MODIFIED FILTERED-X DICHOTOMOUS COORDINATE DESCENT RECURSIVE AFFINE PROJECTION ALGORITHM

Felix Albu[1], *Member, IEEE,* Yuriy Zakharov[2], *Senior Member, IEEE*, and Constantin Paleologu[1], *Member IEEE*

[1]Faculty of Electronics and Telecommunications, "Politehnica" University of Bucharest,
1-3 Iuliu Maniu, Bucharest, Romania, e-mail: {felix, pale}@comm.pub.ro
[2] Electronics Department, University of York
Heslington, York, YO10 5DD, U.K. yz1@ohm.york.ac.uk

ABSTRACT

In this paper, we propose a new multichannel filtered-x affine projection algorithm based on dichotomous coordinate descent (DCD) iterations for active noise control (ANC) systems. It includes a fast recursive filtering procedure with the filter update incorporated in the DCD iterations. It is shown that the proposed algorithm has a lower complexity, and superior convergence properties than the multichannel filtered-x LMS algorithm. Also, it compares favorably to a previously published DCD based algorithm for ANC systems.

Index Terms— adaptive filters, adaptive signal processing, acoustic applications, least mean square methods

1. INTRODUCTION

Active noise control (ANC) systems have been increasingly researched and developed [1]. In such systems, an adaptive controller is used to optimally cancel unwanted acoustic noise. The use of the modified filtered-x structure for ANC using FIR adaptive filtering [2] will be assumed in the rest of this paper (see Fig. 1).

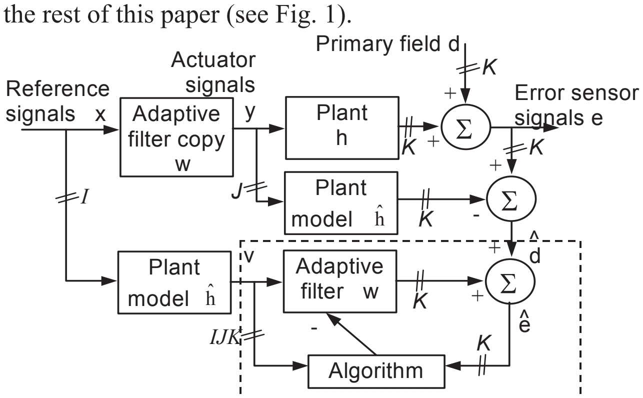

Fig. 1. A delay compensated (or modified filtered-x) structure for active noise control.

The multichannel versions of the filtered-x LMS (FX-LMS) or the modified filtered-x LMS (MFX-LMS) algorithms are the benchmarks to which most adaptive filtering algorithms are compared, because they are widely used [1]-[2]. In the field of adaptive filtering for ANC it is well known that fast affine projection (FAP) algorithms (firstly proposed by Gay and Tavathia [3]) can produce a good tradeoff between convergence speed and computational complexity in ANC systems [4]-[5]. The numerical complexity of affine projection (AP) algorithms for ANC can be further reduced by using the Dichotomous Coordinate Descent (DCD) method proposed in [6]. In [7] it was shown that the Modified Filtered-x Dichotomous Coordinate Descent Affine Projection (MFX-DCDAP) algorithm has similar performance with the more complex Modified Filtered-x Affine Projection (MFX-AP) algorithm. An even simpler version based on approximation of the affine projection, called the Modified Filtered-x Dichotomous Coordinate Descent Pseudo Affine Projection (MFX-DCDPAP) algorithm, has been investigated in [8]. In [9] a novel recursive filtering technique and filtering update that is incorporated in DCD iterations that leads to an important reduction in the number of multiplications is proposed for the AP algorithm.

In section 2, a new algorithm for multichannel active noise control systems called the Modified Filtered-x Dichotomous Coordinate Descent Recursive Affine Projection (MFX-DCDRAP) algorithm is proposed. It uses a variant of the DCD algorithm called the DCD algorithm with a leading element [10]. The computational complexity of the proposed algorithm is evaluated and compared with other algorithms in Section 3. Simulation results comparing the new proposed algorithm with previously published algorithms are presented in Section 4. Section 5 concludes this work.

2. MFX-DCDRAP ALGORITHM

In order to describe the algorithm most of the notations and definitions from [7] are used. The variable *n* refers to the discrete time, I is the number of reference sensors, J represents the number of actuators, K is the number of error sensors, L is the length of the adaptive FIR filters, M is the length of FIR filters modeling the plant, N is the projection order.

The vectors $\mathbf{x}_i = [x_i(n),..., x_i(n-L+1)]^T$ and $\mathbf{x}_i^{'} = [x_i(n),..., x_i(n-M+1)]^T$ consist of the last L and M samples of the reference signal $x_i(n)$, respectively. The vector $\mathbf{y}_j = [y_j(n),..., y_j(n-M+1)]^T$ 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 a $IJ \times K$ matrix $\mathbf{V}_0(n) = \begin{bmatrix} v_{1,1,1}(n) \ldots v_{1,1,K}(n) \\ .... \quad ...... \\ v_{I,J,1}(n) \ldots v_{I,J,K}(n) \end{bmatrix}$, $IJL \times K$ matrix $\mathbf{V}_1(n) = \left[\mathbf{V}_0^T(n),..., \mathbf{V}_0^T(n-L+1)\right]$, and $IJL \times KN$ matrix $\mathbf{V}(n) = [\mathbf{V}_1(n) \cdots \mathbf{V}_1(n-N+1)]$.

Vectors $\hat{\mathbf{d}}(n) = [\hat{d}_1(n), \hat{d}_2(n),..., \hat{d}_K(n)]$ and $\hat{\mathbf{e}}(n) = [\hat{e}_1(n), \hat{e}_2(n),..., \hat{e}_K(n)]$ consist of estimates $\hat{d}_k(n)$ of the primary sound field $d_k(n)$ and alternative error signals samples $\hat{e}_k(n)$, both computed in delay-compensated modified filtered-x structures.

Vectors $\hat{\mathbf{D}}(n) = \left[\hat{\mathbf{d}}(n), \hat{\mathbf{d}}(n-1),..., \hat{\mathbf{d}}(n-N+1)\right]$ and $\hat{\mathbf{E}}(n) = [\hat{\mathbf{e}}(n), \hat{\mathbf{e}}(n-1),..., \hat{\mathbf{e}}(n-N+1)]$ have both $1 \times KN$ size [6]. The vectors $\mathbf{h}_{j,k} = [h_{j,k,1},..., h_{j,k,M}]^T$ consist of taps $h_{j,k,m}$ of the fixed FIR filter modelling the plant between signals $y_j(n)$ and $e_k(n)$. A $IJL \times 1$ vector $\mathbf{w}(n) = [[w_{1,1,1}(n) \ldots w_{I,J,1}(n)] \ldots [w_{1,1,L}(n) \ldots w_{I,J,L}(n)]]$ consists of taps of all the adaptive FIR filters linking the signals $x_i(n)$ and $y_j(n)$. $\mathbf{R}(n)$ is a $KN \times KN$ auto-correlation matrix , $\mathbf{P}(n)$ and $\mathbf{Z}(n)$ are $KN \times 1$ sized initially null vectors, $\mathbf{I}$ is a $KN \times KN$ identity matrix , δ is a regularisation factor and μ is a normalized convergence gain**.** $\mathbf{Y}(n)$ is a $KN \times 1$ sized initial null vector and $\overline{\mathbf{Y}}(n)$ is a vector that keeps the upper $K(N-1) \times 1$ elements of $\mathbf{Y}(n)$.

In the context of ANC systems, a multichannel feedforward system using an adaptive FIR filter with a modified filtered-x structure and with filter weights adapted with a classical AP algorithm can be described by the following equations [5]:

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

$$v_{i,j,k}(n) = \mathbf{h}_{j,k}^T \mathbf{x}_i^{'}(n) \tag{2}$$

$$\hat{d}_k(n) = e_k(n) - \sum_{j=1}^{J} \mathbf{h}_{j,k}^T \mathbf{y}_j(n) \tag{3}$$

$$\hat{\mathbf{E}}^T(n) = \hat{\mathbf{D}}^T(n) + \mathbf{V}^T(n)\mathbf{w}(n) \tag{4}$$

$$\mathbf{w}(n+1) = \mathbf{w}(n) - \mu\mathbf{V}(n)\left(\mathbf{V}^T(n)\mathbf{V}(n) + \delta\mathbf{I}\right)^{-1}\hat{\mathbf{E}}^T(n) \tag{5}$$

Using the original DCD-AP algorithm from [9] and extending the fast recursive techniques and filtering update to multichannel ANC systems as in [5], the multichannel MFX-DCDRAP algorithm for ANC is obtained. The filtering step (4) takes into account previous computed values according to the following equations:

$$\mathbf{Z}(n) = \left[\mathbf{V}_0^T(n)\mathbf{w}(n\text{-}1)\overline{\mathbf{Y}}(n-1)\right] \tag{6}$$

$$\mathbf{G}(n) = \mathbf{V}^T(n)\mathbf{V}(n-1) \tag{7}$$

$$\mathbf{Y}(n) = \mathbf{Z}(n) - \mathbf{G}(n)\mathbf{P}(n-1) \tag{8}$$

$$\hat{\mathbf{E}}^T(n) = \hat{\mathbf{D}}^T(n) + \mathbf{Y}(n) \tag{9}$$

where $\mathbf{G}(n)$ is a $KN \times KN$ matrix.

The filter update (5) is performed by solving the following linear system of equations [7]:

$$(\mathbf{R}(n) + \delta\mathbf{I}) \cdot \mathbf{P}(n) = \hat{\mathbf{E}}^T(n) \tag{10}$$

using the DCD method with a leading element (Table 1), where $\mathbf{R}^{(p)}(n)$ denotes the pth column of the matrix $\mathbf{R}(n)$.

The only values of $\mathbf{R}(n)$ that require calculations are the upper left $K \times K$ elements given by $\mathbf{V}_0^T(n)\mathbf{V}_0(n)$. The other elements of $\mathbf{R}(n)$ can be taken from $\mathbf{R}(n-1)$ and $\mathbf{G}(n)$. Specifically, elements $[\mathbf{R}(n)]_{i,j}, i, j = K+1,..., KN$ are taken from $[\mathbf{R}(n-1)]_{i,j}, i, j = 1,..., K(N-1)$. The elements $[\mathbf{R}(n)]_{i,j}, i = 1,..., K, j = K+1,..., KN$ and $[\mathbf{R}(n)]_{i,j}, i = K+1,..., KN, j = 1,..., K$ are taken from $[\mathbf{G}(n)]_{i,j}, i = 1,..., K, j = K+1,..., KN$. The MFX-DCDRAP algorithm is described by equations (1)-(3), (6)-(10).

The MFX-DCDPAP algorithm uses the original DCD algorithm [6], while the MFX-DCDRAP uses a DCD version with a leading element [10]. The original DCD algorithm updates a solution of a linear system of equations in directions of Euclidian coordinates in the cyclic order and with a step size α that takes one of M_b (number of bits) predefined values corresponding to a binary representation bounded by an interval $[-H, H]$ [6], [9]. The algorithm starts the iterative search from the most significant bits of the solution and continues until the least significant bits were updated. The algorithm complexity is limited by N_u, the maximum number of "successful" iterations [6]. The comparisons are counted as additions, as shown in [10]. With N_u updates, the number of additions of the leading element DCD version is upper limited by $2N_u N + M_b$, while the complexity of the original DCD version is upper limited by $N(N_u + 2M_b - 1) + M_b + 1$ additions. For $M_b = 16$ (which is a typical number of bits used for representation of filter taps) and $N_u < 32$, the maximum number of additions in the DCD algorithm with a leading element is less than that in the original DCD version. It can be seen from Table 1 that the filtering update is incorporated in the DCD procedure, thus resulting in reduction of the number of multiplications per iteration compared to the previous MFX-DCDAP or MFX-DCDPAP algorithm. The memory requirements of the MFX-DCDRAP algorithm are higher than that of the MFX-DCDPAP algorithm because several matrices and vectors from previous iterations are needed at the following iteration.

Initialization: $\mathbf{P}(n) = 0, \mathbf{r} = \mu\hat{\mathbf{E}}(n), \alpha = H/2, m = 1$
For $k = 1,...,N_u$
$p = \arg\max_{i=0,...,KN-1}\{\lvert r_i \rvert\}$
while $\lvert r_p \rvert \le (\alpha/2)[\mathbf{R}(n)]_{p,p}$ & $m \le M_b$
$m = m+1, \alpha = \alpha/2$
if $m > M_b$, go to Eq. (8)
$P_p = P_p + \text{sign}(r_p)\alpha$
$\mathbf{r} = \mathbf{r} - \text{sign}(r_p)\alpha\mathbf{R}^{(p)}(n)$
$\mathbf{w}(n+1) = \mathbf{w}(n) - \text{sign}(r_p)\alpha\mathbf{V}^{(p)^T}(n)$

Table 1: Code describing the dichotomous coordinate descent (DCD) algorithm with 'leading' element and incorporated filter update.

3. COMPUTATIONAL COMPLEXITY

The number of multiplications per algorithm iteration for the MFX-DCDRAP algorithm is:

$$M_{MFX-DCDRAP} = IJK(M + L + 2KN + 2K) + IJL + JKM + KN(KN+1) \tag{11}$$

The number of multiplications per algorithm iteration for the MFX-LMS algorithm is [11]:

$$M_{MFX-LMS} = IJK(M + 2L) + IJL + JKM + K \tag{12}$$

The number of multiplications per algorithm iteration for the MFX-DCDPAP algorithm is [8]:

$$M_{MFX-DCDPAP} = IJK(M + 2L + 3KN) + IJL + JKM \tag{13}$$

The maximum number of additions per algorithm iteration for the MFX-DCDRAP algorithm is:

$$A_{MFX-DCDRAP} = K^2\left(2IJ(N+1) + N^2 - N - 1\right) + K(IJ(M + L - 1) + J(M-1) + N + 1) + IJ(L-1) + 2N_u KN + M_b \tag{14}$$

The number of additions per algorithm iteration for the MFX-LMS algorithm is:

$$A_{MFX-LMS} = IJK(M + 2L) + IJ(L - K - 1) + JK(M-1) \tag{15}$$

The maximum number of additions per algorithm iteration for the MFX-DCDPAP algorithm is:

$$A_{MFX-DCDPAP} = IJK(M + 2L + 3KN - 2) + IJL + JK(M-1) - IJ - K^2N + KN(N_u + 2M_b - 1) + M_b + 1 \tag{16}$$

Table 2 shows the number of multiplications and additions for the MFX-LMS algorithm and the DCD based algorithms when $I = 1, J = 2, K = 2, M = 64, L = 150$, and two values of N ($N = 5$ and $N = 13$).

It can be seen that the MFX-DCDRAP algorithm is less complex than the MFX-LMS algorithm in terms of additions and multiplications. For $N = 13$, its number of multiplies and additions per iteration are smaller than those of the MFX-DCDPAP for a smaller projection order ($N = 5$), and this justify its use in the next section for performance comparison. Usually we have $L >> \{I, J, K, N\}$ in practical implementations and in most cases, the MFX-DCDRAP algorithm is less complex than the MFX-DCDPAP algorithm.

For the investigated parameters, the number of additions of the DCD part in the MFX-DCDRAP algorithm represents only a small fraction of the total number of additions (about 4% for $N = 5$). However, this fraction is several times higher for the MFX-DCDPAP algorithm (about 12% for $N = 5$). This fraction increases with increasing N (e.g. for $N = 13$, the ratio is only about 7% for the MFX-DCDRAP algorithm and more than 27% for the MFX-DCDPAP algorithm).

Algorithm for multichannel ANC	Multiplies per iteration	Additions per iteration
MFX-LMS [10]	3018	3003
MFX-DCDPAP ($N = 5$)	3198	3524
MFX-DCDRAP ($N = 13$)	3156	3311
MFX-DCDRAP ($N = 5$)	2372	2431

Table 2: Comparison of the number of multiplies and additions per iteration of the MFX-LMS, MFX-DCDRAP and MFX-DCDPAP algorithms for ANC ($L = 150, M = 64, N_u = 4$, $I = 1, J = 3, K = 2$).

4. SIMULATIONS

The new MFX-DCDRAP algorithm, and MFX-DCDPAP algorithm were simulated, and compared to the multichannel modified filtered-x LMS algorithm (MFX-LMS, [5]). The simulation was performed with acoustic transfer functions experimentally measured in a duct. The impulse responses used for the multichannel acoustic plant had M=64 taps each, while the adaptive filters had $L = 150$ taps each. In the case of ideal plants, the step size of all algorithms has been chosen in order to have the same final attenuation. For all the affine projection algorithms, the step size was $\mu = 0.5$, while for the MFX-LMS algorithm it was $\mu = 2 \cdot 10^{-5}$. The regularization factor is $\delta = 2 \cdot 10^{3}$. The parameter H of the DCD algorithm was set to 1/128. The performance of the algorithms was measured by

$$Attenuation\left(dB\right) = 10 \cdot \log_{10} \frac{\sum_{k} E\left[e_k^2\left(n\right)\right]}{\sum_{k} E\left[d_k^2\left(n\right)\right]} \tag{17}$$

and have been averaged over 40 simulations.

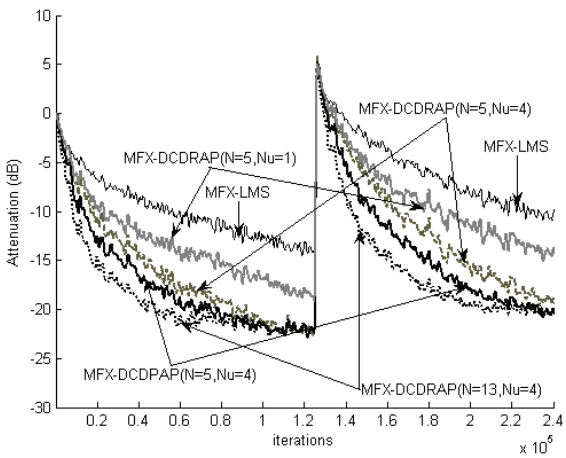

Fig. 2. Convergence curves for multichannel delay-compensated modified filtered-x algorithms for ANC with ideal plant models ($I = 1, J = 3, K = 2, L = 150, M = 64, M_b = 16$).

Figure 2 compares the performance of the selected algorithms, with ideal plant models, for a multichannel system (I=1, J=3, K=2), obtained from Matlab™ simulations. The tracking behavior performance is also investigated by suddenly changing the sign of plant model coefficients after 125000 iterations. As expected, both DCD based algorithms have higher convergence speed and better tracking performances than the MFX-LMS algorithm, which needs many more iterations to reach the same final attenuation. For the projection order $N = 5$ and $M_b = 16$, even one DCD iteration in the MFX-DCDRAP algorithm leads to a superior convergence performance over the MFX-LMS algorithm. As expected the convergence speed increases if the number of iterations is increased (e.g. from 1 to 4 in Figure 2). The MFX-DCDPAP algorithm achieves superior performance over the MFX-DCDRAP algorithm if the same M_b, N_u and H parameters are used. However, for similar number of multiplications, the MFX-DCDRAP algorithm can use a higher projection order (e.g. up to 13 instead of 5 for the investigated *I, J, K, L, M* values – see numerical complexities in Table 2). It can be seen that the MFX-DCDRAP algorithm with $N = 13$ has a faster convergence speed than the MFX-DCDPAP algorithm using $N = 5$.

5. CONCLUSIONS

The multichannel MFX-DCDRAP algorithm has been introduced for practical active noise control systems using FIR adaptive filtering. It has been shown to provide a significant improvement of the convergence speed over the MFX-LMS algorithm, with a smaller computational complexity for typical projection orders. Its performances were also compared favorably with the previously published MFX-DCDPAP algorithm. It was shown that it is a good candidate for practical real-time implementation.

Acknowledgment: This work was supported by the UEFISCSU grant PN-II no. 331/01.10.2007

6. 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.L. Gay and S. Tavathia, “The fast affine projection algorithm”, *Proc. of ICASSP 1995*, Detroit (MI), May 1995, vol. 5, pp. 3023-3026

[4] 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

[5] 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

[6] Y. Zakharov and T. C. Tozer, “Multiplication-free iterative algorithm for LS problem,” *Electronics Letters*, vol. 40, no. 9, pp. 567-569, April 2004.

[7] F. Albu, “Efficient multichannel filtered-x affine projection algorithm for active noise control”, *Electronics Letters*, vol. 42, no. 7, pp. 421 – 423, 2006

[8] 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, pp. 1044-1052, March 2007

[9] Y. Zakharov, “Low complexity implementation of the affine projection algorithm”, *IEEE Signal Processing Letters*, vol. 15, 2008, pp.557-560

[10] Y. Zakharov, G. White, and J. Liu, “Low complexity RLS algorithms using dichotomous coordinate descent iterations,” *IEEE Trans. Signal Processing*, vol.56. No.7, pp.3150—3161. 2008

[11] 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.