

## Pseudo Affine Projection Algorithms for Multichannel Active Noise Control

Felix Albu, Martin Bouchard, and Yuriy Zakharov, *Members, IEEE*

### ABSTRACT

In this paper, we propose two new pseudo affine projection algorithms for multichannel active noise control (ANC) systems: one based on the Gauss-Seidel method and one based on dichotomous coordinate descent (DCD) iterations. It is shown that the proposed algorithms typically have a lower complexity than the previously published multichannel affine projection algorithms for ANC, with similar convergence properties and good numerical stability.

INDEX TERMS— multichannel active noise control, adaptive filtering, affine projection algorithms, Gauss-Seidel solving scheme, dichotomous coordinate descent

EDICS NUMBER: AUD-ANCO

Corresponding author: Felix Albu

Faculty of Electronics, Telecommunications and Information Technology, “Politehnica” University of Bucharest, 1-3 Bd. Iuliu Maniu, Sector 6, B117, Bucharest, Romania

(40) 21 3308449, email: felix\_albu@ieee.org

Martin Bouchard

School of Information Technology and Engineering, Colonel By Hall, University of Ottawa

161 Louis-Pasteur, Ottawa (Ontario), K1N 6N5

(613) 562-5800 ext 6190, e-mail: bouchard@site.uottawa.ca

Yuriy Zakharov

Electronics Department, University of York, Heslington, York YO10 5DD, U.K.

+44 (0) 1904 432396, e-mail: yz1@ohm.york.ac.uk

## I. INTRODUCTION

Active noise control (ANC) systems are being increasingly researched and developed [1]. In such systems, an adaptive controller is used to optimally cancel unwanted acoustic noise. The delay compensated modified filtered-x structure [2] for active noise control systems using FIR adaptive filtering is presented in Fig. 1. This figure shows a block-diagram of a monochannel implementation of feedforward active noise control. The system in Fig.1 is delay-compensated. Since the plant normally includes a propagation delay, there is typically a delay for the effect of changes in the adaptive FIR filter coefficients to become effective in the error sensor signals. The structure in Fig. 1 eliminates this delay by computing an estimate of the primary field signal (which is unaffected by the changes to the adaptive FIR filter coefficients), and by computing an alternative error signal from this estimated primary field signal. The use of this structure will be assumed in the rest of this paper. In the field of adaptive filtering, it is well known that fast affine projection (FAP) algorithms present a good tradeoff between convergence speed and computational complexity. FAP algorithms suitable for active noise control were introduced in [3]-[5]. For ideal (not noisy) plant models, these algorithms typically do not provide the same convergence speed as recursive least-squares (RLS) algorithms. However, they demonstrate a much improved convergence speed compared to stochastic gradient descent algorithms, without the high increase of the computational load or the numerical instability often found in RLS algorithms, especially for multichannel systems [4]-[5]. It was also reported that in the realistic case of noisy plant models (i.e. for algorithms using filtered-x types of structures, which require the use of plant models), FAP algorithms can be more robust to plant model noise than more complex RLS algorithms, and they can improve algorithm convergence performance at a lower cost [4]-[5].

In this paper, two new adaptive algorithms are introduced, both based on the Pseudo Affine Projection (PAP) algorithm, which is derived from the original affine projection algorithm by applying the Levinson-Durbin recursion [6]. Replacing the Levinson-Durbin recursion with the Gauss-Seidel method [7], a simpler algorithm was recently derived in [8], called the Gauss-Seidel Pseudo Affine Projection (GSPAP) algorithm. For a similar convergence performance, the complexity of the GSPAP algorithm is typically lower than that of the previous FAP schemes extended to multichannel ANC systems: the modified filtered-x FAP-RLS (MFX-FAP-RLS) algorithm in [4] and the modified filtered-x Gauss-Seidel Fast Affine Projection (MFX-GSFAP) algorithm in [5]. Therefore, in Section II, we propose an adaptation of the GSPAP algorithm to multichannel ANC systems, termed MFX-GSPAP algorithm, as preliminarily published in [9]. This new algorithm presents a typically reduced computational load, compared to both the MFX-GSFAP [5] and MFX-FAP-RLS [4] algorithms, and a similar convergence performance. However, the MFX-GSPAP still requires at least one inverse matrix computation. This can be very complex for large matrices and prone to numerical instability. Therefore, in Section III, a new algorithm for multichannel ANC systems, termed modified filtered-x Dichotomous Coordinate Descent Pseudo Affine Projection (MFX-DCDPAP) algorithm, is proposed. It is based on the dichotomous coordinate descent method used for solving linear systems [10-11]. This method does not require any matrix inversion, while it also exhibits a convergence speed similar to the previously published MFX-GSFAP and MFX-FAP-RLS algorithms. The computational complexity of the proposed algorithms is evaluated in Section IV. Results of simulations comparing the new proposed algorithms with previously published algorithms are presented in Section V. Section VI concludes this work.

## II. MULTICHANNEL MODIFIED FILTERED-X GAUSS SEIDEL PSEUDO AFFINE PROJECTION ALGORITHM

In the context of ANC systems, a monochannel feedforward system using an adaptive FIR filter with a modified filtered-x structure as in Fig. 1 and with filter weights adapted with a classical affine projection (AP) algorithm [12] can be described by the following equations (1)-(5) [3]-[4]:

$$y(n) = \mathbf{w}^T(n)\mathbf{x}(n) \quad (1)$$

$$v(n) = \mathbf{h}^T \mathbf{x}'(n) \quad (2)$$

$$\hat{d}(n) = e(n) - \mathbf{h}^T \mathbf{y}(n) \quad (3)$$

$$\hat{\mathbf{e}}_N^T(n) = \hat{\mathbf{d}}_N^T(n) + \mathbf{A}(n)\mathbf{w}(n) \quad (4)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \mathbf{A}^T(n) (\mathbf{A}(n)\mathbf{A}^T(n) + \delta \mathbf{I})^{-1} \hat{\mathbf{e}}_N^T(n) \quad (5)$$

The variable  $n$  refers to the discrete time. The column vectors  $\mathbf{x}(n) = [x(n), \dots, x(n-L+1)]^T$  and  $\mathbf{x}'(n) = [x(n), \dots, x(n-M+1)]^T$  consist of the last  $L$  and  $M$  samples of the reference sensor signal  $x(n)$ , respectively (refer to Fig. 1).  $L$  represents the length of the adaptive FIR filter, whose coefficients are represented by the column vector  $\mathbf{w}(n) = [w_1(n) \dots w_L(n)]^T$ , and  $M$  is the length of the (fixed) FIR filter modeling the plant between signals  $y(n)$  and  $e(n)$ , whose coefficients are described by the column vector  $\mathbf{h} = [h_1, \dots, h_M]^T$ . The column vector  $\mathbf{y}(n) = [y(n), \dots, y(n-M+1)]^T$  consists of the last  $M$  samples of the actuator signal  $y(n)$ .  $e(n)$  is the error sensor signal. The samples of the filtered reference signal  $v(n)$  are collected in the column vector  $\mathbf{v}(n) = [v(n) \dots v(n-L+1)]^T$  and the  $N \times L$  matrix  $\mathbf{A}(n) = [\mathbf{v}(n) \dots \mathbf{v}(n-N+1)]^T$ , where  $N$  is the affine projection order [12]. The row vectors  $\hat{\mathbf{d}}_N(n) = [\hat{d}(n) \dots \hat{d}(n-N+1)]$  and  $\hat{\mathbf{e}}_N(n) = [\hat{e}(n) \dots \hat{e}(n-N+1)]$  consist of estimates  $\hat{d}(n)$  of the primary sound field  $d(n)$  and of

alternative error signal samples  $\hat{e}(n)$ , both computed in delay-compensated modified filtered-x structures as in Fig.1. Finally,  $\mu$  is a normalized convergence gain  $0 \leq \mu \leq 1$ ,  $\mathbf{I}$  is an identity matrix of size  $N \times N$  and  $\delta$  is a regularization factor that may be used to help with eventual numerical instability.

In the adaptive filtering literature, it is well known that when the input signal of an adaptive filter trained with an Affine Projection algorithm is a time series (such as  $v(n)$ , which is the actual input signal of the adaptive filter in ANC structures as in Fig. 1), then the redundancy found in the input signal or the data matrix  $\mathbf{A}(n)$  can be exploited, and this results in a family of algorithms called the Fast Affine Projection (FAP) algorithms [13]-[14]. For an adaptive FIR filter of  $L$  coefficients, FAP algorithms include a set of  $N$  linear equations to be solved (with typically  $N \ll L$ ), and this set of equation is often solved by the use of a built-in recursive least-squares (RLS) algorithm [15] or some more efficient built-in fast-RLS algorithm [15] inside the FAP algorithm. In the context of ANC algorithms, a FAP algorithm with a built-in fast-RLS algorithm was first introduced for monochannel ANC systems in [3]. More recently, an AP algorithm and a FAP algorithm were introduced in [4] for the more general case of multichannel ANC. The FAP algorithm in [4] used a built-in RLS algorithm instead of a built-in fast-RLS algorithm as in [3], because the potential numerical instability of the fast-RLS algorithm proved to be even more of a concern in the multichannel case. Thus the FAP algorithm in [4] used the name FAP-RLS, or more precisely modified filtered-x FAP-RLS (MFX-FAP-RLS), since it also made use of the structure of Fig. 1.

In recent years, other schemes have been investigated to replace the built-in RLS or fast-RLS

algorithms inside the FAP algorithm, in order to improve the numerical stability of the algorithms and also possibly reduce the computational load. The Gauss-Seidel inversion scheme [7] is one of those schemes that were successfully applied to the FAP algorithm [16], and the adaptation of that Gauss-Seidel FAP algorithm (GSFAP) to multichannel ANC systems was recently published in [5] as the MFX-GSFAP algorithm, producing a lower complexity and a better numerical stability than the MFX-FAP-RLS, for the same convergence speed. As an attempt to further reduce the complexity, a Pseudo Affine Projection (PAP) algorithm was recently derived from the original AP algorithm by applying a Levinson-Durbin recursion [6]. Replacing the Levinson-Durbin recursion with the Gauss-Seidel method [7], a simpler algorithm was derived in [8], called the Gauss-Seidel Pseudo Affine Projection (GSPAP) algorithm. For a similar convergence performance, the complexity of the GSPAP algorithm is typically lower than for the FAP-RLS and GSFAP algorithms, as will be shown later in this paper. This section will present the extension of the GSPAP algorithm to the case of multichannel ANC systems: the MFX-GSPAP algorithm.

The derivation of the GSPAP algorithm in the context of monochannel ANC using the structure of Fig. 1 is first presented. Let's consider the linear prediction with order  $N-1$  (prediction filter with coefficients  $f_i$   $1 \leq i \leq N-1$ ) of the signal  $v(n)$ , based on the last  $L$  samples of  $v(n)$  and producing a residual prediction signal  $u(n)$ :

$$u(n) = \mathbf{f}^T \mathbf{v}_N(n) \quad (6)$$

with the column vectors  $\mathbf{v}_N(n) = [v(n), \dots, v(n-N+1)]^T$  and  $\mathbf{f} = [1, f_1 \dots, f_{N-1}]^T$ . The coefficients  $f_i$  could be computed from the last  $L$  samples of  $v(n)$  by solving the following Yule-Walker

equation [17]:

$$\mathbf{R}(n)\mathbf{f} = [\mathbf{v}^T(n)\mathbf{u}(n), 0, \dots, 0]^T \quad (7)$$

with  $\mathbf{u}(n) = [u(n), u(n-1), \dots, u(n-L+1)]^T$  and  $\mathbf{R}(n) = \mathbf{A}(n)\mathbf{A}^T(n)$ . Defining  $\mathbf{p}(n)$  as

$$\mathbf{p}(n) = \left[ \frac{1}{\mathbf{v}^T(n)\mathbf{u}(n)}, \frac{f_1}{\mathbf{v}^T(n)\mathbf{u}(n)}, \dots, \frac{f_{N-1}}{\mathbf{v}^T(n)\mathbf{u}(n)} \right]^T = \frac{\mathbf{f}}{\mathbf{v}^T(n)\mathbf{u}(n)},$$

the previous formula can be re-

arranged as:

$$\mathbf{R}(n)\mathbf{p}(n) = [1, 0, \dots, 0]^T \quad (8).$$

One single iteration from the Gauss-Seidel scheme [7] can be used to compute  $\mathbf{p}(n)$ , using as an initial condition for  $\mathbf{p}(n)$  the value of  $\mathbf{p}(n-1)$  from the previous iteration of the GSPAP algorithm.

More details on the implementation of the Gauss-Seidel scheme are provided later in this section.

With the knowledge of  $\mathbf{p}(n)$ , the prediction residual signal  $u(n)$  can be computed by:

$$u(n) = \mathbf{f}^T \mathbf{v}_N(n) = \mathbf{v}_N^T(n)\mathbf{f} = \mathbf{v}_N^T(n)\mathbf{p}(n)(\mathbf{v}^T(n)\mathbf{u}(n)) = \mathbf{v}_N^T(n)\mathbf{p}(n)\bar{p}^{-1}(n) \quad (9)$$

where  $\bar{p}(n)$  is the first component of  $\mathbf{p}(n)$  and  $\bar{p}^{-1}(n)$  is its inverse value.

For the case with  $\mu=1$  (referred to in the literature as case without relaxation), the previous equation

(5) for the AP algorithm can be re-written in terms of the prediction residual signal  $u(n)$  instead of

the original input signal  $v(n)$  [6], with the following simplified equation:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mathbf{u}(n)(\mathbf{u}^T(n)\mathbf{v}(n) + \delta)^{-1} \hat{e}(n) \quad (10)$$

where  $\hat{e}(n)$  is computed as a subset of (4):

$$\hat{e}(n) = \hat{d}(n) + \mathbf{v}^T(n)\mathbf{w}(n) \quad (11).$$

A more efficient implementation can be achieved by further re-arranging the equations. First, the direct computation of  $\mathbf{R}(n) = \mathbf{A}(n)\mathbf{A}^T(n)$  should be avoided and it can be evaluated instead by

considering the special structure of  $\mathbf{R}(n)$ :  $\mathbf{R}(n) = \begin{bmatrix} \bar{\mathbf{r}}(n) & \mathbf{r}^T(n) \\ \mathbf{r}(n) & \bar{\mathbf{R}}(n-1) \end{bmatrix}$ , with  $\bar{\mathbf{R}}(n)$  defined as the top

left  $(N-1) \times (N-1)$  values of  $\mathbf{R}(n)$ .  $\mathbf{r}(n)$  and  $\bar{\mathbf{r}}(n)$  can then be computed by:

$$\mathbf{r}(n) = \mathbf{r}(n-1) + \underline{\mathbf{v}}_N(n)v(n) - \underline{\mathbf{v}}_N(n-L)v(n-L) \quad (12)$$

$$\bar{\mathbf{r}}(n) = \bar{\mathbf{r}}(n-1) + v(n)v(n) - v(n-L)v(n-L) \quad (13)$$

where  $\underline{\mathbf{v}}_N(n)$  corresponds to the last  $N-1$  rows of  $\mathbf{v}_N(n)$  (i.e.

$\underline{\mathbf{v}}_N(n) = [v(n-1), \dots, v(n-N+1)]^T$ ). By introducing a scalar variable  $m(n)$  computed as:

$$m(n) = m(n-1) + u(n)v(n) - u(n-L)v(n-L) \quad (14)$$

then (10) can be re-written as:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \frac{1}{m(n) + \delta} \mathbf{u}(n)\hat{e}(n). \quad (15)$$

Re-introducing a relaxation factor  $\mu$  ( $0 \leq \mu \leq 1$ ) can reduce the error signal  $\hat{e}(n)$  in some situations,

but since the resulting algorithm with the  $\mu$  relaxation factor has not been formally derived with  $\mu$ ,

then the result is called a Pseudo Affine Projection algorithm (PAP) instead of a FAP algorithm:

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \frac{1}{m(n) + \delta} \mathbf{u}(n)\hat{e}(n). \quad (16).$$

To summarize, the equations describing the GSPAP for monochannel ANC are given by (1), (2),

(3), (8) through a Gauss-Seidel iteration, (9), (11), (12), (13), (14) and (16).

Those ten equations can be extended to the case of multichannel ANC systems, still using the

structure of Fig. 1 (i.e. delay-compensated modified filtered-x structure). This results in the multichannel modified filtered-x Gauss-Seidel Pseudo Affine Projection (MFX-GSPAP) algorithm described by the following equations:

$$y_j(n) = \sum_{i=1}^I \mathbf{w}_{i,j}^T(n) \mathbf{x}_i(n) \quad (17)$$

$$v_{i,j,k}(n) = \mathbf{h}_{j,k}^T \mathbf{x}'_i(n) \quad (18)$$

$$\hat{d}_k(n) = e(n) - \sum_{j=1}^J \mathbf{h}_{j,k}^T \mathbf{y}_j(n) \quad (19)$$

$$\hat{\mathbf{e}}^T(n) = \hat{\mathbf{d}}^T(n) + \mathbf{V}^T(n) \mathbf{w}(n) \quad (20)$$

$$\mathbf{R}_0(n) = \mathbf{R}_0(n-1) + \underline{\mathbf{A}}_0(n) \mathbf{V}_0(n) - \underline{\mathbf{A}}_0(n-L) \mathbf{V}_0(n-L) \quad (21)$$

$$\bar{\mathbf{R}}_0(n) = \bar{\mathbf{R}}_0(n-1) + \mathbf{V}_0^T(n) \mathbf{V}_0(n) - \mathbf{V}_0^T(n-L) \mathbf{V}_0(n-L) \quad (22)$$

$$\mathbf{R}(n) \mathbf{P}(n) = \mathbf{C} \quad (\text{to solve with Gauss-Seidel method})$$

(23)

$$\mathbf{U}_0(n) = \mathbf{A}_0^T(n) \mathbf{P}(n) \bar{\mathbf{P}}^{-1}(n) \quad (24)$$

$$\mathbf{M}(n) = \mathbf{M}(n-1) + \mathbf{U}_0^T(n) \mathbf{V}_0(n) - \mathbf{U}_0^T(n-L) \mathbf{V}_0(n-L) \quad (25)$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \mathbf{U}(n) \mathbf{M}^{-1}(n) \hat{\mathbf{e}}^T(n) \quad (26)$$

where the following additional notations are defined :  $I$  - number of reference sensors,  $J$  - number of actuators,  $K$  - number of error sensors, and the variables  $i = 1, \dots, I$ ,  $j = 1, \dots, J$ , and  $k = 1, \dots, K$  refer to the different reference sensors, actuators, and error sensors, respectively.  $\mathbf{R}(n)$  is now a  $KN \times KN$  auto-correlation matrix, initialized as an identity matrix multiplied by a regularization

factor  $\delta_1$ ,  $\mathbf{R}(n) = \begin{bmatrix} \bar{\mathbf{R}}_0(n) & \mathbf{R}_0^T(n) \\ \mathbf{R}_0(n) & \bar{\mathbf{R}}(n-1) \end{bmatrix}$ , where  $\bar{\mathbf{R}}(n)$  is the top left  $K(N-1) \times K(N-1)$  block of

$\mathbf{R}(n)$ , while  $\mathbf{R}_0(n)$  and  $\bar{\mathbf{R}}_0(n)$  are, respectively,  $K(N-1) \times K$  and  $K \times K$  correlation matrices initialized with zeros.  $\mathbf{P}(n)$  is an inverse  $KN \times K$  correlation matrix, while  $\bar{\mathbf{P}}(n)$  is the top  $K \times K$  block of  $\mathbf{P}(n)$ .  $\mathbf{C}$  is a  $KN \times K$  constant matrix whose elements are zeros, except for the top  $K \times K$  block set to an identity matrix. The  $IJL \times K$  matrix  $\mathbf{U}(n) = [\mathbf{U}_0(n), \dots, \mathbf{U}_0(n-L+1)]^T$  consists of decorrelated filtered reference signal matrices  $\mathbf{U}_0(n)$  of size  $IJ \times K$ .  $\mathbf{M}(n)$  is a  $K \times K$  inverse matrix initialized with an identity matrix multiplied by a regularization factor  $\delta_2$ . The vectors  $\mathbf{x}_i = [x_i(n), \dots, x_i(n-L+1)]^T$  and  $\mathbf{x}_i' = [x_i(n), \dots, x_i(n-M+1)]^T$  consist of the last  $L$  and  $M$  samples of the reference signal  $x_i(n)$ , respectively. The vector  $\mathbf{y}_j = [y_j(n), \dots, y_j(n-M+1)]^T$  consists of the last  $M$  samples of the actuator signal  $y_j(n)$ . The samples of the filtered reference signal  $v_{i,j,k}(n)$  are collected in the  $IJ \times K$ ,  $IJL \times K$  and  $KN \times IJ$  matrices

$$\mathbf{V}_0(n) = \begin{bmatrix} v_{1,1,1}(n) & \cdots & v_{1,1,K}(n) \\ \vdots & \ddots & \vdots \\ v_{I,J,1}(n) & \cdots & v_{I,J,K}(n) \end{bmatrix}, \quad \mathbf{V}(n) = [\mathbf{V}_0^T(n) \cdots \mathbf{V}_0^T(n-L+1)]^T, \quad \text{and}$$

$\mathbf{A}_0(n) = [\mathbf{V}_0(n) \cdots \mathbf{V}_0(n-N+1)]^T$ , while the matrix  $\underline{\mathbf{A}}_0(n)$  consists of the last  $K(N-1)$  rows of

$\mathbf{A}_0(n)$ . The  $1 \times K$  vectors  $\hat{\mathbf{d}}(n) = [\hat{d}_1(n), \hat{d}_2(n), \dots, \hat{d}_K(n)]$  and  $\hat{\mathbf{e}}(n) = [\hat{e}_1(n), \hat{e}_2(n), \dots, \hat{e}_K(n)]$  consist

of estimates  $\hat{d}_k(n)$  of the primary sound field  $d_k(n)$  and of alternative error signals samples  $\hat{e}_k(n)$ ,

both computed in delay-compensated modified filtered-x structures, as mentioned earlier. The  $M \times 1$

vector  $\mathbf{h}_{j,k} = [h_{j,k,1}, \dots, h_{j,k,M}]^T$  consists of taps  $h_{j,k,m}$  of the (fixed) FIR filter modeling the plant

between signals  $y_j(n)$  and  $e_k(n)$ . The  $IJL \times 1$  vector

$\mathbf{w}(n) = \left[ \left[ w_{1,1,1}(n) \dots w_{1,J,1}(n) \right] \dots \left[ w_{1,1,L}(n) \dots w_{1,J,L}(n) \right] \right]^T$  consists of the coefficients from all the

adaptive FIR filters linking the signals  $x_i(n)$  and  $y_j(n)$  ( $\forall i, j$ ). Finally,  $e_k(n)$  is the  $k^{\text{th}}$  error sensor

signal and  $\mu$  is again a normalized convergence gain,  $0 \leq \mu \leq 1$ .

To determine  $\mathbf{P}(n)$  needed in (24), equation (23) is solved using the Gauss-Seidel method [7] for each column of  $\mathbf{P}(n)$  and  $\mathbf{C}$ . To do this, equation (23) is transformed in  $K$  independent equations,

$k = 1, \dots, K$ , where  $\mathbf{p}_k(n)$  is the  $k$ th column of  $\mathbf{P}(n)$  and  $\mathbf{c}_k$  is the  $k$ th column of  $\mathbf{C}$ . Since the

filter length  $L$  is usually significantly bigger than the affine projection order  $N$ , i.e.  $L \gg N$ , the

correlation matrix  $\mathbf{R}(n)$  is slowly varying in time, as is the solution of the system (23). Therefore,

assuming that we have already obtained an accurate estimate of the vector  $\mathbf{p}_k(n-1)$  for the time

sample  $(n-1)$ , the vector  $\mathbf{p}_k(n-1)$  can be used as an initial condition in the Gauss-Seidel method.

This is equivalent to solving the system  $\mathbf{R}(n)\mathbf{p}_k(n) = \mathbf{c}_k$  with one GS iteration:

$$[\mathbf{p}_k(n)]_i = \frac{1}{[\mathbf{R}(n)]_{i,i}} \left( [\mathbf{c}_k]_i - \sum_{j=1}^{i-1} [\mathbf{R}(n)]_{i,j} [\mathbf{p}_k(n-1)]_j - \sum_{j=i+1}^{KN} [\mathbf{R}(n)]_{i,j} [\mathbf{p}_k(n)]_j \right) \quad (27)$$

where  $[\mathbf{c}_k]_i$  is the  $i$ th element of the vector  $\mathbf{c}_k$ ,  $[\mathbf{p}_k(n)]_i$  is the  $i$ th element of the vector  $\mathbf{p}_k(n)$ ,

$[\mathbf{R}(n)]_{i,j}$  is the  $(i, j)$ th element of the matrix  $\mathbf{R}(n)$ , and  $i = 1, \dots, KN$ . In the reference [5], it was

shown that one GS iteration per sample was enough for the MFX-GSPAP algorithm to achieve

approximately the same performance as that of the theoretically more accurate MFX-FAP-RLS

algorithm.

Even though  $\mathbf{P}(n)$  is computed recurrently by using Gauss-Seidel iterations, the MFX-GSPAP

(and also the MFX-GSFAP in [5]) computes  $\mathbf{P}(n)$  directly from the correlation matrix  $\mathbf{R}(n)$ , unlike the MFX-FAP-RLS [4] or other RLS-based algorithms. Therefore, it has the potential for an inherently better numerical stability. Moreover, in the MFX-GSPAP and MFX-GSFAP algorithms, it may not be required to invert  $\mathbf{R}(n)$  for each iteration of the algorithm (i.e.  $\mathbf{R}(n)$  has to be always updated but its inverse does not necessarily have to be computed for each iteration). This is not the case in the MFX-FAP-RLS or purely RLS-based algorithms, because the recurrent scheme for inverting  $\mathbf{R}(n)$  cannot miss any update without having undesirable effects caused by discontinuities. For a proper initialization of the MFX-GSPAP algorithm, at the first iteration of the algorithm  $v(n)$  should be non-zero and  $v(n-1)\dots v(n-L+1)$  (or  $v(n-1)\dots v(n-N+1)$ ) should all be zero.

The proposed MFX-GSPAP algorithm directly computes the adaptive filter coefficients  $\mathbf{w}$  in (26), used for the filtering between the reference sensors and the actuators in (1). This is unlike the previously published FAP algorithms for active noise control [3]-[5], which instead compute what is referred to as "auxiliary coefficients". The direct computation of the vector  $\mathbf{w}$  can be useful for a few reasons. For example, the only computation that must absolutely be done in real time in ANC applications is the computation of the actuator values in (1). The other computations could possibly be done offline, at a reduced rate, using recorded blocks of data (at the cost of having reduced tracking capabilities). Also, the direct access to the time domain coefficients can be interesting because it provides more physical insights into the control system, for example to observe its causality and the number of required coefficients, etc.

### III. MULTICHANNEL MODIFIED FILTERED-X DICHOTOMOUS COORDINATE DESCENT PSEUDO AFFINE PROJECTION ALGORITHM

The MFX-GSPAP algorithm introduced in the previous section provides both a direct estimation of the adaptive coefficients  $\mathbf{w}$  and a computation complexity typically lower than the previous FAP algorithms for multichannel ANC [4]-[5]. However, the MFX-GSPAP algorithm still requires at least one inverse matrix computation. This can be very complex for large matrices and prone to numerical instability. Therefore, in this section, a second new pseudo affine projection algorithm called the modified filtered-x Dichotomous Coordinate Descent Pseudo Affine Projection (MFX-DCDPAP) algorithm is introduced. It uses the first seven equations (17)-(23) of the MFX-GSPAP algorithm. However, for solving the linear system in (23), the DCD method [10-11] is used; this multiplication-free and division-less procedure is presented below.

Let a system of equations to be solved be  $\mathbf{R}(n)\mathbf{p}_k(n) = \mathbf{c}_k$ . The DCD algorithm is based on a binary representation of elements of the solution vector with  $M_b$  bits within an amplitude range  $[-H, H]$ . The iterative approximation of the solution vector  $\mathbf{p}_k(n)$  starts by updating the most significant bit of its elements and proceeds to less significant bits. If a bit update happens (such an iteration is called “successful”), the vector  $\mathbf{c}_k$  is also updated. The complexity of the method is mainly due to updates of the vector  $\mathbf{c}_k$ , i.e. due to the “successful” iterations. The parameter  $N_{upd}$ , that represents the maximum number of “successful” iterations, limits the algorithm complexity. Denote  $\mathbf{r}_i(n)$  the  $i$ th column of the matrix  $\mathbf{R}(n)$ . The DCD algorithm can be described as follows.

Initialization:  $\mathbf{p}_k(n) = \mathbf{0}$ ,  $d = H$ ,  $q = 0$ .

for  $m = 1 : M_b$

$d = d/2$   
 (a)  $flag = 0$   
 for  $i = 1:KN$   
     if  $|\lfloor \mathbf{c}_k \rfloor_i| > (d/2)\lfloor \mathbf{R}(n) \rfloor_{i,i}$ , then  
          $flag = 1, q = q + 1$   
          $\lfloor \mathbf{p}_k(n) \rfloor_i = \lfloor \mathbf{p}_k(n) \rfloor_i + \text{sgn}(\lfloor \mathbf{c}_k \rfloor_i) \cdot d$   
          $\mathbf{c}_k = \mathbf{c}_k - \text{sgn}(\lfloor \mathbf{c}_k \rfloor_i) \cdot d \cdot \mathbf{r}_i(n)$   
         if  $q > N_{upd}$ , then the algorithm stops  
     end of the  $i$ -loop  
     if  $flag = 1$ , then go to (a)  
 end of the  $m$ -loop

It can be seen from the algorithm description that if  $H$  is a power of two, then only multiplications by factors of power of two are used; these can be replaced by bit shifts [10]. Thus, the DCD algorithm can be implemented without explicit multiplications and divisions (well known as potential sources of numerical instability). The peak complexity of the DCD algorithm for given  $M_b$  and  $N_{upd}$  is

$$KN(2N_{upd} + M_b) \text{ shift-accumulation (SACs) operations} \quad (28)$$

To complete the MFX-DCDPAP algorithm, in addition to equations (17)-(23) and the DCD algorithm to solve the system (23), the modification proposed in [18] for the GSPAP algorithm is adapted to multichannel ANC systems. If the  $\mathbf{p}(n)$  estimate is exact and there is no regularization [18], the equation (10) with  $\mu$  included can be replaced by

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \frac{\mathbf{A}^T(n)\mathbf{p}(n)}{\mathbf{p}^T(n)\mathbf{A}(n)\mathbf{v}(n)} \hat{e}(n) = \mathbf{w}(n) - \mu \mathbf{A}^T(n)\mathbf{p}(n)\hat{e}(n) \quad (29)$$

Therefore, for the multichannel ANC systems the following computations are performed:

$$\hat{\mathbf{U}}_0(n) = \mathbf{A}^T(n)\mathbf{P}(n)$$

(30)

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu \hat{\mathbf{U}}(n)\hat{\mathbf{e}}^T(n) \quad (31)$$

where  $\hat{\mathbf{U}}(n) = [\hat{\mathbf{U}}_0(n), \dots, \hat{\mathbf{U}}_0(n-L+1)]^T$  and therefore is updated in a similar way to  $\mathbf{U}(n)$ . The remaining two equations of the MFX-DCDPAP algorithm are (30) and (31). The advantage of this new algorithm is that it avoids the matrix inverse, a known source of numerical instability. Also, it has a reduced complexity as will be shown in Section IV. It will also be verified in section V that the modifications proposed for the MFX-DCDPAP do not alter significantly the convergence and steady-state properties of the MFX-DCDPAP algorithm, compared to the MFX-GSPAP algorithm when ideal plant models are used.

Just like the MFX-GSPAP algorithm introduced in the previous section, the MFX-DCDPAP directly computes the adaptive filter coefficients  $\mathbf{w}$  in (31), used for the filtering between the reference sensors and the actuators in (1). It thus presents the same advantages as the ones described at the end of Section II for the MFX-GSPAP algorithm.

#### IV. COMPUTATIONAL COMPLEXITY

The computational complexity of the algorithms considered was estimated by the number of multiplications required per iteration. Matrix inversions were assumed to be performed with standard

LU decomposition that requires  $O\{X^3/2\}$  multiplications, where  $X$  is the size of a square matrix.

The number of multiplications per MFX-GSFAP algorithm iteration is [5]:

$$IJK(2L+M+2(N-1)(K+J)+K+1)+K^2(2N-1) \\ +J(KM+K(N-1)+IL)+KN+K^3N^2/p$$

(32)

where  $p$  is the update period of the GS algorithm [5]. The number of multiplications per MFX-GSPAP algorithm iteration is:

$$IJK(M+2L+3KN+2K+1)+IJL+JKM+K^3\left(\frac{1}{2}+\frac{N^2}{p}\right)+K^2 \quad (33)$$

The number of multiplications per MFX-DCDPAP algorithm iteration is:

$$IJK(M+2L+3KN)+IJL+JKM \quad (34)$$

It can be seen that the MFX-GSPAP algorithm has  $IJK(2K+1)+K^3\left(\frac{1}{2}+\frac{N^2}{p}\right)+K^2$  more multiplications per iteration than the MFX-DCDPAP algorithm. This increase of the numerical complexity is approximately proportional with  $K^2$  and can be reduced by updating less frequently the GS part of the algorithm. For the MFX-GSPAP and MFX-GSFAP algorithms, updating less frequently the GS part produces a reduction of about  $\frac{p-1}{p}K^3N^2$  multiplications and therefore can be important, especially for high values of  $K$  and  $N$ . The performance of those two algorithms is only slightly reduced when the update of the solution for the linear system in (23) is not done at the sample rate [5], [9]. If a proper regularization factor is used, a value of  $p$  up to  $p=10$  can be used safely for updating the solution of (23), without having signs of instability and with an average loss of less than 1 dB in convergence. Updating less frequently the solution of (23) in the MFX-DCDPAP

algorithm doesn't change its complexity given by (34) in terms of multiplications, since this part is computed only with additions and shift operations.

Table 1 evaluates the complexity of the two new introduced PAP algorithms, compared with previously published LMS and FAP based algorithms for multichannel ANC systems [4], [5], [19]. The values between accolades correspond to  $p=10$ . It can be seen that all the affine projection derived algorithms are only slightly more complex than the benchmark MFX-LMS algorithm. For the chosen parameters ( $I, J, K, L, M, N$ ) the complexity of the MFX-DCDPAP algorithm is lower than that of the MFX-GSPAP algorithm updated at the sample rate, both of them being less complex than the MFX-GSFAP algorithm. This is the typical situation in most cases, but for some particular parameter values, especially for low projection orders  $N$  and high updating factors  $p$ , the MFX-GSFAP algorithm can be slightly less complex than the MFX-GSPAP algorithm. If  $I, J, K, L, M$  have fixed values, only the MFX-DCDPAP algorithm complexity is proportional with  $N$ , while the complexities of the MFX-GSPAP and MFX-GSFAP algorithms are proportional with  $N^2$ . It can be seen from Fig. 2 that the MFX-DCDPAP algorithm is the most efficient, particularly for high values of the projection order  $N$ . Similar conclusions can be obtained if  $I, J, L, M, N$  are fixed and  $K$  is variable.

The last two columns of Table 1 give a performance/cost ratio obtained from the attenuation achieved by the algorithms after 50,000 iterations (averaged over the last 5000 iterations) for ideal and noisy plant models, divided by the number of multiplications per iteration. It can be seen that the proposed MFX-DCDPAP and MFX-GSPAP algorithms provide the best performance/cost ratio for the considered ANC system parameters. It can also be seen from Table 1 that the performance/cost ratios decrease when noisy plant models are used, although the two new proposed

algorithms still produce the best performance.

## V. SIMULATIONS

The new MFX-GSPAP and MFX-DCDPAP algorithms were simulated and compared to the previously published multichannel modified filtered-x LMS algorithm (MFX-LMS, [19]) and the multichannel modified filtered-x GS-FAP algorithm (MFX-GSFAP, [5]). We used in our simulation  $I=1$ ,  $J=3$ ,  $K=2$  and the reference signal was a white noise with zero mean and variance one. The simulations were performed with acoustic transfer functions experimentally measured in a duct. The impulse responses used for the multichannel acoustic plant had 64 samples each ( $M=64$ ), while the adaptive filters had 100 coefficients each ( $L=100$ ). For all the affine projection algorithms, a value of 0.9 was used for the step size  $\mu$  and the regularization factors were  $\delta = \delta_2 = 2 \cdot 10^3$  for the ideal case (clean plant models) and  $\delta = 10^4$ ,  $\delta_2 = 2 \cdot 10^3$  for plant models with a signal to noise (SNR) ratio of 10 dB. The step size  $\mu$  for the MFX-LMS algorithm was  $2 \cdot 10^{-5}$  and the parameter  $H$  of the DCD algorithm was set to  $1/128$ . The convergence performances have been averaged over 200 simulations. The performance of the algorithms was measured by

$$Attenuation(dB) = 10 \cdot \log_{10} \frac{\sum_k E[e_k^2(n)]}{\sum_k E[d_k^2(n)]} \quad (35)$$

It was found by simulations that a projection order of size  $N=5$  was sufficient for (fast or pseudo) affine projection algorithms to get a significantly improved convergence performance over the LMS algorithm. Fig. 3 shows that the implementation using 12 DCD iterations and 16 bits provides almost identical performance with the method using the ideal matrix inverse. In this case the

theoretical peak complexity of the DCD algorithm is 200 SACs. Also, it can be seen in the same figure, that if an average loss of about 1 dB is allowed, the number of bits can be reduced to 8 and the peak DCD complexity to 160 SACs. However, the average DCD complexity is around 60% of the theoretical peak complexity in both cases (125 and 90, respectively). The DCD part increases the number of additions, but has no divisions or multiplications. Therefore,  $N_{upd} = 12$  and  $M_b = 16$  were used in the following simulations of the MFX-DCDPAP algorithm.

Fig. 4 compares the performance of the selected algorithms, with ideal plant models, for a multichannel ANC system, obtained from Matlab™ implementations of the algorithms (double precision 64 bits floating point format). It can be seen that the MFX-GSPAP and MFX-DCDPAP algorithms have almost the same performance as the previously published MFX-GSFAP algorithm. As expected, their convergence performance is also better than that of the LMS-based algorithm.

Fig.5 shows the performance when plant models with a 10 dB SNR were used. In this case, noise was added on a frequency by frequency basis to the ideal plant models, i.e. in the frequency response a random complex value with a magnitude of 10 dB less than the original magnitude was added to each frequency. It can be seen from Fig. 5 that, when 10 dB SNR models were used, the MFX-GSPAP algorithm outperformed the other algorithms. Therefore, the approximation used in deriving the MFX-DCDPAP algorithm reduces its robustness and performance with noisy plant models in comparison with that of the MFX-GSPAP algorithm.

## VI. CONCLUSION

The multichannel MFX-GSPAP and MFX-DCDPAP algorithms were introduced for practical active

noise control systems using FIR adaptive filtering. Both algorithms provide a significant improvement of the convergence speed over the MFX-LMS algorithm, with a similar computational complexity. It was shown that the proposed algorithms provide an excellent performance/cost ratio and are good candidates for practical real-time implementations. The advantage of the MFX-DCDPAP algorithm is its reduced numerical complexity, especially for high projection orders. On the other hand, the MFX-GSPAP has shown a better performance when used with noisy plant models.

## VII. ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their useful comments

## REFERENCES

- [1] S.M.Kuo and D.R. Morgan, "Active noise control: a tutorial review", *Proc. of the IEEE*, vol. 87, no. 6, pp. 943-973, June 1999
- [2] E. Bjarnason "Active noise cancellation using a modified form of the filtered-x LMS algorithm", *Proc. 6th Eur. Signal Processing Conf.*, Brussels, Belgium, vol. 2, pp.1053–1056, Aug. 1992
- [3] S.C.Douglas, "The fast affine projection algorithm for active noise control", *Proc. 29th Asilomar Conf. Sign., Syst., Comp.*, Pacific Grove (CA), vol.2, pp.1245-1249, Oct. 1995
- [4] M. Bouchard "Multichannel Affine and Fast Affine Projection Algorithms for Active Noise Control and Acoustic Equalization Systems", *IEEE Trans. on Speech and Audio Processing*, v.11, no.1, pp. 54-60, Jan. 2003

- [5] M. Bouchard, F. Albu, "The Gauss-Seidel fast affine projection algorithm for multichannel active noise control and sound reproduction systems", *Special Issue on Adaptive Control of Sound and Vibration, International Journal of Adaptive Control and Signal Processing*, vol. 19, no. 2-3, pp. 107-123, March-April 2005
- [6] F. Bouteille, P. Scalart, M. Corazza, "Pseudo Affine Projection Algorithm New Solution for Adaptive Identification", *Proc. of Eurospeech 1999*, Budapest, Hungary, vol. 1, pp. 427-430, Sept. 1999
- [7] R. Barret et al. *Templates for the solutions of linear systems: Building blocks for iterative methods*, 2<sup>nd</sup> ed., SIAM, Philadelphia (PA), 1994.
- [8] F. Albu, A. Fagan, "The Gauss-Seidel Pseudo Affine Projection Algorithm and its Application for Echo Cancellation", *Proc. of 37th Asilomar Conf. Sign., Syst., Comp.*, Pacific Grove (CA), vol. 2, pp. 1303-1306, Nov. 2003
- [9] F. Albu and M. Bouchard, "A low-cost and fast convergence Gauss-Seidel pseudo affine projection algorithm for multichannel active noise control", *Proc. of ICASSP 2004*, Montreal, Canada, vol. 4, pp. 121-124, May 2004
- [10] Y. V. Zakharov and T. C. Tozer, "Multiplication-free iterative algorithm for LS problem," *Electronics Letters*, vol. 40, no. 9, pp. 567-569, April 2004.
- [11] Y. Zakharov and F. Albu, "Coordinate descent iterations in fast affine projection algorithm", *IEEE Signal Processing Letters*, vol. 12, no. 5, pp. 353-356, May 2005
- [12] K. Ozeki, and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties", *Elec. Comm. Japan*, vol. J67-A, pp.126-132, Feb. 1984

- [13] S.L. Gay and S. Tavathia, "The fast affine projection algorithm", *Proc. of ICASSP 1995*, Detroit (MI), May 1995, vol. 5, pp. 3023-3026
- [14] M. Tanaka, Y. Kaneda, S. Makino, and J. Kojima, "Fast projection algorithm and its step size control", *Proc. of ICASSP 1995*, Detroit (MI), May 1995, vol. 2, pp. 945-948
- [15] S. Haykin, *Adaptive filter theory*, 3<sup>rd</sup> edition, Englewood Cliffs (NJ), Prentice Hall, 989p., 1996
- [16] F. Albu, J. Kadlec, N. Coleman, and A. Fagan, "The Gauss-Seidel Fast Affine Projection Algorithm", *Proc. of SIPS 2002*, San Diego (CA), USA, pp.109-114, 2002.
- [17] J.G. Proakis and D.G. Manolakis, *Digital signal processing: principles, algorithms and applications*, 3rd edition, Prentice Hall, 1996
- [18] H. Sheikhzadeh, K. L. Whyte, and R. L. Brennan, "Partial update subband implementation of complex pseudo-affine projection algorithm on oversampled filterbanks", *Proc. of ICASSP 2005*, Philadelphia, U.S.A, vol. 4, pp. 373-376, April 2005
- [19] M. Bouchard and S. Quednau, "Multichannel recursive least-squares algorithms and fast-transversal-filter algorithms for active noise control and sound reproduction systems," *IEEE Trans. Speech Audio Processing*, vol. 8, no.5, pp.606-618, Sept. 2000.

## FIGURES

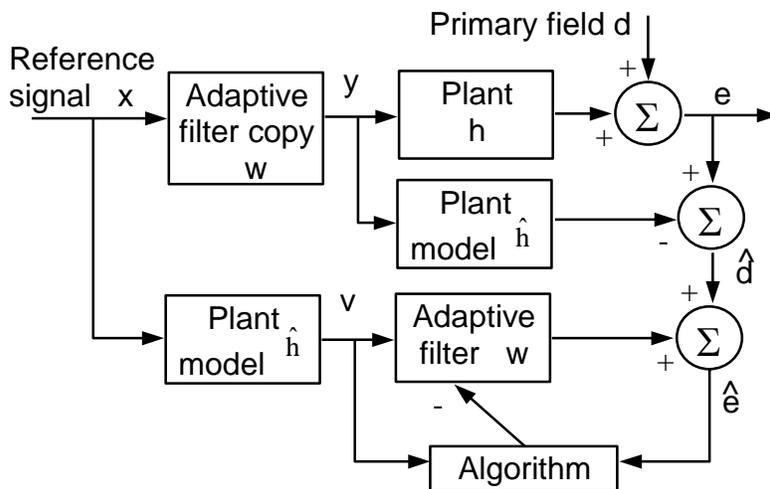


Fig 1 (Albu)

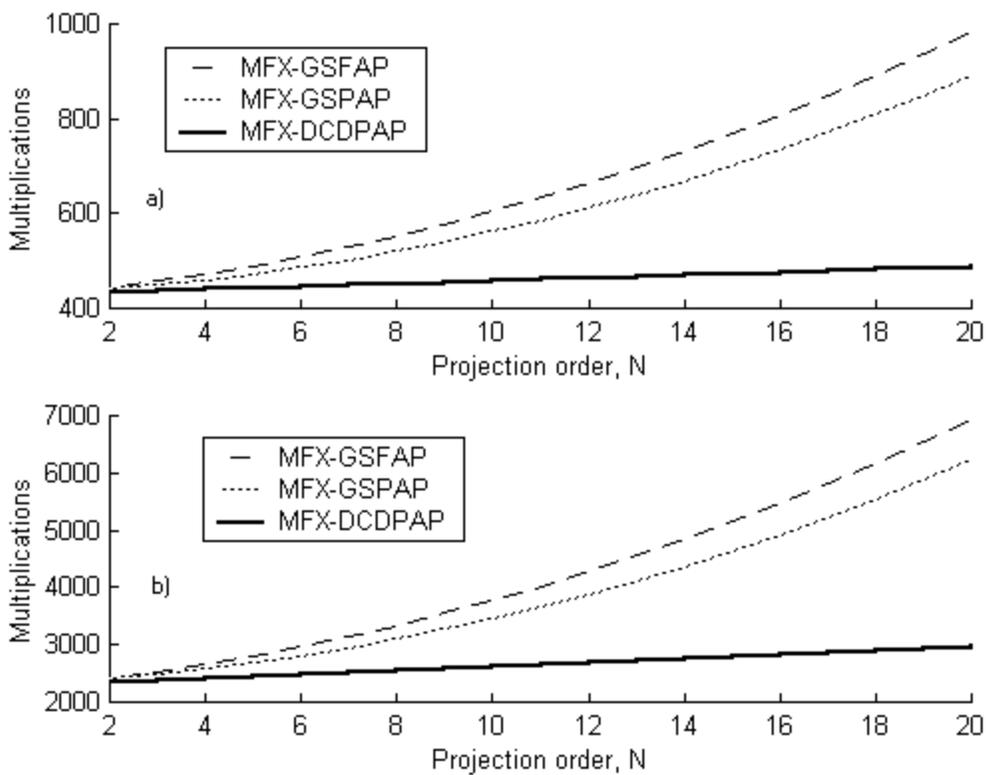


Fig 2 (Albu)

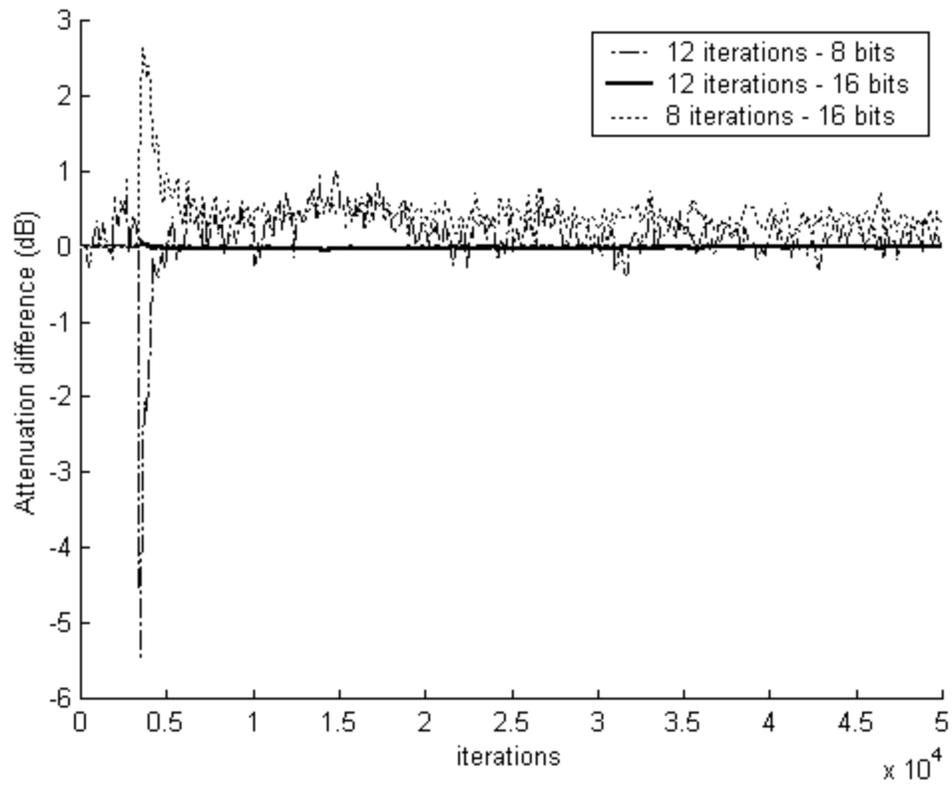


Fig. 3 (Albu)

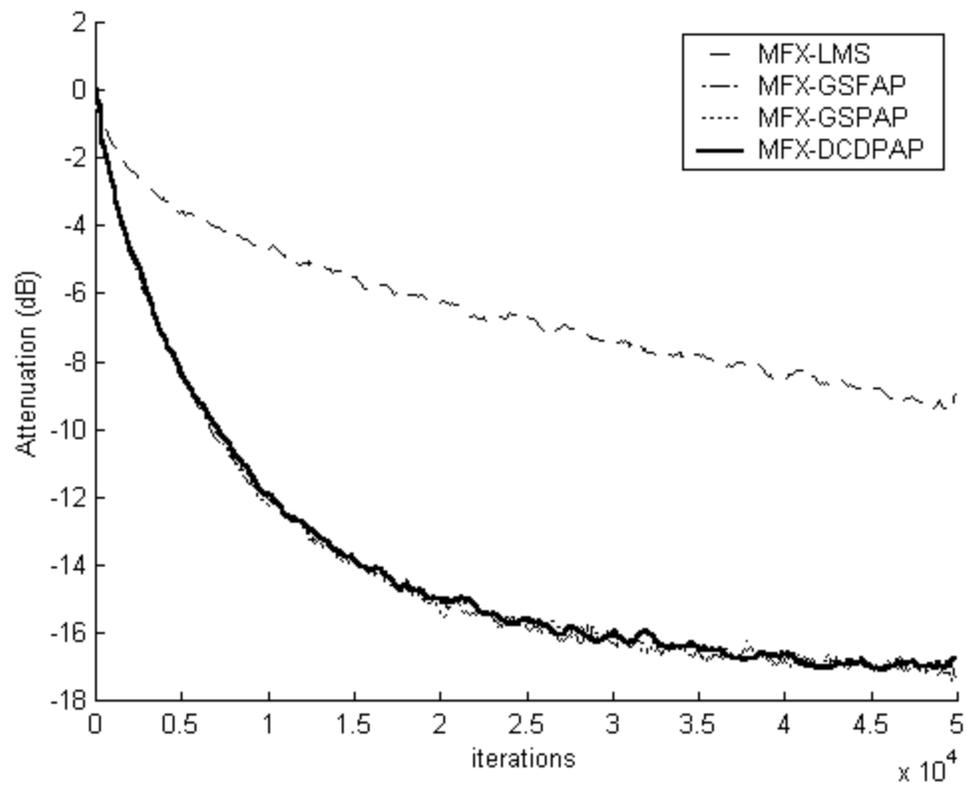


Fig. 4 (Albu)

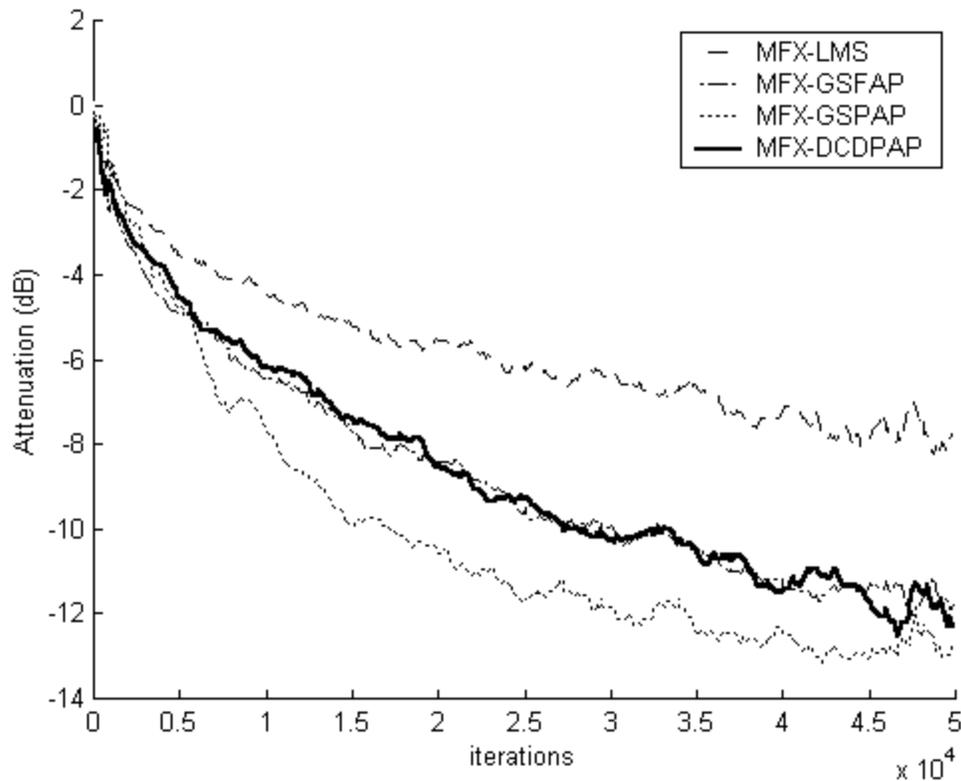


Fig. 5 (Albu)

## FIGURE CAPTIONS

Fig.1. A delay compensated modified filtered-x structure for active noise control.

Fig.2. The number of multiplications per algorithm iteration for the MFX-GSFAP, MFX-GSPAP and MFX-DCDPAP algorithms and for variable projection orders with two situations: a)  $I=1, J=1, K=1, L=100, M=64, p=1$  and b)  $I=1, J=3, K=2, L=100, M=64, p=1$ .

Fig.3. The attenuation difference over 50,000 iterations between the convergence curves of the algorithm using the ideal matrix inverse and the algorithm using different numbers of DCD iterations and bits.

Fig.4. Convergence curves of multichannel delay -compensated modified filtered-x algorithms for ANC, with ideal plant models

Fig.5. Convergence curves of multichannel delay -compensated modified filtered-x algorithms for adaptive noise control, with 10 dB SNR models.

### TABLES

Algorithm for multichannel ANC systems, $L=100, M=64, N=5$	Multiplications per iteration for $I=1, J=1, K=1$	Multiplications per iteration for $I=1, J=3, K=2$	Performance/cost ratio after 50000 iterations, for $I=1, J=3, K=2, L=100, M=64$ and ideal plant models (dB/multiplication )	Performance/cost ratio after 50000 iterations, for $I=1, J=3, K=2, L=100, M=64, 10$ dB SNR plant models (dB/multiplication)
MFX-LMS [19]	428	2,268	-4.0E-03	-3.4E-03
MFX-DCDPAP	443	2,448	-6.9E-03	-4.9E-03
MFX-GSPAP	473 (451)	2,686 (2,506)	-6.3E-03 (-6.6E-03)	-4.7E-03 (-4.9E-03)
MFX-GSFAP [5]	479 (457)	2,796 (2,616)	-6.1E-03 (-6.3E-03)	-4.1E-03 (-4.2E-03)

Table 1 (Albu)

### TABLE CAPTIONS

Table 1: Comparison of the computational load of the MFX-GSPAP and MFX-DCDPAP algorithms with other multichannel delay-compensated modified filtered-x algorithms for multichannel ANC systems, and evaluation of a performance/cost ratio for  $p=1$  (the values between accolades corresponds to  $p=10$ ).

The codes for the proposed algorithms can be obtained from

[http://falbu.50webs.com/List\\_of\\_publications\\_anc.htm](http://falbu.50webs.com/List_of_publications_anc.htm)

The reference for the paper is:

F. Albu, M. Bouchard, Y. Zakharov, “Pseudo Affine Projection Algorithms for Multichannel Active Noise Control”, IEEE Transaction on Audio, Speech and Language Processing, Vol. 15, Issue 3, March 2007, pp. 1044-1052