AN EFFICIENT ALGORITHM FOR ACTIVE NOISE CONTROL

FELIX ALBU, CONSTANTIN PALEOLOGU [1], YURIY ZAKHAROV[2]

Key words: Affine projection algorithms, Dichotomous coordinate descent, Multichannel active noise control.

A multichannel filtered-x affine projection algorithm for active noise control (ANC) systems based on dichotomous coordinate descent (DCD) iterations is proposed. It is shown that it has better convergence properties, lower complexity, and improved robustness to inaccuracies of the plant model than other previously published algorithms for ANC systems.

1. **INTRODUCTION**

Active noise control (ANC) systems have been increasingly researched and developed [1]. The use of the modified filtered-x structure for ANC using finite impulse response (FIR) adaptive filtering [2] will be assumed in the rest of this paper (Fig. 1).

The multichannel versions of the filtered-x least-mean-square (FX-LMS) and the modified FX-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 it is well known that fast affine projection (FAP) algorithms for ANC provide a good tradeoff between convergence speed and computational complexity [3–9, 11, 14]. It was also reported that in realistic cases where noisy plant models are used, FAP algorithms can be much more robust to plant model noise than more complex algorithms based on recursive least-squares, and they can achieve a better convergence performance at a lower cost [4, 6]. In [6], an efficient implementation based on Gauss-Seidel method has been proposed and adapted for nonlinear ANC using Volterra filtering in [15]. Also, a similar approach with that of [6] has been adapted for filtered-X structures in [7–9]. The numerical complexity of previously published affine projection (AP) algorithms for

[1] Faculty of Electronics, Telecommunications and Information Theory / Telecommunications Department, University “Politehnica” of Bucharest, 1-3 Iuliu Maniu, ZIP 061071, Bucharest, Romania, E-mail: {felix, pale}@comm.pub.ro
[2] Electronics Department, University of York, Heslington, York YO10 5DD, U.K. Tel: +44 (0) 1904 432396, E-mail: yz1@ohm.york.ac.uk

Rev. Roum. Sci. Techn. – Électrotechn. et Énerg., **55**, *4*, p. 416–425, Bucarest, 20101

ANC systems was further reduced by using the Dichotomous Coordinate Descent (DCD) method proposed in [10].

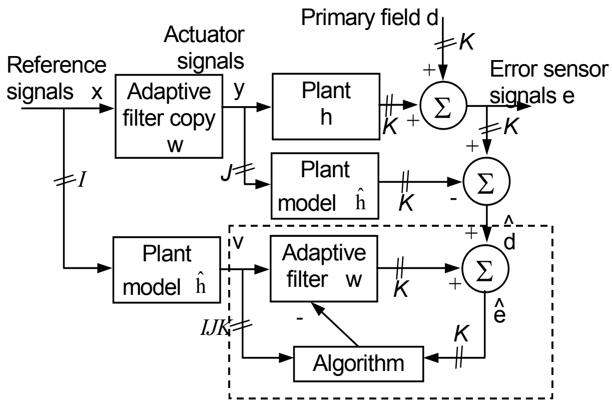

Fig. 1 – A delay compensated or modified filtered-x structure for active noise control [2, 4].

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 [11]. In [12] a novel recursive filtering technique and filtering update that is incorporated in DCD iterations is proposed for the AP algorithm. This leads to an important reduction in the number of multiplications needed by the AP algorithm.

In Section 2, a multichannel ANC system called the Modified Filtered-x Dichotomous Coordinate Descent Recursive Affine Projection (MFX-DCDRAP) algorithm is presented. It uses a variant of the DCD algorithm called the DCD algorithm with a leading element [13]. The computational complexity of the proposed algorithm is evaluated and compared with other algorithms in Section 3. Simulation results comparing the investigated algorithm with previously published MFX-LMS and MFX-DCDPAP 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 [14] 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 the FIR filters modeling the plant, and N is the projection order. The vectors $\mathbf{x}_i = \left[x_i(n),\ldots,x_i(n-L+1)\right]^{\mathrm{T}}$ and

$\mathbf{x}_i^{'} = \left[x_i(n),\ldots,x_i(n-M+1)\right]^{\mathrm{T}}$ consist of the last L and M samples of the reference signal $x_i(n)$, respectively. Superscript "T" denotes transposition. The vector $\mathbf{y}_j = \left[y_j(n),\ldots,y_j(n-M+1)\right]^{\mathrm{T}}$ 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 $\mathbf{V}_0(n) = \begin{bmatrix} v_{1,1,1}(n) \ldots v_{1,1,K}(n) \\ \ldots. \quad \ldots\ldots \\ v_{I,J,1}(n) \ldots v_{I,J,K}(n) \end{bmatrix}$, $IJL \times K$ matrix $\mathbf{V}_1(n) = [\mathbf{V}_0^{\mathrm{T}}(n),\ldots,\mathbf{V}_0^{\mathrm{T}}(n-L+1)]$, and $IJL \times KN$ matrix $\mathbf{V}(n) = \left[\mathbf{V}_1(n) \cdots \mathbf{V}_1(n-N+1)\right]$. The vectors $\hat{\mathbf{d}}(n) = \left[\hat{d}_1(n), \hat{d}_2(n),\ldots,\hat{d}_K(n)\right]$ and $\hat{\mathbf{e}}(n) = \left[\hat{e}_1(n), \hat{e}_2(n),\ldots,\hat{e}_K(n)\right]$ 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 (Fig. 1). Vectors $\hat{\mathbf{D}}(n) = [\hat{\mathbf{d}}(n), \hat{\mathbf{d}}(n-1),\ldots,\hat{\mathbf{d}}(n-N+1)]$ and $\hat{\mathbf{E}}(n) = [\hat{\mathbf{e}}(n), \hat{\mathbf{e}}(n-1),\ldots,\hat{\mathbf{e}}(n-N+1)]$ have both $1 \times KN$ size [14]. The vectors $\mathbf{h}_{j,k} = \left[h_{j,k,1},\ldots,h_{j,k,M}\right]^{\mathrm{T}}$ consist of taps $h_{j,k,m}$ of the fixed FIR filter modelling the plant between the 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 regularization 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 [4]:

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

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

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

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

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

Using the original DCD-AP algorithm from [12] and extending the fast recursive techniques and filtering update to multichannel ANC systems as in [4], the multichannel MFX-DCDRAP algorithm for ANC is obtained.

$$\mathbf{Z}(n) = [\mathbf{V}_0^{\mathrm{T}}(n)\mathbf{w}(n-1)\ \bar{\mathbf{Y}}(n-1)], \tag{6}$$

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

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

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

where $\mathbf{G}(n)$ is a $KN \times KN$ matrix. The filter update (5) is performed by solving the following linear system of equations [14]:

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

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 $\mathbf{V}_0^{\mathrm{T}}(n)\mathbf{V}_0(n)$. The other elements of $\mathbf{R}(n)$ can be taken from $\mathbf{R}(n-1)$ and $\mathbf{G}(n)$. Specifically, elements $[\mathbf{R}(n)]_{i,j}, i,j = K+1,\ldots,KN$ are taken from $[\mathbf{R}(n-1)]_{i,j}, i,j = 1,\ldots,K(N-1)$. The elements $[\mathbf{R}(n)]_{i,j}, i = 1,\ldots,K, j = K+1,\ldots,KN$ and $[\mathbf{R}(n)]_{i,j}, i = K+1,\ldots,KN, j = 1,\ldots,K$ are taken from $[\mathbf{G}(n)]_{i,j}, i = 1,\ldots,K, j = K+1,\ldots,KN$. The MFX-DCDRAP algorithm is described by the equations (1)–(3), (6)–(10). The MFX-DCDPAP algorithm uses the original DCD algorithm [10], while the MFX-DCDRAP uses a DCD version with a leading element [13]. 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 [10] bounded by an interval $[-H, H]$, [12].

Table 1

DCD algorithm with 'leading' element and incorporated filter update

Initialization: $\mathbf{P}(n) = \mathbf{0}, \mathbf{r} = \mu\hat{\mathbf{E}}(n), \alpha = H/2, m = 1$
For $k = 1,\ldots,N_u$
$p = \arg\max_{i=0,\ldots,KN-1}\{\lvert r_i \rvert\}$
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. (1)
$\mathbf{P}_p = \mathbf{P}_p + \text{sign}(r_p)\alpha$
$\mathbf{r} = \mathbf{r} - \text{sign}(r_p)\alpha\mathbf{R}(n)^{(p)}$
$\mathbf{w}(n+1) = \mathbf{w}(n) - \text{sign}(r_p)\alpha\mathbf{V}^T(n)^{(p)}$

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. We are interested in using a smaller number of updates and a more efficient DCD version from this point of view was proposed in [13]. This new version finds a 'leading' (*p*th) element of the solution to be updated (Table 1). With N_u updates, the number of additions of this version is upper limited by $2N_uN + M_b$, 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 iterations compared to the previous MFX-DCDAP or MFX-DCDPAP algorithm.

3. COMPUTATIONAL COMPLEXITY

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

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

The number of multiplications per algorithm iteration for the MFX-LMS algorithm is [4]:

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

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

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

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

$$\begin{aligned} A_{MFX-DCDRAP} &= K^2\left(2IJ(N+1) + N^2 - N - 1\right) + \\ &+ K\left(IJ(M + L - 1) + J(M - 1) + N + 1\right) + IJ(L - 1) + 2N_u KN + M_b. \end{aligned} \quad (14)$$

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

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

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

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

Figure 2a shows the number of multiplications for the MFX-LMS algorithm and the DCD based algorithms when $I = 1, J = 3, K = 2, M = 64, L = 150$, and N is varying; it can be seen that the MFX-DCDRAP algorithm is less complex than the MFX-LMS algorithm for $N \leq 12$ and less complex than the MFX-DCDPAP algorithm for $N \leq 16$.

Usually we have $L >> \{I, J, K, N\}$ in practical implementations and therefore, in terms of multiplications, the MFX-DCDRAP algorithm is less complex than the MFX-DCDPAP algorithm.

Figure 2b shows the number of additions per algorithm iteration for the MFX-LMS, MFX-DCDRAP, and MFX-DCDPAP algorithms in the same situation

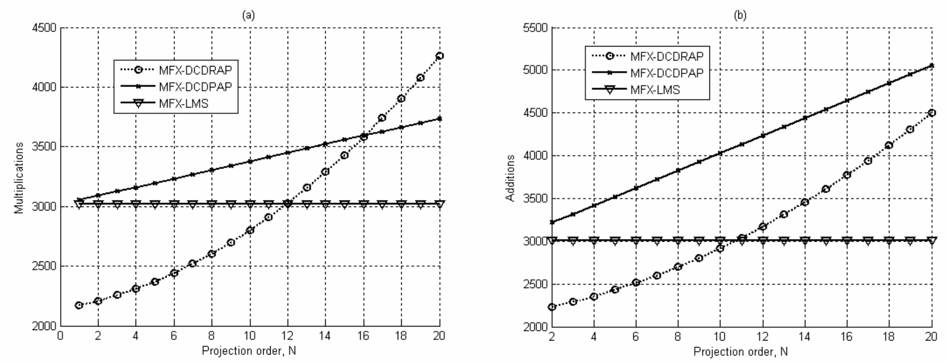

Fig. 2 – Numerical complexity of the MFX-LMS, MFX-DCDRAP and MFX-DCDPAP algorithms ($I=1, J=3, K=2, L=150, M=64, N_u=4, M_b=16$, and N is varying); a) number of multiplications per algorithm iteration; b) number of additions per algorithm iteration.

Table 2

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

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

as above. The MFX-DCDRAP algorithm requires fewer additions per iteration than the MFX-LMS algorithm for $N \le 10$.

Note that 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).

4. **SIMULATION RESULTS**

The MFX-DCDRAP algorithm and the previously published MFX-DCDPAP algorithm were simulated and compared to the MFX-LMS algorithm [4]. 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. The reference signal was a white noise with zero mean and variance one. For all the affine projection algorithms, the step size is $\mu = 1$ in the case of ideal plants. In case of noisy plant models with a signal to noise ratio (SNR) of 10 dB $\mu = 0.5$ was used for improved stability and robustness [14]. The regularization factor is $\delta = 2 \cdot 10^3$ for the ideal plant and $\delta = 10^4$ for plant models with the SNR of 10 dB. As known from [4], higher regularization values could lead to a reduced initial convergence speed. The step size μ for the MFX-LMS algorithm was $2 \cdot 10^{-5}$ [4], [11]. The parameter H of the DCD algorithm is related to the bounds of the solution of the linear system [12] and it was set to 1/128. The choice of the parameters of the algorithms was made by trials in order to obtain the best performance. The performance of the algorithms was measured by

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

and have been averaged over 50 simulations. It was found in previous works that a projection order of size $N = 5$ is sufficient for AP algorithms to achieve a significantly improved convergence performance compared to the LMS algorithm and therefore this value is used as a base for comparing the considered DCD based algorithms [11].

Figure 3a 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. As expected, both DCD based algorithms have higher convergence performances than the MFX-LMS algorithm. 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 Fig. 3a).

If ideal plants are used, the MFX-DCDPAP algorithm achieves superior performance over the MFX-DCDRAP algorithm in terms of convergence speed, 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 $N = 5$.

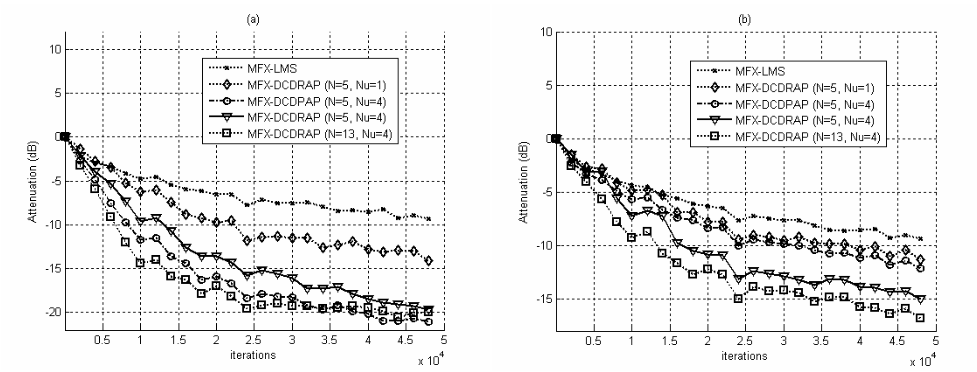

Fig. 3 – Convergence curves for multichannel delay-compensated modified filtered-x algorithms for ANC with ideal plant models ($I = 1, J = 3, K = 2, L = 150, M = 64, M_b = 16$);
a) ideal plant models; b) noisy plant models (10 dB SNR).

Figure 3b shows the performance when plant models with a 10 dB SNR are used. The noisy plant models with 10 dB SNR accuracy were obtained as in [4]. In this case, the behavior of the MFX-DCDRAP algorithm is better than that of the MFX-DCDPAP algorithm with similar DCD parameters and much better than that of the MFX-LMS algorithm. Therefore, the MFX-DCDRAP algorithm is potentially more robust to inaccuracies of the plant model.

5. **CONCLUSIONS**

The multichannel MFX-DCDRAP algorithm has been introduced for practical ANC 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 performance was also compared favourably with the previously published MFX-DCDPAP algorithm. It was shown that it is a good candidate for practical real-time implementation due to its fast convergence, low complexity, and robustness to plant model inaccuracies.

ACKNOWLEDGEMENTS

This work was supported by the UEFISCSU Romania under Grant PN-II-PCE-“Idei” no. 331/01.10.2007.

Received on 11 August 2009

REFERENCES

1. S. Kuo and D. R. Morgan, *Active noise control: a tutorial review,* Proc. of the IEEE, **87**, *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, Aug. 1992, pp. 1053–1056.
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, Oct. 1995, pp. 1245-1249.
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, **11**, *1*, pp. 54-60, Jan. 2003.
5. A. Carini and G. L. Sicuranza, *Filtered-X affine projection algorithms for a class of nonlinear multichannel active noise controllers*, Proceedings of IWAENC 2005, International Workshop on Acoustic Echo and Noise Control, Eindhoven, The Netherlands, Sep. 12-15, 2005.
6. M. Bouchard, F. Albu, *The Multichannel Gauss-Seidel Fast Affine Projection Algorithm for Active Noise Control*, Seventh International Symposium on Signal Processing and its Applications, IEEE – ISSPA 2003, Paris, France, July 2003.
7. M. Ferrer, A. Gonzales, M. Diego, and G. Pinero, *Fast Affine projection algorithms for filtered-x active noise control*, IEEE Transaction on Audio, Speech and Language Processing, **16**, *8*, pp. 1396-1408, Nov. 2008.
8. M. Ferrer, A. Gonzales, M. Diego, and G. Pinero, *Fast Efficient Update for filtered-X Affine Projection algorithms*, Journal of Multimedia, **2**, *2*, pp. 83-89, April 2007.
9. J. M. Wesselink, A. P. Berkhoff, *Fast affine projections and the regularized modified filtered-error algorithm in multichannel active noise control*, The Journal of the Acoustical Society of America, **124**, *2*, pp. 949-960, August 2008.
10. Y. Zakharov and T. Tozer, *Multiplication-free iterative algorithm for LS problem,* Electronics Letters, **40**, *9*, pp. 567-569, April 2004.
11. F. Albu, M. Bouchard, Y. Zakharov, *Pseudo Affine Projection Algorithms for Multichannel Active Noise Control,* IEEE Transaction on Audio, Speech and Language Processing, **15**, *3*, pp. 1044-1052, March 2007.
12. Y. Zakharov, *Low complexity implementation of the affine projection algorithm,* IEEE Signal Processing Letters, **15**, pp. 557-560, 2008.
13. Y. Zakharov, G. White, and J. Liu, *Low complexity RLS algorithms using dichotomous coordinate descent iterations,* IEEE Trans. Signal Processing, **56**, *7*, pp. 3150-3161, July 2008.
14. F. Albu, Y. Zakharov, C. Paleologu, *Modified Filtered-x dichotomous coordinate descent recursive affine projection algorithm*, IEEE – ICASSP 2009, pp. 257-260, Taipei, April 2009.
15. J. B. Seo, K. J. Kim, D. W. Kim, M. O. Oh, W. H. Cho, H. K. Lee, W. Y. Kim, S. W. Nam, *Nonlinear active noise control using Volterra filtering with a variable step-size GS-PAP algorithm*, IEEE – IWAENC 2008, Seattle, U.S.A., September 2008.