Parallel-Cascade Adaptive Volterra Filters

*T. M. Panicker** *V. J. Mathews** *G.L. Sicuranza***
*Department of Electrical Engineering
University of Utah, Salt Lake City, UT 84112, USA
**DEEI - University of Trieste,
Via A. Valerio, 10, 34127 Trieste, Italy

ABSTRACT

Adaptive truncated Volterra filters using parallel-cascade structures are discussed in this paper. Parallel-cascade realizations implement higher-order Volterra systems as a parallel and multiplicative combination of lower-order Volterra systems. A normalized LMS adaptive filter for parallel-cascade structures is developed and its performance is evaluated through simulation experiments. The experimental results indicate that the normalized LMS parallel-cascade Volterra filter has superior convergence over several competing structures.

1 Introduction

Adaptive least-mean-square (LMS) Volterra filters employing direct form realizations have become popular in recent years [1]. Adaptive parallel-cascade filters for quadratic system models have been presented in [2, 3]. The structure of [2] is not constrained to result in a unique solution to the estimation problem. Consequently, this filter exhibits relatively slow convergence behavior. The work in [3] constrains the filter structure so as to provide convergence to a unique solution. However, this adaptive filter requires appropriate training to select its initial settings. 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. Our adaptive filter 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 an adaptive normalized LMS (NLMS) [4] parallel-cascade filter. This algorithm offers significant performance advantages over previously available algorithms.

2 Parallel-cascade realization

The input-output relationship of a homogeneous and causal pth order Volterra system with N-sample memory can be written compactly using matrix notation as [5]

$$y(n) = \mathbf{X}_{N,l}^T(n)\mathbf{H}_{N,l,p-l}\mathbf{X}_{N,p-l}(n), \qquad (1)$$

where the vector $\mathbf{X}_{N,p_1}(n)$ has $\binom{N+p_1-1}{p_1}$ elements and contains all possible p_1th order product signals of the form $x(n-k_1)x(n-k_2)\cdots x(n-k_{p_1})$ and $0\leq k_1, k_2, \cdots, k_{p_1} \leq N-1$. The matrix $\mathbf{H}_{N,l,p-l}$ contains the coefficients of the pth order Volterra kernel [6] arranged in some orderly manner. Using a singular value decomposition on the coefficient matrix $\mathbf{H}_{N,l,p-l}$, we can write the output of the parallel-cascade structure as

$$\begin{aligned} y(n) &= \sum_{i=1}^{r}\sigma_i[\mathbf{X}_{N,l}^T(n)\mathbf{U}_i][\mathbf{V}_i^T\mathbf{X}_{N,p-l}(n)] \\ &= \sum_{i=1}^{r}\sigma_i y_{l,i}(n) y_{p-l,i}(n), \end{aligned} \qquad (2)$$

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

$$y_{l,i}(n) = \mathbf{X}_{N,l}^T(n)\mathbf{U}_i. \qquad (3)$$

The signal $y_{p-l,i}(n)$ is also defined in a similar manner. In the above equations, r is the rank of the coefficient matrix $\mathbf{H}_{N,l,p-l}$, σ_i's are the non-zero 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. 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 Figure 1.

When $l = p - l = p/2$, it is shown in [7] that the output of the pth order Volterra system can be written as 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.*,

$$y(n) = \sum_{i=1}^{r}\sigma_i (y_{p/2,i}(n))^2, \qquad (4)$$

where

$$y_{p/2,i}(n) = \mathbf{X}_{N,p/2}^T(n)\mathbf{L}_i. \qquad (5)$$

The scalar multipliers σ_i's and the vectors $\mathbf{L}_i$'s can be obtained using an LDL^T decomposition on the coefficient matrix. 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.

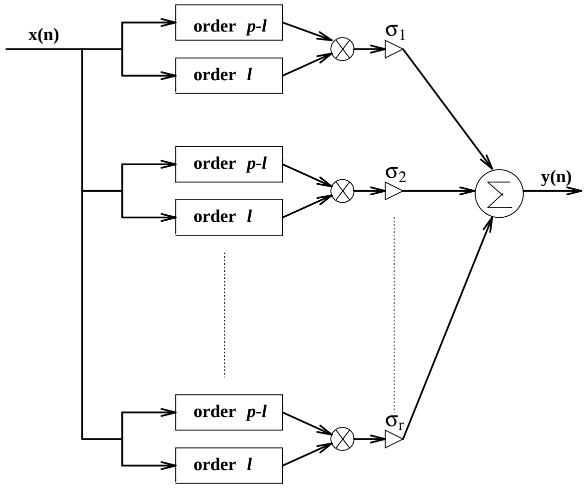

Figure 1: A parallel-cascade realization of a pth order Volterra kernel. Each block represents a homogeneous Volterra system of the order shown within.

3 The adaptive normalized LMS 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

$$\begin{aligned}\hat{d}(n) &= \sum_{i=1}^{r}\sigma_i(n)y_{l,i}(n)y_{p-l,i}(n)\\ &= \sum_{i=1}^{r}\sigma_i(n)[\mathbf{X}_{N,l}^T(n)\mathbf{U}_i(n)]\\ &\quad\times[\mathbf{V}_i^T(n)\mathbf{X}_{N,p-l}(n)], \qquad (6)\end{aligned}$$

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

$$e(n) = d(n) - \hat{d}(n), \qquad (7)$$

denote the estimation error at time n.

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

$$\begin{aligned}d(n) &= \sum_{i=1}^{r}(\sigma_i(n)+\delta\sigma_i(n))[\mathbf{X}_{N,l}^T(n)(\mathbf{U}_i(n)+\delta\mathbf{U}_i(n)]\\ &\quad\times[(\mathbf{V}_i(n)+\delta\mathbf{V}_i(n))^T\mathbf{X}_{N,p-l}(n)], \qquad (8)\end{aligned}$$

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

$$\sum_{i=1}^{r}(\delta\sigma_i(n))^2+\sum_{i=1}^{r}\parallel\delta\mathbf{U}_i(n)\parallel^2+\sum_{i=1}^{r}\parallel\delta\mathbf{V}_i(n)\parallel^2, \quad (9)$$

subject to the equality in (8). Since the above solution may in general result in erratic coefficient behavior because of the presence of the noise in the desired response signal, the normalized LMS parallel-cascade adaptive 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

$$\sigma_i(n+1) = \sigma_i(n)+\mu\delta\sigma_i(n), \qquad (10)$$

$$\mathbf{U}_i(n+1) = \mathbf{U}_i(n)+\mu\delta\mathbf{U}_i(n) \qquad (11)$$

and

$$\mathbf{V}_i(n+1) = \mathbf{V}_i(n)+\mu\delta\mathbf{V}_i(n), \qquad (12)$$

where μ is a small positive constant in the range $0 < \mu < 2$. The constrained optimization problem can be solved using the method of Lagrange multipliers. We define a cost function

$$\begin{aligned}\mathbf{J}(n) &= \sum_{i=1}^{r}(\delta\sigma_i(n))^2+\sum_{i=1}^{r}\parallel\delta\mathbf{U}_i(n)\parallel^2+\\ &+ \sum_{i=1}^{r}\parallel\delta\mathbf{V}_i(n)\parallel^2+\lambda[d(n)-\sum_{i=1}^{r}(\sigma_i+\delta\sigma_i(n))\\ &\quad\times[\mathbf{U}_i(n)+\delta\mathbf{U}_i(n)]^T\mathbf{X}_{N,l}(n)\\ &\quad\times[\mathbf{V}_i(n)+\delta\mathbf{V}_i(n)]^T\mathbf{X}_{N,p-l}(n)], \qquad (13)\end{aligned}$$

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, and equating them to zero we get $3r$ equations to solve for $\delta\sigma_i(n)$, $\delta\mathbf{U}_i(n)$ and $\delta\mathbf{V}_i(n)$. These $3r$ equations and the constraint equation given by (8) gives $(3r+1)$ coupled nonlinear equations to solve for $\delta\sigma_i(n)$, $\delta\mathbf{U}_i(n)$, $\delta\mathbf{V}_i(n)$ and λ.

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, and also assumes that the level of measurement noise in the desired response signal and the level of nonstationarity in the operating environment are relatively low. However, our experience is that the resulting adaptive filter exhibits good convergence properties under a variety of noise conditions.

3.1 The approximate solution

To find an approximate solution, we assume 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 $\mathbf{V}_i(n)+\delta\mathbf{V}_i(n)$ with $\sigma_i(n)$, $\mathbf{U}_i(n)$ and $\mathbf{V}_i(n)$, respectively when necessary. Taking the partial derivative of (13) with respect to $\delta\sigma_i(n)$ and equating it to zero, we get,

$$2\delta\sigma_i(n) = \lambda\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n) \qquad (14)$$

for $i = 1,\cdots,r$. The signals $\tilde{y}_{l,i}(n)$ and $\tilde{y}_{p-l,i}(n)$ are the outputs at the ith branch given by

$$\tilde{y}_{l,i}(n) = \mathbf{X}_{N,l}^T(n)[\mathbf{U}_i(n)+\delta\mathbf{U}_i(n)] \qquad (15)$$

and

$$\tilde{y}_{p-l,i}(n) = \mathbf{X}_{N,p-l}^T(n)[\mathbf{V}_i(n) + \delta\mathbf{V}_i(n)], \qquad (16)$$

respectively. Multiplying (14) with $\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n)$, we get,

$$2\delta\sigma_i(n)\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n) = \lambda\tilde{y}_{l,i}^2(n)\tilde{y}_{p-l,i}^2(n). \qquad (17)$$

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 the following equation:

$$\sum_{i=1}^{r} 2[\sigma_i(n) + \delta\sigma_i(n)]\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n) = \sum_{i=1}^{r} 2\sigma_i(n)\tilde{y}_{l,i}(n)\tilde{y}_{p-l,i}(n) + \lambda\tilde{y}_{l,i}^2(n)\tilde{y}_{p-l,i}^2(n)(18)$$

Since the left-hand-side uses the solution to the constrained optimization problem for all variables, we note that the left-hand-side is identical to $2d(n)$, 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 (18) with $y_{l,i}(n)$ and $y_{p-l,i}(n)$ respectively. This approximation and the earlier observation about the left-hand-side of (18) lead to

$$2d(n) = 2\hat{d}(n) + \lambda\sum_{i=1}^{r} y_{l,i}^2(n)y_{p-l,i}^2(n). \qquad (19)$$

Transposing $2\hat{d}(n)$ to the left-hand-side of the above equation, we get the following simplified result:

$$2e(n) = \lambda\sum_{i=1}^{r} y_{l,i}^2(n)y_{p-l,i}^2(n). \qquad (20)$$

Similarly, taking the partial derivative of (13) with respect to $\delta\mathbf{U}_i(n)$ and $\delta\mathbf{V}_i(n)$ and proceeding in the same manner, we arrive at the equations

$$2e(n) = \lambda\sum_{i=1}^{r} \sigma_i^2(n)y_{p-l,i}^2(n) \parallel \mathbf{X}_{N,l}(n) \parallel^2, \qquad (21)$$

and

$$2e(n) = \lambda\sum_{i=1}^{r} \sigma_i^2(n)y_{l,i}^2(n) \parallel \mathbf{X}_{N,p-l}(n) \parallel^2, \qquad (22)$$

respectively. We can obtain an expression for λ from (20), (21) and (22) as

$$\lambda(n) = \frac{6e(n)}{P_\sigma(n) + P_u(n) + P_v(n)}, \qquad (23)$$

where

$$P_\sigma(n) = \sum_{i=1}^{r} y_{l,i}^2(n)y_{p-l,i}^2(n), \qquad (24)$$

$$P_u(n) = \sum_{i=1}^{r} \sigma_i^2(n)y_{p-l,i}^2(n) \parallel \mathbf{X}_{N,l}(n) \parallel^2 \qquad (25)$$

and

$$P_v(n) = \sum_{i=1}^{r} \sigma_i^2(n)y_{l,i}^2(n) \parallel \mathbf{X}_{N,p-l}(n) \parallel^2 . \qquad (26)$$

We have included the time index n for the Lagrangian multiplier in (23), 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 after taking the derivative of (13) with respect to $\delta\sigma_i(n)$, $\delta\mathbf{U}_i(n)$ and $\delta\mathbf{V}_i(n)$ respectively and then equating to zero. The adaptive normalized LMS parallel-cascade filter is obtained by substituting the solutions so obtained in the update equations (10), (11) and (12) respectively. The relevant equations are

$$\sigma_i(n+1) = \sigma_i(n) + \frac{3\mu}{P_\sigma(n) + P_u(n) + P_v(n)} \times e(n)y_{l,i}(n)y_{p-l,i}(n), \qquad (27)$$

$$\mathbf{U}_i(n+1) = \mathbf{U}_i(n) + \frac{3\mu}{P_\sigma(n) + P_u(n) + P_v(n)} \times e(n)\sigma_i(n)y_{p-l,i}(n)\mathbf{X}_{N,l}(n) \qquad (28)$$

and

$$\mathbf{V}_i(n+1) = \mathbf{V}_i(n) + \frac{3\mu}{P_\sigma(n) + P_u(n) + P_v(n)} \times e(n)\sigma_i(n)y_{l,i}(n)\mathbf{X}_{N,p-l}(n). \qquad (29)$$

In the actual realizations of the adaptive filter, the normalization factor $P_\sigma(n) + P_u(n) + P_v(n)$ was replaced by a smoothed version obtained using a single-pole low pass filter as

$$\xi(n) = \alpha\xi(n-1) + (1-\alpha)[P_\sigma(n) + P_u(n) + P_v(n)], \quad (30)$$

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 ξ and $\xi(n+1)$ closer to each other than 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.

It is quite straightforward to derive similar update equations for the LDL^T decomposition of the coefficient matrix.

4 Experimental results

Several experiments were conducted to evaluate the performance of the algorithm developed in this paper [7]. One of those experiments that illustrate the superior performance of NLMS algorithm in parallel-cascade form with colored Gaussian input is included in this paper. The mean-squared-error curves presented in this paper are averages obtained over one hundred independent simulations, and were further smoothed by time-averaging over fifty consecutive samples. The plots presented were obtained by subsampling the time-averaged data by a factor of fifty.

The purpose of this experiment is to evaluate the performance of the NLMS parallel-cascade adaptive filter in a stationary system identification problem. The system coefficients were chosen as

$$h(k_1, k_2, k_3, k_4) = \frac{20.88}{2\pi[1.5^4 + a_1^4 + a_2^4 + a_3^4 + a_4^4]^{3/4}} + u(k_1, k_2, k_3, k_4), \quad (31)$$

where $a_i = (k_i - 1)$, $0 \leq k_1, k_2, k_3, k_4 \leq 4$ and u is a random variable uniformly distributed between -0.1 and +0.1 that is also symmetric in its indices k_1, k_2, k_3 and k_4. The number of branches in the parallel-cascade realization of the above filter is 15. The input signal was generated as the output of a linear system with input-output relationship given by

$$x(n) = 0.6x(n-1) + 0.8\epsilon(n), \quad (32)$$

where $\epsilon(n)$ was a white Gaussian signal with unit variance and zero mean. The desired response signal was generated by processing the signal with the fourth-order Volterra system of (31) and then corrupting the output with an uncorrelated white Gaussian noise sequence with zero mean value and variance 0.01. Figure 2 displays a plot of the mean-squared estimation error signals obtained using the direct form LMS, parallel-cascade LMS and parallel-cascade NLMS adaptive filters. The step sizes for the three methods were selected so as to get approximately 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 one hundred ensembles. It can be seen from the figure that the NLMS algorithm developed in this paper converges significantly faster than the direct LMS and the unnormalized LMS using the LDL^T decomposition. All the spikes in the figure are due to the direct form LMS adaptive filter, indicating that it is operating very close to the stability bound. Even though the results are not shown here, adaptive NLMS direct form filter performed poorer than all the systems shown in the figure.

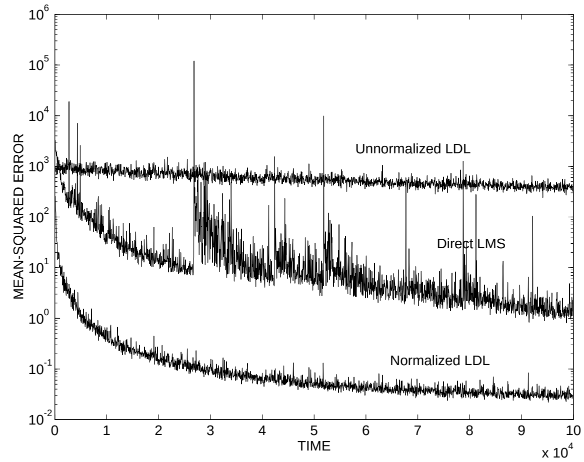

Figure 2: Mean-squared error of the adaptive filters for colored Gaussian input and measurement noise with variance 0.01.

5 Concluding remarks

A normalized LMS adaptive filter employing the parallel-cascade structure of truncated Volterra systems was presented in this paper. The algorithm was experimentally shown to perform better than the direct form and unnormalized parallel-cascade adaptive Volterra filters. 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.

Acknowledgment. This work was supported by NATO Grant CRG. 950379.

References

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

[2] Y. Lou, C. L. Nikias, A. N. Venetsanopoulos, "Efficient VLSI array processing structures for adaptive quadratic digital filters," *J. Circuits Systems Signal Process.*, Vol. 7, No. 2, pp. 253-273, 1988.

[3] S. Marsi and G. L. Sicuranza, "On reduced-complexity approximation of quadratic filters," *Proc. 27th Asilomar Conf. Signals, Systems and Computers*, pp. 1026-1030, Pacific Grove, CA, November 1993.

[4] J. Nagumo and A. Noda, "A learning method for system Identification," *IEEE Trans. on Automatic Control*, Vol. AC 12, No.3, pp. 282-287, June 1967.

[5] T. M. Panicker and V. J. Mathews, "Parallel-cascade realizations and approximations of truncated Volterra systems," *Proc. ICASSP-96*, Atlanta, Georgia.

[6] M. Schetzen, *The Volterra and Wiener Theories of Nonlinear Systems*, New York: Wiley, 1980.

[7] T. M. Panicker, V. J. Mathews and G. L. Sicuranza, "Adaptive parallel-cascade truncated Volterra filters," *submitted to IEEE Trans. on Signal Processing.*