A LOW COMPLEXITY PROPORTIONATE AFFINE PROJECTION ALGORITHM FOR ECHO CANCELLATION

Felix Albu[1], *Constantin Paleologu*[1], *Jacob Benesty*[2], *and Silviu Ciochina*[1]

[1] Department of Telecommunications, University Politehnica of Bucharest
1-3, Iuliu Maniu Blvd., 061071, Bucharest, Romania
email: {felix, pale, silviu}@comm.pub.ro
[2] INRS-EMT, Universite du Quebec
QC H5A 1K6, Montreal, Canada
email: benesty@emt.inrs.ca

ABSTRACT

Proportionate-type affine projection algorithms were developed in the context of echo cancellation, as a generalization of the proportionate-type normalized least-mean-square algorithms. A matrix inversion is required within the affine projection algorithm (APA). In the case of proportionate-type APAs, the update of the matrix to be inverted is very computationally expensive. In this paper, an efficient update of this matrix is proposed and the procedure is applied for a recently developed proportionate-type APA. It is shown that the proposed algorithm achieves similar performance but significantly lowers numerical complexity as compared to known proportionate-type APAs.

1. INTRODUCTION

Many interesting adaptive algorithms have been proposed for echo cancellation [1], [2]. The main goal is to identify an unknown system, i.e., the echo path, providing at the output of the adaptive filter a replica of the echo. Nevertheless, the echo paths (for both network and acoustic echo cancellation scenarios) have a specific property, which can be used in order to help the adaptation process. These systems are sparse in nature, i.e., a small percentage of the impulse response components have a significant magnitude while the rest are zero or small. The sparseness character of the echo paths inspired the idea to "proportionate" the algorithm behavior, i.e., to update each coefficient of the filter independently of the others, by adjusting the adaptation step-size in proportion to the magnitude of the estimated filter coefficient. In this manner, the adaptation gain is "proportionately" redistributed among all the coefficients, emphasizing the large ones in order to speed up their convergence, and consequently to increase the overall convergence rate. The proportionate normalized least-mean-square (PNLMS) algorithm [3] proposed by Duttweiler almost a decade ago, was one of the first proportionate-type algorithms. An insightful overview of the proportionate-type algorithms can be found in [2] (chapter 5); also, some recently proposed proportionate-type NLMS algorithms can be found in [4] and [5].

In the context of echo cancellation, the affine projection algorithm (APA) [6] and its fast versions, i.e., the fast affine projection (FAP) algorithms (e.g., [7]–[11]), were found to be very attractive choices. Also, several proportionate-type APAs were developed [12]–[14], as a straightforward extension of the proportionate-type NLMS algorithms. It is known that a matrix inversion is required within the APA. Most of the FAP algorithms implement this operation in a computationally efficient manner, by taking into account the properties of the matrix to be inverted, i.e., time-shift character and symmetrical structure. Unfortunately, this is not valid in the case of proportionate-type APAs. The only proportionate-type APA with a matrix to be inverted having a time-shift character (but not symmetric) was recently proposed in [15]. This algorithm was called the "memory"-improved proportionate APA (MIPAPA) and it was derived as a version of the improved proportionate APA (IPAPA) [14] [which is a generalization of the improved proportionate NLMS (IPNLMS) algorithm [16]].

The contribution of this paper is that an approximation is proposed for the MIPAPA in order to update in a computationally efficient manner the matrix to be inverted. The paper is organized as follows. Section 2 represents an overview of the proportionate-type algorithms for echo cancellation; also, an efficient implementation of the MIPAPA is presented. In Section 3, an approximated MIPAPA (AMIPAPA) is derived. The numerical complexity of these algorithms is investigated in Section 4. The simulation results presented in Section 5 compare the proposed algorithm with IPAPA and MIPAPA in the context of echo cancellation. Finally, the conclusions are given in Section 6.

2. OVERVIEW OF THE PROPORTIONATE-TYPE ALGORITHMS FOR ECHO CANCELLATION

In the context of echo cancellation, an adaptive filter is used to model an unknown system, i.e., the echo path. Both systems are driven by the same input, i.e., the far-end signal $x(n)$, where n is the time index. The reference signal of the adaptive filter, $d(n)$, contains the output of the echo path (i.e., the echo signal) and the near-end signal. Let us assume an adaptive finite-impulse-response filter defined by the real-valued coefficients vector $\hat{\mathbf{h}}(n) = [\hat{h}_0(n), \hat{h}_1(n),\ldots, \hat{h}_{L-1}(n)]^T$, where L is the length of the adaptive filter and superscript T denotes transposition. The error signal is defined as

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

where $\mathbf{x}(n) = [x(n), x(n-1),\ldots, x(n-L+1)]^T$ is a real-valued vector containing the L most recent samples of the input signal. A proportionate-type NLMS algorithm [3] updates its coefficients according to

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \frac{\mu \mathbf{G}(n-1)\mathbf{x}(n)e(n)}{\delta + \mathbf{x}^T(n)\mathbf{G}(n-1)\mathbf{x}(n)}, \quad (2)$$

where μ is the normalized step-size parameter, δ is the regularization constant, and $\mathbf{G}(n-1)$ is an L x L diagonal matrix which assigns an individual step-size to each filter coefficient. This type of algorithms converges faster than NLMS especially for sparse impulse responses. The diagonal elements of $\mathbf{G}(n-1)$, which allocate a certain gain for each filter coefficient, should be evaluated based on the adaptive filter coefficients only. In the case of the IPNLMS algorithm [16], the diagonal elements of $\mathbf{G}(n-1)$, denoted in the following by $g_l(n-1)$, with $0 \le l \le L-1$, are evaluated as

$$g_l(n-1) = \frac{1-\alpha}{2L} + (1+\alpha)\frac{\left|\hat{h}_l(n-1)\right|}{2\sum_{i=0}^{L-1}\left|\hat{h}_i(n-1)\right| + \xi}, \quad (3)$$

where $-1 \le \alpha < 1$ and the small positive constant ξ avoids division by zero (especially at the beginning of the adaptation when all the filter taps are initialized to zero); in practice, good choices for the parameter α are 0 or –0.5.

Due to its convergence performance (especially for correlated inputs), the APA is also frequently used for echo cancellation. The equations of the classical APA [6] are

$$\mathbf{e}(n) = \mathbf{d}(n) - \mathbf{X}^T(n)\hat{\mathbf{h}}(n-1), \quad (4)$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{X}(n)\left[\delta\mathbf{I}_p + \mathbf{X}^T(n)\mathbf{X}(n)\right]^{-1}\mathbf{e}(n), \quad (5)$$

where $\mathbf{d}(n) = \left[d(n), d(n-1),\ldots,d(n-p+1)\right]^T$ is the reference signal vector of length p, with p denoting the projection order, $\mathbf{X}(n) = \left[\mathbf{x}(n), \mathbf{x}(n-1),\ldots,\mathbf{x}(n-p+1)\right]$ is the input signal matrix, and $\mathbf{I}_p$ is the p x p identity matrix. Most of the proportionate-type APAs (e.g., [12]–[14]) were straightforwardly obtained from the proportionate-type NLMS algorithms, by a simple extension of the proportionate idea. Their coefficients are updated by

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{G}(n-1)\mathbf{X}(n) \times\left[\delta\mathbf{I}_p + \mathbf{X}^T(n)\mathbf{G}(n-1)\mathbf{X}(n)\right]^{-1}\mathbf{e}(n). \quad (6)$$

When the elements of $\mathbf{G}(n-1)$ are evaluated as in (3), the IPAPA is obtained [14]. Let us denote

$$\mathbf{P}(n) = \mathbf{G}(n-1)\mathbf{X}(n) = \left[\mathbf{g}(n-1)\odot\mathbf{x}(n)\ldots\mathbf{g}(n-1)\odot\mathbf{x}(n-p+1)\right], \quad (7)$$

where $\mathbf{g}(n-1)$ is a vector containing the diagonal elements of $\mathbf{G}(n-1)$; the operator $\odot$ denotes the Hadamard product, i.e., $\mathbf{a}\odot\mathbf{b} = \left[a(1)b(1), a(2)b(2),\ldots,a(L)b(L)\right]^T$, where $\mathbf{a}$ and $\mathbf{b}$ are two vectors of length L. The other equations of IPAPA are

$$\mathbf{S}(n) = \delta\mathbf{I}_p + \mathbf{X}^T(n)\mathbf{P}(n), \quad (8)$$

$$\text{Solve } \mathbf{S}(n)\boldsymbol{\varepsilon}(n) = \mathbf{e}(n) \quad \Rightarrow \quad \boldsymbol{\varepsilon}(n), \quad (9)$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{P}(n)\boldsymbol{\varepsilon}(n), \quad (10)$$

where the solving of the linear system of (9) is performed in the classical manner using $\mathbf{LDL}^T$ method [17].

Following a general framework for the derivation of proportionate-type APAs, it was shown in [15] that a more proper approach instead of (7) is

$$\mathbf{P}'(n) = \left[\mathbf{g}(n-1)\odot\mathbf{x}(n)\ldots\mathbf{g}(n-p)\odot\mathbf{x}(n-p+1)\right], \quad (11)$$

where $\mathbf{g}(n-k)$ are the vectors containing the diagonal elements of the matrixes $\mathbf{G}(n-k)$, with $k = 1, 2, \ldots, p$. The advantage of this modification is twofold. First, this algorithm takes into account the "history" of the proportionate factors from the last p steps. Second, the computational complexity is lower as compared to (7), because (11) can be written as

$$\mathbf{P}'(n) = \left[\mathbf{g}(n-1)\odot\mathbf{x}(n) \quad \mathbf{P}'_{-1}(n-1)\right], \quad (12)$$

where the matrix

$$\mathbf{P}'_{-1}(n-1) = \left[\mathbf{g}(n-2)\odot\mathbf{x}(n-1)\ldots\mathbf{g}(n-p)\odot\mathbf{x}(n-p+1)\right] \quad (13)$$

contains the first $p-1$ columns of $\mathbf{P}'(n-1)$. Thus, the matrix $\mathbf{P}'(n)$ has the time-shift character, requiring only L multiplications to be evaluated. However, the matrix to be inverted [i.e., $\delta\mathbf{I}_p + \mathbf{X}^T(n)\mathbf{P}'(n)$] is not symmetric. When the elements of the vectors $\mathbf{g}(n-k)$, with $k = 1, 2, \ldots, p$, are evaluated as in (3), the MIPAPA is obtained [15]. For the MIPAPA, the matrix $\mathbf{X}^T(n)\mathbf{P}'(n)$ has the time-shift property. The time-shift property can be exploited in order to reduce the complexity of updating the matrix $\mathbf{S}'(n) = \delta\mathbf{I}_p + \mathbf{X}^T(n)\mathbf{P}'(n)$. Only the first row and column of $\mathbf{S}'(n)$ are computed, and the bottom-right $(p-1)\times(p-1)$ submatrix of $\mathbf{S}'(n)$ is replaced with the top-left $(p-1)\times(p-1)$ submatrix of $\mathbf{S}'(n-1)$. Important computational savings are obtained because only a part of the matrix $\mathbf{S}'(n)$ is re-computed. The first column is given by $\mathbf{X}^T(n)\cdot\left[\mathbf{g}(n-1)\odot\mathbf{x}(n)\right]$, while the first row is computed as $\mathbf{x}^T(n)\mathbf{P}'(n)$. The other equations of MIPAPA are

$$\text{Solve } \mathbf{S}'(n)\boldsymbol{\varepsilon}(n) = \mathbf{e}(n) \quad \Rightarrow \quad \hat{\boldsymbol{\varepsilon}}(n), \quad (14)$$

$$\hat{\mathbf{h}}(n) = \hat{\mathbf{h}}(n-1) + \mu\mathbf{P}'(n)\hat{\boldsymbol{\varepsilon}}(n). \quad (15)$$

3. LOW COMPLEXITY PROPORTIONATE-TYPE APA

In the IPAPA, the step required in (8) is computationally expensive, because the matrix $\mathbf{X}^T(n)\mathbf{P}(n)$ does not have the time-shift property. For the MIPAPA, the matrix $\mathbf{S}'(n)$ has the time-shift property, but it is not symmetric. Important computational savings, especially for large filter lengths and projection orders, can be achieved if an approximation is made in order to obtain a symmetric matrix. This new

matrix, $\mathbf{S}''(n)$, is obtained by updating both its first row and its first column with $\mathbf{X}^T(n)\cdot\left[\mathbf{g}(n-1)\odot\mathbf{x}(n)\right]=\mathbf{X}^T(n)\mathbf{P}'_{:,1}(n)$ and adding δ to the first element [where $\mathbf{P}'_{:,1}(n)$ denotes the first column of $\mathbf{P}'(n)$]. A similar updating procedure was extensively used in FAP algorithms, e.g., [7]–[11], for updating the correlation matrix. Like for the MIPAPA, the bottom-right $(p-1)\times(p-1)$ submatrix of $\mathbf{S}''(n)$ is replaced with the top-left $(p-1)\times(p-1)$ submatrix of $\mathbf{S}''(n-1)$. The resulted algorithm, called the approximated MIPAPA (AMIPAPA), will require the following system of equations to be solved:

$$\mathbf{S}''(n)\boldsymbol{\varepsilon}(n)=\mathbf{e}(n) \quad \Rightarrow \quad \hat{\boldsymbol{\varepsilon}}(n). \tag{16}$$

The other equations of AMIPAPA are identical with those of MIPAPA.

Fig. 1 shows the error norm between the vectors containing the first row using the updating procedure of AMIPAPA [i.e. $\mathbf{X}^T(n)\mathbf{P}'_{:,1}(n)$] and MIPAPA [i.e. $\mathbf{P}'^T(n)\mathbf{x}(n)$], respectively.

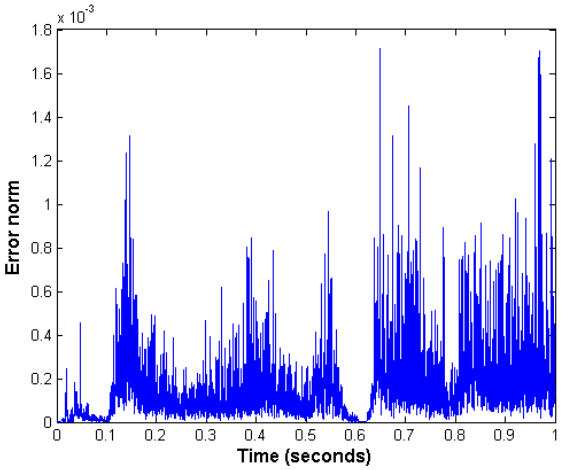

Fig. 1. The error norm between first row updating methods of MIPAPA and AMIPAPA algorithms.

It can be seen from Fig. 1 that error norm is very small. It will be shown in Section 5 that this fact leads to almost identical performance of AMIPAPA and MIPAPA algorithms in the investigated cases.

4. NUMERICAL COMPLEXITY COMPARISON

The numerical complexity of the investigated algorithms in terms of multiplications is the following:

$$C_{\text{IPAPA}}=L\left(p^2+3p+1\right)+p+P_m, \tag{17}$$

$$C_{\text{MIPAPA}}=L\left(4p+1\right)+p+P_m, \tag{18}$$

$$C_{\text{AMIPAPA}}=L\left(3p+2\right)+p+P_m. \tag{19}$$

The notation $P_m = O(p^3)$ indicates the numerical complexity in terms of multiplications associated with solving the linear systems of equations using the $\mathbf{LDL}^T$ method ([9], [17]).

Fig. 2 shows the numerical complexity comparison in two situations: a) as a function of L and fixed $p = 8$ and b) as a function of p and fixed $L = 512$. The number of multiplications varies linearly with the filter length for all the considered algorithms. Also, the variation of the term that multiplies the filter length, L, is proportional to p^2 for the IPAPA, while it is proportional to p for MIPAPA and AMIPAPA. It can be seen from Fig. 2 that AMIPAPA is the least complex in terms of multiplications, especially for large filter lengths or projection orders. The MIPAPA follows after AMIPAPA, with the IPAPA being the most complex algorithm.

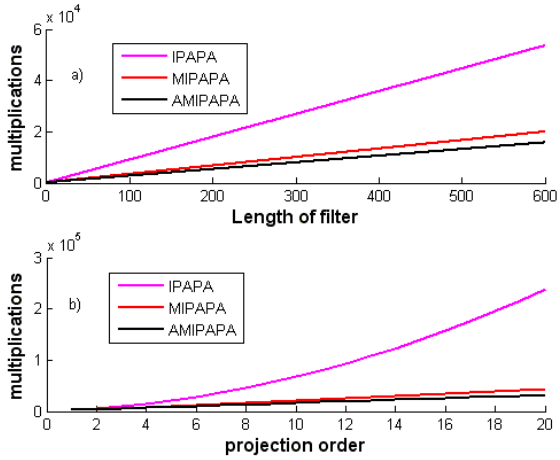

Fig. 2. Numerical complexity of the considered algorithms in terms of multiplications for $p = 8$ in two situations: a) variable L; b) variable p.

The numerical savings are important. For example, with $L = 512$, $p = 8$, the IPAPA needs 45716 multiplications, while the MIPAPA is about 60% less complex (i.e., it requires 17044 multiplications). The least complex algorithm is the AMIPAPA, which needs 13460 multiplications; therefore, it is about 67% less complex than the IPAPA and about 21% less complex than MIPAPA in terms of multiplications.

5. SIMULATION RESULTS

Simulations were performed in the context of echo cancellation. The network echo path from Fig. 3a [18] is used in most of the experiments; for the last experiment, the acoustic echo path from Fig. 3b is used. The length of the adaptive filter is $L = 512$. The input signal is a speech signal. The output of the echo path is corrupted by an independent white Gaussian noise with different values of the signal-to-noise ratio (SNR). The measure of performance is the normalized misalignment (in dB); it is defined as $20\log_{10}(\|\mathbf{h} - \hat{\mathbf{h}}(n)\|_2/\|\mathbf{h}\|_2)$, where $\mathbf{h}$ is the true impulse response of the echo path and $\|\cdot\|_2$ denotes the l_2 norm. In the undermodelling scenario (Fig. 7), the expression of the normalized misalignment is evaluated by padding the vector of the adaptive filter coefficients (256 coefficients) with 256 zeros. Also, the abrupt change of the echo path is introduced at time 0.5 by shifting the impulse response to the right by 12 samples.

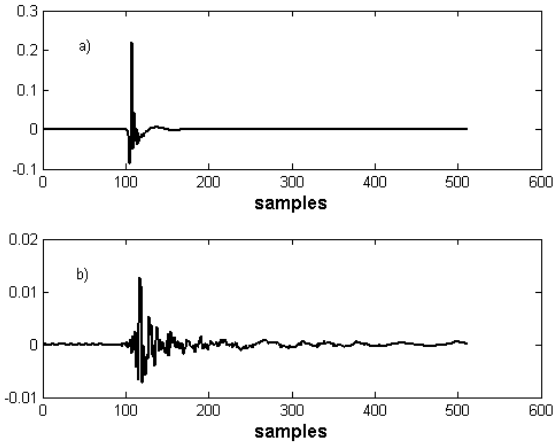

Fig. 3. Echo paths used in simulations. a) Sparse network echo path impulse response (used in Figs. 5 and 6); b) Acoustic echo path (used in the experiment from Figs. 7 and 8).

All the algorithms use the same values for their parameters, i.e., the step-size is $\mu = 0.2$, the regularization constant is $\delta = 50\sigma_x^2/2L$ (where σ_x^2 is the input signal variance), and $\alpha = 0$. A speech sequence was used as input, and $p = 8$. It was shown in [15] that both IPAPA and MIPAPA outperform the classical APA in the context of network echo cancellation; therefore, the APA is not included for comparison. Also, since the numerous FAP algorithms are approximations of the classical APA, their performance was not investigated as well.

In Fig. 4, SNR = 30 dB and echo path changes. Fig. 4 shows the better performance of both AMIPAPA and MIPAPA, as compared to the IPAPA. The same conclusions can be obtained in the case of variable background noise (at the near-end). In Fig. 5, the SNR decreases from 30 dB to 10 dB between times 0.25 and 0.5.

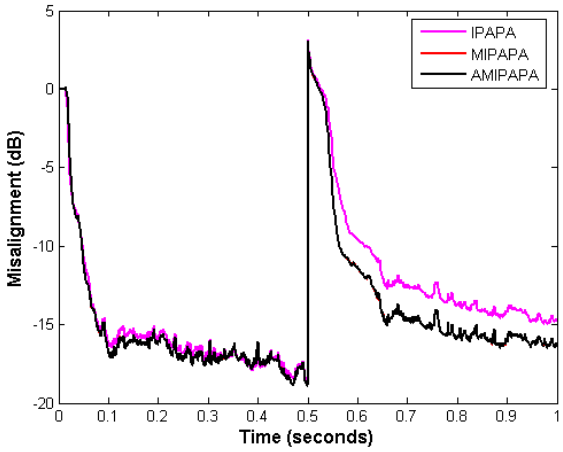

Fig. 4. Misalignment of the IPAPA, MIPAPA, and AMIPAPA. The input signal is a speech sequence, $p = 8$, $L = 512$, SNR = 30 dB, and echo path (from Fig. 3a) changes at time 0.5.

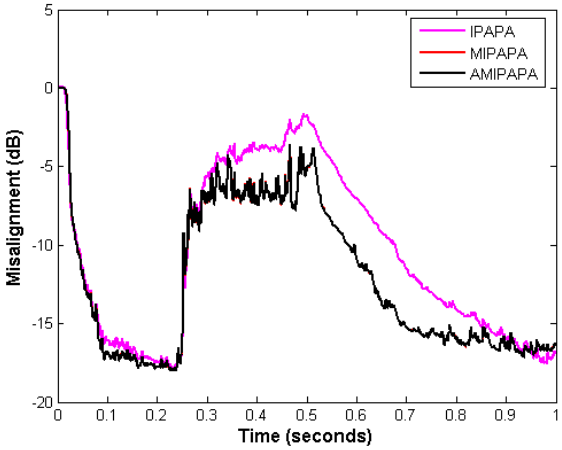

Fig. 5. Misalignment of the IPAPA, MIPAPA, and AMIPAPA. The input signal is a speech sequence, $p = 8$, $L = 512$, echo path (from Fig. 3a), and variable background noise (SNR decreases from 30 dB to 10 dB between times 0.25 and 0.5, otherwise is 30 dB).

Next, the performance of the algorithms is evaluated using the acoustic echo path from Fig. 3b and SNR = 30 dB. It can be seen from Fig. 6 that both AMIPAPA and MIPAPA achieve faster tracking and lower misalignment than the IPAPA. However, the misalignment difference between the IPAPA and MIPAPA/AMIPAPA is higher than that obtained in Fig. 4, using the echo path from Fig. 3a.

Fig. 7 considers the under-modelling case ($L = 256$ coefficients are considered to model the echo path from Fig. 3b) and the acoustic path changes at time 0.5. It can be seen from Fig. 7 that the algorithms are robust, and similar conclusions as above can be drawn (regarding the convergence and tracking abilities). As expected, the misalignment performance is worse than that of the exact modelling case.

Fig. 8 confirms that the performance of MIPAPA and AMIPAPA is virtually the same in case of simulations from Figs. 4-7. Because of the good approximation proved in Fig. 1, the misalignment difference between AMIPAPA and MIPAPA is limited in absolute value to 0.15 dB.

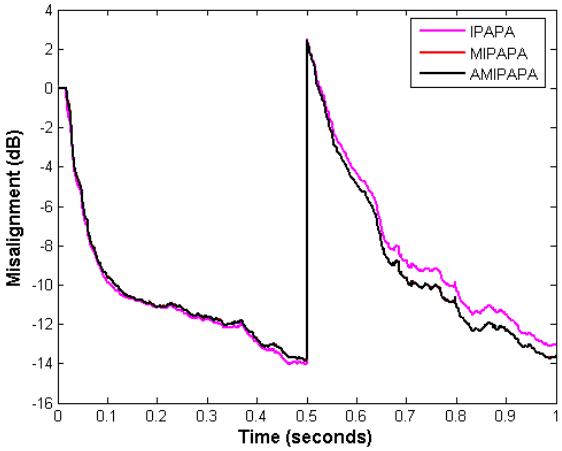

Fig. 6. Misalignment of the IPAPA, MIPAPA, and AMIPAPA. The input signal is a speech sequence, $p = 8$, $L = 512$, SNR = 30 dB, echo path (from Fig 3b) changes at time 0.5.

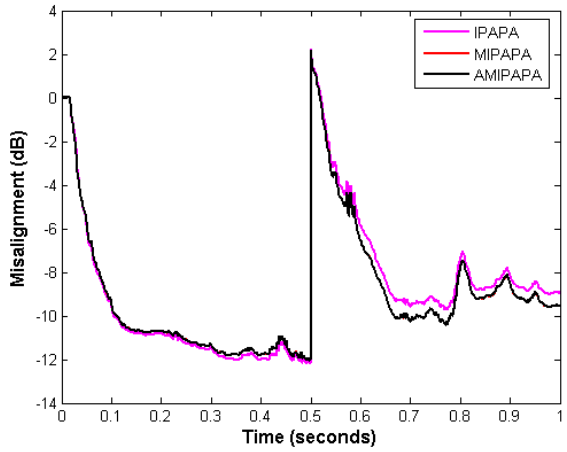

Fig. 7. Misalignment of the IPAPA, MIPAPA, and AMIPAPA. The input signal is a speech sequence, p = 8, L = 512, SNR = 30 dB, echo path (from Fig 3b) changes at time 0.5, under-modelling scenario.

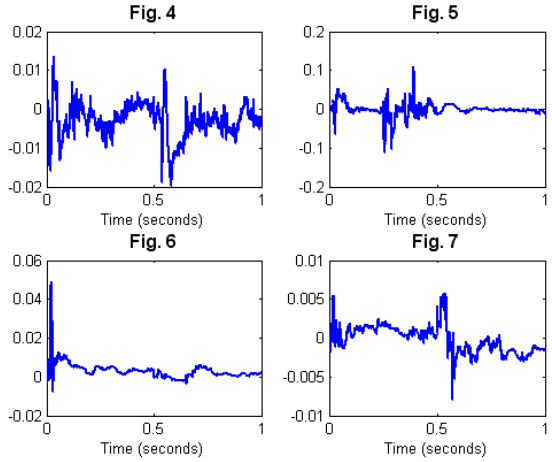

Fig. 8. Misalignment difference between MIPAPA, and AMIPAPA for Figs. 4-7.

6. CONCLUSIONS

In this paper, a low complexity proportionate-type APA was proposed. An approximation was performed within the MIPAPA in order to update in a computationally efficient manner the matrix to be inverted. The proposed AMIPAPA offers good performance *versus* complexity ratios as compared to the original IPAPA and MIPAPA. Therefore, longer filter lengths and higher projection orders can be used by AMIPAPA for a similar complexity as compared to the other algorithms, but with improved convergence performance.

Acknowledgment: This work was supported by the UEFISCSU under Grants PN-II "Idei" no. 331 / 01.10.2007 and no. 65 / 01.10.2007. The authors also thank to the anonymous reviewers for their valuable comments.

REFERENCES

[1] J. Benesty, T. Gaensler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, Advances in Network and Acoustic Echo Cancellation. Berlin, Germany: Springer-Verlag, 2001.
[2] E. Haensler and G. Schmidt, Eds., Topics in Acoustic Echo and Noise Control. Berlin, Germany: Springer-Verlag, 2006.
[3] D. L. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancellers," IEEE Trans. Speech Audio Process., vol. 8, no. 5, pp. 508–518, Sept. 2000.
[4] H. Deng and M. Doroslovački, "Proportionate adaptive algorithms for network echo cancellation," IEEE Trans. Signal Process., vol. 54, no. 5, pp. 1794–1803, May 2006.
[5] P. Loganathan, A. W. H. Khong, and P. A. Naylor, "A class of sparseness-controlled algorithms for echo cancellation," IEEE Trans. Audio, Speech, Audio Process., vol. 17, no. 8., pp. 1591–1601, Nov. 2009.
[6] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties," Electron. Commun. Jpn., vol. 67-A, no. 5, pp. 19–27, May 1984.
[7] S. L. Gay, "A fast converging, low complexity adaptive filtering algorithm," Proc. IWAEC, 1993, pp. 223–226.
[8] F. Albu, J. Kadlec, N. Coleman, and A. Fagan, "The Gauss-Seidel fast affine projection algorithm," in Proc. IEEE SIPS, 2002, pp. 109–114.
[9] H. Ding, "Fast affine projection adaptation algorithms with stable and robust symmetric linear system solvers," IEEE Trans. Signal Process., vol. 55, no. 5, pp. 1730–1740, May 2007.
[10] 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.
[11] Y. V. Zakharov, "Low complexity implementation of the affine projection algorithm," IEEE Signal Processing Letters, vol. 15, pp. 557–560, 2008.
[12] T. Gänsler, S. L. Gay, M. M. Sondhi, and J. Benesty, "Double-talk robust fast converging algorithms for network echo cancellation," IEEE Trans. Speech Audio Process., vol. 8, no. 6, pp. 656–663, Nov. 2000.
[13] S. Werner, J. A. Apolinário Jr., and P. S. R. Diniz, "Set-membership proportionate affine projection algorithms," EURASIP J. Audio, Speech, Music Process., vol. 2007, no. 1, pp. 1–10, 2007.
[14] O. Hoshuyama, R. A. Goubran, and A. Sugiyama, "A generalized proportionate variable step-size algorithm for fast changing acoustic environments," in Proc. IEEE ICASSP, 2004, pp. IV-161–IV-164.
[15] C. Paleologu, S. Ciochină, and J. Benesty, "An efficient proportionate affine projection algorithm for echo cancellation," IEEE Signal Processing Letters, vol. 17, no. 2, pp. 165–168, Feb. 2010.
[16] J. Benesty and S. L. Gay, "An improved PNLMS algorithm," in Proc. IEEE ICASSP, 2002, pp. II-1881–II-1884.
[17] G. H. Golub and C. F. Van Loan, Matrix computation, 3rd edition. Baltimore, MD: The John Hopkins Univ. Press, 1996.
[18] Digital Network Echo Cancellers, ITU-T Rec. G.168, 2002