Filtered-X Affine Projection Algorithms for Active Noise Control Using Volterra Filters

Alberto Carini

Institute of Science and Information Technologies, University of Urbino, 61029 Urbino, Italy
Email: carini@sti.uniurb.it

Giovanni L. Sicuranza

Department of Electrical, Electronic and Computer Engineering, University of Trieste, 34127 Trieste, Italy
Email: sicuranza@univ.trieste.it

We consider the use of adaptive Volterra filters, implemented in the form of multichannel filter banks, as nonlinear active noise controllers. In particular, we discuss the derivation of filtered-X affine projection algorithms for homogeneous quadratic filters. According to the multichannel approach, it is then easy to pass from these algorithms to those of a generic Volterra filter. It is shown in the paper that the AP technique offers better convergence and tracking capabilities than the classical LMS and NLMS algorithms usually applied in nonlinear active noise controllers, with a limited complexity increase. This paper extends in two ways the content of a previous contribution published in *Proc. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing (NSIP '03)*, Grado, Italy, June 2003. First of all, a general adaptation algorithm valid for any order L of affine projections is presented. Secondly, a more complete set of experiments is reported. In particular, the effects of using multichannel filter banks with a reduced number of channels are investigated and relevant results are shown.

Keywords and phrases: active noise control, adaptive Volterra filters, affine projection algorithms.

1. INTRODUCTION

Methods for active noise control are nowadays intensively studied and have already provided promising applications in vibration and acoustic noise control tasks. The initial activities originated in the field of control engineering [1, 2], while in recent years a signal processing approach has been successfully applied. This approach strongly benefited of the advances in electroacoustic transducers, flexible digital signal processors, and efficient adaption algorithms [3, 4]. The technique used in a single-channel active noise controller is based on the destructive interference in a given location of the noise produced by a primary source and the interfering signal generated by a secondary source.

Most of the studies presented in the literature refer to linear models, while it is often recognized that nonlinear effects can affect actual applications [5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. Such effects may arise from the behavior of the noise source which rather than a stochastic process may be depicted as a nonlinear deterministic noise process, sometimes of chaotic nature. Moreover, the primary path may exhibit a nonlinear behavior thus motivating the use of a nonlinear controller.

Recently, a model for a nonlinear controller based on Volterra filters [15] has been presented in [9] with interest-

ing results. The model actually exploits the so-called diagonal representation introduced in [16]. This representation allows a truncated Volterra system to be described by the "diagonal" entries of its kernels. In fact, if the pth-order kernel is represented as a sampled hypercube of the same order, the diagonal representation implies the change of the Cartesian coordinates to coordinates that are aligned along the diagonals of the hypercube. In this way, the Volterra filter can be represented in the form of a filter bank, where each filter corresponds to a diagonal of the hypercube. This representation is particularly useful for processing carrier-based input signals, since the frequency content of the output signal is directly related to the frequency response of the diagonal elements of the kernel. It has been exploited in the derivation of efficient implementations of Volterra filters for processing carrier-based input signals using fast convolution techniques [16].

A similar representation has been used in [17, 18] to define the so-called simplified Volterra filters (SVF). Even though this model can be applied to kernels of arbitrary order, the simplest example of a Volterra filter, that is, the homogeneous quadratic filter, has been specifically considered in [17, 18] with reference to the acoustic echo cancellation problem. An SVF is still implemented as a filter bank, but here the stress is on the fact that, according to

the characteristics of the measured second-order kernel, it is possible to use a reduced number of active channels. In fact, it has been noted that the relevance of the quadratic kernel elements is strongly decreasing moving far from the main diagonal. Therefore, remarkable savings in implementation complexity can be achieved. It is worth noting that similar behaviors have been observed also in other real-world nonlinear systems.

The other relevant idea in [17, 18] is that, exploiting the SVF structure, it is possible to extend the affine projection (AP) adaptation algorithm, originally proposed by Ozeki and Umeda [19] for linear filters, to quadratic filters. It has been shown that the AP technique offers also for quadratic filters better convergence and tracking capabilities than the classical LMS and NLMS.

Based on these premises, we present in this paper a couple of novel filtered-X AP (F-X AP) algorithms for nonlinear active noise control. The derivation of these algorithms is given according to the SVF model of a homogeneous quadratic filter. Extensions to higher-order Volterra filters can be obtained using the general diagonal representation mentioned above. The advantages given by the AP algorithms can be appreciated when changes in the characteristics of the noise source or of the room acoustics occur. It is also shown that the complexity increase with respect to the F-X LMS algorithm may be relatively small.

The outline of our paper is the following. In Section 2, the quadratic model used for the nonlinear active noise control is briefly described. The derivation of the novel F-X AP algorithms for homogeneous quadratic filters is presented in Section 3. Results of extensive computer simulations are presented in Section 4 for typical nonlinear situations and a few concluding remarks are given in Section 5.

2. MODELING THE NONLINEAR ACTIVE NOISE CONTROLLER

A single-channel acoustic noise controlling scheme is depicted in Figure 1. The corresponding block diagram is shown in Figure 2. The noise source is sensed by a reference microphone and the primary path P consists of the acoustic response from the reference microphone to the error microphone located at the canceling point. The signal to be attenuated is marked as $d_p(n)$. The reference microphone collects the samples $x(n)$ of the noise source and feeds them as input to the adaptive controller. The controller is adapted according to the feedback signal $e(n)$ coming from the error microphone. The controller output $y(n)$ generates an acoustic signal that, traveling through the secondary path S, gives a signal $d_s(n)$ which destructively interferes with the undesired signal $d_p(n)$. It is usually assumed that the secondary path is linear and time invariant, and that its impulse response $s(n)$ has been obtained by separate estimation procedures. Then, the signal $d_s(n)$ is given as the linear convolution of $s(n)$ with the signal $y(n)$ and thus the error signal can be expressed as $e(n) = d_p(n) + d_s(n) = d_p(n) + s(n) * y(n)$, where $*$ indicates the operation of linear convolution. In the nonlinear situation we are dealing with, the controller is described as
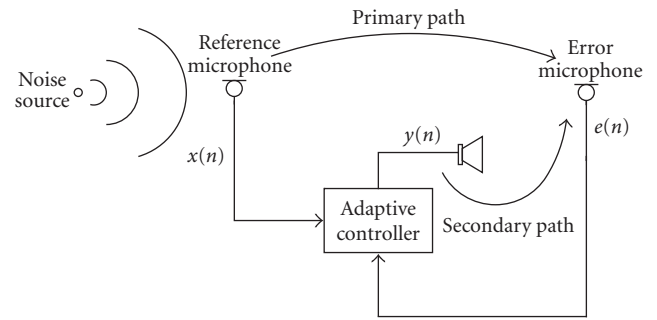

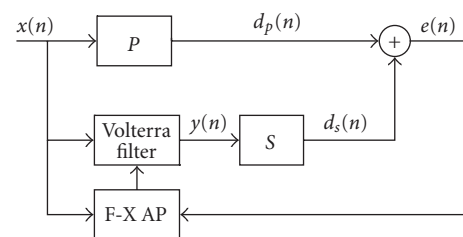FIGURE 1: Single-channel adaptive controller.

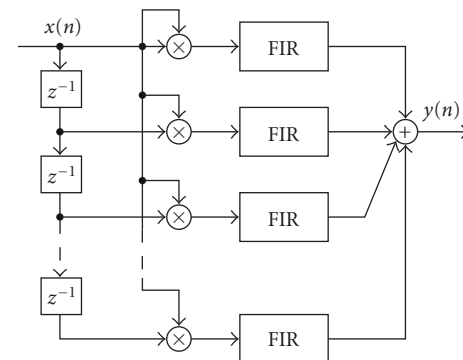FIGURE 2: F-X AP adaptive nonlinear controller.

FIGURE 3: Simplified Volterra filter structure.

a Volterra filter implemented by a multichannel filter bank. The adaptation of the nonlinear filter is controlled by means of the F-X AP algorithms introduced in the next Section.

3. SINGLE-CHANNEL F-X AP ALGORITHMS

We illustrate here the main steps leading to the derivation of F-X AP algorithms for a homogeneous quadratic filter represented as a filter bank according to the SVF implementation shown in Figure 3. The output $y(n)$ in Figure 3 is obtained as

$$y(n) = \sum_{i=1}^{M} y_i(n), \tag{1}$$

where M is the number of channels actually used, with $M \leq N$. N is the memory length of the quadratic filter and $y_i(n)$ is the output of the FIR filter in the generic channel i in

Figure 3, given by the following equation:

$$y_i(n) = \sum_{k=0}^{N-i} h(k, k+i-1)x(n-k)x(n-k-i+1). \quad (2)$$

Using a vector notation, (2) becomes

$$y_i(n) = \mathbf{h}_i^T(n)\mathbf{x}_i(n), \quad (3)$$

where $\mathbf{h}_i(n)$ is the vector formed with the $N-i+1$ coefficients of the ith channel $(1 \leq i \leq M)$,

$$\mathbf{h}_i(n) = [h(0, i-1) \ h(1, i) \ \cdots \ h(N-i, N-1)]^T. \quad (4)$$

The corresponding input vector $\mathbf{x}_i(n)$, again formed with $N-i+1$ entries, is defined as

$$\mathbf{x}_i(n) = \begin{bmatrix} x(n)x(n-i+1) \\ x(n-1)x(n-i) \\ \vdots \\ x(n-N+i)x(n-N+1) \end{bmatrix}. \quad (5)$$

If we now define two vectors of $K = \sum_{k=1}^{M}(N-k+1)$ elements

$$\begin{aligned} \mathbf{h}(n) &= [\mathbf{h}_1^T(n) \ \cdots \ \mathbf{h}_M^T(n)]^T, \\ \mathbf{x}(n) &= [\mathbf{x}_1^T(n) \ \cdots \ \mathbf{x}_M^T(n)]^T \end{aligned} \quad (6)$$

formed, respectively, with the vectors $\mathbf{h}_i(n)$ and $\mathbf{x}_i(n)$ related to single channels of the filter bank, then the output of the homogeneous quadratic filter can be written as

$$y(n) = \mathbf{h}^T(n)\mathbf{x}(n). \quad (7)$$

While the F-X LMS algorithm minimizes, according to the stochastic gradient approximation, the single error at time n,

$$e(n) = d_p(n) + d_s(n) = d_p(n) + s(n) * [\mathbf{h}^T(n)\mathbf{x}(n)], \quad (8)$$

the aim of an F-X AP algorithm of order L is to minimize the last L F-X errors. The desired minimization can be obtained by finding the minimum norm of the coefficient increments that set to zero the last L a posteriori errors. More in detail, the a posteriori error at time $n-j+1$, with $j = 1,\ldots,L$, is defined as

$$\begin{aligned} \epsilon(n-j+1) = d_p(n-j+1) + s(n-j+1) \\ * [\mathbf{h}^T(n+1)\mathbf{x}(n-j+1)]. \end{aligned} \quad (9)$$

The set of constraints for an Lth order F-X AP algorithm is written in the following form:

$$\begin{aligned} d_p(n) + s(n) * \mathbf{h}^T(n+1)\mathbf{x}(n) &= 0, \\ d_p(n-1) + s(n-1) * \mathbf{h}^T(n+1)\mathbf{x}(n-1) &= 0, \\ &\vdots \\ d_p(n-L+1) + s(n-L+1) * \mathbf{h}^T(n+1)\mathbf{x}(n-L+1) &= 0. \end{aligned} \quad (10)$$

The function J to be minimized can then be defined as

$$\begin{aligned} J = \delta\mathbf{h}^T(n+1)\delta\mathbf{h}(n+1) \\ + \sum_{j=1}^{L} \lambda_j (d_p(n-j+1) + s(n-j+1) \\ * \mathbf{h}^T(n+1)\mathbf{x}(n-j+1)), \end{aligned} \quad (11)$$

where

$$\delta\mathbf{h}(n+1) = \mathbf{h}(n+1) - \mathbf{h}(n) \quad (12)$$

and λ_j are Lagrange's multipliers. By differentiating J with respect to $\delta\mathbf{h}(n+1)$, the following set of K equations is obtained

$$2\delta\mathbf{h}(n+1) = -\sum_{j=1}^{L} \lambda_j s(n-j+1) * \mathbf{x}(n-j+1), \quad (13)$$

where the linearity of the convolution operation has been exploited. Equation (13) can be also rewritten as

$$2\delta\mathbf{h}(n+1) = -\mathbf{G}(n)\mathbf{\Lambda}, \quad (14)$$

where the $K \times L$ matrix $\mathbf{G}(n)$ is defined as

$$\begin{aligned} \mathbf{G}(n) \\ = [s(n)*\mathbf{x}(n) \ s(n-1)*\mathbf{x}(n-1) \ \cdots \ s(n-L+1)*\mathbf{x}(n-L+1)], \end{aligned} \quad (15)$$

$$\mathbf{\Lambda} = [\lambda_1\lambda_2\cdots\lambda_L]^T. \quad (16)$$

By premultiplying equation (14) by $\mathbf{G}^T(n)$, the following equation is obtained:

$$\mathbf{\Lambda} = -(\mathbf{G}^T(n)\mathbf{G}(n))^{-1}\mathbf{G}^T(n)2\delta\mathbf{h}(n+1). \quad (17)$$

The $L \times 1$ vector $\mathbf{G}^T(n)2\delta\mathbf{h}(n+1)$ can be also written as

$$\begin{aligned} \mathbf{G}^T(n)2\delta\mathbf{h}(n+1) \\ = 2\begin{bmatrix} \{s(n) * \mathbf{x}^T(n)\}\delta\mathbf{h}(n+1) \\ \{s(n-1) * \mathbf{x}^T(n-1)\}\delta\mathbf{h}(n+1) \\ \vdots \\ \{s(n-L+1) * \mathbf{x}^T(n-L+1)\}\delta\mathbf{h}(n+1) \end{bmatrix}. \end{aligned} \quad (18)$$

Apart from the factor 2, the jth element of this vector $(j = 1,\ldots,L)$, can be written, after some manipulations, as

$$\begin{aligned} [s(n-j+1) * \mathbf{x}^T(n-j+1)][\mathbf{h}(n+1) - \mathbf{h}(n)] \\ = s(n-j+1) * [\mathbf{h}^T(n+1)\mathbf{x}(n-j+1)] \\ - s(n-j+1) * [\mathbf{h}^T(n)\mathbf{x}(n-j+1)] \\ = -d_p(n-j+1) - s(n-j+1) \\ * [\mathbf{h}^T(n)\mathbf{x}(n-j+1)] = -e_j(n). \end{aligned} \quad (19)$$

In deriving this expression, the linearity of the convolution operation and the constraints given by (10) have been

---

Initialization: $\mathbf{h}_i = \mathbf{0} \ \forall i$

$y(n) = \sum_{i=1}^{M} \mathbf{h}_i^T(n)\mathbf{x}_i(n)$

$e_j(n) = d_p(n-j+1) + s(n-j+1) * \sum_{i=1}^{M} \mathbf{h}_i^T(n)\mathbf{x}_i(n-j+1)$

for $j = 1,\ldots,L$

$\mathbf{e}(n) = [e_1(n) \ \ e_2(n) \ \ \cdots \ \ e_L(n)]^T$

$\mathbf{G}_i(n) =$

$[s(n)*\mathbf{x}_i(n) \ \ s(n-1)*\mathbf{x}_i(n-1) \ \ \cdots \ \ s(n-L+1)*\mathbf{x}_i(n-L+1)]$

$\mathbf{h}_i(n+1) = \mathbf{h}_i(n) - \mu_i \mathbf{G}_i(n)(\mathbf{G}^T(n)\mathbf{G}(n))^{-1}\mathbf{e}(n)$ for $i = 1,\ldots,M$

ALGORITHM 1: F-X AP adaptive algorithm of order L for an SVF using the direct matrix inversion.

exploited. As a conclusion, (18) can be rewritten as

$$\mathbf{G}^T(n)2\delta\mathbf{h}(n+1) = -2\mathbf{e}(n), \qquad (20)$$

where $\mathbf{e}(n)$ is the $L \times 1$ vector of the F-X a priori estimation errors,

$$\mathbf{e}(n) = [e_1(n) \ \ e_2(n) \ \ \cdots \ \ e_L(n)]^T. \qquad (21)$$

By combining (14), (17), and (20), the following relation is derived:

$$\delta\mathbf{h}(n+1) = -\mathbf{G}(n)(\mathbf{G}^T(n)\mathbf{G}(n))^{-1}\mathbf{e}(n). \qquad (22)$$

By splitting the vector $\delta\mathbf{h}(n+1)$ in its components, that is,

$$\delta\mathbf{h}(n+1) = [\delta\mathbf{h}_1^T(n+1)\cdots\delta\mathbf{h}_M^T(n+1)]^T, \qquad (23)$$

and accordingly partitioning the matrix $\mathbf{G}(n)$ in submatrices $\mathbf{G}_i(n)$ of congruent dimensions, the following set of equations is obtained:

$$\delta\mathbf{h}_i(n+1) = -\mathbf{G}_i(n)(\mathbf{G}^T(n)\mathbf{G}(n))^{-1}\mathbf{e}(n) \qquad (24)$$

for $i = 1,\ldots,M$. As a consequence, the updating relations for the coefficients of each branch are given by

$$\mathbf{h}_i(n+1) = \mathbf{h}_i(n) - \mu_i\mathbf{G}_i(n)(\mathbf{G}^T(n)\mathbf{G}(n))^{-1}\mathbf{e}(n) \qquad (25)$$

for $i = 1,\ldots,M$, where μ_i is a parameter that controls both the convergence rate and the stability of the F-X AP algorithm. The $L \times L$ matrix $\mathbf{G}^T(n)\mathbf{G}(n)$ represents an estimate of the filtered-X autocorrelation matrix of the signal formed with products of couples of input samples, obtained using the last L input vectors. The computation of its inverse is required at any time n.

Since this step is often a critical one, we can distinguish the solution for the F-X AP algorithms of low orders, that is, with $L = 2, 3$ from that for greater orders. In fact, for $L = 2, 3$, even the direct inversion of the matrix is an affordable task. The only necessary care in order to avoid possible numerical instabilities is to add a diagonal matrix δI, where δ is a small positive constant, to the matrix $\mathbf{G}^T(n)\mathbf{G}(n)$. The equations employed for updating the coefficients and filtering the input signal using SVFs are summarized in Algorithm 1.

A general and efficient solution which can be applied to any order L of affine projections is derived by resorting to a simpler and more stable estimate for the inverse of the matrix $\mathbf{G}^T(n)\mathbf{G}(n)$. We introduce the vectors

$$\tilde{\mathbf{x}}_i(n) = s(n) * \begin{bmatrix} x(n)x(n-i+1) \\ x(n-1)x(n-i) \\ \vdots \\ x(n-L+1)x(n-i-L+2) \end{bmatrix} \qquad (26)$$

for $i = 1,\ldots,M$, and the $M \times L$ matrix

$$\tilde{\mathbf{X}}(n) = [\tilde{\mathbf{x}}_1(n) \ \ \tilde{\mathbf{x}}_2(n) \ \ \cdots \ \ \tilde{\mathbf{x}}_M(n)]^T. \qquad (27)$$

Then a recursive approximation of the $\mathbf{G}^T(n)\mathbf{G}(n)$ matrix is given by

$$\mathbf{R}(n) = \lambda\mathbf{R}(n-1) + (1-\lambda)\tilde{\mathbf{X}}^T(n)\tilde{\mathbf{X}}(n), \qquad (28)$$

where λ is a forgetting factor ($0 < \lambda < 1$) which determines the temporal memory length in the estimation of the auto-correlation matrix. The higher the forgetting factor, the more insensitive to noise is the estimate. In practice, λ is always taken close to 1. The recursive estimate can now be used in the coefficient updating equation (25), where the computation of the inverse matrix is still required. To avoid such an inversion, it is convenient to directly update the inverse matrix $\mathbf{R}^{-1}(n)$, as done for the recursive least square (RLS) algorithm. We define the following matrices:

$$\mathbf{R}_0(n) = \lambda\mathbf{R}(n-1), \qquad (29)$$

$$\mathbf{R}_l(n) = \mathbf{R}_{l-1}(n) + (1-\lambda)\tilde{\mathbf{x}}_l(n)\tilde{\mathbf{x}}_l^T(n) \qquad (30)$$

for $l = 1,\ldots,M$. Since, from (28),

$$\mathbf{R}(n) = \lambda\mathbf{R}(n-1) + (1-\lambda)[\tilde{\mathbf{x}}_1(n)\tilde{\mathbf{x}}_1^T(n) + \tilde{\mathbf{x}}_2(n)\tilde{\mathbf{x}}_2^T(n) \\ + \cdots + \tilde{\mathbf{x}}_M(n)\tilde{\mathbf{x}}_M^T(n)], \qquad (31)$$

it immediately follows that $\mathbf{R}(n) = \mathbf{R}_M(n)$. By using the matrix inversion lemma [20], it is possible to derive from (30) the following updating rule:

$$\mathbf{R}_l^{-1}(n) = \mathbf{R}_{l-1}^{-1}(n-1) \\ - \frac{\mathbf{R}_{l-1}^{-1}(n-1)\tilde{\mathbf{x}}_l(n)\tilde{\mathbf{x}}_l^T(n)\mathbf{R}_{l-1}^{-1}(n-1)}{1/(1-\lambda) + \tilde{\mathbf{x}}_l^T(n)\mathbf{R}_{l-1}^{-1}(n-1)\tilde{\mathbf{x}}_l(n)}. \qquad (32)$$

These expressions can be written in a more compact form by defining

$$\mathbf{k}_l(n) = \frac{\mathbf{R}_{l-1}^{-1}(n-1)\tilde{\mathbf{x}}_l(n)}{1/(1-\lambda) + \tilde{\mathbf{x}}_l^T(n)\mathbf{R}_{l-1}^{-1}(n-1)\tilde{\mathbf{x}}_l(n)}, \qquad (33)$$

$\mathbf{P}(n) = \mathbf{R}^{-1}(n)$, and $\mathbf{P}_l(n) = \mathbf{R}_l^{-1}(n)$. Therefore, $\mathbf{P}(n) = \mathbf{P}_M(n)$ and $\mathbf{P}_0(n) = (1/\lambda)\mathbf{P}(n-1)$. As a consequence, the following recursive estimate for $\mathbf{P}_l(n)$ is derived:

$$\mathbf{P}_l(n) = \mathbf{P}_{l-1}(n) - \mathbf{k}_l(n)\tilde{\mathbf{x}}_l^T(n)\mathbf{P}_{l-1}(n). \qquad (34)$$

Initialization: $\mathbf{P}(-1) = \delta\mathbf{I}$, $\mathbf{h}_i = \mathbf{0}\ \forall i$

$y(n) = \sum_{i=1}^{M} \mathbf{h}_i^T(n)\mathbf{x}_i(n)$

$e_j(n) = d_p(n-j+1) + s(n-j+1) * \sum_{i=1}^{M} \mathbf{h}_i^T(n)\mathbf{x}_i(n-j+1)$

for $j = 1,\ldots,L$

$\mathbf{e}(n) = [e_1(n)\ \ e_2(n)\ \ \cdots\ \ e_L(n)]^T$

$\mathbf{G}_i(n) =$

$[s(n) * \mathbf{x}_i(n)\ \ s(n-1) * \mathbf{x}_i(n-1)\ \ \cdots\ \ s(n-L+1) * \mathbf{x}_i(n-L+1)]$

$\mathbf{P}_0(n) = (1/\lambda)\mathbf{P}(n-1)$

$\mathbf{k}_l(n) = \mathbf{P}_{l-1}(n)\bar{\mathbf{x}}_l(n)/(1/(1-\lambda) + \bar{\mathbf{x}}_l^T(n)\mathbf{P}_{l-1}(n)\bar{\mathbf{x}}_l(n))$

$\mathbf{P}_l(n) = \mathbf{P}_{l-1}(n) - \mathbf{k}_l(n)\bar{\mathbf{x}}_l^T(n)\mathbf{P}_{l-1}(n)$,

for $l = 1,\ldots,M$

$\mathbf{P}(n) = \mathbf{P}_M(n)$

$\mathbf{h}_i(n+1) = \mathbf{h}_i(n) - \mu_i\mathbf{G}_i(n)\mathbf{P}(n)\mathbf{e}(n)$

for $i = 1,\ldots,M$

ALGORITHM 2: Filtered-X AP adaptive algorithm of order L for an SVF using the matrix inversion lemma.

Finally, by replacing in (25) $(\mathbf{G}^T(n)\mathbf{G}(n))^{-1}$ with the matrix $\mathbf{P}(n) = \mathbf{P}_M(n)$, the following updating expression for the ith branch of an SVF is obtained:

$$\mathbf{h}_i(n+1) = \mathbf{h}_i(n) - \mu_i\mathbf{G}_i(n)\mathbf{P}(n)\mathbf{e}(n) \qquad (35)$$

for $i = 1,\ldots,M$. The corresponding updating algorithm is described in Algorithm 2. Its complexity is given by $O(ML^2 + KL)$. Since $O(K) = O(MN)$ and usually $L < M$, the number of multiplications needed to implement SVFs equipped with this adaptive AP algorithm is $O(LMN)$, Therefore, its complexity is of the order of L times that of the corresponding LMS algorithm. More specifically, the complexity of the F-X AP algorithm for a quadratic filter with $M = N$ is $O(LN^2)$ per sample, while the complexity of the F-X LMS algorithm is $O(N^2)$, as also reported in [9]. On the other hand, for high values of L, the F-X AP algorithms tend to behave as the RLS algorithms with similar convergence rates and tracking capabilities. However, the complexity of RLS algorithms for quadratic filters is $O(N^4)$ or $O(N^3)$ for their fast versions [15, page 271]. In addition, it is worth noting that often even small values of L, that is, $L = 2, 3$, are sufficient to obtain remarkable convergence improvements with respect to the F-X LMS algorithm. Moreover, while the F-X AP algorithm can be applied to complete quadratic filters simply by setting $M = N$, using a small number of channels M often permits to obtain good adaptation performances with a reduced computational complexity, as shown in the next Section. In fact, with reference to these aspects, the implementation complexity $O(LMN)$ indicates a sort of tradeoff between the number L of APs used and the number M of active channels M in the filter bank realization.

Finally, it is worth noting that it is easy to pass from the algorithm for a homogeneous second-order Volterra filter to that of a generic Volterra filter. The SVF structure can be completed with the branches associated with the linear term and the higher-order Volterra operators according to their di-

agonal representation. Each of these channels is then treated by the algorithms of Tables 1 and 2 in a way similar to the channels of the homogeneous second-order Volterra filter.

4. SIMULATION RESULTS

In this section, we present some simulation results obtained with the F-X AP algorithms of Tables 1 and 2.

In the first set of simulations, we consider the same experimental conditions of [9, Section IV-A]. The source noise is a logistic chaotic noise, that is, a second-order white and predictable nonlinear process, generated with the recursive law,

$$\xi(i+1) = 4\xi(i)(1 - \xi(i)), \qquad (36)$$

where $\xi(0)$ is a real number between 0 and 1 different from $k/4$ with $k = 0, 1, \ldots, 4$. The nonlinear process is then normalized in order to have a unit signal power $x(i) = \xi(i)/\sigma_\xi$. The primary and secondary paths are modeled with the following FIR filters, respectively,

$$P(z) = z^{-5} - 0.3z^{-6} + 0.2z^{-7}, \qquad (37)$$

$$S(z) = z^{-2} + 1.5z^{-3} - z^{-4}. \qquad (38)$$

The system is identified with a second-order Volterra filter with a linear part of memory length 10 and a quadratic part of memory length 10 and 10 diagonals ($M = 10$). Figures 4 and 5 plot the ensemble average of the resulting mean square error for 100 runs of the simulation system, using the direct matrix inversion as in Algorithm 1 and the recursive technique in Algorithm 2, respectively. The four curves refer to different values of the affine projection order L. The order $L = 1$ corresponds to a normalized LMS adaptation algorithm, which is the same adaptation algorithm employed in [9] apart from the normalization. In the experiments of Figure 4, the step size was equal to 0.005. In the experiments of Figure 5, the λ factor was equal to 0.9 and the step size value was set to 0.0009 in order to obtain the same convergence characteristics of the first set of experiments when the affine projection order L equals 1. For higher orders of affine projections, the improvement in the convergence behavior of the algorithm is evident. Moreover, the adaptation curves of Figure 5 indicate a slight but steady reduction of the asymptotic error for increasing values of L. This fact confirms the reliability of the recursive approximation leading to the algorithm of Algorithm 2.

In the second set of experiments, we simulate a sudden change in the noise source and its propagation model and we investigate the ability of the F-X AP algorithm to track the noise conditions. We employ the same experimental conditions of the first set of simulations, but after 100000 signal samples we modify the primary path model according to the following equation:

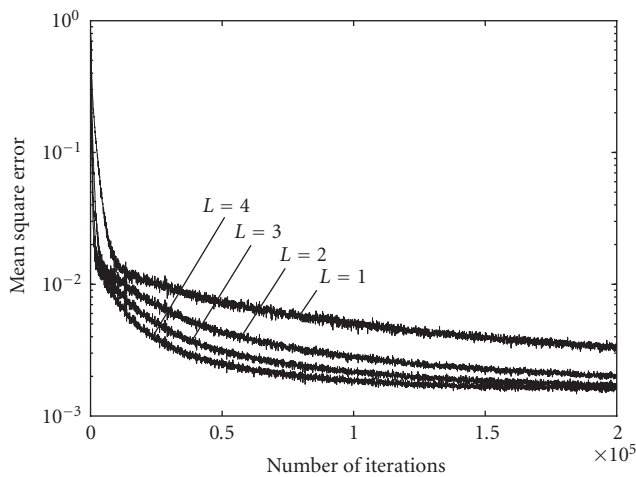$$P(z) = z^{-5} + 0.3z^{-6} - 0.2z^{-7} \qquad (39)$$

FIGURE 4: Adaptation curves for different orders of affine projections L using the method in Algorithm 1.
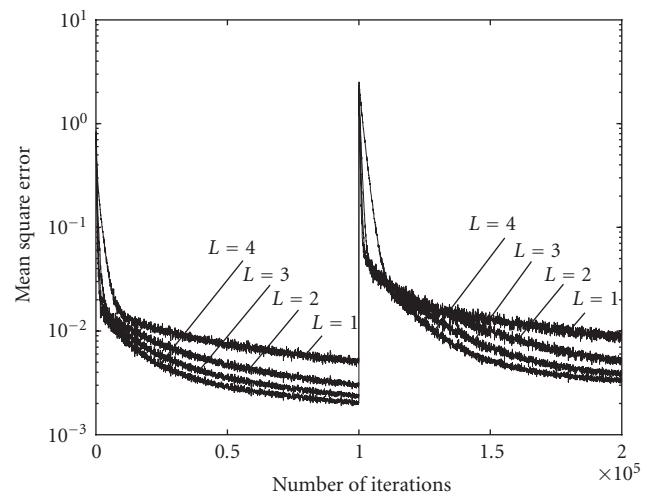

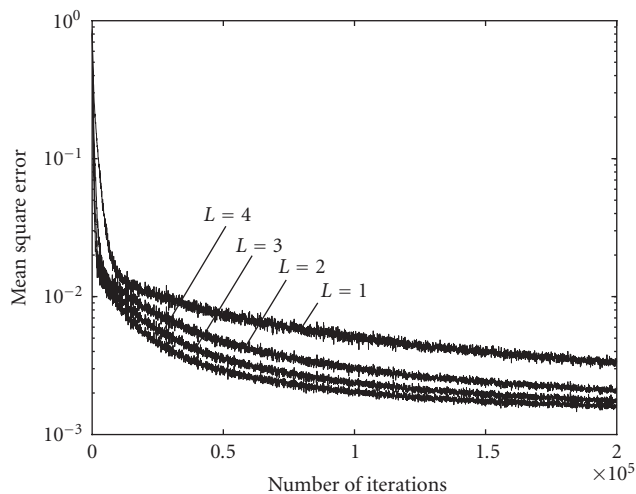FIGURE 6: Adaptation curves with a sudden modification in the noise conditions.

FIGURE 5: Adaptation curves for different orders of affine projections L using the method in Algorithm 2.
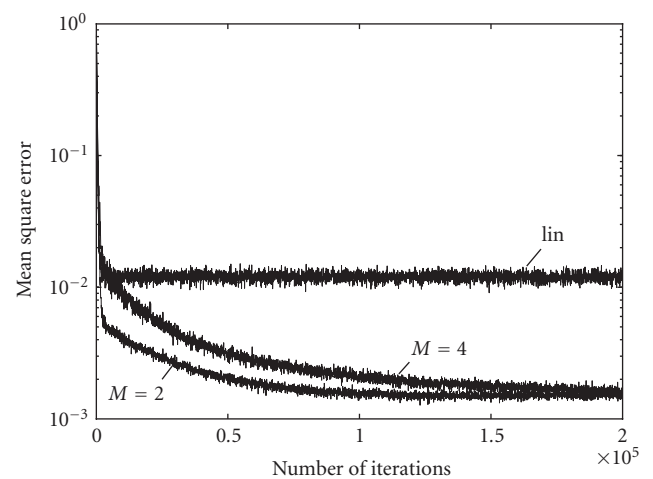

FIGURE 7: Adaptation curves with different number of channels and linear primary path.

and we use as input signal $\hat{x}(i) = x^2(i)/2$, where $x(i)$ is the normalized logistic noise of the previous experiments. Figure 6 plots the resulting adaptation curves for different orders of affine projections when the algorithm of Algorithm 2 is applied for the filter adaptation. Again, we can observe the improvement in the convergence behavior determined by the AP algorithm.

In the last set of experiments, we investigate the effects of modeling an active noise controller as a multichannel filter bank with a reduced number of channels. The noise source is the logistic chaotic noise of the first set of experiments. The primary and secondary paths are modeled as in (37), (38), respectively. The system is identified with an SVF with a linear part and a quadratic part, both of memory length 10. Figures 7 and 8 plot the ensemble average

of the resulting mean square error for 100 runs of the simulation system. The algorithm of Algorithm 2 was applied with an affine projection order L set to 4 and with $\lambda = 0.9$ and $\mu = 0.0005$. The curves in Figure 7 refer to the linear controller and the quadratic controller with $M = 2$ and $M = 4$. The convergence improvement with respect to the linear case can be easily appreciated, especially for $M = 2$. It has been experimentally verified that the curves for $4 \leq M \leq 10$ converge with a progressively slower behavior to about the residual error of the curve for $M = 2$. Therefore, using the full model, as done in [9], does not allow any improvement. It has been also observed that using a number L of APs equal to 2 gives, as expected, the same global performances, but with a reduced convergence speed.
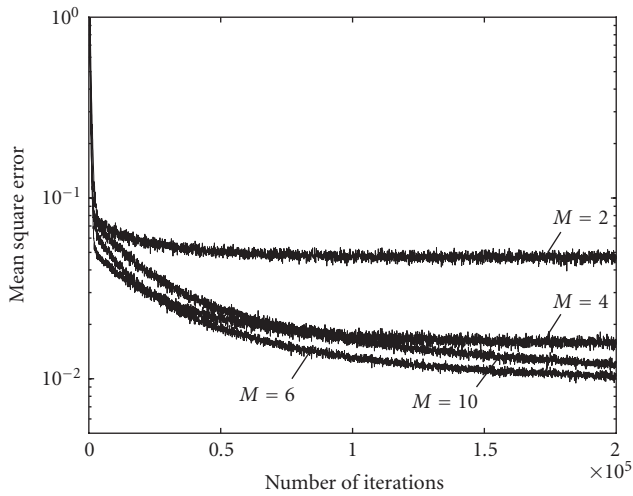
FIGURE 8: Adaptation curves with different number of channels and nonlinear primary path.

To complete this set of experiments, the primary path has been then replaced by the following second-order Volterra filter:

$$
\begin{aligned}
y(n) = {} & x(n-5) - 0.3x(n-6) + 0.2x(n-7) \\
& + 0.5x(n-5)x(n-5) - 0.1x(n-6)x(n-6) \\
& + 0.1x(n-7)x(n-7) - 0.2x(n-5)x(n-6) \\
& + 0.05x(n-6)x(n-7) - 0.02x(n-7)x(n-8) \\
& + 0.1x(n-5)x(n-7) - 0.02x(n-6)x(n-8) \\
& + 0.01x(n-7)x(n-9) + 0.5x(n-5)x(n-8) \\
& - 0.1x(n-6)x(n-9) + 0.1x(n-7)x(n-10) \\
& - 0.2x(n-5)x(n-9) + 0.05x(n-6)x(n-10) \\
& - 0.02x(n-7)x(n-11) + 0.1x(n-5)x(n-10) \\
& - 0.02x(n-6)x(n-11) + 0.01x(n-7)x(n-12)
\end{aligned}
\tag{40}
$$

and all the simulations have been repeated with the same parameters. The results obtained for different numbers of branches M of the quadratic part of the SVF are shown in Figure 8. Of course, the best approximation result is now that for $M = 6$, since in this case the SVF exactly corresponds to the system to be modeled. We observe that when $M = 2$, the resulting SVF is inadequate to model the noise generation system, while for $M = 4$ a better approximation is obtained. This case can be considered as a compromise in terms of modeling accuracy, speed of convergence, and computational cost. From Figure 8, it can be noted again that overdimensioning the model using a complete second-order Volterra filter, $M = 10$, does not offer particular advantages. In fact, this filter is able to model the noise generation system with slightly reduced accuracy and convergence speed at an increased computational cost with respect to the reference case $M = 6$.

5. CONCLUSIONS

In practical applications, methods for active noise control have often to deal with nonlinear effects. In such environments, nonlinear controllers based on Volterra filters implemented in the form of multichannel filter banks can be usefully exploited. One of the crucial aspects is the derivation of efficient adaptation algorithms. Usually, the so-called filtered-X LMS or NLMS algorithms are used. In this paper we proposed the use of the affine projection technique, and we derived in detail the so-called filtered-X AP algorithms for homogeneous quadratic filters. According to the multichannel approach, these derivations can be easily extended to a generic Volterra filter. The extensive experiments we report confirm that the AP technique offers better convergence and tracking capabilities than the classical LMS and NLMS algorithms with a limited increase of the computational complexity.

ACKNOWLEDGMENT

REFERENCES

[1] G. E. Warnaka, "Active attenuation of noise—the state of the art," *Noise Control Engineering*, vol. 18, no. 3, pp. 100–110, 1982.

[2] S. J. Elliot and P. A. Nelson, *Active Control of Sound*, Academic Press, New York, NY, USA, 3rd edition, 1995.

[3] S. J. Elliott and P. A. Nelson, "Active noise control," *IEEE Signal Processing Magazine*, vol. 10, no. 4, pp. 12–35, 1993.

[4] S. M. Kuo and D. R. Morgan, *Active Noise Control Systems: Algorithms and DSP Implementations*, John Wiley & Sons, New York, NY, USA, 1996.

[5] D. Pavisic, L. Blondel, J.-P. Draye, G. Libert, and P. Chapelle, "Active noise control with dynamic recurrent neural networks," in *Proc. European Symposium on Artificial Neural Networks (ESANN '95)*, pp. 45–50, Brussels, Belgium, April 1995.

[6] T. Matsuura, T. Hiei, H. Itoh, and K. Torikoshi, "Active noise control by using prediction of time series data with a neural network," in *Proc. IEEE International Conference on Systems, Man and Cybernetics (SMC '95)*, vol. 3, pp. 2070–2075, Vancouver, BC, Canada, October 1995.

[7] 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, 1998.

[8] L. Tan and J. Jiang, "Filtered-X second-order Volterra adaptive algorithms," *Electronics Letters*, vol. 33, no. 8, pp. 671–672, 1997.

[9] L. Tan and J. Jiang, "Adaptive Volterra filters for active control of nonlinear noise processes," *IEEE Trans. Signal Processing*, vol. 49, no. 8, pp. 1667–1676, 2001.

[10] M. Bouchard, B. Paillard, and C. T. L. Dinh, "Improved training of neural networks for the nonlinear active control of sound and vibration," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 391–401, 1999.

[11] C. A. Silva, J. M. Sousa, and J. M. G. Sá da Costa, "Active noise control based on fuzzy models," in *Proc. 4th European Conference on Noise Control (EURONOISE '01)*, pp. 1–14, Patras, Greece, January 2001.

[12] J. M. Conchinha, M. A. Botto, J. M. Sousa, and J. M. G. Sá da Costa, "The use of neural network models in an active noise control applications," in *Proc. 4th European Conference on*

Noise Control (EURONOISE '01), pp. 1–13, Patras, Greece, January 2001.

[13] M. H. Costa, J. C. Bermudez, and N. J. Bershad, "Stochastic analysis of the filtered-X LMS algorithm in systems with nonlinear secondary paths," *IEEE Trans. Signal Processing*, vol. 50, no. 6, pp. 1327–1342, 2002.

[14] O. J. Tobias and R. Seara, "Performance comparison of the FXLMS, nonlinear FXLMS and leaky FXLMS algorithms in nonlinear active control applications," in *Proc. European Signal Processing Conference (EUSIPCO '02)*, pp. 1–4, Toulouse, France, September 2002.

[15] V. J. Mathews and G. L. Sicuranza, *Polynomial Signal Processing*, John Wiley & Sons, New York, NY, USA, 2000.

[16] G. M. Raz and B. D. van Veen, "Baseband Volterra filters for implementing carrier based nonlinearities," *IEEE Trans. Signal Processing*, vol. 46, no. 1, pp. 103–114, 1998.

[17] A. Fermo, A. Carini, and G. L. Sicuranza, "Simplified Volterra filters for acoustic echo cancellation in GSM receivers," in *Proc. European Signal Processing Conference (EUSIPCO '00)*, Tampere, Finland, September 2000.

[18] A. Fermo, A. Carini, and G. L. Sicuranza, "Low-complexity nonlinear adaptive filters for acoustic echo cancellation in GSM hand-set receivers," *European Transactions on Telecommunications*, vol. 14, no. 2, pp. 161–169, 2003.

[19] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan*, vol. 67-A, no. 5, pp. 19–27, 1984.

[20] C. F. N. Cowan and P. M. Grant, *Adaptive Filters*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1985.

Alberto Carini was born in Trieste, Italy, in 1967. He received the Laurea degree (summa cum laude) in electronic engineering in 1994 and the "Dottorato di Ricerca" degree (Ph.D.) in information engineering in 1998, both from the University of Trieste, Italy. He has received the Zoldan Award for the best Laurea degree in electronic engineering at the University of Trieste during the academic year 1992–1993. In 1996 and 1997, during his Ph.D. studies, he spent several months as a Visiting Scholar at the University of Utah, Salt Lake City, USA. From 1997 to 2003, he worked as a DSP engineer with Telit Mobile Terminals SpA, Trieste, Italy, where he was leading the audio processing R&D activities. In 2003, he worked with Neonseven srl, Trieste, Italy, as audio and DSP expert. From 2001 to 2004, he collaborated with the University of Trieste as a Contract Professor of Digital Signal Processing. Since 2004, he is an Associate Professor at the Information Science and Technology Institute (ISTI), University of Urbino, Urbino, Italy. His research interests include adaptive filtering, nonlinear filtering, nonlinear equalization, acoustic echo cancellation, and active noise control.

Giovanni L. Sicuranza is Professor of signal and image processing and Head of the Image Processing Laboratory at DEEI, University of Trieste (Italy). His research interests include multidimensional digital filters, polynomial filters, processing of images and image sequences, image coding, and adaptive algorithms for echo cancellation and active noise control. He has published a number of papers in international journals and conference proceedings. He contributed in chapters of six books and is the Coeditor, with Professor Sanjit Mitra, University of California at Santa Barbara, of the books *Multidimensional Processing of Video Signals*, (Kluwer Academic Publisher, 1992), and *Nonlinear Image Processing*, (Academic Press, 2001). He is the coauthor with Professor V. John Mathews, University of Utah at Salt Lake City, of the book *Polynomial Signal Processing*, (J. Wiley, 2000). Dr. Sicuranza has been a member of the technical committees of numerous international conferences and Chairman of EUSIPCO-96 and NSIP-03. He is an Associate Editor of "Multidimensional Systems and Signal Processing" and a Member of the Editorial Board of "Signal Processing" and "IEEE Signal Processing Magazine." Dr. Sicuranza is currently the Awards Chairman of the Administrative Committee of EURASIP and a Member of the IMDSP Technical Committee of the IEEE Signal Processing Society. He has been one of the founders and the first Chairman of the Nonlinear Signal and Image Processing (NSIP) Board of which he is still a Member.