b}_1[n] + \left(\underline{k}_2^d\right)^T \underline{b}_2[n] \tag{34}$$

be the best estimate so obtained, where $\underline{k}_0^d, \underline{k}_1^d$, and $\underline{k}_2^d$ are appropriate coefficient vectors from which the possible time dependence has been suppressed. One of the biggest advantages of the lattice structure is that since $\underline{b}_0[n]$, $\underline{b}_1[n]$, and $\underline{b}_2[n]$ are orthogonal to each other, the coefficient vector $\underline{k}_i^d$ can be computed solely from the joint statistics of d[n] and $\underline{b}_i[n]$. For example, the min-

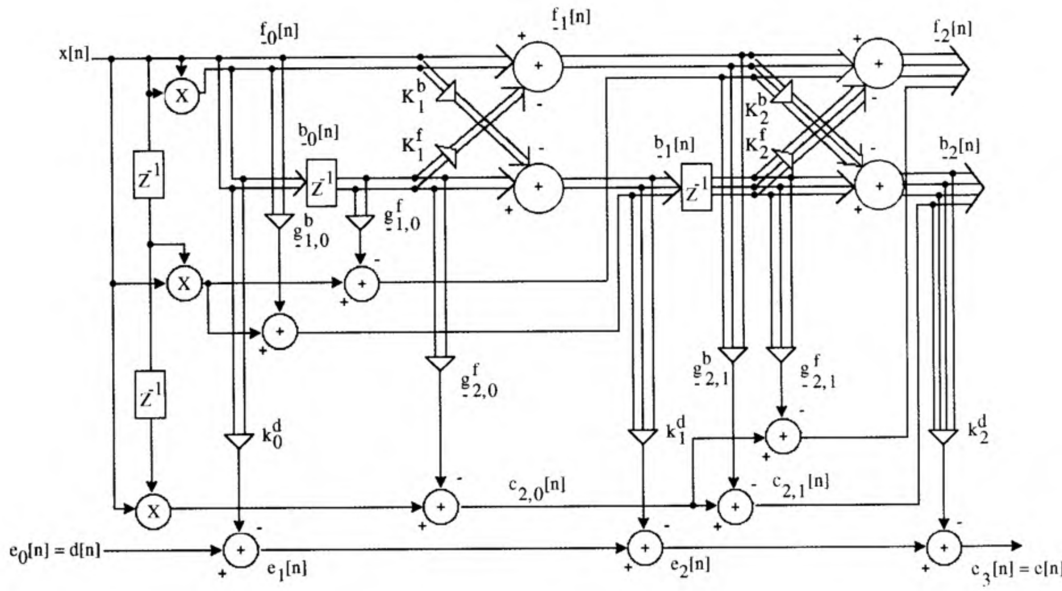

Fig. 10. Block diagram of a lattice filter structure for Volterra systems with N = 3 and P = 2. The number of lines going into and out of a system component indicates the number of input and output signals, respectively, of that component. The backward prediction error vectors $\underline{b}_0[n]$, $\underline{b}_1[n]$, and $\underline{b}_2[n]$ are orthogonal to each other, and the components of these vectors span the whole space spanned by the elements of $X[n]=(x[n],x^2[n],x[n-1],x^2[n-1],x[n]x[n-1],x[n-2], x^2[n-2],x[n-1]x[n-2],x[n]x[n-2])^T$. (Note that the elements of each of these vectors are not orthogonal to each other. This can be achieved by doing a Gram-Schmidt orthogonalization of the elements of each vector.) At each stage of the lattice, the prediction error vector has one more element than the previous stage. This prediction error signal (that corresponds to estimating x(n)x(n - i) for the i-th stage) must be computed outside the basic lattice structure. The coefficients denoted using the letter g are used to compute these additional prediction-error signals. Efficient computation of the backward prediction-error vectors requires computation of the forward prediction-error vectors $\underline{f}_0[n]$, $\underline{f}_1[n]$, and $\underline{f}_2[n]$ also. (See text for details.) For joint process estimation (for estimating a different signal d(n) using elements of X[n]), we need only to find the appropriate linear combination of the components of the backward prediction-error vectors. Development of gradient and least-squares adaptive algorithms based on this lattice structure is now relatively straightforward.

imum mean-squared solution for k_0^d is given by

$$k_0^d = E\{\underline{b}_0[n]\,\underline{b}_0^T[n]\}^{-1}\,E\{d[n]\,\underline{b}_0[n]\} \tag{35}$$

and does not depend on $\underline{b}_1$[n] or $\underline{b}_2$[n].

It is well known that one way of obtaining $\underline{b}_0$[n], $\underline{b}_1$[n], and $\underline{b}_2$[n] is to define $\underline{b}_i$[n] as the i-th order backward prediction error vector for $\underline{x}_i^b[n]$. $\underline{b}_i$[n] would then be the estimation error when $\underline{x}_i^b[n]$ is estimated using the previous column vectors of (33). $\underline{b}_0$[n] is defined to be

$$\underline{b}_0[n] = \underline{x}_0^b[n] = \begin{pmatrix} x[n] \\ x^2[n] \end{pmatrix} \tag{36}$$

$\underline{b}_1$[n] and $\underline{b}_2$[n] are defined to be the estimation error vector when column 1 and column 2, respectively, of (33) are estimated using elements of all the previous columns. Given $\underline{b}_0$[n] and $\underline{b}_1$[n], computation of $\underline{b}_2$[n] can be done from knowledge of $\underline{b}_1$[n - 1], the first-order "forward prediction error" vector (to be defined shortly), $\underline{f}_1$[n], and some allied quantities. To see this, note that $\underline{b}_1$[n - 1] is the error vector obtained when we estimate

$$\underline{x}_1^b[n-1] = \begin{pmatrix} x[n-2] \\ x^2[n-2] \\ x[n-1]\,x[n-2] \end{pmatrix} \tag{37}$$

using the elements of the set {x[n - 1], x^2[n - 1]}, i.e., $\underline{x}_0^b[n-1]$. The key point is that $\underline{x}_1^b[n-1]$ is nothing but the top three elements of $\underline{x}_2^b[n]$ (similarly, $\underline{x}_0^b[n-1]$ appears as the top two elements of $\underline{x}_1^b[n]$) and $\underline{b}_2$[n] is the prediction error vector when we estimate $\underline{x}_2^b[n]$ using $\underline{x}_1^b[n]$ and $\underline{x}_0^b[n]$. In $\underline{b}_1$[n - 1], we have all the information about the top three elements of $\underline{x}_2^b[n]$ that we can extract from the top two elements of $\underline{x}_1^b[n]$. Now, the problem is to find out how much additional information is contained in $\underline{x}_0^b[n]$ and x[n] x[n - 1], the last element of $\underline{x}_1^b[n]$. The "new" information is present in that part of

$$\begin{pmatrix} x[n] \\ x^2[n] \\ x[n]\,x[n-1] \end{pmatrix} \quad (38)$$

that is "not related to" or orthogonal to

$$\begin{pmatrix} x[n-1] \\ x^2[n-1] \end{pmatrix} \quad (39)$$

This component is precisely the estimation error obtained when the three-element vector in (38) is estimated using the two-element vector in (39). The estimation error vector that is produced in the process is nothing but the first-order forward prediction error vector. In general, the i-th order forward prediction error vector $f_i[n]$ is defined as the error vector produced when the data vector

$$\begin{pmatrix} x[n] \\ x^2[n] \\ x[n]\,x[n-1] \\ x[n]\,x[n-2] \\ \cdot \\ \cdot \\ \cdot \\ x[n]\,x[n-i] \end{pmatrix} \quad (40)$$

is estimated using all possible linear and quadratic terms formed using the elements of the set {x[n - 1], x[n - 2], ..., x[n - i]}. Let $\bar{b}_2[n]$ represent the top three elements of $\underline{b}_2[n]$. Based on the above discussion, we can express $\bar{b}_2[n]$ as

$$\bar{b}_2[n] = \underline{b}_1[n-1] - K_2^b \underline{f}_1[n] \quad (41)$$

where K_2^b is the appropriate coefficient matrix and the possible time dependence has been suppressed. Similarly, one can show that the top three elements of $\underline{f}_2[n]$ can be evaluated as

$$\bar{f}_2[n] = \underline{f}_1[n] - K_2^f \underline{b}_1[n-1] \quad (42)$$

where the notation is similar to that used in (41).

The last element of $\underline{b}_2[n]$, which is the error in estimating x[n]x[n - 2] using the same five input elements in the first two columns of (33), has to be computed separately. This element can be computed by subtracting a linear combination of all the elements of $\underline{b}_0[n]$ and $\underline{f}_1[n]$ from x[n]x[n - 2] since the components of $\underline{b}_0[n]$ and $\underline{f}_1[n]$ do span the same space spanned by the elements of the first two columns of (33). In general, the last element of $\underline{b}_i[n]$ can be obtained by subtracting from x[n]x[n - i] an appropriate linear combination of the elements of the vectors $\underline{b}_0[n-1]$, $\underline{b}_1[n-1]$, ..., $\underline{b}_{i-2}[n-1]$, and $\underline{f}_{i-1}[n]$. Similarly, the last element of $\underline{f}_i[n]$ can be obtained by subtracting an estimate of x[n]x[n - i] obtained as a linear combination of the elements of $\underline{b}_0[n-1]$, $\underline{b}_1[n-1]$, ..., $\underline{b}_{i-1}[n-1]$ from x[n]x[n - i]. The basic lattice predictor algorithm for a second-order Volterra system with N - 1 delays is given in Table III.

Once the lattice structure has been developed, deriving an adaptive filter based on this structure is not very

TABLE III

LATTICE FILTER STRUCTURE

Structure shown is for a second-order Volterra system with arbitrary N. Possible time dependence of the coefficients has been suppressed.

<u>Initialization</u>

$$\underline{f}_0[n] = \underline{b}_0[n] = \begin{pmatrix} x[n] \\ x^2[n] \end{pmatrix}$$

$$e_0[n] = d[n]$$

$$\underline{b}_1[n] = \begin{pmatrix} \underline{b}_0[n-1] - \left(K_1^b\right)^T \underline{f}_0[n] \\ \text{-----} \\ x[n]\,x[n-1] - \left(\underline{g}_{1,0}^b\right)^T \underline{f}_0[n] \end{pmatrix}$$

$$\underline{f}_1[n] = \begin{pmatrix} \underline{f}_0[n] - \left(K_1^f\right)^T \underline{b}_0[n-1] \\ \text{-----} \\ x[n]\,x[n-1] - \left(\underline{g}_{1,0}^f\right)^T \underline{b}_0[n-1] \end{pmatrix}$$

$$c_{i,0}[n] = x[n]\,x[n-i] - \left(\underline{g}_{i,0}^f\right)^T \underline{b}_0[n-1]; \quad i = 2, 3, \ldots, N-1$$

$$e_1[n] = e_0[n] - \left(k_0^d\right)^T \underline{b}_0[n]$$

<u>Lattice Sections 2 thru N-1</u>

DO FOR i= 2, 3, ..., N-1

<u>Backward Prediction Error Update</u>

$$\underline{b}_i[n] = \begin{pmatrix} \underline{b}_{i-1}[n-1] - \left(K_i^b\right)^T \underline{f}_{i-1}[n] \\ \text{-----} \\ c_{i,i-2}[n] - \left(\underline{g}_{i,i-1}^b\right)^T \underline{f}_{i-1}[n] \end{pmatrix}$$

<u>Forward Prediction Error Update</u>

$$\underline{f}_i[n] = \begin{pmatrix} \underline{f}_{i-1}[n] - \left(K_i^f\right)^T \underline{b}_{i-1}[n-1] \\ \text{-----} \\ c_{i,i-2}[n] - \left(\underline{g}_{i,i-1}^f\right)^T \underline{b}_{i-1}[n-1] \end{pmatrix}$$

<u>Auxiliary Variable Update</u>

$$c_{j,i-1}[n] = c_{j,i-2}[n] - \left(\underline{g}_{j,i-1}^f\right)^T \underline{b}_{i-1}[n-1]; \quad j = i+1, i+2, \ldots, N-1$$

<u>Joint Process Estimation Error Update</u>

$$e_i[n] = e_{i-1}[n] - \left(k_{i-1}^d\right)^T \underline{b}_{i-1}[n]$$

END LOOP

<u>Final Joint Process Estimation Error</u>

$$e[n] = e_N[n] = e_{N-1}[n] - \left(k_{N-1}^d\right)^T \underline{b}_{N-1}[n]$$

Notes: K_i^f and K_i^b are (i+1)× (i+1) matrices.
$\underline{g}_{i,j}^f$ and $\underline{g}_{i,j}^b$ are vectors with (j+1) elements.
$c_{i,j}[n]$ are scalar signals and are used to compute the last elements of $\underline{f}_i[n]$ and $\underline{b}_i[n]$.

difficult. Gradient algorithms like those presented in [Gr77] can be easily extended to the nonlinear case. The key idea employed in LMS-type lattice filters is that the coefficients in each stage can be optimized independently of later stages. This is because of the orthogonality of the relevant signals in different stages when optimal lattice coefficients are used. For example, consider the discussion surrounding equations (34) and (35). It is apparent that each coefficient k_i^d can be evaluated as the optimal coefficient for estimating the desired response signal d[n] as a linear combination of the elements of $\underline{b}_i[n]$. Let

$$e_i[n] = d[n] - \sum_{j=0}^{i-1} \left(k_j^d\right)^T \underline{b}_j[n] \tag{43}$$

Observe that since $\underline{b}_0[n], \ldots, \underline{b}_{i-1}[n]$ are orthogonal to $\underline{b}_i[n]$, $e_i[n]$ contains all the components of d[n] that can be estimated using $\underline{b}_i[n]$. Therefore, if we try to estimate $e_i[n]$ using $\underline{b}_i[n]$, we will get the same result that we would have obtained if we tried to estimate d[n] using $\underline{b}_i[n]$. Thus adaptation of k_i^d can be considered as a separate adaptive filtering problem where $e_i[n]$ is the desired response signal, $\underline{b}_i[n]$ is the input to the adaptive filter, and e_{i+1} is the estimation error. The relevant equations for the LMS-type adaptive filter are

$$e_{i+1}[n] = e_i[n] - \left(k_i^d[n]\right)^T \underline{b}_i[n] \tag{44}$$

and

$$k_i^d[n+1] = k_i^d[n] + \mu\, e_{i+1}[n]\, \underline{b}_i[n] \tag{45}$$

Different values of μ may be used for different stages.

Similar to the above development, the adaptation of the coefficients in each stage of the predictor part of the lattice structure can be done independently of the other stages. The coefficient matrix K_i^b for the backward prediction problem can be updated by realizing that we can view this as a separate adaptive filtering problem with $\underline{f}_{i-1}[n]$ as the input signal, $\underline{b}_{i-1}[n-1]$ as the desired response signal, and $\overline{b}_i[n]$ as the error signal. The corresponding adaptation algorithm is

$$\overline{b}_i[n] = \underline{b}_{i-1}[n-1] - \left(K_i^b[n]\right)^T \underline{f}_{i-1}[n] \tag{46}$$

and

$$K_i^b[n+1] = K_i^b[n] + \mu\, \underline{f}_{i-1}[n]\, \overline{b}_i^T[n] \tag{47}$$

Again, μ can be different for each stage or even time-varying. One may also use a matrix in place of the scalar quantity.

The update equations for the coefficients of the forward predictor sections and the auxiliary quantities associated with computation of the last elements of $\underline{f}_i[n]$ and $\underline{b}_i[n]$ can be derived in a similar fashion. The derivation of these equations, which is straightforward and omitted here, will complete the development of the LMS adaptive Volterra lattice filter.

One of the disadvantages of the lattice structure when compared with the direct form structure is the fact that it requires $O(N^3)$ coefficients to completely describe a second-order Volterra system with N delays while the direct form structure needs only $O(N^2)$ coefficients. Therefore, the computational complexity of the gradient adaptive algorithms based on the lattice structure will also be proportional to N^3 operations per instant. This complexity is comparable to those of fast RLS algorithms (even though it will still be lower than most RLS adaptive Volterra filters), and consequently the computational advantage the gradient adaptive lattice Volterra filters enjoy over the RLS adaptive Volterra filters is not as significant as in the case of direct-form implementations.

Least-squares adaptive lattice Volterra filters with $O(N^3)$ computational complexity and extremely good numerical properties have recently been developed [Sy90]. However, an exposition of the ideas employed in the derivation of such algorithms is beyond the scope of this introductory paper. The interested reader is referred to [Sy90] for details.

Algorithms for adaptive least-squares lattice bilinear filters have also been developed [Ba90a, Ba90b]. Another related work is [Pa81]. Korenberg has developed algorithms using Gram-Schmidt orthogonalization of the input data and that can be applied to the general class of polynomial system models [Ko88]. For the Volterra and bilinear system models, the lattice filter structure discussed in this paper turns out to be quite a bit more efficient than Korenberg's approach.

CONCLUDING REMARKS

This tutorial article presented an introduction to adaptive nonlinear filtering theory. The emphasis in the first part of the paper was on system models using truncated Volterra series expansions and adaptive filters based on such models. Also presented was a brief introduction to adaptive filtering using recursive nonlinear system models. This paper also described the basics of a lattice nonlinear filter structure. Obviously, there is no one general theory of nonlinear system analysis and we had to restrict ourselves to just these two nonlinear models. Consequently, adaptive nonlinear filters based on other nonlinear models were not discussed in the paper. Adaptive nonlinear filtering is an exciting and challenging area with a wide variety of applications. While quite some progress has been made in recent years, much needs to be still done. There is a fairly large amount of research activity going on in this area at present and we can expect to see a substantial number of new techniques being developed, and potential breakthroughs with great impact on practical applications occurring in the near future.

Acknowledgments

This work was supported in part by NSF under Grant Nos. MIP-8708970 and MIP-8922146 and a University of Utah Faculty Fellow Award. I have learned quite a lot from my students at the University of Utah. In particular, I would like to acknowledge very useful interactions with Heung Ki Baik, Junghsi Lee, and Mushtaq Syed as well as their contributions to the development of many of the ideas discussed in the paper. Junghsi Lee provided the results of all the experiments described in the paper.

V. John Mathews (S '82, M '85, SM '90) was born in Nedungadappally, Kerala, India in 1958. He received his B.E. (Hons.) degree in electronics and communication engineering from the University of Madras, India, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Iowa, Iowa City, in 1980, 1981, and 1984, respectively. From 1980 to 1984 he held a Teaching-Research Fellowship at the University of Iowa, where he also worked as a Visiting Assistant Professor with the Department of Electrical and Computer Engineering from 1984 to 1985. He is currently an Assistant Professor with the Department of Electrical Engineering, University of Utah, Salt Lake City. His research interests include adaptive filtering, spectrum estimation, and data compression. Dr. Mathews is an associate editor of the *IEEE Transactions on Signal Processing*.

REFERENCES

[Ag82]Agazzi, O., D. G. Messerschmitt and D. A. Hodges, "Nonlinear echo cancellation of data signals," *IEEE Trans. Communications*, Vol. COM-30, pp. 2421-2433, November 1982.

[Al86]Alexander, S. T., "Fast adaptive filters: A geometrical approach," *IEEE ASSP Magazine*, Vol. 3, No. 4, pp. 18-28, October 1986.

[Ba63]Barret, J. F., "The use of functionals in the analysis of nonlinear system," *Journal of Electronics, Controls*, Vol. 15, pp. 567-615, No. 6, 1963.

[Ba64]Balakrishnan, A. V., "A general theory of nonlinear estimation problems in control systems," *Journal of Mathematics Analysis, Applications*, Vol. 8, No. 1, pp. 4-30, 1964.

[Ba90a]Baik, H. K., V. J. Mathews, and R. T. Short, "Adaptive lattice bilinear filters," *Proc. SPIE's 1990 Int. Symp. on Optics and Optoelectronic Applied Science and Engineering — Conference on Advanced Signal Processing Algorithms, Architectures and Implementations*, San Diego, California, July 1990.

[Ba90b]Baik, H. K. and V. J. Mathews, "Adaptive lattice bilinear filters," submitted to *IEEE Trans. Signal Processing*, December 1990.

[Be76]Benedetto, S., E. Biglieri and R. Daffara, "Performance of multilevel baseband digital systems in nonlinear environment," *IEEE Trans. Communications*, Vol. COM-24, pp. 1166-1175, 1976.

[Be79]Benedetto, S., E. Biglieri and R. Daffara, "Modeling and performance evaluation of nonlinear satellite links - A Volterra series approach," *IEEE Trans. Aerospace and Electronic Systems*, Vol. AES-15, pp. 494-507, 1979.

[Be83]Benedetto, S. and E. Biglieri, "Nonlinear equalization of digital satellite channels," *IEEE J. Selected Areas on Communications*, Vol. SAC-1, pp. 57-62, 1983.

[Be85]Bellafemina, M. and S. Benedetto, "Identification and equalization of nonlinear channels for digital transmission," *Proc. IEEE Int. Symp. Circuits and Systems*, Kyoto, Japan, pp. 1477-1480, June 1985.

[Be87]Benedetto, S., E. Biglieri, and V. Castellini, *Digital Transmission Theory*, Prentice Hall, Englewood Cliffs, New Jersey, 1987.

[Bi80]Billings, S. A., "Identification of nonlinear systems - A survey," *IEE Proceedings*, Vol. 127, Pt. D, No. 6, pp. 272-285, November 1980.

[Bi84a]Biglieri, E., A. Gersho, R. D. Gitlin and T. L. Lim, "Adaptive cancellation of nonlinear, intersymbol interference for voiceband data transmission," *IEEE J. Selected Areas on Communications*, Vol. SAC-2, pp. 765-77, 1984.

[Bi84b]Billings, S. A. and W. S. F. Voon, "Least squares parameter estimation algorithms for nonlinear systems," *Int. J. System Sci.*, Vol. 15, No. 6, pp. 601-615, 1984.

[Bi84c]Billings, S. A., "Identification of nonlinear systems," In *Nonlinear System Design*, (Ed. S. A. Billings, J. O. Gray and D. H. Owens) Peter Peregrinus, Ltd., London, UK, 1984.

[Bo83a]Boyd, S. , Y. S. Tang, and L. O. Chua, "Measuring Volterra kernels," *IEEE Transactions on Circuits and Systems*, Vol. CAS-30, No. 8, pp. 571-577, August 1983.

[Bo83b]Bovik, A. C., T. S. Huang, and D. C. Munson, "A generalization of median filtering using linear combinations of order statistics," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. ASSP-31, No. 6, pp. 1342-1349, December 1983.

[Bo85]Boyd, S. and L. O. Chua, "Fading memory and the problem of approximating nonlinear operators with Volterra series," *IEEE Transactions on Circuits and Systems*, Vol. CAS-32, No. 11, pp. 1150-1161, November 1985.

[Br70]Brillinger, D., "The identification of polynomial systems by means of higher order spectra," *J. Sound and Vibrations*, Vol. 12, pp. 301-313, 1970.

[Br74]Bruni, C., G. DiPillo, and G. Koch, "Bilinear systems: An appealing class of 'nearly linear systems' in theory and applications," *IEEE Trans. Automatic Control*, Vol. AC-19, pp. 334-348, 1974.

[Br76]Brockett, R., "Volterra series and geometric control theory," *Automatica*, Vol. 12, pp. 167-176, 1976.

[Ca85]Casar-Corredera, J. R., M. Garcia-Otero, and A. R. Figueiras-Vidal, "Data echo nonlinear cancellation," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Tampa, Florida, pp. 32.4.1-4, March 1985.

[Ci84]Cioffi, J. M . and T. Kailath, "Fast, recursive least-squares transversal filters for adaptive filtering," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-32, No. 2, pp. 304-337, April 1984.

[Cl81]Claasen, T. A. C. M. and W. F. G. Mecklenbrauker, "Comparison of the convergence of two algorithms for adaptive FIR digital filters," *IEEE Transactions Acoust., Speech, Signal Proc.*, Vol. ASSP-29, pp. 670-678, June 1981.

[Co80]Coker, M. J. and D. N. Simkins, "A nonlinear adaptive noise canceller," *Proceedings of the 1980 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 470-473, 1980.

[Da87]Davila, C. E., A. J. Welch, and H. G. Rylander, III, "A second-order adaptive Volterra filter with rapid convergence," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-35, No. 9, pp. 1259-1263, September 1987.

[Da89]Dai, H. and N. K. Sinha, "Robust recursive least-squares method with modified weights for bilinear system identification," *IEE Proceedings*, Vol. 136, Part D, No. 3, pp. 122-126, May 1989.

[Di88]Diaz, H. and A. A. Desrochers, "Modeling of nonlinear discrete-time systems from input-output data," *Automatica*, Vol. 24, No. 5, pp. 629-641, 1988.

[Ew80]Ewen, E. J. and D. D. Weiner, "Identification of weakly nonlinear systems using input and output measurements," *IEEE Transactions on Circuits and Systems*, Vol. CAS-27, No. 12, pp. 1255-1261, December 1980.

[Ey63]Eykhoff, P., "Some fundamental aspects of process-parameter estimation," *IEEE Transactions on Automatic Control*, Vol. AC-8, pp. 347-357, October 1963.

[Fa78]Falconer, D. D., "Adaptive equalization of channel nonlinearities in QAM Data Transmission Systems," *Bell Systems Tech. J.*, Vol. 57, No. 7, pp. 2589-2611, September 1978.

[Fa80]Fakhouri, S. Y., "Identification of the Volterra kernels of nonlinear systems," *IEE Proceedings*, Vol. 127, Pt. D, No. 6, pp. 296-304, November 1980.

[Fn87]Fnaiech, F. and L. Ljung, "Recursive identification of bilinear systems," *Int. J. Control*, Vol. 45, No. 2, pp. 453-470, 1987.

[Ga89]Gao, X. Y., W. M. Snelgrove, and D. A. Johns, "Nonlinear IIR adaptive filtering using a bilinear structure," *Proc. IEEE Int. Symp. Circuits and Systems*, Portland, Oregon, May 1989.

[Gr77]Griffiths, L. J., "A continuously adaptive filter implemented as a lattice structure," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Hartford, CT, pp. 683-686, May 1977.

[Ha86]Haykin, S., *Adaptive Filter Theory*, Prentice Hall, Englewood Cliffs, New Jersey, 1986.

[Hu86]Hunter, I. W. and M. J. Korenberg, "The identification of nonlinear biological systems: Wiener and Hammerstein cascade models," *Biological Cybernetics*, Vol. 55, pp. 135-144, 1986.

[Ja70]Jazwinsky, A. H., *Stochastic Processes and Filtering Theory*, Academic Press, New York, 1970.

[Ja77]Javed, A., B. A. Syrett, and P. A. Goud, "Intermodulation distortion analysis of reflection-type IMPATT amplifiers using Volterra series representation," *IEEE Transactions on Microwave Theory and Techniques*, Vol. MTT-25, No. 9, pp. 729-733, September 1977.

[Ja84]Jacobsen, S. C., S. G. Meek, and R. R. Fullmer, "An adaptive myoelectric filter," 6th IEEE Conf. Eng. in Med. and Biol. Soc., 1984.

[Ke85]Kenefic, R. J. and D. D. Weiner, "Application of the Volterra functional expansion in the detection of nonlinear functions of Gaussian processes," *IEEE Transactions on Communications*, Vol. COM-33, No. 3, pp. 276-279, March 1985.

[Ki83]Kim, T. L. and J. K. Omura, "Error-rate estimates in digital communication over a nonlinear channel with memory," *IEEE Transactions in Communications*, Vol. COM-31, No. 3, pp. 407-412, March 1983.

[Ko83a]Koh, T. and E. J. Powers, "An adaptive nonlinear digital filter with lattice orthogonalization," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Boston, Massachusetts, pp. 37-40, April 1983.

[Ko83b]Koh, T., E. J. Powers, and R. W. Miksad, "An approach to time-domain modeling of nonlinear drift oscillations in random seas," *Proceedings of the International Symposium on Offshore Engineering*, Rio de Janeiro, Brazil, 1983.

[Ko84a]Korenberg, M. J., "Fast orthogonal identification of nonlinear difference equation and functional expansion models," *Proc. 30th Midwest Symp. Circuits and Systems*, pp. 570-575, 1984.

[Ko84b]Korenberg, M. J., "Statistical identification of Volterra kernels of high-order systems," *Proc. IEEE Int. Symp. Circuits and Systems*, pp. 570-575, 1984.

[Ko85]Koh, T. and E. J. Powers, "Second-order Volterra filtering and its application to nonlinear system identification," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-33, No. 6, pp. 1445-1455,December 1985.

[Ko86]Korenberg, M. J. and I. W. Hunter, "The identification of nonlinear biological systems: LNL cascade models," *Biological Cybernetics*, Vol. 55, pp. 125-134, 1986.

[Ko88]Korenberg, M. J., "Identifying nonlinear difference equation and function expansion representations: The fast orthogonal algorithm," *Annals of Biomedical Engineering*, Vol. 16, pp. 123-142, 1988.

[La81]Lawrence, P. J., "Estimation of the Volterra functional series of a nonlinear system using frequency-response data," *IEE Proceedings*, Vol. 128, Pt. D, No. 5, pp. 206-210, September 1981.

[Le78]Lesiak, C. and A. Krener, "The existence and uniqueness of Volterra series for nonlinear systems," *IEEE Trans. Automatic Control*, Vol. AC-23, pp. 1090-1095, 1978.

[Le85]Lee, Y. H. and S. A. Kassam, "Generalized median filtering and related nonlinear filtering technique," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. ASSP-33, No. 3, pp. 672-683, June 1985.

[Le86]Lenk, P. J. and S. R. Parker, "Nonlinear modeling by discrete orthogonal lattice structure," *Proc. IEEE Int. Symp. Circuits and Systems*, 1986.

[Le91]Lee, J. and V. J. Mathews, "A fast recursive least-squares second-order Volterra filter and its performance analysis," submitted to *IEEE Trans. Signal Proc.*, January 1991.

[Li84]Ling, F. and J. G. Proakis, "A generalized multichannel least squares lattice algorithm based on sequential processing stages," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. ASSP-32, No. 2, pp. 381-390, April 1984.

[Lj78]Ljung, L., M. Morf, and D. Falconer, "Fast calculation of gain matrices for recursive estimation," *Int. J. Control*, Vol. 27, pp. 1-19, January 1978.

[Lo88]Lou, Y., C. L. Nikias, and A. N. Venetsanopoulos, "Efficient VLSI array processing structures for adaptive quadratic digital filters," *Circuits, Systems, Signal Process.*, Vol. 7, No. 2, 1988.

[Lu75]Lucky, R. W., "Modulation and detection for data transmission on the telephone channel," In *New Directions in Signal Processing in Communication and Control*, J. K. Skwirzynski, ed., Leiden, Holland. Noordhoff, 1975.

[Ma78]Marmarelis, P. Z. and V. Z. Marmarelis, *Analysis of Physiological Systems*, Plenum, New York, 1978.

[Ma85]Maqusi, M., "Performance of baseband digital data transmission in nonlinear channels with memory," *IEEE Transactions on Communications*, Vol. COM-33, No. 7, pp. 715-719, July 1985.

[Ma87a]Maragos, P. and R. W. Schafer, "Morphological filters, part I: Their set theoretic analysis and relations to linear shift invariant filters," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. ASSP-35, No. 8, pp. 1153-1169, August 1987.

[Ma87b]Maragos, P. and R. W. Schafer, "Morphological filters, part II: Their relation to median, order static and stack filters," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. ASSP-35, No. 8, pp. 1170-1184, August 1987.

[Ma87c]Mathews, V. J., "Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm," *IEEE Transactions on Acoust., Speech, Signal Proc.*, Vol. ASSP-35, No. 4, pp. 450-454, April 1987.

[Ma88]Mathews, V. J. and J. Lee, "A fast recursive least-squares second-order Volterra filter," *Proceedings of IEEE Int. Conf. Acoust., Speech, Signal Proc.*, New York, pp. 1383-1386, April 1988.

[Mo80]Mohler, R. R. and W. J. Kolodziej, "An overview of bilinear system theory and applications," *IEEE Trans. Systems, Man, and Cybernetics*, Vol. SMC-10, pp. 683-688, October 1980.

[Mo82]Moore, J. B., "Global convergence of output error recursions in colored noise," *IEEE Trans. Automatic Control*, Vol. AC-27, No. 6, pp. 1189-1199, December 1982.

[Na67]Narayanan, S., "Transistor distortion analysis using Volterra series representation," *Bell Systems Technical Journal*, Vol. 46, pp. 991-1204, May-June 1967.

[Na70]Narayanan, S., "Application of Volterra series to intermodulation distortion of transistor feedback amplifiers," *IEEE Transactions on Circuit Theory*, Vol. CT-17, pp. 518-527, November 1970.

[No82]Nodes, T. A. and N. C. Gallagher, "Median filters: Some modifications and their properties," *IEEE Trans. Acoust., Speech, Signal Proc.*, Vol. ASSP-30, No. 5, pp. 739-746, October 1982.

[Op68]Oppenheim, A. V., R. W. Schafer, and T. G. Stockham, Jr., "Nonlinear filtering of multiplied and convolved signals," *Proceedings IEEE*, Vol. 56, No. 8, pp. 1264-1291, August 1968.

[Pa81]Parker, S. R. and F. A. Perry, "A discrete ARMA model for nonlinear system identification," *IEEE Trans. Circuits and Systems*, Vol. CAS-28, No. 3, March 1981.

[Pa88]Palmieri, F. and C. G. Boncelet Jr., "A class of nonlinear adaptive filters," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, New York, pp. 1483-1486, April 1988.

[Pi90a]Pitas, I. and A. N. Venetsanopoulos, *Nonlinear Digital Filters — Principles and Applications*, Kluwer Academic Publishers, Boston, MA, 1990.

[Pi90b]Pitas, I. and A. N. Venetsanopoulos, "LMS and RLS adaptive L-filters," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, Albuquerque, NM, pp. 1389-1392, April 1990.

[Pr75]Prochazka, A. and R. Neumann, "High-frequency distortion analysis of a semiconductor diode for CATV applications," *IEEE Transactions on Consumer Electronics*, Vol. CE-21, No. 2, pp. 120-129, May 1975.

[Ra87]Ramponi, G. and G. L. Sicuranza, "Decision-directed nonlinear filters for image processing," *Electronic Letters*, Vol. 23, No. 23, pp. 1218-1219, November 5, 1987.

[Re84]Reiss, W., "Nonlinear distortion analysis of P-I-N diode attenuators using Volterra series representation," *IEEE Transactions on Circuits and Systems*, Vol. CAS-31, No. 6, pp. 535-542, June 1984.

[Ru81]Rugh, W. J., *Nonlinear System Theory. The Volterra-Wiener Approach*, Johns Hopkins University Press, Baltimore, Maryland, 1981.

[Sa83a]Sandberg, I. W., "On Volterra expansions for time-varying nonlinear systems," *IEEE Transactions on Circuits and Systems*, Vol. CAS-30, No. 2, pp. 61-67, February 1983.

[Sa83b]Sandberg, I. W., "Series expansions for nonlinear systems," *Circuits, Systems and Signal Processing*, Vol. 2, No. 1, pp. 77-87, 1983.

[Sa83c]Sandberg, I. W., "The mathematical foundations of associated expansions of mildly nonlinear systems," *IEEE Trans. Circuits and Systems*, Vol. CAS-30, pp. 441-455, July 1983.

[Sc80]Schetzen, M., *The Volterra and Wiener Theory of the Nonlinear Systems*, Wiley and Sons, New York, 1980.

[Sc81]Schetzen, M., "Nonlinear system modeling based on the Wiener theory," *Proceedings IEEE*, Vol. 69, No. 12, pp. 1557-1573, December 1981.

[Sh89]Shynk, J. J., "Adaptive IIR filtering," *IEEE ASSP Magazine*, Vol. 6, No. 2, pp. 4-21, April 1989.

[Si84]Sicuranza, G. L., A. Bucconi, and P. Mitri, "Adaptive echo cancellation with nonlinear digital filters," *Proc. IEEE Int. Conf. Acoust., Speech, Signal Proc.*, San Diego, California, pp. 3.10.1-4, March 1984.

[Si86]Sicuranza, G. L. and G. Ramponi, "Adaptive nonlinear digital filters using distributed arithmetic," *IEEE Trans. on Acoust. Speech, and Signal Proc.*, Vol. ASSP-34, No. 3, June 1986.

[Si87]Sicuranza, G. L. and G. Ramponi, "A variable-step adaptation algorithm for memory-oriented Volterra filters," *IEEE Trans. on Acoust. Speech, and Signal Proc.*, Vol. ASSP-35, No. 10, October 1987.

[Sm88]Smith, M. J., C. F. N. Cowan, and P. F. Adams, "Nonlinear echo cancellers based on transpose distributed arithmetic," *IEEE Transactions on Circuits and Systems*, Vol. CAS-35, No. 1, pp. 6-18, January 1988.

[St72]Stockham, T. G. Jr., "Image processing in the context of a visual model," *Proceedings IEEE*, Vol. 60, No. 7, pp. 828-842, July 1972.

[St85]Stapleton, J. C. and S. C. Bass, "Adaptive noise cancellation for a class of nonlinear, dynamic reference channels," *IEEE Transactions on Circuits and Systems*, Vol. CAS-32, No. 2, pp. 143-150, February 1985.

[Su81]Subba Rao, T., "On the theory of bilinear time series models," *J. Royal Statistical Society*, Series B, Vol. 43, 1981.

[Sy90]Syed, M. A. and V. J. Mathews, "Lattice and QR decomposition-based algorithms for recursive least squares adaptive nonlinear filters," *Proc. IEEE Int. Symp. Circuits and Systems*, New Orleans, Louisiana, May 1990.

[Th71]Thomas,E. J., "Some considerations on the application of the Volterra representation of nonlinear networks to adaptive echo cancellers," *Bell. Syst. Tech. J.*, Vol. 50, pp. 2979-2805, 1971.

[Th84]Thapar, H. K. and B. J. Leon, "Transform-domain and time-domain characterization of nonlinear systems with Volterra series," *IEEE Transactions on Circuits and Systems*, Vol. CAS-31, No. 10, pp. 906-912, October 1984.

[Ts88]Tsai, T. C. and D. Anastassiou, "Nonlinear recursive filters and applications to image processing," *Proc. ICASSP '88*, pp. 828-831, New York, April 1988.

[Wi58]Wiener, N., *Nonlinear Problems in Random Theory*, Wiley and Sons, New York, 1958.

[Za85]Zarzycki, J., *Nonlinear Prediction Ladder Filters for Higher Order Stochastic Sequences*, Springer-Verlag, Berlin, 1985.