FILTERED-X AFFINE PROJECTION ALGORITHMS FOR A CLASS OF NONLINEAR MULTICHANNEL ACTIVE NOISE CONTROLLERS

[1]*Alberto Carini*, [2]*Giovanni L. Sicuranza*

[1]`carini@sti.uniurb.it`
[1]University of Urbino, ISTI, Piazza della Repubblica 13, 61029 Urbino, Italy
[2]University of Trieste, DEEI, Via A. Valerio 10, 34127 Trieste, Italy

ABSTRACT

The paper extends the use of multichannel filtered-x affine projection algorithms suitable for feed-forward active noise control to a broad class of nonlinear filter structures, which comprises also Volterra filters and Functional Link Artificial Neural Networks (FLANN). An analysis of the transient and steady-state behavior of the resulting algorithms is provided. Some experimental results that compare multichannel filtered-x affine projection algorithms for Volterra filters and for FLANN are discussed.

1. INTRODUCTION

The problem of nonlinear active noise control has recently attracted the attention of many researchers. While most of active noise control systems applied in practice are linear, in the last years it has been recognized that nonlinear effects can affect actual applications [1, 2, 3]. 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 acoustic paths may exhibit a nonlinear behavior thus motivating the use of a nonlinear controller. Different nonlinear filter structures have been proposed in the literature to cope with these nonlinearities. The first nonlinear active noise control structures were based on multilayer neural networks [1], but very soon Volterra filters were also adopted [2, 3] and they were subsequently used in many contributions. More recently, Functional Link Artificial Neural Network (FLANN) were proposed by Das e Panda in [4] as an alternative to Volterra filters. Like the Volterra filters, the output of a FLANN structure depends linearly from the filter coefficients, but the terms multiplied by these coefficients, rather than product of input samples, are nonlinear functions of these samples. In [4] it was also shown that FLANN employing trigonometric functional expansion and adapted with a filtered-x LMS algorithm can provide similar or better behavior than Volterra filters adapted with an equivalent adaptation algorithm.

This work was supported by the Grant MIUR PRIN 2004092314.

More recently, different singlechannel and multichannel filtered-x affine projection (AP) algorithms suitable for feedforward active noise control employing both linear and Volterra filters have been studied [5, 6]. The transient and steady-state behavior of these algorithms were analyzed in [7] and [8]. In this paper, we first show that singlechannel and multichannel filtered-x affine projection algorithms can also be used for the adaptation of FLANN structures and of all filters whose output depend linearly from the filter coefficients. Then the analysis technique developed in [7] and [8] is applied and the transient and steady-state behavior of the resulting algorithms is discussed. Eventually, some experimental results that compare multichannel filtered-x affine projection algorithms for Volterra filters and for FLANN are discussed.

2. FILTERED-X AP ALGORITHMS

The general scheme of a multichannel feed-forward active noise controller is shown in Figure 1 in the form of the so-called *delay-compensation* scheme adopted throughout the paper. It is assumed here that a perfect model of the secondary paths is available, i. e. $\tilde{s}_{k,j}(z) = s_{k,j}(z)$, but this limitation can be easily removed by following the same methodology of [8]. In this paper the j-th actuator

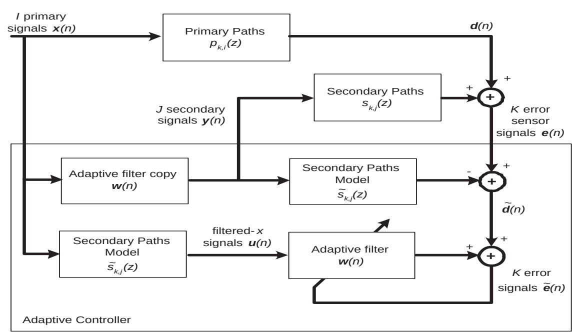

Figure 1: Delay-compensated filtered-x structure for active noise control.

output is modeled as

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

where $\mathbf{w}_{j,i}(n)$ is the coefficient vector of the filter connecting the input i to the output j of the adaptive controller, and $\mathbf{x}_i(n)$ is the i-th primary source input signal vector. In our approach $\mathbf{x}_i(n)$ is expressed as a vector function of the signal samples $x_i(n)$ whose general form is given by

$$\mathbf{x}_i(n) = \left[f_1\left[x_i(n)\right], f_2\left[x_i(n)\right], \ldots, f_N\left[x_i(n)\right]\right]^T, \tag{2}$$

where $f_i[\cdot]$, for any $i = 1 \ldots N$, is a time invariant functional of its argument. Equations (1) and (2) include linear filters, truncated Volterra filters of any order p and other nonlinear functionals as the FLANN structure of [4]. As an example, $\mathbf{x}_i(n)$ for FLANN employing trigonometric functional expansion with memory length N_i and finite order P_i is given by

$$\begin{aligned}\mathbf{x}_i(n) = \big[&x_i(n), x_i(n-1), \ldots, x_i(n-N_i+1),\\ &\sin[\pi x_i(n)], \ldots, \sin[\pi x_i(n-N_i+1)],\\ &\cos[\pi x_i(n)], \ldots, \cos[\pi x_i(n-N_i+1)], \ldots\\ &\sin[P_i\pi x_i(n)], \ldots, \sin[P_i\pi x_i(n-N_i+1)],\\ &\cos[P_i\pi x_i(n)], \ldots, \cos[P_i\pi x_i(n-N_i+1)]\big]^T. \end{aligned} \tag{3}$$

To introduce the filtered-x AP algorithms the following notations are used:
I is the number of primary source signals,
J is the number of secondary source signals,
K is the number of error sensors,
L is the affine projection order,
N is the number of elements of vectors $\mathbf{x}_i(n)$ and $\mathbf{w}_{j,i}(n)$,
$M = N \cdot I \cdot J$ is the number of coefficients of $\mathbf{w}(n)$,
$s_{k,j}(n)$ is the impulse response of the secondary path connecting the j-th secondary source to the k-th error sensor,
$\mathbf{x}_i(n)$ is the i-th primary source input signal vector,
$\mathbf{x}(n) = \left[\mathbf{x}_1^T(n), \ldots, \mathbf{x}_I^T(n)\right]^T$, is the full primary source input signal vector,
$\mathbf{w}_{j,i}(n)$ is the coefficient vector of the filter connecting the input i to the output j of the controller,
$\mathbf{w}_j(n) = \left[\mathbf{w}_{j,1}^T(n), \ldots, \mathbf{w}_{j,I}^T(n)\right]^T$ is the aggregate of the coefficient vectors at the output j of the controller,
$\mathbf{w}(n) = \left[\mathbf{w}_1^T(n), \ldots, \mathbf{w}_I^T(n)\right]^T$ is the full coefficient vector of the controller,
$y_j(n) = \mathbf{w}_j^T(n)\mathbf{x}(n)$ is the j-th secondary source signal,
$d_k(n)$ is the output of the k-th primary path,
$\mathbf{d}_k(n) = \left[d_k(n), \ldots, d_k(n-L+1)\right]^T$ is the vector of the L past outputs of the k-th primary path,
$\mathbf{d}(n) = \left[\mathbf{d}_1^T(n), \ldots, \mathbf{d}_K^T(n)\right]^T$ is the full vector of the L past outputs of the primary paths,
$\mathbf{u}_{k,j}(n) = s_{k,j}(n) \odot \mathbf{x}(n)$ is the filtered-x vector obtained by filtering, sample by sample, $\mathbf{x}(n)$ with $s_{k,j}(n)$,
$\mathbf{u}_k(n) = \left[\mathbf{u}_{k,1}^T(n), \ldots, \mathbf{u}_{k,J}^T(n)\right]^T$ is the aggregate of the filtered-x vectors associated with the output k,
$\mathbf{U}_k(n) = \left[\mathbf{u}_k(n), \mathbf{u}_k(n-1), \ldots, \mathbf{u}_k(n-L+1)\right]$ is the matrix constituted by the last L filtered-x vectors $\mathbf{u}_k(n)$,
$\mathbf{U}(n) = \left[\mathbf{U}_1(n), \ldots, \mathbf{U}_K(n)\right]$ is the full matrix of filtered-x vectors,
$e_k(n) = d_k(n) + \sum_{j=1}^{J} s_{k,j}(n) \odot y_j(n)$ is the k-th error sensor signal,
$\mathbf{I}$ indicates an identity matrix of appropriate dimensions,
$\odot$ denotes the linear convolution,
vec$\{\cdot\}$ indicates the vector operator,
vec$^{-1}\{\cdot\}$ is the inverse vector operator,
diag$\{\ldots\}$ is a block-diagonal matrix of the entries $\{\ldots\}$,
$\otimes$ denotes the Kronecker product.
In this paper we apply to the class of filters described by (1) and (2) the exact AP algorithm and an approximate AP algorithm introduced in [7] for linear filters. The adaptation rules of the two algorithms can be put in the same form as follows

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \boldsymbol{\mu}\overline{\mathbf{U}}(n)\mathbf{e}(n), \tag{4}$$

where $\boldsymbol{\mu}$ is a diagonal step-size matrix and the matrix $\overline{\mathbf{U}}(n)$ is defined according to equations (5) and (6) in case of the exact and the approximate algorithms respectively,

$$\overline{\mathbf{U}}(n) = \mathbf{U}(n)\left[\mathbf{U}^T(n)\mathbf{U}(n) + \delta\mathbf{I}\right]^{-1}, \tag{5}$$

$$\begin{aligned}\overline{\mathbf{U}}(n) = \mathbf{U}(n) \cdot \text{diag}\Big\{&\left[\mathbf{U}_1^T(n)\mathbf{U}_1(n) + \delta\mathbf{I}\right]^{-1}, \ldots\\ &\ldots, \left[\mathbf{U}_K^T(n)\mathbf{U}_K(n) + \delta\mathbf{I}\right]^{-1}\Big\},\end{aligned} \tag{6}$$

where δ is a small positive constant. Note that in the singlechannel case, where $I = J = K = 1$, the adaptation equation of (4) and (6) reduces to the adaptation rule of the exact AP algorithm given by (4) and (5).
In [7] we have shown that in the case of linear filters the AP algorithms in (4) do not converge to the minimum-mean-square (MMS) solution of the ANC problem but to the asymptotic coefficient vector given by

$$\mathbf{w}_\infty = -E\left[\mathbf{P}(n)\right]^{-1}E\left[\overline{\mathbf{U}}(n)\mathbf{d}(n)\right]. \tag{7}$$

with $\mathbf{P}(n) = \overline{\mathbf{U}}(n) \cdot \mathbf{U}^T(n)$. The same conclusions apply when we adapt with the filtered-x AP algorithms the filters described by equations (1) and (2).

3. TRANSIENT AND STEADY-STATE ANALYSIS

The aim of the transient and steady-state analysis is to study the time evolution of the expectation of the weighted Euclidean norm of the coefficient vector $E\left[\|\mathbf{w}(n)\|_{\boldsymbol{\Sigma}}^2\right] = \mathbf{w}(n)^T\boldsymbol{\Sigma}\mathbf{w}(n)$ for some choices of the symmetric positive definite matrix $\boldsymbol{\Sigma}$ [9].
By applying the approach of [9], the following result that describes the transient behavior of the AP algorithms was

proved in [8]. This result holds for the whole class of filters described by (1) and (2).

Theorem 1 *Under the assumption that $\mathbf{w}(n)$ is uncorrelated with $\mathbf{P}(n)$ and with $\mathbf{q}_{\boldsymbol{\Sigma}}(n) = \left(\mathbf{I} - \mathbf{P}^T(n)\,\boldsymbol{\mu}\right)\boldsymbol{\Sigma}\cdot \cdot\boldsymbol{\mu}\overline{\mathbf{U}}(n)\mathbf{d}(n)$, the transient behavior of the filtered-x AP algorithms with updating rule given by (4) is described by the state recursions*

$$E\left[\mathbf{w}(n+1)\right] = \mathbf{M}\,E\left[\mathbf{w}(n)\right] - E\left[\boldsymbol{\mu}\overline{\mathbf{U}}(n)\mathbf{d}(n)\right]$$

and $$\boldsymbol{\mathcal{W}}(n+1) = \boldsymbol{\mathcal{F}}\,\boldsymbol{\mathcal{W}}(n) + \boldsymbol{\mathcal{Y}}(n),$$

where $$\mathbf{M} = E\left[\left(\mathbf{I} - \boldsymbol{\mu}\mathbf{P}(n)\right)\right],$$

$$\boldsymbol{\mathcal{F}} = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ -p_0 & -p_1 & -p_2 & \dots & -p_{M^2-1} \end{bmatrix},$$

$$\boldsymbol{\mathcal{W}}(n) = \begin{bmatrix} E[\|\mathbf{w}(n)\|_{vec^{-1}\{\boldsymbol{\sigma}\}}] \\ E[\|\mathbf{w}(n)\|_{vec^{-1}\{\mathbf{F}\boldsymbol{\sigma}\}}] \\ \vdots \\ E[\|\mathbf{w}(n)\|_{vec^{-1}\{\mathbf{F}^{M^2-1}\boldsymbol{\sigma}\}}] \end{bmatrix},$$

$$\boldsymbol{\mathcal{Y}}(n) = \begin{bmatrix} \left(\boldsymbol{\gamma}^T + 2\,E[\mathbf{w}^T(n)]\mathbf{Q}\right)\boldsymbol{\sigma} \\ \left(\boldsymbol{\gamma}^T + 2\,E[\mathbf{w}^T(n)]\mathbf{Q}\right)\mathbf{F}\boldsymbol{\sigma} \\ \vdots \\ \left(\boldsymbol{\gamma}^T + 2\,E[\mathbf{w}^T(n)]\mathbf{Q}\right)\mathbf{F}^{M^2-1}\boldsymbol{\sigma} \end{bmatrix},$$

$\mathbf{F}$ *is the $M^2 \times M^2$ matrix defined by*

$$\mathbf{F} = E\left[\left(\mathbf{I} - \mathbf{P}^T(n)\,\boldsymbol{\mu}\right) \otimes \left(\mathbf{I} - \mathbf{P}^T(n)\,\boldsymbol{\mu}\right)\right],$$

$\mathbf{Q}$ *is the $M \times M^2$ matrix given by*

$$\mathbf{Q} = E\left[\left(\boldsymbol{\mu}\overline{\mathbf{U}}(n)\mathbf{d}(n)\right)^T \otimes \left(\mathbf{I} - \mathbf{P}^T(n)\,\boldsymbol{\mu}\right)\right],$$

the $M^2 \times 1$ vector $\boldsymbol{\gamma}$ is

$$\boldsymbol{\gamma} = vec\left\{E\left[\boldsymbol{\mu}\overline{\mathbf{U}}(n)\mathbf{d}(n)\mathbf{d}^T(n)\overline{\mathbf{U}}^T(n)\boldsymbol{\mu}\right]\right\},$$

the p_i are the coefficients of the characteristic polynomial of $\mathbf{F}$, i. e. $p(x) = x^{M^2} + p_{M^2-1}x^{M^2-1} + \ldots + p_1 x + p_0 = \det(x\mathbf{I} - \mathbf{F})$, and $\boldsymbol{\sigma} = vec\{\boldsymbol{\Sigma}\}$.

According to Theorem 1, the transient behavior of the filtered-x AP algorithms is described by the cascade of two linear systems, with system matrices $\mathbf{M}$ and $\boldsymbol{\mathcal{F}}$, respectively. The stability in the mean sense and mean-square sense can be deduced by the stability properties of these two linear systems. Indeed, the filtered-x AP algorithm will converge in the mean for any step-size matrix $\boldsymbol{\mu}$ such that $|\lambda_{\max}(\mathbf{M})| < 1$. The algorithm will converge in the mean-square sense if, in addition, $|\lambda_{\max}(\mathbf{F})| < 1$.

With the steady-state analysis we are interested in evaluating the mean-square-deviation (MSD) and the mean-square-error (MSE) at steady-state, which are defined by equations (8) and (9), respectively,

$$\begin{aligned} \text{MSD} &= \lim_{n\to+\infty} E\left[\|\mathbf{w}(n) - \mathbf{w}_\infty\|^2\right] \\ &= \lim_{n\to+\infty} E\left[\mathbf{w}^T(n)\mathbf{w}(n)\right] - \|\mathbf{w}_\infty\|^2, \quad (8) \end{aligned}$$

$$\text{MSE} = \lim_{n\to+\infty} E\left[\sum_{k=1}^{K} e_k^2(n)\right]. \quad (9)$$

In the hypothesis that $\mathbf{w}(n)$ is independent from $\sum_{k=1}^{K}\mathbf{u}_k(n)\mathbf{u}_k^T(n)$ and from $\sum_{k=1}^{K} d_k(n)\mathbf{u}_k(n)$, the MSE can be expressed as

$$\text{MSE} = S_d + 2\,\mathbf{R}_{ud}^T\mathbf{w}_\infty + \lim_{n\to+\infty} E\left[\mathbf{w}^T(n)\mathbf{R}_{uu}\mathbf{w}(n)\right], \quad (10)$$

where $S_d = E\left[\sum_{k=1}^{K} d_k^2(n)\right]$, $\mathbf{R}_{uu} = E\left[\sum_{k=1}^{K}\mathbf{u}_k(n)\mathbf{u}_k^T(n)\right]$ and $\mathbf{R}_{ud} = E\left[\sum_{k=1}^{K}\mathbf{u}_k(n)d_k(n)\right]$.

The computations in (8) and (10) require the evaluation of $\lim_{n\to+\infty} E\left[\|\mathbf{w}(n)\|_{\boldsymbol{\Sigma}}\right]$, where $\boldsymbol{\Sigma} = \mathbf{I}$ in (8) and $\boldsymbol{\Sigma} = \mathbf{R}_{uu}$ in (10). This limit can be estimated with the same methodology of [9] and thus the following expressions for the MSD and MSE are obtained

$$\text{MSD} = \left(\boldsymbol{\gamma}^T - 2\mathbf{w}_\infty^T\mathbf{Q}\right)\left(\mathbf{I} - \mathbf{F}\right)^{-1}\text{vec}\{\mathbf{I}\} - \|\mathbf{w}_\infty\|^2, \quad (11)$$

$$\begin{aligned} \text{MSE} = {} & S_d + 2\,\mathbf{R}_{ud}^T\mathbf{w}_\infty + \\ & \left(\boldsymbol{\gamma}^T - 2\,\mathbf{w}_\infty^T\mathbf{Q}\right)\left(\mathbf{I} - \mathbf{F}\right)^{-1}\text{vec}\{\mathbf{R}_{uu}\}. \quad (12) \end{aligned}$$

4. EXPERIMENTAL RESULTS

In this section we provide a few experimental results that compare filtered-x AP algorithms for FLANN and Volterra filters. We consider a multichannel active noise controller with $I = 1$, $J = 2$, $K = 2$. The transfer functions of the primary paths are given by

$$\begin{aligned} p_{1,1}(z) &= 1.0z^{-2} - 0.3z^{-3} + 0.2z^{-4}, \\ p_{2,1}(z) &= 1.0z^{-2} - 0.2z^{-3} + 0.1z^{-4}. \end{aligned}$$

and the transfer functions of the secondary paths are

$$\begin{aligned} s_{1,1}(z) &= 1.0z^{-2} + 1.5z^{-3} - 1.0z^{-4}, \\ s_{1,2}(z) &= 1.0z^{-2} + 1.3z^{-3} - 1.0z^{-4}, \\ s_{2,1}(z) &= 1.0z^{-2} + 1.3z^{-3} - 1.0z^{-4}, \\ s_{2,2}(z) &= 1.0z^{-2} + 1.2z^{-3} - 1.0z^{-4}. \end{aligned}$$

The input signal is the normalized logistic noise, which has been generated by scaling the signal $\xi(n)$ obtained from the logistic recursion $\xi(n+1) = \lambda\xi(n)(1 - \xi(n))$, with $\lambda = 4$ and $\xi(0) = 0.9$, and by adding a white Gaussian noise to get a 30 dB SNR. The controller is either a two-channel FLANN structure with memory length N=5 and order P=1 or a two-channel second-order Volterra filter with memory length N=5 for the linear and quadratic parts and two diagonals for the quadratic kernel. Therefore, the two controllers have the same memory length and

Table 1: First five coefficients of the MMS solution and of the asymptotic solutions.

FLANN			Volterra		
$\mathbf{w}_o$	$\mathbf{w}_{\infty,e}$	$\mathbf{w}_{\infty,a}$	$\mathbf{w}_o$	$\mathbf{w}_{\infty,e}$	$\mathbf{w}_{\infty,a}$
7.79	1.08	8.06	-3.54	2.54	-0.63
-5.16	-0.90	-4.84	-19.21	-0.92	-17.34
-0.43	0.31	-0.54	12.03	0.47	4.93
-1.63	0.28	-1.92	-6.65	0.11	-3.22
0.01	0.07	-0.07	3.62	0.17	2.27

15 and 14 coefficients per channel, respectively. Moreover, a zero mean, white Gaussian noise is added to the error microphone signals $d_k(n)$ to get a 40 dB SNR, the parameter δ is set to 0.001 and the same step-size is used for all the filter coefficients.

Table 1 compares the MMS solution ($\mathbf{w}_o$) and the asymptotic solution of the exact ($\mathbf{w}_{\infty,e}$) and approximate ($\mathbf{w}_{\infty,a}$) filtered-x AP algorithms with an AP order L=2 for FLANN and Volterra filters. The exact AP algorithm always provides a very biased estimate of the MMS solution. On the contrary, the approximate algorithm obtains a better estimation of the MMS coefficient vector. In all our experiments, even the simplest FLANN structure with P=1 always provided bias, steady-state MSE and MSD lower than those of second-order Volterra filters. For example, Figure 2 diagrams for FLANN and Volterra filters the MSE of the exact and approximate filtered-x AP algorithms, estimated with (12) or obtained from simulations with time averages over 1 billion samples, at different values of step-size μ and with AP order L=1, 2 and 3. It is apparent that the approximate AP algorithm outperforms the exact algorithm in terms of residual error but we must point out that it provides also a slower convergence speed. Indeed, Figure 3 diagrams the learning curves of the residual error of the FLANN structure adapted with the exact algorithm and the approximate algorithm with a step-size equal to 0.128. Each point of Figure 3 represents the ensemble average, estimated over 100 runs of the algorithm, of the mean value of the residual error computed on 100 successive samples. In the figure the asymptotic values (dashed lines) of the residual errors estimated with (12) are also shown. While the exact algorithm converges in less than 100,000 samples, the approximate algorithm

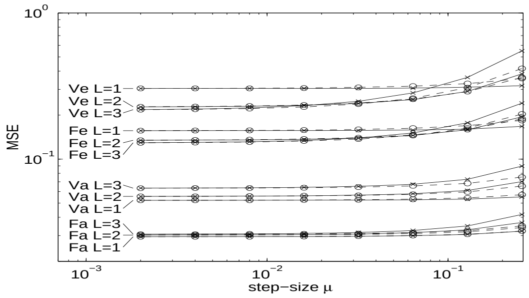

Figure 2: Theoretical (–) and simulation values (- -) of steady-state MSE with FLANN (F) and Volterra filters (V) for the exact (e) and approximate (a) algorithms versus step-size.

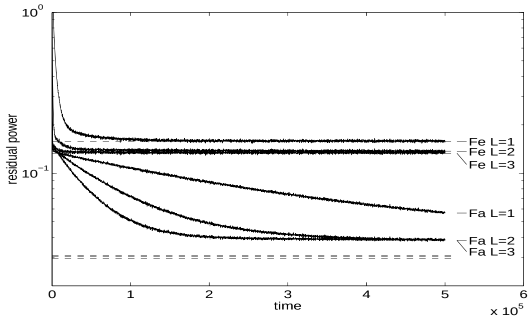

Figure 3: Evolution of the residual error with FLANN of the exact (Fe) and approximate (Fa) algorithm (–) and corresponding steady-state MSE values (- -).

reaches the steady-state MSE (the dashed line) only after several billions of samples. The learning curves of the residual error for Volterra filters provide similar convergence behavior. For each algorithm and for each AP order, the convergence speed of the Volterra controller is lower than that of the FLANN structure.

In conclusion, FLANN structures with P=1 can be profitably employed for nonlinear multichannel ANC with performances in most cases better than those of second-order Volterra filters.

5. REFERENCES

[1] D. Pavisic et alt., "Active noise control with dynamic recurrent neural networks," in *Proc. ESANN 95*, Brussels, Belgium, Apr. 1995, pp. 45–50.

[2] 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.

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

[4] D. P. Das and G. Panda, "Active mitigation of nonlinear noise processes using a novel filtered-s LMS algorithm," *IEEE Trans. Speech and Audio Processing*, vol. 12, no. 3, pp. 313–322, May 2004.

[5] A. Carini and G. L. Sicuranza, "Filtered-X affine projection algorithm for active noise control using second-order Volterra filters," *EURASIP Journal on Applied Signal Processing*, vol. 2004, no. 12, pp. 1841–1848, Sept. 2004.

[6] 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, no. 11, pp. 853–857, Nov. 2004.

[7] A. Carini and G. L. Sicuranza, "Steady-state and transient analysis of multichannel filtered-X affine projection algorithms," in *Proc. ICASSP, Int. Conference on Acoustic, Speech, Signal Processing*, Philadelphia, PA, Mar. 2005.

[8] A. Carini and G. L. Sicuranza, "Transient and steady-state analysis of filtered-X affine projection algorithms," *IEEE Trans. Signal Processing*, to be published.

[9] H. C. Shin and A. H. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Trans. Signal Processing*, vol. 52, no. 1, pp. 90–102, Jan. 2004.