AFFINE PROJECTION ALGORITHMS: EVOLUTION TO SMART AND FAST ALGORITHMS AND APPLICATIONS

Alberto Gonzalez [a], Miguel Ferrer [a], Felix Albu [b], and Maria de Diego [a]

[a] Institute of Telecommunications and Multimedia Applications (iTEAM)
Universitat Politècnica de València, Spain
e-mail: {agonzal,mferrer,mdediego}@dcom.upv.es
[b] Department of Electronics & Telecommunications
Valahia University of Targoviste, Romania
e-mail: felix.albu@valahia.ro

ABSTRACT

Affine projection algorithm encompasses a family of configurable algorithms designed to improve the performance of other adaptive algorithms, mainly LMS based ones, especially when input data is highly correlated. The computational cost of the affine projection algorithm depends largely on the projection order, which in turn conditions the speed of convergence, thus high speed of convergence implies usually high computational cost. Some real-time applications (especially multichannel) using the affine projection algorithm can not be implemented in the existing general-purpose hardware, because of this several improvements of the affine projection algorithm have been proposed to make it more computationally efficient and more versatile in terms of performance. This paper outlines the evolution of the affine projection algorithm and its variants, in order to get an efficient and self-reconfigurable algorithm. Furthermore new improvements over the existing low cost and variable step size and projection order versions are proposed to give examples of the new generation of affine projection algorithms.

1. INTRODUCTION

Originally affine projection (AP) algorithms emerged to improve speed of convergence of gradient based algorithms when the input signals did not exhibit flat spectrum, since the speed of convergence of these algorithms decreased substantially in this cases [1].

A common feature that encompasses all AP algorithms is the filter update equation, which uses N (called projection order) vectors of the input signal instead of a single vector as the NLMS algorithm. Therefore, these algorithms could be understood as an extension of the algorithm NLMS, or more generally, as a minimization problem with constraints that can be expressed mathematically as follows.

Let (1) be the change of the L adaptive filter coefficients between successive algorithm iterations,

$$\Delta \mathbf{w}_L[n] = \mathbf{w}_L[n] - \mathbf{w}_L[n-1]. \quad (1)$$

To develop the algorithm the expression (2) has to be minimized under N constraints given by (3):

$$\|\Delta \mathbf{w}_L[n]\|^2 = \Delta \mathbf{w}_L^T[n]\Delta \mathbf{w}_L[n], \quad (2)$$

This work was supported by a grant of the Romanian National Authority for Scientific Research, CNCS-UEFISCDI, project number PN-II-ID-PCE-2011-3-0097, Spanish Ministerio de Ciencia e Innovacion TEC2009-13741 and Generalitat Valenciana PROMETEO 2009/2013.

$$\mathbf{w}_L^T[n]\mathbf{x}_L[n-k] = d[n-k] \ \text{ para k} = 0,1,...,\mathrm{N}-1. \quad (3)$$

$\mathbf{x}_L[n]$ is a vector that comprises the last L samples of the input vector and $d[n]$ represent the desired signal as it is illustrated in Fig. 1.

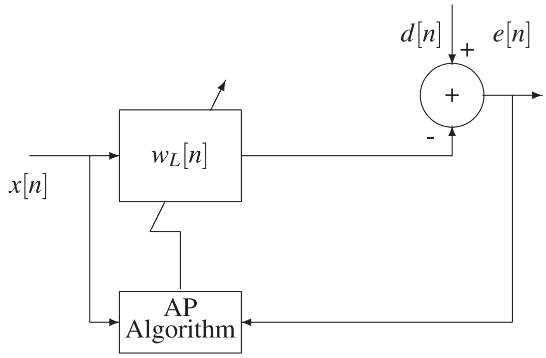

Figure 1: Adaptive scheme and signals involved using the affine projection algorithm.

The solution of the presented problem leads to the AP update equation given by

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \mathbf{A}^T[n](\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N[n], \quad (4)$$

where

$$\mathbf{A}[n] = (\mathbf{x}_L[n], \mathbf{x}_L[n-1], ..., \mathbf{x}_L[n-N+1])^T, \quad (5)$$

$\mathbf{e}_N(n)$ is a vector of size $N \times 1$ given by

$$\mathbf{e}_N[n] = \mathbf{d}_N[n] - \mathbf{A}[n]\mathbf{w}_L[n-1], \quad (6)$$

and $\mathbf{d}_N[n]$ represents the desired signal vector of size $N \times 1$

$$\mathbf{d}_N^T[n] = (d[n], d[n-1], ..., d[n-N+1]). \quad (7)$$

Equation (4) can be expressed by a general form, see expression. (8), which includes the entire affine projection family.

$$\begin{aligned} \mathbf{w}_L[n] &= \mathbf{w}_L[n-1-\alpha(N-1)] \\ &+ \mu \mathbf{A}_\tau{}^T[n](\mathbf{A}_\tau[n]\mathbf{A}_\tau{}^T[n] + \delta \mathbf{I})^{-1}\mathbf{e}_{N\tau}[n] \end{aligned} \quad (8)$$

with $\mathbf{e}_{N\tau}[n] = \mathbf{d}_{N\tau}[n] - \mathbf{A}_\tau \mathbf{w}_L[n-1-\alpha(N-1)]$,

$$\mathbf{A}_\tau[n] = (\mathbf{x}_L[n], \mathbf{x}_L[n-\tau], ..., \mathbf{x}_L[n-(N-1)\tau])^T \quad (9)$$

and

$$\mathbf{d}_{N\tau}^T[n] = (d[n], d[n-\tau], ..., d[n-(N-1)\tau]). \qquad (10)$$

As it can be observed in (8), the N data vectors that are used to update the coefficients could not necessarily be the most recent ones. Thus different versions of the algorithms can be developed choosing the input data vectors and using them in different ways within (8). As examples we can cite: the NLMS with orthogonal correction factors (NLMS-OCF) [2], the partial rank affine projection algorithms (PRA) [3], and the most frequently used standard affine projection algorithms (APA) [4], which is given when $\delta = 0$, $\alpha = 0$ and $\tau = 1$. The last one with $\delta \neq 0$ leads to the affine projection algorithm with regularization (R-APA) [5], whose coefficient update equation can be considered as a special case of LevenbergMarquardt regularized APA (LMR-APA)[6].

The different configurations of this family of algorithms can fit the needs of convergence of different applications and the nature of the signals involved. However, despite that good performance can be obtained, two factors can be improved: their computational cost, which can be unaffordable for high projection orders, and the little configuration flexibility of the algorithm. It seems clear that the algorithm could improve its performance if it could efficiently adjust their parameters (μ, N, δ, α or τ) during execution.

2. LOW COST AFFINE PROJECTION ALGORITHM

Two main procedures to reduce the computational cost of the AP family algorithms have been proposed: either making use of algebraic relations that allow calculate some parameters using computational resources or using accurate approximations to avoid costly calculations. The main strategies that have reduced the computational cost by following these guidelines are explained within the following sections.

2.1 Fast Affine Projection Algorithm

The fast affine projection algorithm [7, 8] (FAP) reduces the cost of the AP algorithm by N, thus it is suitable for higher projection orders. It makes use of algebraic and matrix relations and assumes $\delta \approx 0$ to reduce the computational cost of

$$\varepsilon_N[n] = (\mathbf{A}[n]\mathbf{A}^T[n])^{-1}\mathbf{e}_N[n], \qquad (11)$$

that can be simplified even more when $\mu = 1$. The FAP algorithm needs to calculate the forward and backward linear prediction filters and the minimum value of the sum of prediction-error squares, whose values are recursively computed. It must be noted that the main cost of the algorithm is due to the calculation of the inverse matrix that appears in (4) and (11). It is easy to realize that the main matrix and its inverse can be recursively calculated thus reducing the computational cost from $N^2L + O(N^3)$ to $6(N^2+2N)$ multiplications [9, 10]. Furthermore the FAP algorithm optimizes even more the cost making use of auxiliary coefficients in the coefficient update equation, thus it does not use (4). The additional cost reduction reaches $(L-3)N+2$ [11]. The relation between the auxiliary coefficients and the original ones are given by the following expressions

$$\widehat{\mathbf{w}}_L[n] = \widehat{\mathbf{w}}_L[n-1] + \mu\mathbf{x}_L[n-N+1]E_{N-1}[n-1] \qquad (12)$$

$$y[n] = \widehat{\mathbf{w}}_L^T[n]\mathbf{x}_L[n] + \mu\mathbf{x}_L^T[n]\overline{\mathbf{A}}^T[n]\overline{\mathbf{E}}[n] \qquad (13)$$

where E_{N-1}[n] and $\overline{\mathbf{E}}[n]$ are the last element and the $N-1$ last elements, respectively, of the auxiliary error vector, which can be recursively calculated by

$$\mathbf{E}[n] = \varepsilon_N[n] + \begin{bmatrix} 0 \\ \overline{\mathbf{E}}[n-1] \end{bmatrix}. \qquad (14)$$

The performance of this algorithm is similar to the original AP algorithm. They differ in the transient period when δ is not small enough or non-accurate initial values of the linear prediction filters or other recursively calculated parameters are chosen. Despite this algorithmic variant leads to meaningful computational savings, its computational cost and its steady state behavior depend on its initial configuration, mainly its projection order. Therefore both aspects can be improved by decreasing its projection order during algorithm operation.

2.2 Gauss Seidel pseudo Affine Projection Algorithm

The pseudo affine projection (PAP) algorithms are derived using realistic approximations of the original sample-by-sample AP algorithm. The first version, proposed in [12] computes the weight update equations assuming $\mu = 1$ and the stationarity of the input signal. It derives the relation between the optimal forward linear prediction coefficient error, $[1, a_1, \dots a_{N-1}]^T$ and the energy of the prediction error. It is shown in [12] that

$$\mathbf{A}[n]\mathbf{A}^T[n] \begin{pmatrix} 1 \\ a_1 \\ \vdots \\ a_{N-1} \end{pmatrix} = \begin{pmatrix} \mathbf{x}_L^T[n]\mathbf{u}[n] \\ 0 \\ \vdots \\ 0 \end{pmatrix} \qquad (15)$$

where $\mathbf{u}[n] = \mathbf{x}_L[n] + \sum_{i=1}^{N-1} a_i\mathbf{x}_L[n-1]$. The coefficient update equation is the following:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \frac{\mu}{\mathbf{u}^T[n]\mathbf{x}_L[n] + \delta}\mathbf{u}[n]e[n], \qquad (16)$$

where $e[n] = d[n] - \mathbf{x}_L^T[n]\mathbf{w}_L[n-1]$. The Gauss-Seidel PAP algorithm (GSPAP) [13] proposes to use the one Gauss Seidel iteration to solve (1) and use the previous forward linear prediction coefficient error solution as an initialization for the current iteration. The computational load is optimized by the following approximation

$$\mathbf{u}[n] \approx (\mathbf{s}^T[n]\mathbf{x}_N[n]/s_0[n], \overline{\mathbf{u}}[n-1]), \qquad (17)$$

where $\mathbf{s}[n] = (s_0[n], s_1[n], \dots, s_{N-1}[n])^T$ is the solution of the system $(\mathbf{R}[n] + \delta\mathbf{I}_N)\mathbf{s}[n] = \mathbf{b}$ with $\mathbf{b} = (1, 0 \dots 0)^T$, $\mathbf{R}[n] = \mathbf{A}[n]\mathbf{A}^T[n]$, $\mathbf{x}_N[n] = (x[n], x[n-1], \dots, x[n-N+1])^T$, $\overline{\mathbf{u}}[n-1]$ denotes the vector consisting of the uppermost $L-1$ elements of $\mathbf{u}[n-1]$ and $\mathbf{I}_N$ is the $N \times N$ unity matrix. Another variant of PAP algorithm has been derived in [14] inspired from the adaptive algorithms with decorrelating properties investigated in [15]. The coefficient update is slightly changed as follows:

$$\mathbf{w}_L[n] = \mathbf{w}_L[n-1] + \frac{\mu}{\mathbf{u}^T[n]\mathbf{u}[n] + \delta}\mathbf{u}[n]e[n]. \qquad (18)$$

If a similar approximation of $\mathbf{u}[n]$ with a tapped-delay line as in (17) and one Gauss-Seidel iteration is performed, the GS-PAP algorithm introduced in [16] is obtained. Further adaptations of the PAP algorithms were investigated for active noise control in [17] and hearing aids in [18]. Unlike the FAP algorithm the PAP algorithms updates the weight coefficients, not the auxiliary ones. Their numerical complexity can be further reduced by the block exact based versions (e.g. [19]-[21]). Also, variable regularization [22] and variable step size version were investigated in [23] and [24].

2.3 Dichotomous Coordinate Descent Affine Projection Algorithm

The DCD algorithm was proposed in [25] in order to iteratively solve linear systems using only additions and bit-shift operations. 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 that takes one of (number of bits) predefined values corresponding to a binary representation bounded by an interval $[-H,H]$ [25]. 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. A more efficient DCD version was proposed in [26]. This new version finds a 'leading' (pth) element of the solution to be updated. More details about the DCD algorithms version can be found in [25] and [26]. The DCD algorithms are typically used to solve implicit linear systems of equations, and therefore to avoid complicated matrix inversions. Important complexity reductions were reported for various DCD based AP, FAP or PAP algorithms (see for example [27]-[30]).

3. VARIABLE STEP SIZE AFFINE PROJECTION ALGORITHMS

The step size μ commands the convergence speed and strongly influences the steady state performance of the AP algorithm. As μ increases the convergence is faster but the Mean Squared Error (MSE) in steady state worsens [31]. Variable step size algorithms allow to adjust the step size to the algorithm needs within both transient or steady states.

The theoretical MSE for the AP algorithm can be approximated by [31, 32]

$$\mathrm{MSE} \approx \frac{\mu \mathrm{N} \sigma_{\mathrm{v}}^2}{2-\mu} + \sigma_{\mathrm{v}}^2 \tag{19}$$

where σ_v^2 represents the measurement noise variance. From (19) the MSE can be approximated when $\mu \approx 0$ [31] by

$$\mathrm{MSE} \approx \frac{\mu \sigma_{\mathrm{v}}^2}{2-\mu} + \sigma_{\mathrm{v}}^2 \tag{20}$$

that coincides with the MSE of the NLMS algorithm.

In summary, only the AP algorithms with low step size exhibit optimal behavior in steady state. However low values of μ slow down their convergence speed. It exists several algorithms to dynamically adjust the step size [33]-[36], all of them improve the MSE at steady state and do not worsen the convergence speed, but their computational cost is similar to the original AP.

4. VARIABLE ORDER AFFINE PROJECTION ALGORITHMS

The variable order affine projection algorithms adjust its projection order to their convergence needs and therefore decrease their computational cost. Furthermore they reach a good MSE at steady state as it can be expected from (19). These algorithms decrease the projection order when they reach the steady state or lower convergence speed is allowed. A first version of these algorithms was proposed in [37] where the number of input data vectors to update the filter coefficients are selected within each algorithm iteration. Other examples of this kind of algorithms are given in [32] and [38, 39]. All these strategies guarantee a good behavior at both steady and transient states, but mainly they try to optimize the computational cost when the algorithm does not need to work with high projection orders.

5. AFFINE PROJECTION ALGORITHMS WITH VARIABLE BOTH PROJECTION ORDER AND STEP SIZE

Variable step size algorithms exhibit good steady state performance and the algorithms that adjust their projection orders reduce the computational cost when the convergence speed can be decreased. As it can be observed by (19) and (20), the AP algorithm with $N = 1$ and low step size would exhibit the lower cost and better steady state. On the other hand high projection order and step size values improve the convergence properties of the algorithm. Thus algorithms that suitably adapt both parameters would eventually improve algorithm performance in almost any issue. An example of this kind of algorithm was proposed in [40].

In this work we propose an alternative algorithm that dynamically and simultaneously adjusts N and μ. We have used some of the AP variants named within sections 3 and 4. We start from the variable step size AP algorithm (VSSAP) proposed in [33]. The step size follows an update rule that maximizes the change of the coefficients between iterations as

$$\mu[n] = \mu_{max} \frac{\|\mathbf{p}[n]\|^2}{\|\mathbf{p}[n]\|^2 + C}, \tag{21}$$

where $\mathbf{p}[n]$ represents an estimation of the mean value of $\varepsilon_N[n]$, which was defined in (11), and it is recursively obtained from $\mathbf{p}[n] = \alpha \mathbf{p}[n-1] + (1-\alpha)\varepsilon[n]$, with $0 < \alpha < 1$ and $C \approx \frac{N}{\mathrm{SNR}}$, where SNR is the signal to noise ratio $\frac{\sigma_x^2}{\sigma_v^2}$, and σ_x^2 is the variance of $x[n]$.

Following the rules given above the algorithm would reach the steady state for a given projection order N, then it would be convenient to decrease the projection order to get lower cost and better MSE. In order to know when the algorithm should change its projection order we propose to use the condition presented in [39], where it is stated that an AP algorithm of order N has reached its steady state when

$$\gamma = \frac{R}{N} \leq 0.32, \tag{22}$$

being R the number of elements of vector $\mathbf{e}_N[n] = (e_1[n], e_2[n] \cdots e_{\mathrm{N}}[n])^T$ in (6) that fulfils

$$e_i^2[n] \geq \frac{\mu[n] N \sigma_v^2}{2-\mu[n]} + \sigma_v^2. \tag{23}$$

When the projection order decreases it is advisable to adjust the step size as well, in order to get a meaningful convergence speed. The new step size should fulfill [32]

$$\mu[n] = \frac{2\mu[n-1]N[n-1]}{\mu[n-1](N[n-1]-N[n])+2N[n]}. \tag{24}$$

Since the new projection order in time n is $N-1$ (it decrements a unit) when (22) is fulfilled, the proposed algorithm would carry out the step size readjustment following

$$\mu = \frac{2\mu N}{\mu + 2N - 2}. \tag{25}$$

Thus the algorithm would change to the following projection order until $N = 1$, at this stage it would work as the variable step size NLMS.

The performance of the proposed algorithm compared with: AP ($N = 10$), the evolving AP [38] (as an example of variable order projection AP algorithm) and the VSSAP [33] (as an example of variable step size), is illustrated in Fig. 2. It shows the learning curves for an identification FIR system (20 taps), averaging 3000 tries and considering $\sigma_v^2 = 0.001$ and a colored Gaussian noise generated by filtering white Gaussian noise (of zero mean and unit variance) with a first order autoregressive filter of transfer function $\sqrt{1-0.9^2}/(1-0.9z^{-1})$, as reference signal. The MSE of

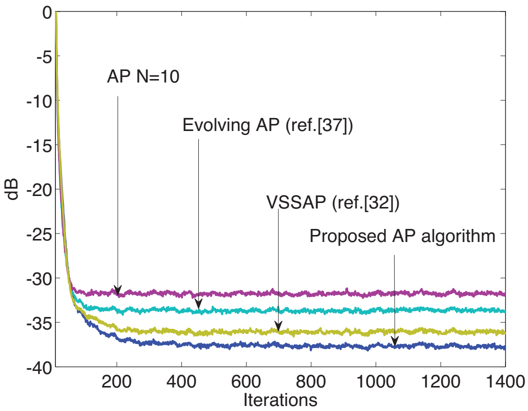

Figure 2: Comparative performance of different AP algorithm. Learning curves for an identification FIR system.

the proposed algorithm is the lowest and the transient is as good as the other algorithms. Moreover, the computational cost decreases with n because the projection order becomes smaller as well. The number of multiplications per iteration for each algorithm is illustrated in Fig. 3. The total number of multiplications accumulated within the whole simulation (3000 iterations) for each algorithm is summarized as:

AP N=10	VSSAP	Evolving AP	Proposed AP
55200000	55510000	2111290	1249394

6. CONCLUSIONS

This work has revised the main existing alternatives that allow the AP algorithm to become more computationally efficient and versatile. These alternatives also improve the convergence properties of the algorithm and its behavior at steady state. The proposed algorithms work suitably even when there is an uncorrelated near-speech sound. To better illustrate the work, a novel version of the AP algorithm that updates both the step size and the projection order has been proposed and compared with other existing versions. This algorithm is based on the VSSAP and it carries out a sequential decrement of the projection order when the algorithm reaches its steady state. Then it also readjusts its step size.

These strategies can also be applied to the low cost AP algorithms and thus optimize even more its required computational resources.

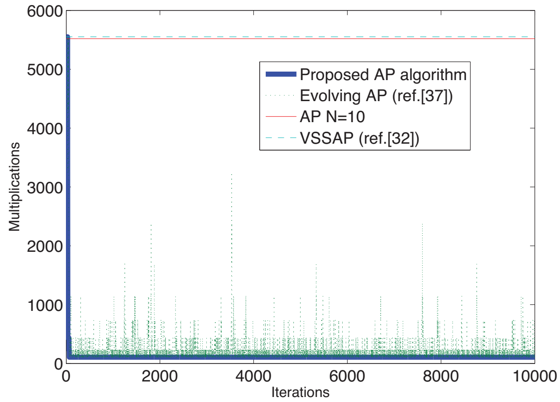

Figure 3: Number of multiplications of the different AP algorithm shown in Fig. 2.

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*. Fourth edition, Prentice-Hall, Ed., Upper Saddle River, NJ, 2002

[2] S. G. Sankaran and A. A. Louis Beex, “Normalized LMS algorithm with orthogonal correction factors,” in *Proc, 31 Asilimar Conference*,Nov. 1997, pp.1670–1673.

[3] S. G. Kratzer and D. R. Morgan, “The partial Rank Algorithm for adaptive beamforming,” in *Proc SPIE*, 1985, vol. 564, pp. 9–14..

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

[5] S. L. Gay and J. Benesty, *Acoustic signal processing for telecommunication*. Second edition, Kubler Academic Publishers, 2001.

[6] T. Van Waterschoot, G. Rombouts, and M. Moonen, “Optimally regularized adaptive filtering algorithms for room acoustic signal enhancement”, Signal Processing, vol. 88, no. 3, Mar. 2008, pp. 594–611.

[7] S. L. Gay, and S. Tavathia, “The fast affine projection algorithm,” in *Proc. ICASSP-95*, Detroit, USA, May 1995, vol.5, pp. 3023–3026.

[8] M. Tanaka, Y. Kaneda, S. Makino and J. Kojima, “Fast projection algorithm and its step size control,” in *Proc. ICASSP-95*, Detroit, USA, May 1995, vol.2, pp. 945–948.

[9] Y. Kaneda, M. Tanaka and J. Kojima, “An Adaptive Algorithm with Fast Convergence for Multi-input Sound

Control," in *Proc, Active95*, Newport Beach, California, USA, Jul 1995, pp. 993–1004.

[10] M. Ferrer, M. de Diego, A. Gonzalez and G. Piñero, "Efficient implementation of the affine projection algorithms for Active Noise Control aplication, " in *Proc. 12th European Signal Process. Conf. (EUSIPCO)*,Sep. 2004.

[11] A. Gonzalez, M. Ferrer, M. de Diego, and G. Piñero, "Fast filtered-x affine projection algorithm with efficient computation of coefficient update,"in *Proc. ICASSP06*, Toulouse, Francia, May 2006 pp. 285–288.

[12] F. Bouteille, P. Scalart and M. Corazza, "Pseudo affine projection algorithm: new solution for adaptive identification," *Eurospeech*, 1999, Volume 1, Page 427–430.

[13] F. Albu and A. Fagan, "The Gauss-Seidel pseudo affine pro-jection algorithm and its application for echo cancellation," *Conference Record of the Thirty-Seventh Asilomar Conference on Signals, Systems and Computers 2003*, 9-12 Nov. 2003, Vol. 2, pp. 1303–1306.

[14] P. Scalart and F. Bouteille, "On Integrating Speech Coding Functions into Echo Canceling Filters with Decorrelating Properties," *Proceedings of ICASSP*, Florida, May 2002, Vol. 2, pp. 1905–1908.

[15] M. Rupp, "A Family of Adaptive Filter Algorithms with Decorrelating Properties," *IEEE Trans. Signal Processing*, Vol.46, No.3, pp. 771–775, 1998.

[16] F. Albu and H.K. Kwan, "Combined echo and noise cancellation based on Gauss-Seidel pseudo affine projection algorithm," *Proc. IEEE ISCAS 2004*, Vancouver, Canada, pp. 505–508.

[17] F. Albu, M. Bouchard and Y. Zakharov, "Pseudo-affine projection algorithms for multichannel active noise control," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15 No. 3, pp. 1044–1052, March 2007.

[18] S. Lee, I. Kim and Y. Park, "Approximated affine projection algorithm for feedback cancellation in hearing aids," *Computer Methods and Programs in Biomedicine*, Volume 87, Issue 3, pp. 254–261, September 2007.

[19] F. Albu, and H.K. Kwan, "Fast block exact Gauss-Seidel pseudo affine projection algorithm," *Electronics Letters*, Vol. 40, Issue 22, pp. 1451–1453, Oct. 2004.

[20] M. Tanaka, S. Makino and Y. Kojima, "A block exact fast affine projection algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 7, pp. 79–86, Jan 1999.

[21] G. Rombouts and M. Moonen, "A sparse block exact affine projection algorithm," *IEEE Transactions on Speech and Audio Processing*, Vol. 10, no. 2, pp. 100–108, Feb. 2002.

[22] S.-J. Ban, C.-W. Lee and S.-W Kim "Adaptive Regularization Parameter for Pseudo Affine Projection Algorithm," *IEEE Signal Processing Letters*, Vol. 16, No. 5, pp. 382–385, May 2009.

[23] J. Lee, Y.-C. Park and D.-H Youn, "Robust pseudo affine projection algorithm with variable step-size," *Electronics Letters* vol. 44, no. 3, pp. 250–252, 2008.

[24] F. Albu, C. Paleologu and J. Benesty, "Gauss-Seidel Based Variable Step-Size Affine Projection Algorithms for Acoustic Echo Cancellation," *Proc. of EUSIPCO 2009*, Glasgow, Scotland, August 24-28, 2009, pp. 2007–2011.

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

[26] 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, July 2008.

[27] Y. Zakharov, "Low complexity implementation of the affine projection algorithm," *IEEE Signal Processing Letters*, vol. 15, pp. 557–560.

[28] F. Albu, C. Paleologu, J. Benesty and Y.V. Zakharov, "Variable Step Size Dichotomous Coordinate Descent Affine Projection Algorithm," *EUROCON 2009*, Saint Petersburg, Russia, May 2009, pp. 1366–1371.

[29] Y. V. Zakharov and F. Albu, "Coordinate descent iterations in fast affine projection algorithm," *IEEE Signal Processing Letters*, vol. 12, pp. 353–356, May 2005.

[30] F. Albu and C. Kotropoulos, "Coordinate descent iterations in pseudo affine projection algorithm," *Proceedings of ISCAS 2005*, Kobe, Japan, May 2005, pp. 2283–2286.

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

[32] N. W. Kong J. W. Shin and P.G. Park, "A two-stage affine projection algorithm with mean square error matching step sizes," *Signal Processing*, vol. 91, no. 11, pp. 2639–2646, Nov. 2011.

[33] H. Shin and A. Sayed, "Variable step-size NLMS and affine projection algorithms," *IEEE Signal Process. Lett.*, vol. 11, no. 2, pp. 132–135, Feb. 2004.

[34] K. Mayyas, "A variable step-size affine projection algorithm," *Digital Signal Processing*, vol. 20, no. 2, pp. 502–510, Mar. 2010.

[35] L. R. Vega, H. Rey, and J. Benesty, "A robust variable step-size affine projection algorithm," *Signal Processing*, vol. 90, no. 9, pp. 2806–2810, Sep. 2010.

[36] C. Liu, Z. He, H. Z Hang, H. Liao and W. Xia, "A variable step size subband affine projection algorithm with dynamic selection of subband filters," in *IEICE Electronis Express*, vol. 8 No.10, pp.715–721, May 2011

[37] S. J. Kong, K. Y. Hwang and W. J. Song, "An affine projection algorithm with dynamic selection of input vectors," *IEEE Signal Process. Lett.*, vol. 14, no. 8, pp. 529–532, Aug. 2007.

[38] S. E. Kim, S. J. Kong and W. J. Song, "An affine projection algorithm with evolving order," *IEEE Signal Process. Lett.*, vol. 16, no. 11, pp. 937–940, Nov. 2009.

[39] M. Chang, N. Kong and P. Park, "An Affine Projection Algorithm Based on Reuse Time of Input Vectors," *IEEE Signal Process. Lett.*, vol.17, no.8, pp.750–753, Aug. 2010.

[40] A. Gonzalez, M. Ferrer, M. de Diego and G. Piñero, "An affine projection algorithm with variable step-size and projection order," *Digital Signal Processing*, vol 22, no.4, pp. 586-592, Jul. 2012.