Adaptive Parallel-Cascade Truncated Volterra Filters

Thomas M. Panicker, V. John Mathews, *Senior Member, IEEE*, and Giovanni L. Sicuranza, *Senior Member, IEEE*

***Abstract*—This paper studies adaptive truncated Volterra filters employing parallel-cascade structures. Parallel-cascade realizations implement higher order Volterra systems as a parallel connection of multiplicative combinations of lower order truncated Volterra systems. A normalized LMS adaptive filter is developed, and its performance capabilities are evaluated using a series of simulation experiments. The experimental results indicate that the normalized LMS adaptive parallel-cascade Volterra filter has superior convergence properties over several competing structures. This paper also includes an experiment that demonstrates the capability of the parallel-cascade adaptive system to reduce its implementation complexity by using fewer than the maximum number of branches required for the most general realization of the system.**

I. Introduction

THE OBJECTIVE of this paper is to present stochastic gradient adaptive Volterra filters employing parallel-cascade realizations of the system model. Parallel-cascade realizations implement higher order Volterra systems as a parallel connection of multiplicative combinations of lower order truncated Volterra systems. Such algorithms are attractive because of the modularity of the parallel-cascade realizations as well as the capability of such realizations to approximate nonlinear systems efficiently using a reduced number of branches.

Least-mean-square (LMS) adaptive Volterra filters employing direct-form realizations have become popular in recent years [1]–[3]. Adaptive parallel-cascade filters for quadratic system models were presented in [4] and [5]. The structure of [4] is not constrained to result in a unique solution to the estimation problem. Consequently, this filter exhibits relatively slow convergence behavior. The work in [5] constrains the filter structure to provide convergence to a unique solution. However, this adaptive filter requires appropriate training to select its initial settings. Our method is different from the previous works in many ways. As far as we are aware of, this is the first time that an adaptive parallel-cascade Volterra filter has been developed for nonlinearity orders larger than two. The adaptive filter presented in this paper is capable of converging to a unique solution and does not require the use of a training signal to initialize the algorithm. Finally, we derive a normalized LMS (NLMS) adaptive parallel-cascade Volterra filter. This algorithm is somewhat different from traditional normalized LMS adaptive filters [6]–[8] and offers significant performance advantages over previously available algorithms.

Manuscript received May 11, 1996; revised March 6, 1998. This work was supported in part by NATO Grant CRG. 950379. The associate editor coordinating the review of this paper and approving it for publication was Dr. Akihiko Sugiyama.

T. M. Panicker is with Rockwell Semiconductor Systems, San Diego, CA 92121 USA (tom.panicker@rss.rockwell.com).

V. J. Mathews is with the Department of Electrical Engineering, University of Utah, Salt Lake City, UT 84112 USA.

G. L. Sicuranza is with DEEI—University of Trieste, Trieste, Italy.

Publisher Item Identifier S 1053-587X(98)07068-8.

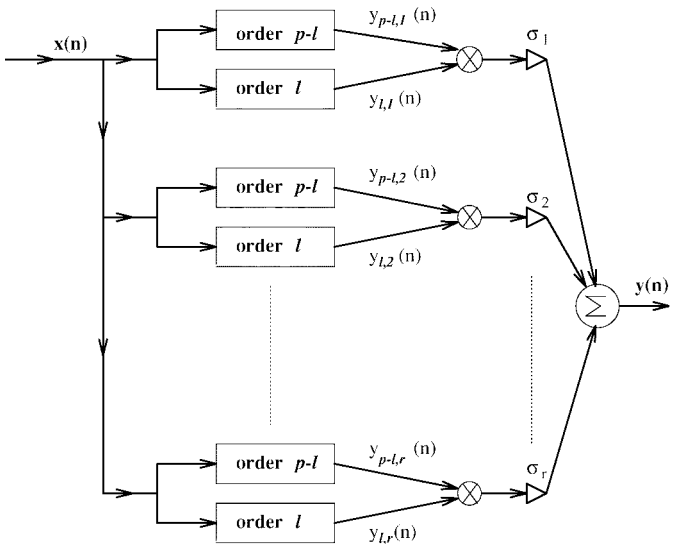

Fig. 1. Parallel-cascade realization of a homogeneous pth-order Volterra kernel. Each block represents a homogeneous Volterra system of the order shown within.

The rest of the paper is organized as follows. The parallel-cascade realizations of a homogeneous truncated Volterra system is discussed in Section II. An LMS adaptive filter for truncated Volterra systems in parallel-cascade form is derived in Section III. The NLMS adaptive filter for parallel-cascade realizations is derived in Section IV. Section V contains the results of several simulation experiments. Finally, concluding remarks are made in Section VI.

II. Parallel-Cascade Realization of Truncated Volterra Systems

The output of a homogeneous and causal pth-order Volterra system with N-sample memory is related to its input as [9], [10]

$$y(n) = \sum_{m_1=0}^{N-1} \sum_{m_2=m_1}^{N-1} \cdots \sum_{m_p=m_{p-1}}^{N-1} h_p(m_1, m_2, \ldots, m_p) \times x(n-m_1)x(n-m_2)\cdots x(n-m_p) \quad (1)$$

where $h_p(m_1, m_2, \ldots, m_p)$ denotes the coefficients of the pth-order kernel. In the above equation, we have explicitly made use of the invariance of the coefficients with respect to permutations of their indices $m_1, m_2, \ldots, m_p$. A pth-order homogeneous Volterra system can be realized using lth-order and $(p-l)$th-order Volterra systems as in Fig. 1. The results are

applicable to any p and $l < p$, and each component can be further decomposed using lower order components. Furthermore, the results can be easily extended to inhomogeneous Volterra systems by considering their homogeneous components of each order individually. The input–output relationship in (1) can be written compactly using matrix notation as

$$y(n) = \mathbf{X}_{N,l}^T(n)\mathbf{H}_{N,l,p-l}\mathbf{X}_{N,p-l}(n) \tag{2}$$

where $\mathbf{H}_{N,p_1,p_2}$ is a matrix of dimension $\binom{N+p_1-1}{p_1} \times \binom{N+p_2-1}{p_2}$ elements in which the coefficients in (1) are arranged in some orderly manner. The column vector $\mathbf{X}_{N,p}(n)$ has $\binom{N+p-1}{p}$ elements and contains all possible pth-order product signals of the form $x(n-m_1)x(n-m_2)\cdots x(n-m_p)$, where $0 \le m_1 \le m_2 \le \cdots \le m_p \le N-1$. Let $x(n-m_1)$ $x(n-m_2)\cdots x(n-m_l)$ be the ith element of $\mathbf{X}_{N,l}(n)$, and let $x(n-k_1)x(n-k_2)\cdots x(n-k_{p-l})$ be the jth element of $\mathbf{X}_{N,p-l}(n)$, where $0 \le k_1 \le k_2 \le \cdots \le k_{p-l} \le N-1$. Then, the (i,j)th element of the coefficient matrix $\mathbf{H}_{N,l,p-l}$ scales $x(n-m_1)x(n-m_2)\cdots x(n-m_l)x(n-k_1)x(n-k_2)\cdots x(n-k_{p-l})$ in (1). Since $\mathbf{H}_{N,p_1,p_2}$ contains more elements than there are independent coefficients, several entries of the coefficient matrix are zero. Let the rank of $\mathbf{H}_{N,l,p-l}$ be r. Then, it is well known that we can express $\mathbf{H}_{N,l,p-l}$ as [11]

$$\mathbf{H}_{N,l,p-l} = \sum_{i=1}^{r} \sigma_i \mathbf{U}_i \mathbf{V}_i^T \tag{3}$$

using singular value decomposition. In the above equation, σ_i's are the nonzero singular values of $\mathbf{H}_{N,l,p-l}$, and $\mathbf{U}_i$'s and $\mathbf{V}_i$'s are the left and right singular vectors, respectively, of the matrix. Substituting (3) in (2), we get

$$\begin{aligned} y(n) &= \sum_{i=1}^{r} \sigma_i \left[\mathbf{X}_{N,l}^T(n)\mathbf{U}_i\right]\left[\mathbf{V}_i^T\mathbf{X}_{N,p-l}(n)\right] \\ &= \sum_{i=1}^{r} \sigma_i y_{l,i}(n) y_{p-l,i}(n) \end{aligned} \tag{4}$$

where we have defined $y_{l,i}(n)$ as the output of a homogeneous lth-order Volterra system given by

$$y_{l,i}(n) = \mathbf{X}_{N,l}^T(n)\mathbf{U}_i. \tag{5}$$

The signal $y_{p-l,i}(n)$ is also defined in a similar manner. From the above analysis, it is clear that the left and right singular vectors define the coefficients of the lower order components used in the decomposition shown in Fig. 1. This structure is applicable for any $l < p$. Each lower order block can be further decomposed into realizations employing even lower order components.

A. Some Special Decompositions

Under some mild conditions, we can express the coefficient matrix using the LU decomposition in which $\mathbf{H}_{N,l,p-l}$ is expressed as

$$\mathbf{H}_{N,l,p-l} = \sum_{i=1}^{r} \mathbf{L}_i \mathbf{U}_i^T \tag{6}$$

where the leading $(i-1)$ elements of $\mathbf{L}_i$ and $\mathbf{U}_i$ are zero, and the ith element of $\mathbf{L}_i$ is one. Such a decomposition is possible if all the leading principal submatrices of $\mathbf{H}_{N,l,p-l}$ are nonsingular [11]. A realization of the Volterra filter using the above decomposition results in a small amount of computational savings.

When $l = p - l = p/2$, it is possible to arrange the coefficients of the filter such that $\mathbf{H}_{N,p/2,p/2}$ is a symmetric matrix. It is then possible to obtain an LDL^T decomposition [11] of the form

$$\mathbf{H}_{N,p/2,p/2} = \sum_{i=1}^{r} \sigma_i \mathbf{L}_i \mathbf{L}_i^T. \tag{7}$$

Substituting this expression into (2) and manipulating the results as before, we can see that the pth-order Volterra filter can be exactly realized as a parallel-cascade structure in which each branch contains a $(p/2)$th-order Volterra filter whose output is squared and weighted by a constant multiplier σ_i, i.e.,

$$y(n) = \sum_{i=1}^{r} \sigma_i (y_{p/2,i}(n))^2 \tag{8}$$

where

$$y_{p/2,i}(n) = \mathbf{X}_{N,p/2}^T(n)\mathbf{L}_i. \tag{9}$$

In most situations, the coefficient vector $\mathbf{L}_i$ can be constrained to have zero value for its $(i-1)$ leading elements and one for its ith element [11].

III. The LMS Adaptive Parallel-Cascade Volterra Filter

Consider the problem of estimating a desired response signal $d(n)$ as the output of a homogeneous pth-order adaptive Volterra system employing the parallel-cascade structure with r branches. The output of the adaptive filter may be written as

$$\begin{aligned} \hat{d}(n) &= \sum_{i=1}^{r} \sigma_i(n) y_{l,i}(n) y_{p-l,i}(n) \\ &= \sum_{i=1}^{r} \sigma_i(n) \left[\mathbf{X}_{N,l}^T(n)\mathbf{U}_i(n)\right]\left[\mathbf{V}_i^T(n)\mathbf{X}_{N,p-l}(n)\right] \end{aligned} \tag{10}$$

where we have explicitly shown the time dependence of the parameters $\sigma_i(n)$, $\mathbf{U}_i(n)$, and $\mathbf{V}_i(n)$ in the above expression. Let

$$e(n) = d(n) - \hat{d}(n) \tag{11}$$

denote the estimation error at time n. The coefficient update strategy for the LMS adaptive parallel-cascade Volterra filter can be derived as

$$\begin{aligned} \sigma_i(n+1) &= \sigma_i(n) - \frac{\mu_1}{2}\frac{\partial}{\partial \sigma_i(n)} e^2(n) \\ &= \sigma_i(n) + \mu_1 e(n) y_{l,i}(n) y_{p-l,i}(n) \end{aligned} \tag{12}$$

$$\begin{aligned} \mathbf{U}_i(n+1) &= \mathbf{U}_i(n) - \frac{\mu_2}{2}\frac{\partial}{\partial \mathbf{U}_i(n)} e^2(n) \\ &= \mathbf{U}_i(n) + \mu_2 e(n)\sigma_i(n) y_{p-l,i}(n)\mathbf{X}_{N,l}(n) \end{aligned} \tag{13}$$

and

$$\mathbf{V}_i(n+1) = \mathbf{V}_i(n) - \frac{\mu_3}{2}\frac{\partial}{\partial \mathbf{V}_i(n)}e^2(n) = \mathbf{V}_i(n) + \mu_3 e(n)\sigma_i(n)y_{l,i}(n)\mathbf{X}_{N,p-l}(n). \quad (14)$$

In the above equations, the step sizes μ_1, μ_2, and μ_3 are small positive constants that control the speed of convergence and the steady-state characteristics of the adaptive filter.

When the adaptive filter employs the LU decomposition of the coefficient matrix as given by (6), the coefficient update equations may be modified as

$$\mathbf{L}_i(n+1) = \mathbf{L}_i(n) + \mu_1 e(n)y_{p-l,i}(n)\mathbf{X}_{N,l}(n) \quad (15)$$

and

$$\mathbf{U}_i(n+1) = \mathbf{U}_i(n) + \mu_2 e(n)y_{l,i}(n)\mathbf{X}_{N,p-l}(n) \quad (16)$$

where μ_1 and μ_2 are positive step size values for the update equations. Recall from Section II-A that the first $(i-1)$ elements of the coefficient vectors $\mathbf{L}_i(n)$ and $\mathbf{U}_i(n)$ are, by definition, zero and that the ith element of $\mathbf{L}_i(n)$ is one, as shown by their structure

$$\mathbf{L}_i(n) = [0 \quad \cdots \quad 0 \quad 1 \quad l_{i,i+1}(n) \quad \cdots \quad l_{i,M}(n)]^T \qquad i = 1, 2, \ldots, r \quad (17)$$

and

$$\mathbf{U}_i(n) = [0 \quad \cdots \quad 0 \quad u_{i,i}(n) \quad u_{i,i+1}(n) \quad \cdots \quad u_{i,N}(n)]^T \qquad i = 1, 2, \ldots, r \quad (18)$$

respectively. In the above definitions, the parameters $M = \binom{N+l-1}{l}$ and $N = \binom{N+(p-l)-1}{(p-l)}$ represent the number of coefficients in the lth-order system and $(p-l)$th-order system, respectively. It is important to recognize that the zero and unity coefficients of (17) and (18) are not updated during each iteration. However, the matrix representation in (15) and (16) does not explicitly show this fact.

Similarly, we can derive the coefficient update strategy for realizations employing the LDL^T decomposition of (7) as

$$\sigma_i(n+1) = \sigma_i(n) + \mu_1 e(n)y^2_{p/2,i}(n) \quad (19)$$

and

$$\mathbf{L}_i(n+1) = \mathbf{L}_i(n) + \mu_2 e(n)\sigma_i(n)y_{p/2,i}(n)\mathbf{X}_{N,p/2}(n) \quad (20)$$

where $\mathbf{L}_i(n)$ has the same form given by (17). Once again, μ_1 and μ_2 are the convergence parameters of the adaptive filter, and we do not update the first i entries of $\mathbf{L}_i(n)$. The above derivations are similar to that of [4] in which an adaptive parallel-cascade quadratic filter was derived. For the quadratic estimation problem, a bound on the step size for convergence was derived in [4]. Because of the nonlinearities in the system model, derivation of the stability bounds for the step sizes is a very difficult problem. In the next section, we derive a normalized LMS adaptation algorithm for the parallel-cascade realization. Experimental results have shown that this system has significant performance advantage over the algorithm derived in this section. Consequently, we do not attempt a rigorous stability analysis of the LMS adaptive parallel-cascade Volterra filters in this paper. A heuristic bound on the step size of the normalized LMS adaptive parallel-cascade filter is derived in the next section.

IV. The Normalized LMS Adaptive Parallel-Cascade Volterra Filter

One of the difficulties with the conventional LMS adaptive filter is that its performance depends on the input signal power. The normalized LMS adaptive filter is employed to provide robustness to variations in the input signal strength. Furthermore, the step-size selection is, in general, a simpler task with the normalized LMS adaptive filter than the conventional LMS adaptive filters. We derive a normalized LMS adaptive parallel-cascade truncated Volterra filter in this section. The derivation follows the work in [6]. In order to obtain an easily realizable adaptive filter, we resort to several simplifying approximations during the derivation.

The principle behind the derivation of the normalized LMS adaptive parallel-cascade filter may be described as follows. At each iteration, we solve for

$$d(n) = \sum_{i=1}^{r}(\sigma_i(n) + \delta\sigma_i(n))\left[\mathbf{X}^T_{N,l}(n)(\mathbf{U}_i(n) + \delta\mathbf{U}_i(n))\right] \times \left[(\mathbf{V}_i(n) + \delta\mathbf{V}_i(n))^T\mathbf{X}_{N,p-l}(n)\right] \quad (21)$$

where $\delta\sigma_i(n)$, $\delta\mathbf{U}_i(n)$, and $\delta\mathbf{V}_i(n)$ are the increments in the coefficient values that provide the exact solution to the above equation. Since there are an infinite number of solutions for the above problem, we seek the solution that minimizes the magnitude of the increments defined as

$$\sum_{i=1}^{r}(\delta\sigma_i(n))^2 + \sum_{i=1}^{r}\|\delta\mathbf{U}_i(n)\|^2 + \sum_{i=1}^{r}\|\delta\mathbf{V}_i(n)\|^2 \quad (22)$$

subject to the equality in (21). Such a solution may, in general, result in erratic coefficient behavior because of the presence of the noise in the desired response signal. Consequently, the normalized LMS adaptive parallel-cascade filter updates the coefficients by moving a fraction of the distance suggested by the solution to the optimization problem. Thus, the update equations are given by

$$\sigma_i(n+1) = \sigma_i(n) + \mu\delta\sigma_i(n) \quad (23)$$

$$\mathbf{U}_i(n+1) = \mathbf{U}_i(n) + \mu\delta\mathbf{U}_i(n) \quad (24)$$

and

$$\mathbf{V}_i(n+1) = \mathbf{V}_i(n) + \mu\delta\mathbf{V}_i(n) \quad (25)$$

where μ is a small positive constant. Because the step sizes in this algorithm correspond to a fraction of the difference between the current step size and the solution to the optimization problem, it is not necessary to use three different step sizes for updating the three components of the adaptive filter. A heuristic bound for the step size is provided later in this section. The constrained optimization problem can be solved using the method of Lagrange multipliers. We define a scalar

cost function

$$J(n) = \sum_{i=1}^{r} (\delta\sigma_i(n))^2 + \sum_{i=1}^{r} \|\delta\mathbf{U}_i(n)\|^2 + \sum_{i=1}^{r} \|\delta\mathbf{V}_i(n)\|^2 + \lambda\Bigg[d(n) - \sum_{i=1}^{r} (\sigma_i + \delta\sigma_i(n))\big[\mathbf{X}_{N,l}^T(n)(\mathbf{U}_i(n) + \delta\mathbf{U}_i(n))\big]\big[(\mathbf{V}_i(n) + \delta\mathbf{V}_i(n))^T\mathbf{X}_{N,p-l}(n)\big]\Bigg] \quad (26)$$

where λ is a Lagrange multiplier. Taking the partial derivative of the above expression with respect to $\delta\sigma_i(n)$, $\delta\mathbf{U}_i(n)$, and $\delta\mathbf{V}_i(n)$, respectively, we get

$$\frac{\partial}{\partial\delta\sigma_i(n)} J(n) = 2\delta\sigma_i(n) - \lambda\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n) \quad (27)$$

and

$$\frac{\partial}{\partial\delta\mathbf{U}_i(n)} J(n) = 2\delta\mathbf{U}_i(n) - \lambda(\sigma_i(n) + \delta\sigma_i(n))\tilde{y}_{p-l,i}(n)\mathbf{X}_{N,l}(n) \quad (28)$$

as well as

$$\frac{\partial}{\partial\delta\mathbf{V}_i(n)} J(n) = 2\delta\mathbf{V}_i(n) - \lambda(\sigma_i(n) + \delta\sigma_i(n))\tilde{y}_{l,i}(n)\mathbf{X}_{N,p-l}(n) \quad (29)$$

for $i = 1, \ldots, r$, where

$$\tilde{y}_{l,i}(n) = \mathbf{X}_{N,l}^T(n)[\mathbf{U}_i(n) + \delta\mathbf{U}_i(n)] \quad (30)$$

and

$$\tilde{y}_{p-l,i}(n) = \mathbf{X}_{N,p-l}^T(n)[\mathbf{V}_i(n) + \delta\mathbf{V}_i(n)] \quad (31)$$

are the output signals at the ith branch when the coefficient vectors are the solution to the optimization problem.

Solving for the unknown variables after setting (27)–(29) to zero requires the simultaneous solution for $(3r+1)$ sets of unknowns from $(3r+1)$ sets of coupled nonlinear equations. In the following, we pursue an approximate solution to the optimization problem. The approximations employed are valid during the final stages of adaptation when the coefficients are close to the optimal solution. They also assume that the level of measurement noise in the desired response signal and the level of nonstationarity in the operating environment are relatively low. Even though the approximations employed can be rigorously justified only for the situations described above, our experience is that the resulting adaptive filter exhibits good convergence properties under a variety of noise conditions. Results of experiments illustrating this observation are included in Section V-A.

A. An Approximate Solution to the Constrained Optimization Problem

To find an approximate solution, we substitute (30) and (31) in (27)–(29) and then use the simplification that $\delta\sigma_i(n)$, $\delta\mathbf{U}_i(n)$, and $\delta\mathbf{V}_i(n)$ are small so that we can approximate $\sigma_i(n) + \delta\sigma_i(n)$, $\mathbf{U}_i(n) + \delta\mathbf{U}_i(n)$, and $\mathbf{V}_i(n) + \delta\mathbf{V}_i(n)$ on the right-hand sides of the resulting equations with $\sigma_i(n)$, $\mathbf{U}_i(n)$, and $\mathbf{V}_i(n)$, respectively. Multiplying (27) with $\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n)$ and equating the result to zero, we get

$$2\delta\sigma_i(n)\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n) = \lambda\tilde{y}_{l,i}^2(n)\tilde{y}_{p-l,i}^2(n). \quad (32)$$

In order to proceed, we add $2\sigma_i(n)\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n)$ to both sides of the above equation and add the result over all i. These operations result in

$$\sum_{i=1}^{r} 2[\sigma_i(n) + \delta\sigma_i(n)]\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n) = \sum_{i=1}^{r} \big[2\sigma_i(n)\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n) + \lambda\tilde{y}_{l,i}^2(n)\tilde{y}_{p-l,i}^2(n)\big]. \quad (33)$$

Since the left-hand side uses the solution to the constrained optimization problem for all variables, it is identical to $2d(n)$, which is twice the desired response signal. By using the approximation that $\delta\sigma_i(n)$, $\delta\mathbf{U}_i(n)$, and $\delta\mathbf{V}_i(n)$ have small magnitudes, we can approximate $\tilde{y}_{l,i}(n)$ and $\tilde{y}_{p-l,i}(n)$ on the right-hand side of (33) with $y_{l,i}(n)$ and $y_{p-l,i}(n)$, respectively. This approximation and the earlier observation about the left-hand side of (33) lead to

$$2d(n) = 2\sum_{i=1}^{r} \sigma_i(n)y_{l,i}(n)y_{p-l,i}(n) + \lambda\sum_{i=1}^{r} y_{l,i}^2(n)y_{p-l,i}^2(n) = 2\hat{d}(n) + \lambda\sum_{i=1}^{r} y_{l,i}^2(n)y_{p-l,i}^2(n). \quad (34)$$

Subtracting $2\hat{d}(n)$ from both sides of the above equation, we get

$$2e(n) = \lambda\sum_{i=1}^{r} y_{l,i}^2(n)y_{p-l,i}^2(n). \quad (35)$$

Similarly, equating (28) to zero after multiplying both sides by $[\sigma_i(n) + \delta\sigma_i(n)]\tilde{y}_{p-l,i}(n)\mathbf{X}_{N,l}^T(n)$, we get

$$2[\sigma_i(n) + \delta\sigma_i(n)]\tilde{y}_{p-l,i}(n)\mathbf{X}_{N,l}^T(n)[\mathbf{U}_i(n) + \delta\mathbf{U}_i(n) - \mathbf{U}_i(n)] = \lambda[\sigma_i(n) + \delta\sigma_i(n)]^2\tilde{y}_{p-l,i}^2(n)\|\mathbf{X}_{N,l}(n)\|^2. \quad (36)$$

Using approximations similar to those employed for deriving (35), we arrive at

$$2e(n) = \lambda\|\mathbf{X}_{N,l}(n)\|^2\sum_{i=1}^{r} \sigma_i^2(n)y_{p-l,i}^2(n). \quad (37)$$

Similarly, manipulation of (29) leads to

$$2e(n) = \lambda\|\mathbf{X}_{N,p-l}(n)\|^2\sum_{i=1}^{r} \sigma_i^2(n)y_{l,i}^2(n). \quad (38)$$

We can obtain an expression for λ from (35), (37), and (38) as

$$\lambda(n) = \frac{6e(n)}{P_\sigma(n) + P_u(n) + P_v(n)} \quad (39)$$

where

$$P_\sigma(n) = \sum_{i=1}^{r} y_{l,i}^2(n)y_{p-l,i}^2(n) \quad (40)$$

and

$$P_u(n) = \|\mathbf{X}_{N,l}(n)\|^2 \sum_{i=1}^{r} \sigma_i^2(n) y_{p-l,i}^2(n) \tag{41}$$

as well as

$$P_v(n) = \|\mathbf{X}_{N,p-l}(n)\|^2 \sum_{i=1}^{r} \sigma_i^2(n) y_{l,i}^2(n). \tag{42}$$

Under the approximations we have employed, $P_\sigma(n)$, $P_u(n)$, and $P_v(n)$ are close to each other. A solution for λ using all three quantities as in (39) provides a smoother approximation to the optimal solution. We have included the time index n for the Lagrangian multiplier in (39) since the solution changes with time. We can solve for $\delta\sigma_i(n)$, $\delta\mathbf{U}_i(n)$, and $\delta\mathbf{V}_i(n)$ by substituting for λ in (27)–(29) after equating them to zero. The normalized LMS adaptive parallel-cascade filter is obtained by substituting the solutions so obtained in the update equations (23)–(25), respectively. The relevant equations are

$$\sigma_i(n+1) = \sigma_i(n) + \frac{3\mu}{P_\sigma(n) + P_u(n) + P_v(n)} \times e(n) y_{l,i}(n) y_{p-l,i}(n), \tag{43}$$

and

$$\mathbf{U}_i(n+1) = \mathbf{U}_i(n) + \frac{3\mu}{P_\sigma(n) + P_u(n) + P_v(n)} \times e(n)\sigma_i(n) y_{p-l,i}(n) \mathbf{X}_{N,l}(n) \tag{44}$$

and, in addition

$$\mathbf{V}_i(n+1) = \mathbf{V}_i(n) + \frac{3\mu}{P_\sigma(n) + P_u(n) + P_v(n)} \times e(n)\sigma_i(n) y_{l,i}(n) \mathbf{X}_{N,p-l}(n). \tag{45}$$

The normalization factor contains higher order powers of the input signal and, therefore, often has very large statistical variations associated with it. In order to mitigate the ill effects such large dynamics have on the update equation, the normalization factor $P_\sigma(n) + P_u(n) + P_v(n)$ may be replaced by a smoothed version obtained using a single-pole lowpass filter as

$$\xi(n) = \alpha\xi(n-1) + (1-\alpha)[P_\sigma(n) + P_u(n) + P_v(n)] \tag{46}$$

where α is a constant between 0 and 1 and is usually very close to 1. In addition to the smoothing, using $\xi(n)$ also has the effect of making $\xi(n)$ and $\xi(n+1)$ closer to each other than they are to their unsmoothed counterpart. Recall that the approximations employed in solving the constrained optimization problem assumes that the normalization factors are close to each other at successive times. The smoothed normalization factor $\xi(n)$ may be initialized using a positive constant. However, we initialized $\xi(n)$ as zero in all the experiments in Section V that involved the normalized parallel-cascade structure. The complete algorithm is tabulated in Table I. The computational complexity of each branch corresponds to $O(N^l)$ multiplications per iteration, where we have assumed without loss of generality that $l \geq p - l$. The overall complexity of the system is therefore $O(rN^l)$ multiplications per iteration. It is possible in many applications to choose the number of branches r to be much smaller than the maximum possible $O(N^{p-l})$ branches. In such situations, the computational complexity of adaptive parallel-cascade Volterra filters is much smaller than that of adaptive direct form Volterra filters.

TABLE I
NORMALIZED LMS ADAPTIVE PARALLEL-CASCADE TRUNCATED VOLTERRA FILTER

Initialization

for $i = 1 : r$,

$\sigma_i(0)$ = a small positive constant

$l_{i,j}(0) = \begin{cases} 1 & ;\text{if } i = j \\ 0 & ;\text{otherwise} \end{cases}$

$u_{i,j}(0) = \begin{cases} 1 & ;\text{if } i = j \\ 0 & ;\text{otherwise} \end{cases}$

end

$\xi(0)$ = a small positive constant

Main Iteration

for $i = 1 : r$,

$y_{l,i}(n) = \mathbf{U}_i^T(n)\mathbf{X}_{N,l}(n)$

$y_{p-l,i}(n) = \mathbf{V}_i^T(n)\mathbf{X}_{N,p-l}(n)$

end

$e(n) = d(n) - \sum_{i=1}^{r} \sigma_i(n) y_{l,i}(n) y_{p-l,i}(n)$

$P_\sigma(n) = \sum_{i=1}^{r} y_{l,i}^2(n) y_{p-l,i}^2(n)$

$P_u(n) = \| \mathbf{X}_{N,l}(n) \|^2 \sum_{i=1}^{r} \sigma_i^2(n) y_{p-l,i}^2(n)$

$P_v(n) = \| \mathbf{X}_{N,p-l}(n) \|^2 \sum_{i=1}^{r} \sigma_i^2(n) y_{l,i}^2(n)$

$\xi(n) = \alpha\xi(n-1) + (1-\alpha)[P_\sigma(n) + P_u(n) + P_v(n)]$

for $i = 1 : r$,

$\sigma_i(n+1) = \sigma_i(n) + \frac{3\mu}{\xi(n)} e(n) y_{l,i}(n) y_{p-l,i}(n)$

$\mathbf{U}_i(n+1) = \mathbf{U}_i(n) + \frac{3\mu}{\xi(n)} e(n)\sigma_i(n) y_{p-l,i}(n) \mathbf{X}_{N,l}(n)$

$\mathbf{V}_i(n+1) = \mathbf{V}_i(n) + \frac{3\mu}{\xi(n)} e(n)\sigma_i(n) y_{l,i}(n) \mathbf{X}_{N,p-l}(n)$

end

B. The NLMS Algorithm for LDL^T Decomposition

We can derive a coefficient update strategy for an adaptive filter that uses the LDL^T decomposition using a similar set of approximations and derivations as employed in the previous section. The coefficients are updated in this case as

$$\sigma_i(n+1) = \sigma_i(n) + \frac{2\mu}{P_\sigma(n) + P_l(n)} e(n) y_{p/2,i}^2(n) \tag{47}$$

and

$$\mathbf{L}_i(n+1) = \mathbf{L}_i(n) + \frac{2\mu}{P_\sigma(n) + P_l(n)} e(n)\sigma_i(n) y_{p/2,i}(n) \mathbf{X}_{N,p/2}(n) \tag{48}$$

where $P_\sigma(n)$ and $P_l(n)$ are given by

$$P_\sigma(n) = \sum_{i=1}^{r} y_{p/2,i}^4(n) \tag{49}$$

and

$$P_l(n) = \|\mathbf{X}_{N,p/2}(n)\|^2 \sum_{i=1}^{r} \sigma_i^2(n) y_{p/2,i}^2(n) \qquad (50)$$

respectively. For the same reasons as those explained in the previous subsection, the normalization factor $P_\sigma(n) + P_l(n)$ may be replaced by a smoothed version similar to that described in (46).

C. A Bound on the Step Size

As discussed earlier, deriving an exact bound on the step size for providing stable operation of the adaptive parallel-cascade filter is a very difficult problem. However, we can argue in a heuristic manner that a selection of the step size in the range $0 < \mu < 2$ should result in stable operation of the system. Since the solution to the constrained optimization problem is obtained by choosing $\mu = 1$, we can argue that the estimation error is bounded for this choice of the step size as long as the input signal and the desired response signal are bounded in some sense. Intuitively, a choice of μ in the range $0 < \mu < 2$ will move the parameters closer to the solution of the constrained optimization problem from its current value. Obviously, the above is not a rigorous analysis. However, this result agrees with the rigorous result obtained in [6] and [8] for normalized LMS adaptive filters that employ linear system models. Because of the approximations employed in the derivation of the adaptive filter, the above bound may only be approximately met. In the next section, we have included the result of an experiment demonstrating stable operation of the adaptive filter for values of μ as high as 1.5.

V. Experimental Results

The results of several experiments that demonstrate the good properties of the NLMS adaptive parallel-cascade Volterra filters are presented in this section. The adaptive filter was used for identifying an unknown system from measurements of its input and output signals in these experiments. The unknown system used in all the experiments was a homogeneous truncated fourth-order Volterra system. The adaptive filter was implemented by realizing the fourth-order system model using second-order components. The structure was based on the LDL^T decomposition and, therefore, exploited the symmetry of the coefficient matrix.

A. Experiment 1

The objectives of the experiments described in this subsection are

1) to compare the NLMS adaptive parallel-cascade filters with some other competing algorithms available in the literature;
2) to demonstrate the robustness of the NLMS adaptive filter to observation noise present in the desired response signal,
3) to verify the usefulness of the heuristic bound given in Section IV-C.

We first compare the performance of the NLMS adaptive filter, the unnormalized LMS adaptive filter employing the LDL^T decomposition, and the unnormalized LMS adaptive filter employing the direct form realization of the system model in a stationary system identification problem. The normalized LMS adaptive Volterra filter implemented using the direct-form realization surprisingly resulted in much poorer performance than the other three structures and, therefore, is not included in the comparisons presented here. The coefficients of the direct-form LMS adaptive filter were initialized to zero. The scalar multipliers $\sigma_i(n)$ and the ith element of the coefficient vector $\mathbf{L}_i(n)$ were initialized to 1 in the parallel-cascade filter. The rest of the elements of $\mathbf{L}_i(n)$ were initialized to zero. This particular initialization ensures nonzero initial values for $P_\sigma(n)$ and $P_l(n)$. Furthermore, an all-zero coefficient set will not allow the parallel-cascade system to adapt. The initial values of the mean-square estimation error are slightly different because of the differences in the initialization process. The parameter α in (46) was chosen as 0.9 in the experiments involving normalized parallel-cascade structure. Since we chose a relatively large value for $(1 - \alpha)$ in the experiments, we initialized $\xi(0)$ to be zero. For values of α very close to one, it is advisable to chose $\xi(0)$ to be a small positive constant in order to avoid numerical problems during normalization in the early stages of adaptation.

The unknown system was a homogeneous fourth-order truncated Volterra system with five-sample memory whose coefficients were given by (51), shown at the bottom of the page, where $a_i = (m_i - 1)$, and $u(m_1, m_2, m_3, m_4)$ is a random variable that is uniformly distributed between -0.1 and $+0.1$. The number of branches in the parallel-cascade realization of the above filter is $\binom{5+2-1}{2} = 15$. The input signal was generated as the output of a linear system with input–output relationship given by

$$x(n) = 0.6x(n-1) + 0.8\epsilon(n) \qquad (52)$$

where $\epsilon(n)$ was a white Gaussian signal with unit variance and zero mean value. The desired response signal was generated by processing this signal with the fourth-order Volterra system of (51) and then corrupting the output with an uncorrelated white Gaussian noise sequence with zero mean value and variance σ_n^2. Fig. 2 displays a plot of the mean-square estimation error signals obtained using the direct form LMS, parallel-cascade LMS, and parallel-cascade NLMS adaptive filters for $\sigma_n^2 = 0.01$. The adaptive filters employed system models that were exactly matched to that of the unknown system. The step sizes for the three methods were selected to get approximately

$$h_4(m_1, m_2, m_3, m_4) = \begin{cases} \frac{100}{2\pi[1.5^4 + a_1^4 + a_2^4 + a_3^4 + a_4^4]^{3/4}} + u(m_1, m_2, m_3, m_4); & 0 \le m_1 \le m_2 \le m_3 \le m_4 \le 4 \\ 0; & \text{otherwise} \end{cases} \qquad (51)$$

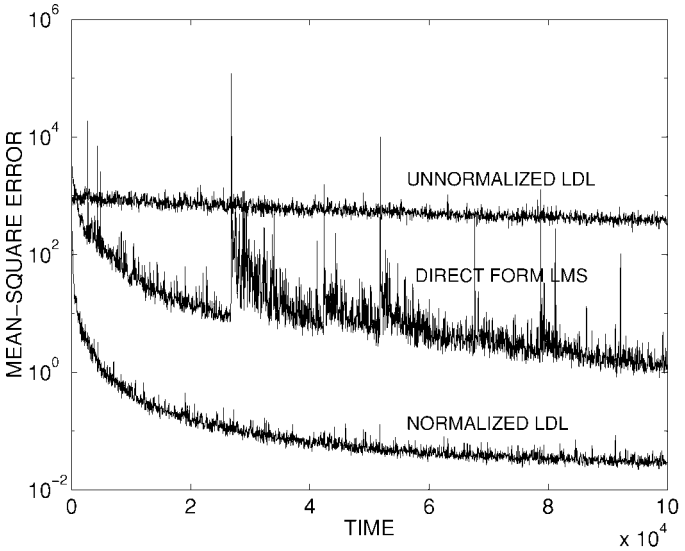

Fig. 2. Mean-square error of the adaptive filters for colored Gaussian input and measurement noise with variance 0.01.

TABLE II
PARAMETERS AND EXCESS MEAN-SQUARE ERRORS OF EXPERIMENT 1

type	σ_n^2	μ	*excess mse*
Direct form LMS	0.01	8.3×10^{-7}	0.003694
Normalized LDL^T	0.01	0.074	0.003666
Unnormalized LDL^T	0.01	6.3×10^{-9}	0.003757

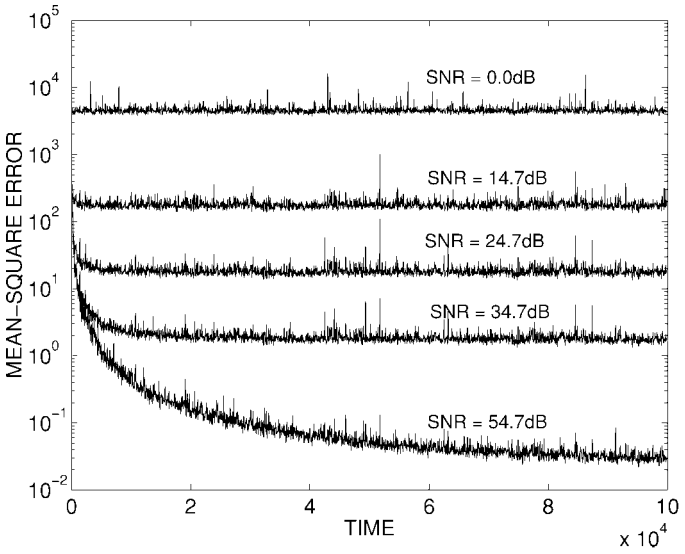

Fig. 3. Mean-square error of the NLMS adaptive filter with colored Gaussian input signal and different noise levels.

the same steady-state excess mean-square estimation error. The selection was performed numerically by initializing the three algorithms using the true values of the unknown system and letting the adaptive filters run for 100 000 samples for several step sizes. The excess mean-square errors were measured by averaging the excess estimation errors over the last 20 000 samples over 100 independent experiments. The step sizes and the measured values of the corresponding excess mean-square errors are displayed in Table II. It can be seen from

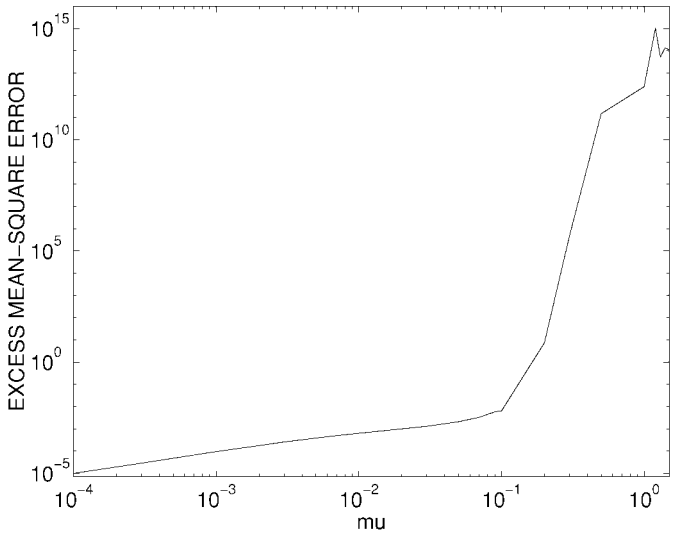

Fig. 4. Excess mean-square error as a function of μ in Experiment 1.

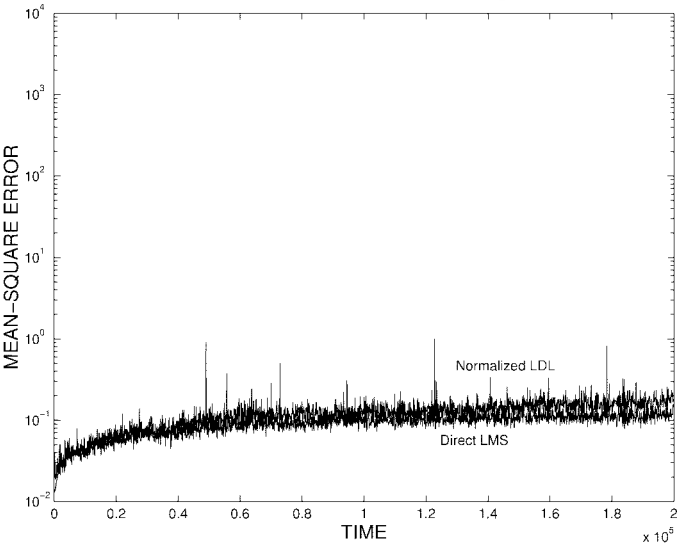

Fig. 5. Mean-square error of the adaptive filters in Experiment 2.

the figure that the NLMS algorithm developed in this paper converges significantly faster than the direct form LMS and the unnormalized LMS using the LDL^T decomposition. All the spikes in the figures are due to the direct form LMS adaptive filter, indicating that this filter is operating near the stability bound for the step size.

Fig. 3 displays the evolution of the mean-square estimation error of the NLMS algorithm for different output signal-to-noise ratios (SNR's), and $\mu = 0.074$. We can see from the figure that the convergence characteristics of the NLMS adaptive filter is not significantly changed by increasing the noise in the desired response signal. These results demonstrate the usefulness of the adaptive filter, in spite of the fact that its derivations assumed low noise levels.

Finally, we consider the performance of the adaptive filter for several values of the step size. Fig. 4 displays the excess mean-square error in the NLMS algorithm as a function of the convergence parameter μ. The excess mean-square error was evaluated by varying μ in the range $0 < \mu < 2$ for SNR =

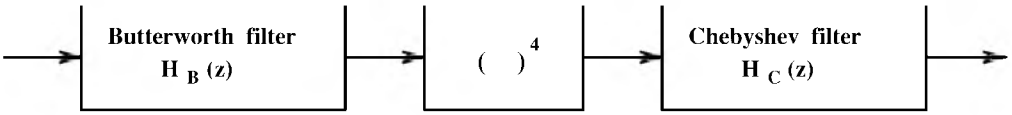

Fig. 6. System model in Experiment 3.

54.7 dB ($\sigma_n^2 = 0.01$). The excess mean-square errors were evaluated after initializing the algorithm with the true values of the unknown system as described in the first experiment in this subsection. It was observed for this experiment that the algorithm diverged for μ greater than 1.5. Even though the excess mean-square error becomes significantly large for $\mu > 0.2$, we observe that the algorithm provided finite values for the steady-state excess mean-square error in the range $0 < \mu \leq 1.5$. The difference in the empirical performance and the heuristic analysis of Section IV-C can be attributed to the approximations employed in the derivations.

B. Experiment 2

The purpose of this experiment was to evaluate the performance of the parallel-cascade NLMS adaptive Volterra filter in a nonstationary system identification problem. The optimum filter coefficients at time n were evolved according to the recursive equation

$$h_{\text{opt},n}(m_1, m_2, m_3, m_4) = h_{\text{opt},n-1}(m_1, m_2, m_3, m_4) + \zeta_n(m_1, m_2, m_3, m_4) \tag{53}$$

where $h_{\text{opt},n}(m_1, m_2, m_3, m_4)$ denotes the optimum coefficients at time equal to n, and $\zeta_n(m_1, m_2, m_3, m_4)$ denotes a zero-mean and white Gaussian process that is symmetric in its arguments and has variance $\sigma_e^2 = 10^{-8}$. The system was initialized with the same coefficients as in (51). The input signal and the adaptive filter structure were identical to that in Experiment 1. Because of its poor performance in the previous experiments, the unnormalized parallel-cascade system was not included in this experiment. The noise level in the desired response signal corresponds to $\sigma_n^2 = 0.01$. The step sizes for the direct-form LMS and the NLMS adaptive algorithms were chosen as 8.3×10^{-7} and 0.074, respectively.

Since we were interested in the steady-state tracking performance of the adaptive filters in this experiment, we initialized the adaptive filter coefficients to match the coefficients of the unknown system. Fig. 5 displays the evolution of the mean-square errors of the adaptive filters. We see from the results that the two algorithms have similar tracking capabilities. The curve for the parallel-cascade system is slightly above that of the direct-form filter. However, the two systems will eventually reach steady-state values that are very close to each other. The more significant spikes in Fig. 5 are from the direct form LMS adaptive filter. We observe a slight increase in the mean-square error over the corresponding steady-state values for the stationary system identification experiment. This increase is due to the contribution from the lag error in the nonstationary environment.

C. Experiment 3

This experiment evaluates the effects of using a fewer number of branches in the parallel-cascade adaptive Volterra filters than the maximum necessary in order to reduce their complexity. Since the usefulness of this procedure is most in approximating nonlinear systems with long memory spans, we consider the identification of a homogeneous fourth-order Volterra system with infinite memory created as a cascade of a fourth-order Butterworth lowpass filter with a memoryless fourth-power operator followed by a fourth-order Chebychev lowpass filter, as shown in Fig. 6. This system is similar to models of satellite communication systems in which the linear filters model the dispersive transmission paths to and from the satellite and the memoryless nonlinearity models amplifiers operating near the saturation region in the satellite [12]. The transfer functions of the lowpass Butterworth and Chebychev filters in Fig. 6 are given by (54) and (55), respectively, shown at the bottom of the page.

In the experiment, we attempt to identify the above system using a memory span of 15 samples. The memory span of the system model was determined by finding the minimum of

$$J(N) = \epsilon(N) + 0.0001N \tag{56}$$

where $\epsilon(N)$ is the modeling error for a system with N-sample memory, and the second term is a penalty term for the increased complexity that accompanies increasing memory lengths. The modeling error was estimated by exciting the system in Fig. 6 with a white Gaussian sequence with zero mean value and unit variance and then finding the time average of the squared error between the output of the model with the reduced memory and the actual system over 20 000 samples. The error $\epsilon(N)$ was obtained by normalizing this average value with the time-averaged value of the square of the output of the actual system.

Fig. 7 displays a semilogarithmic plot of the normalized mean-square error between the outputs of the system and

$$H_B(z) = \frac{(0.2851 + 0.5704z^{-1} + 0.2851z^{-2})(0.2851 + 0.5701z^{-1} + 0.2851z^{-2})}{(1.0 - 0.1024z^{-1} + 0.4475z^{-2})(1.0 - 0.0736z^{-1} + 0.0408z^{-2})} \tag{54}$$

$$H_C(z) = \frac{(0.2025 + 0.288z^{-1} + 0.0.2025z^{-2})(0.2025 + 0.0034z^{-1} + 0.2025z^{-2})}{(1.0 - 1.01z^{-1} + 0.5861z^{-2})(1.0 - 0.6591z^{-1} + 0.1498z^{-2})} \tag{55}$$

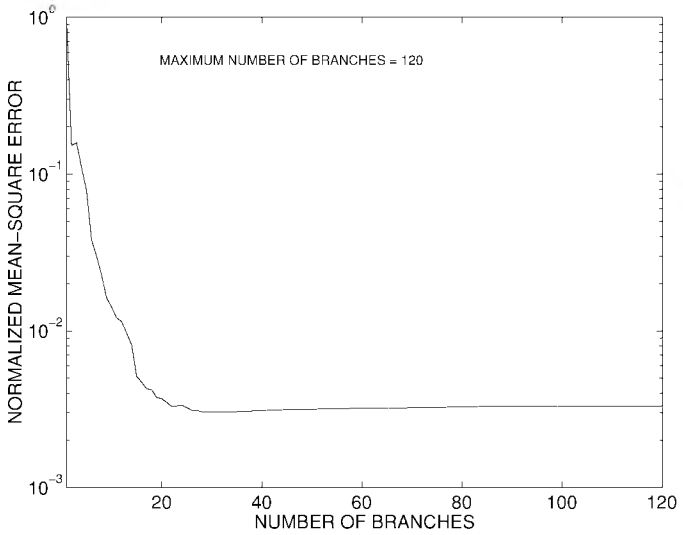

Fig. 7. Mean-square error between the approximated system and theactual system as a function of the number of branches of the parallel-cascade realization of the system in Experiment 3.

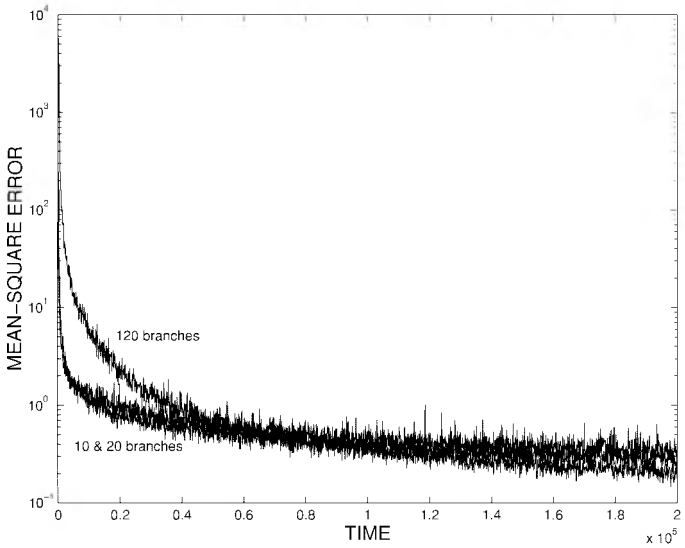

Fig. 8. Mean-square error of the adaptive filters with discarded branches for Experiment 3.

its approximation obtained using a parallel-cascade system with 15-sample memory and reduced number of branches. The approximations were based on a realization involving second-order Volterra filters obtained using the singular value decomposition of the estimated coefficient matrix with 15-sample memory. The input signal to both the original system and its approximation was a white Gaussian sequence with zero mean value and unit variance. The mean-square error at the output was calculated by time averaging the squared differences between the output signal of the approximate system and the actual system over 20 000 samples and then normalizing the result with the time average of the squared output of the actual system. The maximum number of branches required to implement the parallel-cascade system using a 15-sample memory is 120. It is evident from Fig. 7 that very good approximations can be obtained by retaining less than 20 branches in this case.

On the basis of the results described above, we employed the NLMS adaptive filters with 120, 20, and ten branches to identify the unknown system. The step size μ was chosen as 0.1 in all the experiments. Fig. 8 shows the transient behavior of the mean-square estimation error for the three cases. It may appear from Fig. 8 that the systems with fewer branches converge faster. This discrepancy is due to the differences in the initial errors of the three estimates. Since one of the coefficients in each branch is fixed to one, the initializations of the three filters are different, causing a relatively large difference in the initial errors. We observe that the three systems show almost the same convergence behavior after the initial phase. However, the three systems converge to slightly different steady-state values where the system with the least number of branches converge to the highest steady-state value. The steady-state performance of the truncated systems were evaluated by running a single experiment over a long interval of time till the time average of the squared error over successive blocks of 100 000 iterations showed negligible variation. The excess mean-square error evaluated as a time average over 100 000 samples after the system has reached the steady-state is shown in Table III for the three cases. It is clear from the results that the adaptive filter with as few as 20 branches may be adequate in this case. A similar experiment that demonstrates the possible simplifications for a system model similar to that given by (51) is provided in [13].

TABLE III
EXCESS MEAN-SQUARE ERRORS IN EXPERIMENT 3

No. of branches	*excess mse*
120	0.084246
20	0.087432
10	0.092697

VI. CONCLUDING REMARKS

A novel, normalized LMS adaptive filter employing a parallel-cascade structure of truncated Volterra systems was presented in this paper. This algorithm was shown to perform better than the direct-form and unnormalized adaptive parallel-cascade Volterra filters through experimental performance evaluation. Furthermore, it was shown that the complexity of the adaptive filter can be reduced significantly when the coefficient matrix is of low rank. The good characteristics of the NLMS parallel-cascade truncated Volterra filters demonstrated through the experiments makes us believe that the new system is an attractive alternate to currently-available stochastic gradient adaptive truncated Volterra filters in practical applications.

REFERENCES

[1] M. J. Coker and D. N. Simkins, "A nonlinear adaptive noise cancellor," in *Proc. 1980 IEEE Int. Conf. Acoust., Speech, Signal Process.*, Denver, CO, Apr. 1980, pp. 470–473.
[2] T. Koh and E. J. Powers, "Second-order Volterra filtering and its applications to nonlinear system identification," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-33, pp. 1445–1455, Dec. 1985.
[3] V. J. Mathews, "Adaptive polynomial filters," *IEEE Signal Processing Mag.*, vol. 8, pp. 10–26, July 1991.

[4] Y. Lou, C. L. Nikias, and A. N. Venetsanopoulos, "Efficient VLSI array processing structures for adaptive quadratic digital filters," *Circuits Syst. Signal Process.*, vol. 7, no. 2, pp. 253–273, 1988.
[5] S. Marsi and G. L. Sicuranza, "On reduced-complexity approximation of quadratic filters," in *Proc. 27th Asilomar Conf Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 1993, pp. 1026–1030.
[6] J. Nagumo and A. Noda, "A learning method for system identification," *IEEE Trans. Automat. Contr.*, vol. AC-12, pp. 282–287, June 1967.
[7] S. Haykin, *Adaptive Filter Theory*, 3rd ed. Upper Saddle River, NJ: Prentice-Hall, 1996.
[8] R. R. Bitmead and B. D. O. Anderson, "Performance of adaptive estimation algorithms in dependent random environments," *IEEE Trans. Automat. Contr.*, vol. AC-25, pp. 788–794, Aug. 1980.
[9] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*. New York: Wiley, 1980.
[10] W. J. Rugh, *Nonlinear System Theory*. Baltimore, MD: Johns Hopkins Univ. Press, 1981.
[11] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1989.
[12] S. Benedetto, E. Biglieri, and V. Castellini, *Digital Transmission Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1987.
[13] T. M. Panicker, "Parallel-cascade realization of truncated Volterra systems," Ph.D. dissertation, Univ. Utah, Salt Lake City, UT, June 1998.

Thomas M. Panicker was born in Trivandrum, India, in 1962. He received the Bachelor of Technology degree in electronics and communication from the College of Engineering, Trivandrum, in 1985. He received the Master of Technology degree from the Indian Institute of Technology, Delhi, in 1992 and the Ph.D. degree in electrical engineering from the University of Utah, Salt Lake City, in 1998.

After completing one year of a fellowship course at Institute of Armament Technology, Pune, India, he was employed by the Defense Research and Development Organization, Cochin, India, from 1986 to 1990, where he was involved in the design of sonar systems. He is currently employed at Rockwell Semiconductor Systems, San Diego, CA. His current interests cover adaptive signal processing, detection and estimation theory, and channel coding.

V. John Mathews (SM'90) received the B.E. (Hons.) degree in electronics and communication engineering from the University of Madras, Madras, India, in 1980, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Iowa, Iowa City, in 1981 and 1984, respectively.

He joined the Department of Electrical Engineering at the University of Utah, Salt Lake City, in 1985, where he is now a Professor. His research interests span adaptive and nonlinear filtering, image processing, and applications of signal processing techniques in communication systems, and biomedical engineering.

Dr. Mathews is a member of the Digital Signal Processing Technical Committee, the Education Committee, and the Conference Board of the IEEE Signal Processing Society. He is a past associate editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING and is currently on the editorial board of the IEEE SIGNAL PROCESSING LETTERS. He has served or is currently serving on the organization committees of several international technical conferences including as General Chairman of the *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2001*, and the *IEEE DSP Workshop 1998*.

Giovanni L. Sicuranza (SM'85) is Professor of Signal and Image Processing and Head of the Image Processing Laboratory at the Department of Elettrotecnica Elettronica Informatica, University of Trieste, Triests, Italy, where he has given courses on analog circuits, digital circuits, and digital signal and image processing. He was also a teacher in European courses and the organizer of the EURASIP course on Linear and Nonlinear Filtering of Multidimensional Signals in Trieste in 1990. His research interests include multidimensional digital filters, polynomial filters, processing of images and image sequences, image coding, and neural networks for signal processing. He has published a number of papers in international journals and conference proceedings, and he is the co-editor of the book *Multidimensional Processing of Video Signals* (Boston, MA: Kluwer, 1992). He has served as a Project Manager of European ESPRIT and COST research projects and as a consultant to several industries.

Dr. Sicuranza has been a member of the technical committees of numerous international conferences and Chairman of the *VIII European Signal Processing Conference (EUSIPCO-96)* (Trieste, Italy). He is a member of the editorial board of *Signal Processing* and an Associate Editor of *Multidimensional Systems and Signal Processing*. He is currently a member of the EURASIP AdCom and Chair of the Nonlinear Signal and Image Processing Board.