**Proceedings of the Internoise 2012/ASME NCAD meeting**
**August 19-22, 2012, New York City, NY, USA**

# IN12-134

# NEW PROPORTIONATE AFFINE PROJECTION ALGORITHM

**Felix Albu**
Valahia University of Targoviste
Targoviste, Romania
felix.albu@valahia.ro

## ABSTRACT

A new proportionate-type affine projection algorithm with intermittent update of the weight coefficients is proposed. It takes into account the "history" of the proportionate factors and uses a fast recursive filtering procedure. Also, the effect of using dichotomous coordinate descent iterations is investigated. Simulation results indicate that the proposed algorithm has improved performance and much lower computational complexity than other proportionate affine projection algorithms. Therefore it represents a practical solution for acoustic echo cancellation systems.

## INTRODUCTION

There are many adaptive algorithms proposed for echo cancellation [1], [2]. Among the most used algorithms are the normalized least mean square (NLMS) algorithm, the affine projection algorithm (APA) [3], and its fast versions, (e.g., [4]–[9]). They have also proved useful for other applications such as hearing aids [6] and active noise control [7]. It is known that the echo paths are often sparse [1]. The sparseness character of the echo paths has been exploited by updating filter coefficients independently and proportionally to their estimated magnitude. One of the first proportionate-type algorithms was proposed by Duttweiler [10], and it was termed the proportionate normalized least-mean-square (PNLMS) algorithm. An improved version has been proposed in [11]. Also, several proportionate-type APAs were developed, (e.g., μ-law PAPA [12], improved PAPA (IPAPA) [13], memory IPAPA (MIPAPA) [14], μ-law MIPAPA (MMIPAPA) [15], and AMIPAPA [16]). In Section 2 a presentation of the dichotomous coordinate descent (DCD) method is made. The DCD method was first time used in an affine projection algorithm proposed in 2005 [5]. Three years later, a fast recursive filtering proved useful in reducing the complexity of the affine projection algorithm [17]. In [18] a proportionate APA using fast recursive filtering and DCD methods, called FMIPAPA-DCD, was introduced. One of the contributions of [19] was to use an approximation for the output error computation of AMIPAPA. However, this complexity reduction led to a reduced performance by several dBs, especially when using speech signals and sparse echo paths. Therefore, a new algorithm with little performance degradation that incorporates an intermittent update of filter coefficients depending on a computed threshold is proposed. Although the threshold is derived for the affine projection algorithm [21] the simulations show that it is effective enough for proportionate affine projection algorithms. The algorithm also uses a new combination of recursive filtering, dichotomous coordinate descent iterations and an approximation of a matrix in order to further reduce its numerical complexity in terms of multiplications. The new algorithm is termed intermittently update approximated FMIPAPA-DCD (IUAFMIPAPA-DCD).

The paper is organized as follows. A short overview of the proportionate-type algorithms for echo cancellation is given and a new algorithm is derived afterwards. The simulations compare the proposed algorithm with other proportionate AP algorithms in the context of echo cancellation. Finally, the conclusions are presented.

## THE PROPOSED ALGORITHM

In an echo cancellation system, the adaptive filter and the unknown system are driven by the same input, the far-end signal $x(n)$, where $n$ is the time index. The reference signal is $d(n)$. The adaptive FIR filter is defined by the real-valued coefficients vector $\hat{\mathbf{h}}(n)=\left[\hat{h}_0(n),\hat{h}_1(n),...,\hat{h}_{L-1}(n)\right]^T$, where $L$ is the length of the adaptive filter and superscript $T$ denotes transposition. The error signal is given by

$$e(n)=d(n)-\hat{\mathbf{h}}^T(n-1)\mathbf{x}(n) \quad (1)$$

where $\mathbf{x}(n) = [x(n), x(n-1),\ldots, x(n-L+1)]^T$ is a vector containing the $L$ most recent samples of the input signal. The error signal vector is given by

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}^T(n)\hat{\mathbf{h}}(n-1) \tag{2}$$

where $\mathbf{d}(n) = \left[d(n), d(n-1),\ldots, d(n-p+1)\right]^T$ is the reference signal vector, with $p$ being the projection order, $\mathbf{e}(n) = \left[e(n), e(n-1),\ldots, e(n-p+1)\right]^T$ is the error vector and $\mathbf{X}(n) = \left[\mathbf{x}(n), \mathbf{x}(n-1),\ldots, \mathbf{x}(n-p+1)\right]$ is the input signal matrix. Proportionate-type affine projection algorithms (PAPA) update their coefficients according to

$$\begin{aligned}\hat{\mathbf{h}}(n) &= \hat{\mathbf{h}}(n-1) + \mu\mathbf{G}(n-1)\mathbf{X}(n)\times \\ &\left[\delta\mathbf{I}_p + \mathbf{X}^T(n)\mathbf{G}(n-1)\mathbf{X}(n)\right]^{-1}\mathbf{e}(n)\end{aligned} \tag{3}$$

where $\delta$ is a regularization constant, $\mu$ is the normalized step-size parameter, $\mathbf{G}(n-1)$ is an $L$ x $L$ diagonal matrix, and $\mathbf{I}_p$ is the $p$ x $p$ identity matrix. In the case of the memory improved PAPA (IPAPA) [13], the diagonal elements of $\mathbf{G}(n-1)$, denoted by $g_l(n-1)$, are evaluated as

$$g_l(n-1) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{\left|\hat{h}_l(n-1)\right|}{2\sum_{i=0}^{L-1}\left|\hat{h}_i(n-1)\right| + \xi} \tag{4}$$

where $-1 \le \alpha < 1$, $0 \le l < L-1$ and $\xi$ is a small positive constant. Let us denote [14]

$$\begin{aligned}\mathbf{P}(n) &= \mathbf{G}(n-1)\mathbf{X}(n) \\ &= \left[\mathbf{g}(n-1)\odot\mathbf{x}(n)\ldots\mathbf{g}(n-1)\odot\mathbf{x}(n-p+1)\right]\end{aligned} \tag{5}$$

where $\mathbf{g}(n-1)$ is a vector containing the diagonal elements of $\mathbf{G}(n-1)$ and the operator $\odot$ denotes the Hadamard product [14]. Following a similar idea used to derive the pseudo affine projection algorithm [7], [9], $\mathbf{P}(n)$ is approximated with

$$\mathbf{P}'(n) = \left[\mathbf{g}(n-1)\odot\mathbf{x}(n)\ldots\mathbf{g}(n-p)\odot\mathbf{x}(n-p+1)\right] \tag{6}$$

where $\mathbf{g}(n-k)$ are the vectors containing the diagonal elements of the matrixes $\mathbf{G}(n-k)$, with $k = 1, 2, \ldots, p$ [14]. The computational complexity is lower if compared to (5), because (6) can be written as

$$\mathbf{P}'(n) = \left[\mathbf{g}(n-1)\odot\mathbf{x}(n) \quad \mathbf{P}'_{-1}(n-1)\right] \tag{7}$$

where the matrix

$$\begin{aligned}&\mathbf{P}'_{-1}(n-1) = \\ &\left[\mathbf{g}(n-2)\odot\mathbf{x}(n-1)\ldots\mathbf{g}(n-p)\odot\mathbf{x}(n-p+1)\right]\end{aligned} \tag{8}$$

contains the first $p-1$ columns of $\mathbf{P}'(n-1)$. The MIPAPA equations are written as in [16]:

$$\mathbf{S}_1(n) = \delta\mathbf{I}_p + \mathbf{X}^T(n)\mathbf{P}'(n) \tag{9}$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{P}'(n)\mathbf{S}_1^{-1}(n)\mathbf{e}(n) \tag{10}$$

Important computational savings for large projection orders and filter lengths can be achieved if the approximation from [16] is used. The coefficients of the approximated MIPAPA (AMIPAPA) are given by [16]

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{P}'(n)\mathbf{S}_2^{-1}(n)\mathbf{e}(n) \tag{11}$$

where, $\mathbf{S}_2(n)$, is updated by changing both its first row and column with $\mathbf{x}^T(n)\mathbf{P}'(n)$. The bottom-right $(p-1)\times(p-1)$ submatrix of $\mathbf{S}_2(n)$ is replaced with the top-left $(p-1)\times(p-1)$ submatrix of $\mathbf{S}_2(n-1)$. The regularization factor, $\delta$, is added at the end on the diagonal of $\mathbf{S}_2(n)$. This is a different matrix update procedure than that of [16].

A recursive filtering approach similar to that of [17] can exploit the time-shift property of $\mathbf{P}'(n)$. The following equation is obtained:

$$\begin{aligned}\mathbf{y}(n-1) &= \mathbf{X}^T(n-1)\hat{\mathbf{h}}(n-2) \\ &= \left[\mathbf{x}^T(n-1)\hat{\mathbf{h}}(n-2)\ \ldots\ \mathbf{x}^T(n-p)\hat{\mathbf{h}}(n-2)\right]^T \\ &= \left[y^0(n-1)\ y^1(n-1)\ \ldots\ y^{p-1}(n-1)\right]^T\end{aligned} \tag{14}$$

where $y^k(n-1) = \mathbf{x}^T(n-1-p)\hat{\mathbf{h}}(n-2)$. Also, we obtain:

$$\mathbf{y}(n)=\mathbf{X}^T(n)\hat{\mathbf{h}}(n-1)=\mathbf{z}(n)+\mathbf{F}(n)\hat{\boldsymbol{\varepsilon}}(n-1) \tag{15}$$

where

$$\mathbf{F}(n)=\mathbf{X}^T(n)\mathbf{P}'(n-1) \tag{16}$$

and

$$\begin{aligned}\mathbf{z}(n)&=\mathbf{X}^T(n)\hat{\mathbf{h}}(n-2)\\ &=\left[\mathbf{x}^T(n)\hat{\mathbf{h}}(n-2)\ y^0(n-1)\ \ldots y^{p-2}(n-1)\right]^T\end{aligned} \tag{17}$$

If we define the filter output vector $\mathbf{y}(n)=\left[y(n),y(n-1),\ldots,y(n-p+1)\right]^T$, and $\hat{\boldsymbol{\varepsilon}}(n)$ is the solution of the linear system

$$\mathbf{S}_2(n)\hat{\boldsymbol{\varepsilon}}(n)=\mathbf{e}(n) \tag{18}$$

If Eqn. (18) is solved with DCD iterations we obtain:

$$\mathbf{y}(n)=\mathbf{z}(n)+\mathbf{F}(n)\hat{\boldsymbol{\varepsilon}}(n-1) \tag{19}$$

$$\hat{\mathbf{h}}(n)=\hat{\mathbf{h}}(n-1)+\mu\mathbf{P}'(n)\hat{\boldsymbol{\varepsilon}}(n) \tag{20}$$

The update of $\mathbf{F}(n)$ involves only the computation of its first row and column and the bottom-right $(p-1)\times(p-1)$ submatrix of $\mathbf{F}(n)$ is replaced with the top-left $(p-1)\times(p-1)$ submatrix of $\mathbf{F}(n-1)$ ([16] and [17]).

However, the complexity can be further reduced using the intermittently updated procedure proposed in [20]. Thus, the update equation of (20) can be replaced by

$$\hat{\mathbf{h}}(n)=\begin{cases}\hat{\mathbf{h}}(n-1)+\mu\mathbf{P}'(n)\hat{\boldsymbol{\varepsilon}}(n), & \text{if } n \bmod i_n=0\\ \hat{\mathbf{h}}(n-1) & \text{otherwise}\end{cases} \tag{21}$$

where $i_n$ is the variable update interval at time *n.* Starting with an initial update interval of 1, $i_n$ is given by

$$i_n=\begin{cases}\max\left[1,i_{n-1}-1\right], & \text{if } e^2(n)\geq\gamma\\ \min\left[i_{n-1}+1,i_M\right] & \text{otherwise}\end{cases} \tag{22}$$

where $i_M$ is the maximum update interval and $\gamma$ is the threshold [20] computed as in (18)

$$\gamma=\frac{\mu\sigma_v^2 p}{2-\mu}+\sigma_v^2 \tag{23}$$

where $\sigma_v^2$ is estimated during silences [21]. Although the threshold is computed for the affine projection algorithm [21], [22], the simulations from the next section show that it is effective enough for proportionate affine projection algorithms too. It can be noticed that the numerical savings can be important since Eqn. (20) requires at least $Lp$ multiplications and the filter can have hundreds of coefficients in echo cancellation systems. The update of the filter coefficients from Eqn. (21) is performed only when $n \bmod i_n=0$ and not at every iteration like in Eqn. (20). The new algorithm is termed intermittently updated approximated FMIPAPA-DCD (IUAFMIPAPA-DCD). The algorithm can have a periodic update if the update interval is fixed to $i_n>1$.

## DICHOTOMOUS COORDINATE DESCENT ALGORITHM

In order to solve the linear system, $\mathbf{S}_2(n)\hat{\boldsymbol{\varepsilon}}(n)=\mathbf{e}(n)$, many direct or iterative methods can be used [23]. The 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 $\alpha$ that takes one of $M_b$ (number of bits) predefined values corresponding to a binary representation bounded by an interval $[-H,H]$ [5], [17]. 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. If H is a power of 2, the multiplications are replaced by bit shifts. The algorithm has only shift and accumulate operations (SAC) and no divisions. As shown in previous papers (e.g. [5], [17] and [18]) the DCD method approximates very well the exact solution of a linear system if enough DCD iterations are executed. The maximum complexity of DCD part for our cases is $p(2N_u+M_b)$ SACs.

## SIMULATION RESULTS

Simulations were performed in the context of echo cancellation, where the input signal is either white Gaussian noise or speech. The first impulse response from ITU-T G168

Recommendation [24] is padded with zeros in order to have 512 coefficients. The adaptive filter has 512 taps. For the first five figures, a white Gaussian noise with a SNR = 30 dB is added at the output of the echo path. The performance measure used is the normalized misalignment (in dB), defined as $20\log_{10}(\|\mathbf{h} - \hat{\mathbf{h}}(n)\|_2/\|\mathbf{h}\|_2)$, where $\mathbf{h}$ denotes the true impulse response of the echo path. In the simulations with white noise, the performance curves are averaged over 10 independent trials. The regularization constant is $\delta = 0.01$, $p = 8$ and $\alpha = 0$. In all the simulations where the input signal is a white signal, the step size of all algorithms is 0.1.

The first five figures don't take into account the DCD iterations for solving Eqn. (18). This corresponds to the ideal system solution that DCD iterations try to approximate. Therefore the investigated algorithms are termed AFMIPAPA and IUAFMIPAPA respectively.

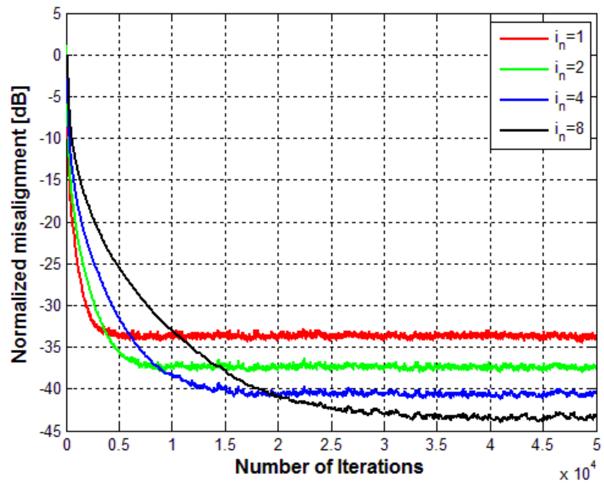
Figure 1 shows the misalignment performance of the periodic AFMIPAPA with fixed periodical update of filter coefficients. Similar conclusions as those of [20] are obtained, i.e., the larger the update interval, the lower steady-state error and the slower the convergence speed. This indicates that a variable updating interval for AFMIPAPA could lead to a good compromise between fast convergence and low steady-state error.

Figure 2 shows misalignment curves for the proposed IUAFMIPAPA ($i_M = 8$), AFMIPAPA, and the periodic AFMIPAPA with $i_n = 8$. The tracking ability of the algorithms was verified by introducing an abrupt change of the echo path after 25000 iterations by shifting the impulse response to the right by 12 samples. As can be seen, the proposed IUAFMIPAPA has roughly the same initial convergence as AFMIPAPA ($i_n = 1$).

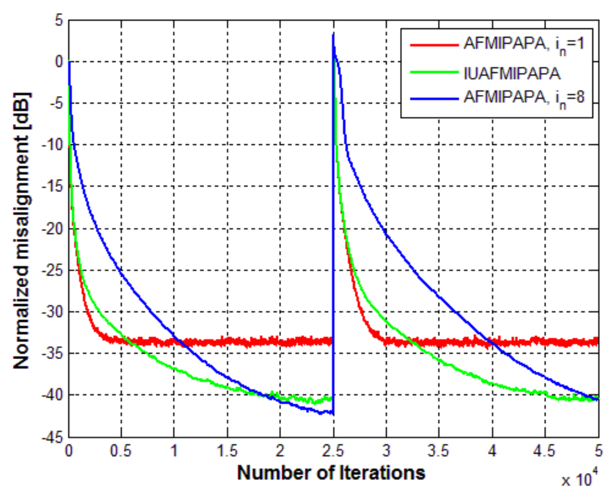
Figure 3 shows misalignment curves for IUAFMIPAPA for different update intervals. Similar conclusions with those obtained in [20] are obtained regarding the influence of $i_M$. The time to reach steady-state increases with $i_M$ and the percentage of updates reduces with $i_M$ (e.g. by 65% from $i_M = 8$ to $i_M = 32$). As in [20], setting $i_M$ to the projection order leads to a satisfactory convergence rate. The maximum update interval is set to the projection order in the following simulations.

An example of computed $i_n$ values and their histogram for the case $i_M = 8$ (Figure 3) is shown in Figure 4. It can be seen that during the initial convergence, the updating intervals are closer to 1, while they are closer to 8 in the steady-state region.

In Figure 5 the input signal is speech. The output of the echo path is corrupted by independent white Gaussian noise SNR = 30 dB. The echo path changes after 0.5 seconds and $p$ = 8. It was shown in [16] that MIPAPA and AMIPAPA have virtually the same performance, both being superior to the IPAPA. Therefore only AMIPAPA curve is shown and compared with that of IUAFMIPAPA. The step-size for all algorithms is 0.2. Figure 5 shows that, in case of a speech signal input, the approximations used by IUAFMIPAPA and the intermittent update of filter weights lead to slightly reduced performance (around 2 dB for this example) in comparison with AMIPAPA.

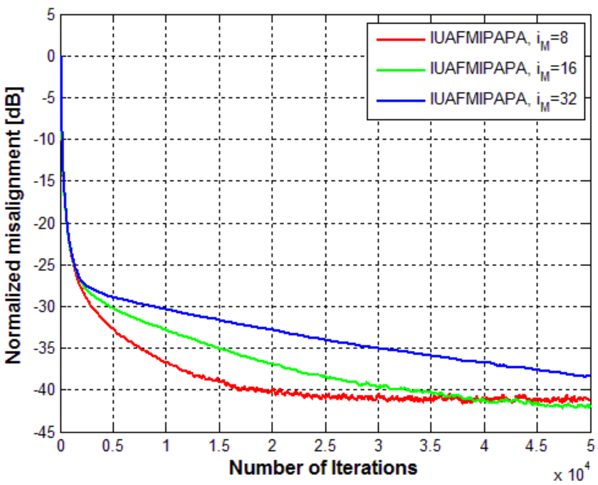



- **Figure 1. Misalignment of periodic AFMIPAPA for different update intervals, *i*. White noise, *p* = 8, *L* = 512, SNR = 30 dB.**

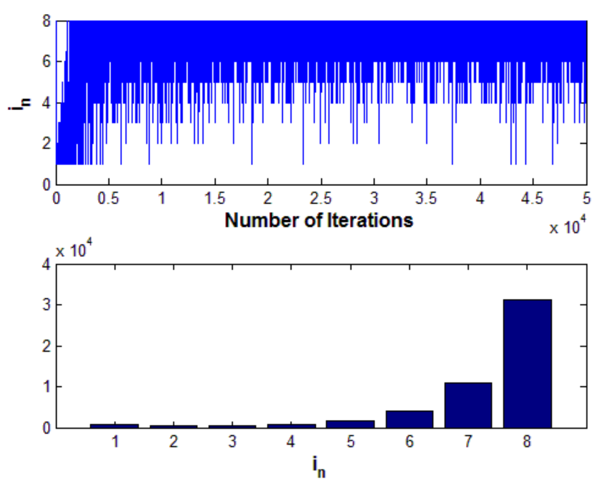



- **Figure 2. Misalignment of AFMIPAPA ($i_n = 1$), periodic AFMIPAPA with $i_n = 8$, and IUAFMIPAPA $i_M = 8$. Other conditions are the same as in Figure 1.**

The effect of the number of DCD iterations ( $M_b = 16$ , $H = 256$ ) on IUAFMIPAPA-DCD is investigated in the next simulation. Figure 6 shows the misalignment curves of the IUFMIPAPA and the IUFMIPAPA-DCD for two particular values of $N_u$ ( $N_u = 8$ and $N_u = 16$ ).

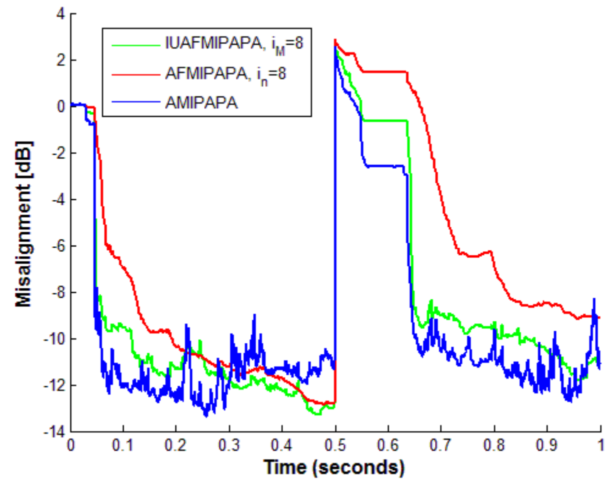



- **Figure 3. Misalignment of IUSAMIPAPA with $i_M = 8$, $i_M = 16$ and $i_M = 32$ respectively. Other conditions are the same as in Figure 1.**

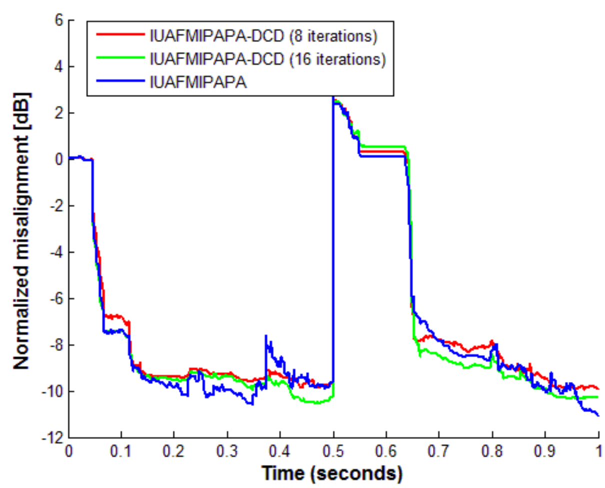



- **Figure 4. (a) Computed update interval values; (b) histogram of computed $i_n$ values**

A speech sequence is used as input, SNR = 20 dB, and echo path changes at 0.5 seconds, and $p$ = 8. If less than 1.5 dB misalignment differences is allowed, 8 DCD iterations are enough for the DCD based algorithm. In case of using 16 DCD iterations in IUAFMIPAPA-DCD the misalignment difference is smaller than for the case of 8 DCD iterations. The performance improvement obtained with more DCD iterations can be explained by examining the error norm between the exact solution and the solution obtained with different number of DCD iterations as shown in [18].




- **Figure 5. Misalignment of the AMIPAPA, AFMIPAPA ( $i_n = 8$ ) and IUAFMIPAPA ( $i_M = 8$ ). Speech sequence, *p* = 8, *L* = 512, SNR = 30 dB, and echo path changes at time 0.5s.**




- **Figure 6. Misalignment of IUAFMIPAPA and IUAFMIPAPA-DCD with different number of DCD iterations (8 and 16 respectively).**

On average, the update of the filter weights in IUAFMIPAPA-DCD is performed only on 20% of the number of iterations. Overall, for the investigated cases, IUAFMIPAPA-DCD obtains a complexity reduction over FMIPAPA in terms of multiplications in the range 15%-25 %, depending on the filter length, projection order and input signals. Therefore, IUAFMIPAPA-DCD offers a better performance/complexity tradeoff than FMIPAPA/MIPAPA, due to its reduced numerical complexity. It is particularly more efficient for higher projection orders, when the size of the linear system increases. Future work will be focused in adapting the techniques proposed in [25] – [27] and developing new variable step size and variable projection order of the proposed algorithm. Also, a logarithmic number system implementation is envisaged [28] – [29].

## SUMMARY AND CONCLUSIONS

A proportionate affine version using intermittent update of filter coefficients, fast recursive filtering, dichotomous coordinate descent iterations and a matrix approximation is proposed. It is shown that IUAFMIPAPA-DCD offers a good convergence performance/numerical complexity compromise in comparison with other proportionate AP algorithms.

## ACKNOWLEDGMENTS

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.

## REFERENCES

1. Benesty, J. Gaensler, T., Morgan, D.R., Sondhi, M.M, 2001, “Advances in network and acoustic echo cancellation,” *Springer-Verlag, Berlin, Germany.*

2. Haensler, E. and Schmidt, G., Eds., 2006, “Topics in Acoustic Echo and Noise Control,” *Springer-Verlag, Berlin, Germany.*

3. Ozeki, K., and Umeda, T., 1984, “An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties,” in *Electronics Communications Japan,* vol. 67-A, (5), 19–27.

4. Gay, S. L., and Tavathia, S., 1995, “The fast affine projection algorithm,” in *Proc. of IEEE ICASSP*, Detroit, Michigan, U.S.A, 3023 – 3026.

5. Zakharov, Y. and Albu, F., 2005, “Coordinate descent iterations in fast affine projection algorithm,” *IEEE Signal Processing Letters*, vol. 12, 353–356.

6. Lee, S., Kim, I., Park, Y., 2007, “Approximated affine projection algorithm for feedback cancellation in hearing aids” *Computer Methods and Programs in Biomedicine*, Volume 87, (3), 254-261.

7. Bouchard, M., and Albu, F., 2003, “The Multichannel Gauss-Seidel Fast Affine Projection Algorithm for Active Noise Control”, in *Proc. of IEEE ISSPA*, vol. 2, 579-582.

8. Albu, F., and Fagan, A., 2003, “The Gauss-Seidel pseudo affine projection algorithm and its application for echo cancellation”, in *Proc. of ASILOMAR* 2003, Vol. 2, 1303-1306.

9. Albu, F., and Kwan, H.K., 2004, “Combined echo and noise cancellation based on Gauss-Seidel pseudo affine projection algorithm”, in *Proc. of IEEE ISCAS* 2004, Vancouver, Canada, 505-508.

10. Duttweiler, D. L., 2000, “Proportionate normalized least-mean-squares adaptation in echo cancellers,” *IEEE Transactions on Speech and Audio Processing*, vol. 8, (5), 508–518.

11. Benesty, J., and Gay, S. L., 2002, “An improved PNLMS algorithm,” in *Proc. of IEEE ICASSP* 2002, vol. II, 1881–1884.

12. Deng, H., and Doroslovački, M., 2006, “Proportionate adaptive algorithms for network echo cancellation,” *IEEE Transactions on Signal Processing*, vol. 54, (5), 1794–1803.

13. Hoshuyama, O., Goubran, R. A., and Sugiyama, A., 2004, “A generalized proportionate variable step-size algorithm for fast changing acoustic environments,” in *Proc. of IEEE ICASSP* 2004, vol IV, 161-164.

14. Paleologu, C., Ciochina, S., and Benesty, J., 2010, “An efficient proportionate affine projection algorithm for echo cancellation,” *IEEE Signal Processing Letters*, vol. 17, (2), 165–168.

15. Yang, J., and Sobelman, G.E., 2010, “Efficient $\mu$ -law improved proportionate affine projection algorithm for echo cancellation”, *Electronics Letters,* vol. 47, (2), 73 – 74.

16. Albu, F., Paleologu, C., Benesty, J., and Ciochina, S., 2010, “A low complexity proportionate affine projection algorithm for echo cancellation,” in *Proc. of EUSIPCO 2010*, Aalborg, Denmark, 6-10.

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

18. F. Albu, 2011, “A Proportionate Affine Projection Algorithm using Fast Recursive Filtering and Dichotomous Coordinate Descent Iterations”, in *Proc. of SPAMEC 2011*, Cluj, Romania, 93-96

19. Albu, F., 2012, “A simplified proportionate affine projection algorithm”, in *Proc. of IWSSIP 2012*, Vienna, Austria, 382-385.

20. Kim, K.-H., Choi, Y.-S., Kim, S.-E., and Song, W. -J., 2011, “An Affine Projection Algorithm with Periodically Evolved Update Interval”, *IEEE Trans on Circuits and Systems-II*, vol. 58, (11), 763-767.

21. Benesty, J., Rey, H., Rey Vega, L., and Tressens, S., 2006, “A nonparametric VSS NLMS algorithm,” *IEEE Signal Processing Letters*, vol. 13, (10), 581–584.

22. Shin, H.-C., Sayed, A. H. and Song, W.-J., 2004, “Variable step-size NLMS and affine projection algorithms,” *IEEE Signal Processing Leters*, vol. 11, (2), 132–135.

23. Golub, G. H., and Van Loan, C. F., 1996, Matrix computation, 3rd edition, Baltimore, MD, *The John Hopkins Univ. Press*.

24. *Digital Network Echo Cancellers*, 2002, ITU-T Rec. G.168

25. Albu F., Paleologu C., Benesty J., 2011, “A Variable Step Size Evolutionary Affine Projection Algorithm”, in *Proc. of ICASSP 2011*, Prague, Czech Republic, pp. 429-432.

26. Albu F., Paleologu C., J. Benesty, 2010, “Efficient Implementation of a Variable Projection Order Affine Projection Algorithm”, *International Symposium on Wireless Communication Systems (ISWCS)*, York, U.K., pp. 369-373.

27. Albu F., Paleologu C., 2009, “The Variable Step-Size Gauss-Seidel Pseudo Affine Projection Algorithm”, in *Proc. of WASET*, vol. 37, Dubai, UAE, pp. 642-645.

28. Albu F., Kadlec J., Fagan A., Hermanek A., Coleman N., 2002, " Analysis of the LNS Implementation of the Fast Affine Projection algorithms”, in *Proc. of ISSC 2002*, Cork, Ireland, pp. 251-255.

29. Albu F., Kadlec J., Coleman N., Fagan A., 2002, “Pipelined Implementations of the *A Priori* Error-Feedback LSL Algorithm Using Logarithmic Number System”, in *Proc. of ICASSP 2002*, Orlando, Florida, U.S.A., pp. 2681-2684.