NONLINEAR MULTICHANNEL ACTIVE NOISE CONTROL USING PARTIAL UPDATES

Giovanni L. Sicuranza

DEEI - University of Trieste
Via A. Valerio, 10 - 34127 Trieste, Italy

Alberto Carini

STI - University of Urbino
Piazza della Repubblica, 13 - 61029 Urbino, Italy

ABSTRACT

We consider in this paper a model for nonlinear multichannel active noise controllers based on truncated Volterra filters. A previously proposed approximate Affine Projection (AP) adaptation algorithm is revisited with the aim of reducing its implementation complexity by applying suitable partial update strategies. The experimental results confirm that remarkable computational reductions are achieved with limited degradations of the convergence behavior.

1. INTRODUCTION

Methods for active noise control have been intensively studied in the last three decades and have already provided useful applications in vibration and acoustic noise control tasks [1]. To spatially extend the silenced region, a multichannel approach can be applied using sets of reference sensors, actuator sources and error microphones. A general scheme describing the so-called feed-forward approach is shown in Fig. 1, where I input sensors are used to collect the corresponding input signals. In the controller, any input i, $1 \leq i \leq I$, is usually connected to any output j, $1 \leq j \leq J$, with an FIR filter. The controller computes J output signals which are propagated to the K error sensors. Since the input signals filtered by the impulse responses of the secondary paths are used, the coefficients of the FIR filters are updated by means of the so-called Filtered-X versions of the standard adaptation algorithms. The main drawbacks of the multichannel approach are the complexity of the coefficient updates, the data storage requirements and the slow convergence of the adaptive algorithms [2]. In order to improve the convergence speed, Affine Projection (AP) algorithms have been used [3] in place of the usual LMS algorithms but at the expense of a further, even though limited, increment of the complexity of updates. It is thus necessary to resort to appropriate techniques for the reduction of the update complexity. This requirement becomes even more stringent if nonlinearities are considered. Presently, most of the studies presented in the literature refer to linear models

This work was supported in part by the MIUR under Grant PRIN 2004092314.

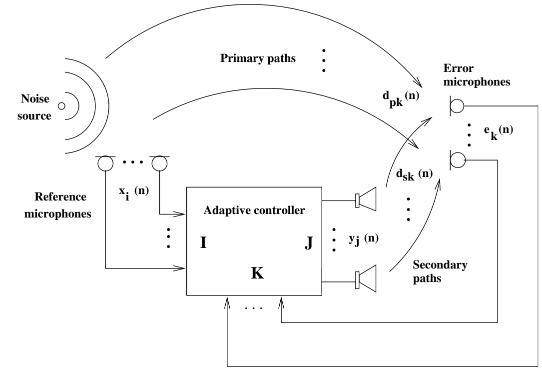

Fig. 1. Multichannel active noise control

and linear time-invariant controllers represent the state-of-the-art in this field. On the other hand, nonlinear modeling techniques may bring new insights and suggest new developments offering an increase of performance. In fact, it is often recognized that nonlinearities affect actual applications [4, 5]. To keep low the updating complexity without negatively affecting the performance of adaptation algorithms it is necessary to choose an appropriate model. Our choice is for truncated Volterra filters which can efficiently model a large class of nonlinear systems [6]. These filters share with the linear ones the property that their output is linear with respect to the filter coefficients. As a consequence, the adaptation algorithms derived for linear filters can be appropriately extended to the nonlinear case. Moreover, truncated Volterra filters can be implemented in the form of multichannel filter banks involving FIR filters. This possibility is granted by the so-called diagonal representation [7] that allows a truncated Volterra filter to be described by the "diagonal" entries of its kernels. This representation has been used in [5] to derive a Filtered-X LMS algorithm for single-channel controllers equipped with Volterra filters.

The model we are proposing here for nonlinear multichannel active noise control is based on Volterra filters connecting any input i to any output j. Such a system has been implemented in [8] where an approximate AP algorithm has been derived. This algorithm is revisited in this paper by exploiting partial update techniques that can offer remarkable reductions in the implementation complexity at a limited expense in the convergence characteristics.

Various techniques have been proposed in the literature to keep low the implementation complexity of adaptive FIR filters having long impulse responses. Most of them can be usefully applied to the Filtered-X algorithms, too, especially in the multichannel situations. A first approach is based on the so-called interpolated FIR filters [9], where a few impulse response samples are removed and then their values are derived using some interpolation scheme. However, the success of this implementation is based on the hypothesis that practical FIR filters have an impulse response with a smooth predictable envelope. Another set of well-established techniques is based on selective partial updates (PU) where selected blocks of filter coefficients are updated at every iteration in a sequential or periodic manner [10] or by using an appropriate selection criterion [11]. Finally, a recently proposed approach is based on data-selective updates which are sparse in time. This approach can be suitably described in the framework of the set-membership filtering (SMF) where a filter is designed to achieve a specified bound on the magnitude of the output error [12]. The rational for using such techniques for nonlinear controllers equipped with $I \times J$ Volterra filters is that their implementation based on the diagonal representation involves FIR filters. On the other hand, the minimization of the K errors at the silenced zone involve a large number of coefficients so that updating rules are usually computationally intensive and thus strongly benefit of partial update strategies.

The outline of this paper is the following: in Section 2, the derivation of the approximate Filtered-X AP algorithm of [8] is briefly described. In Section 3 appropriate partial update techniques are applied. Section 4 includes some simulation results and final remarks are given in Section 5.

2. THE REFERENCE UPDATING ALGORITHM

In this section we briefly summarize the approximate AP algorithm described in [8]. For sake of simplicity, each filter connecting any input i to any output j is initially assumed to be a homogeneous quadratic filter implemented in the form of a filter bank. To reduce the computational complexity, the quadratic kernel is described by means of the so-called triangular representation [6, p. 35]. Each channel of the filter bank contains an FIR filter formed with the coefficients of the corresponding upper diagonal of the quadratic kernel. As a consequence, any output $y_j(n)$ from the multichannel quadratic controller of Fig. 1 can be written as

$$y_j(n) = \sum_{i=1}^{I} \mathbf{h}_{i,j}^T(n)\mathbf{x}_i(n). \tag{1}$$

The vectors $\mathbf{h}_{i,j}$ are formed by $Q_i = \sum_{m=1}^{M_i}(N_i - m + 1)$ elements as shown in [8], where N_i is the memory length of the homogeneous quadratic filter and M_i is the number of channels actually used, with $M_i \leq N_i$. The vectors $\mathbf{x}_i(n)$ are formed with the corresponding products of two input samples collected at the microphone i, $1 \leq i \leq I$. The Filtered-X AP algorithm of order L minimizes the coefficient variations within the constraint that the last L *a posteriori* errors are set to zero. The cost function to be minimized is given by

$$\varepsilon = \sum_{k=1}^{K} \mathbf{e}_k^T(n)\mathbf{e}_k(n), \tag{2}$$

where $\mathbf{e}_k(n) = [e_{k,1}(n)\ e_{k,2}(n) \cdots e_{k,L}(n)]^T$ is the vector collecting the errors at the K error sensors. Using the derivations in [8], the approximate updating relationship for each quadratic filter in the controller is written as

$$\mathbf{h}_{i,j}(n+1) = \mathbf{h}_{i,j}(n) -$$

$$\sum_{k=1}^{K} \mu_k \mathbf{G}_{i,j,k}(n) \left[\mathbf{G}_{i,j,k}^T(n)\mathbf{G}_{i,j,k}(n) + \delta\mathbf{I}\right]^{-1} \mathbf{e}_k(n). \tag{3}$$

The step sizes μ_k are the parameters that control both the convergence rate and the stability of the algorithm. The $Q_i \times L$ matrices $\mathbf{G}_{i,j,k}(n)$ are formed with the input signals filtered by the impulse responses $s_{k,j}(n)$ of the secondary paths

$$\mathbf{G}_{i,j,k}(n) = [s_{k,j}(n) * \mathbf{x}_i(n)\ \cdots$$
$$\cdots\ s_{k,j}(n-L+1) * \mathbf{x}_i(n-L+1)]\,. \tag{4}$$

The $L \times L$ inverse matrices in (3) actually alleviate the convergence difficulties due to the correlations existing in the input vectors to the multichannel filter banks implementing the Volterra filters. In fact, since the entries in these vectors are given in form of products of input samples, correlations exist among them even when the input signals are white [6, p. 253]. Moreover, the small positive constant δ is used to avoid possible numerical instabilities. The computation of the inverse of the matrices is required at each time n. Even though this step is often a critical one, it should be noted that for Filtered-X AP algorithms of low orders, *i. e.* with $L = 2, 3, 4$, direct matrix inversion is still an affordable task. Within these conditions, the complexity of the Filtered-X AP algorithm is $O(KLQ_i)$ per sample for each quadratic filter in the controller. It can also be noted that while the Filtered-X AP algorithm can be applied to quadratic filters characterized by full triangular representations by simply setting $M_i = N_i$, using a smaller number of channels often permits one to still obtain good adaptation performance with a remarkably reduced computational complexity. Moreover, in case of a non-homogeneous quadratic filter, the linear term can be considered as an additional channel. The vectors $\mathbf{h}_{i,j}(n)$ and $\mathbf{x}_i(n)$ are modified by inserting on the top the coefficients of the linear filter and the samples of the input signal, respectively. As a consequence, the updating rule of (3) still applies. In a similar way we can deal with Volterra kernels of any order p. In fact, if the pth-order kernel is represented as a sampled hypercube of the same order, the diagonal representation [7] implies a change of Cartesian coordinates to coordinates that are aligned along the diagonals of the hypercube. In this way, the Volterra filter can be again represented in the form of a filter bank where each filter corresponds to a diagonal of the hypercube.

3. THE PARTIAL UPDATE METHODS

We describe here the partial update strategies applied to the updating rule (3).

The *Partial Error* (PE) technique consists in using sequentially at each iteration only one of the K error vectors $\mathbf{e}_k(n)$ at a time in place of their combination as in (3). This method has been used in [10] for linear multichannel controllers equipped with Filtered-X LMS algorithms. This updating rule requires approximately $\frac{1}{K}$ operations with respect to the original one. This advantage is obtained at the expense of the convergence speed, even though in our case we have not experienced reductions proportional to the decrease of the computational load, as noted in [10].

In the *Partial Error - Partial Update* $(PE - PU)$ case, each measurement of the error vectors $\mathbf{e}_k(n)$ is used to adapt a block of coefficients. According to the multichannel filter bank implementation of the Volterra filters, the natural choice here is to consider as a block the elements of any of the kernel diagonals of each Volterra filter and thus sequentially adapt them. The procedure is repeated for any of the Volterra filters in the controller. With reference to the quadratic case, the updating equation becomes

$$\mathbf{h}_{m,i,j}(n+1) = \mathbf{h}_{m,i,j}(n) -$$

$$\mu_k \mathbf{G}_{m,i,j,k}(n) \left[\mathbf{G}^T_{m,i,j,k}(n)\mathbf{G}_{m,i,j,k}(n) + \delta\mathbf{I}\right]^{-1} \mathbf{e}_k(n), \tag{5}$$

where $\mathbf{h}_{m,i,j}(n)$ is the vector formed with the $N_i - m + 1$ coefficients of the m-th channel, with $1 \leq m \leq M_i$, and $\mathbf{G}_{m,i,j,k}(n)$ are the corresponding partitions of $\mathbf{G}_{i,j,k}(n)$. The complexity of this updating rule is reduced by another factor depending on the number of blocks used.

The previous updating rule can be finally modified using the concept of *Set-Membership Filtering* (SMF). Forcing the last L *a posteriori* errors to be zero, the diagonal of each Volterra kernel in the controller is updated according to the following rule

$$\mathbf{h}_{m,i,j}(n+1) = \begin{cases} \text{right term of (5)} & \text{if } |e_{k,1}(n)| > e_b \\ \mathbf{h}_{m,i,j}(n) & \text{otherwise} \end{cases} \tag{6}$$

for a suitable choice of the threshold error e_b [12].

4. SIMULATION RESULTS

We compare in this section some simulation results obtained with the reference algorithm in (3) and the $PE-PU-SMF$ algorithm in (6). As shown in [4] for the single-channel case, a nonlinear controller is beneficial if the secondary path is modeled as a non-minimum-phase FIR filter and the input signal is a nonlinear and deterministic process of chaotic nature. Such a noise can be efficiently modeled by a second-order white and predictable nonlinear process as the logistic noise generated by the equation

$$\xi(n+1) = \lambda\xi(n)[1 - \xi(n)], \tag{7}$$

with $\lambda = 4$. In the multichannel case we are considering here, the controller has one input $(I = 1)$ and two outputs $(J = 2)$. Two error microphones $(K = 2)$ are used. The noise source is the logistic chaotic noise of (7) with $\xi(0) = 0.9$. The nonlinear process has been normalized in order to have unit signal power $x(i) = \xi(i)/\sigma_\xi$ and a random noise has been added at the error microphones so that the SNR is equal to $30dB$. The primary paths are modeled by two FIR filters with 8 taps while the secondary paths are modeled by four non-minimum-phase FIR filters with 5 taps. The system is identified using two second-order Volterra filters with linear and quadratic parts of memory length $N_i = 10$. The quadratic kernels contain only two channels $(M_i = 2)$ corresponding to the principal diagonal and the adjacent one in the triangular representation. The constant δ has been set equal to 10^{-3}. The step sizes of the linear and quadratic parts have been fixed so that the residual errors in the experiments (a) and (b) in Figure 2 are close as much as possible. This figure plots the ensamble average of the mean attenuation at the error microphones for 50 runs of the simulation system using (a) the updating rule of (3) and the $PE - PU - SMF$ method of (6) with (b) quadratic and (c) linear controllers. The four curves refer to different values of the affine projection order L. Three blocks of lengthes 10, 10, 9 have been used for each Volterra filter. The bound e_b has been chosen equal to $\sqrt{5\sigma_n^2}$, where σ_n^2 is the variance of the additional noise [12], and thus in our experiments $e_b = 0.0707$. It can be seen that the convergence speed is slightly decreased but the computational load is approximately reduced by a factor 2×6, since K=2 and we use 3 coefficient blocks for each quadratic filter, and then by an additional factor ranging from $\simeq 2$ to $\simeq 13$ according to the SMF strategy, as shown in Table I. This Table reports the average number of updates and the residual error e_r for the $PE - PU - SMF$ algorithm with fixed error bounds e_b and AP orders $L = 2$ and $L = 4$. By increasing e_b, remarkable reductions in the number of updates can be observed at the expense of limited increases of the residual errors. Finally, Figure 2 (c) shows for comparison the results of the $PE - PU - SMF$ algorithm for a controller

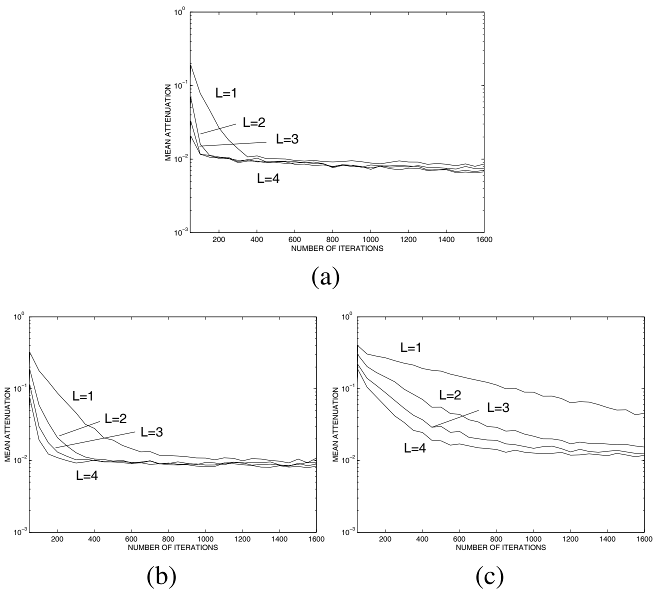

Fig. 2. Mean attenuation at the error microphones using (a) the reference method (b) the $PE - PU - SMF$ method, (c) the $PE - PU - SMF$ method with linear controllers.

Table 1. Average number of updates N_u in 1600 iterations and residual errors e_r for the $PE - PU - SMF$ algorithm.

e_b	L	N_u	e_r	L	N_u	e_r
0.0707	2	851	0.0090	4	794	0.0080
0.1000	2	599	0.0091	4	530	0.0084
0.2000	2	188	0.0098	4	124	0.0093

equipped with two FIR filters of equivalent complexity, *i. e.* 29 coefficients, each one subdivided in 3 blocks of 10, 10, 9 coefficients, respectively. The step size is the same as for the linear part of the quadratic case of Figure 2 (b) . Figure 2 (c) confirms the results of [4] where it is shown the advantage of using a nonlinear control model when the input signals are deterministic processes of chaotic nature.

5. CONCLUDING REMARKS

In this paper it has been shown how partial update techniques can be applied to reduce the computational load of an approximate AP adaptation algorithm for multichannel nonlinear active noise control. These strategies can be used for controllers equipped with any order of Volterra filters. Finally, it is worth mentioning that the ongoing research on this topic recently allowed us to prove that filtered-X AP algorithms provide a biased estimate of the minimum MSE solution. Nevertheless, the estimation errors for the algorithms presented here have been shown to be very low.

6. REFERENCES

[1] P. A. Nelson and S. J. Elliott, *Active Control of Sound*, 3rd ed. London: Academic Press, 1995.

[2] S. C. Douglas, “Fast implementations of the Filtered-X LMS and LMS algorithms for multichannel active noise control,” *IEEE Trans. Speech Audio Processing*, vol. 7, pp. 454-465, July 1999.

[3] M. Bouchard, “Multichannel affine and fast affine projection algorithms for active noise control and acoustic equalization systems,” *IEEE Trans. Speech Audio Processing*, vol. 11, pp. 54-60, Jan. 2003.

[4] P. Strauch and B. Mulgrew, “Active control of nonlinear noise processes in a linear duct,” *IEEE Trans. Signal Processing*, vol. 46, No. 9, pp. 2404-2412, Sept. 1998.

[5] L. Tan and J. Jiang, “Adaptive Volterra filters for active noise control of nonlinear processes,” *IEEE Trans. Signal Processing*, vol. 49, pp. 1667-1676, Aug. 2001.

[6] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*, New York: Wiley, 2000.

[7] G. M. Raz and B. D. Van Veen, “Baseband Volterra filters for implementing carrier based nonlinearities,” *IEEE Trans. Signal Processing*, vol. 46, pp. 103-114, Jan. 1998.

[8] G. L. Sicuranza and A. Carini, “Filtered-X Affine Projection Algorithm for Multichannel Active Noise Control Using Second-order Volterra Filters.” *IEEE Signal Processing Lett.*, vol. 11, pp. 853-857, Nov. 2004.

[9] Y. Neuvo, D. Cheng-Yu and S. K. Mitra, “Interpolated Finite Impulse Response Filters,” *IEEE Trans. Acoust., Speech, Signal Processing*, vol. 32, pp. 563-570, June 1984.

[10] S. C. Douglas, “Adaptive Filters Employing Partial Updates,” *IEEE Trans. Circuits Syst. II*, vol. 44, pp. 209-216, March 1997.

[11] K. Doğançay and O. Tanrikulu, “Adaptive Filtering Algorithms With Selective Partial Updates,”*IEEE Trans. Circuits Syst. II*, vol. 48, pp. 762-769, Aug. 2001.

[12] S. Werner and P. S. R. Diniz, “Set-Membership Affine Projection Algorithm,” *IEEE Signal Processing Lett.*, vol. 8, pp. 231-235, Aug. 2001.