SIMPLIFIED PROPORTIONATE AFFINE PROJECTION ALGORITHMS

Felix Albu

Faculty of Electrical Engineering, Valahia University of Targoviste, Romania

ABSTRACT

In this paper, new efficient proportionate affine projection algorithms are proposed. They use a simplified way of computing the output error vector. It is shown that the simplified approximated memory improved proportionate affine projection algorithm (SAMIPAPA) offers the best compromise between complexity and performance if compared with other proportionate-type APAs. Also, it is shown that, if the logarithmic proportionate scheme is integrated in SAMIPAPA, the convergence speed and tracking abilities are worse.

Index Terms—acoustic echo cancellation, proportionate affine projection algorithm, logarithmic proportionate updating scheme

1. INTRODUCTION

Many 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. In the context of echo cancellation, the normalized least mean square (NLMS) algorithm, the affine projection algorithm (APA) [3] and its fast versions, the fast affine projection (FAP) algorithms (e.g., [4]–[7]), were found to be very attractive choices. They proved useful for other applications such as hearing aids [6] and active noise control [7]. Nevertheless, the echo paths (for both network and acoustic echo cancellation scenarios) 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 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. One of the first proportionate-type algorithms was proposed by Duttweiler in [8]. It was called the proportionate normalized least-mean-square (PNLMS) algorithm and numerous such algorithms have been proposed afterwards [9], [10]. Also, several proportionate-type APAs were developed (e.g., improved PAPA (IPAPA) [11], memory IPAPA (MIPAPA) [12], μ-law MIPAPA (MMIPAPA) [13]). In [14] a very efficient implementation of MIPAPA taking into account the time-shift character and symmetrical structure, termed approximated MIPAPA (AMIPAPA) was proposed.

The paper is organized as follows. Section 2 represents an overview of the proportionate-type algorithms for echo cancellation. In Section 3, a simplified approximated MIPAPA (SAMIPAPA) is derived. One of the contributions of this paper is that a further approximation is proposed for AMIPAPA in order to reduce the output error computation. Also, the logarithmic proportionate updating scheme used in [4] and [13] is integrated in SAMIPAPA, leading to μ-law SAMIPAPA, called MSAMIPAPA. The numerical complexity of these algorithms is investigated in Section 4. The simulation results presented in Section 5 compare the proposed algorithms with AMIPAPA 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, 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 of the adaptive filter, \( d(n) \), contains the output of the echo path and the near-end signal. The adaptive FIR filter is defined by the real-valued coefficients vector \( \mathbf{\beta}(n) = [\beta_1(n), \beta_2(n), \ldots, \beta_L(n)]^T \), where \( L \) is the length of the adaptive filter and superscript \( T \) denotes transposition. The error signal is defined as

\[
e(n) = d(n) - \mathbf{h}^T(n-1)x(n)
\] (1)

where \( x(n) = [x(n), x(n-1), \ldots, x(n-L+1)]^T \) is a real-valued vector containing the \( L \) most recent samples of the input signal. The error signal vector is defined as

\[
e(n) = \mathbf{d}(n) - \mathbf{X}^T(n)\mathbf{h}(n)\]

(2)

where \( \mathbf{d}(n) = [d(n), d(n-1), \ldots, d(n-p+1)]^T \) is the reference signal vector of length \( p \), with \( p \) denoting the projection order, \( \mathbf{X}(n) = [x(n), x(n-1), \ldots, x(n-p+1)] \) is the input signal matrix, and

\[
e(n) = [e(n), e(n-1), \ldots, e(n-p+1)]^T.
\]

The proportionate-type affine projection algorithms (PAPA) [11] update their coefficients according to

This work was supported by the UEFISCDI grant PN-II-ID-PCE-2011-3-0097
\[
\hat{h}(n) = \hat{h}(n-1) + \mu G(n-1)X(n)
\]
\[
X\left[ \delta I_p + \hat{X}^T(n)G(n-1)X(n) \right]^{-1} e(n).
\]
where \(\mu\) is the normalized step-size parameter, \(G(n-1)\) is the regularization constant, and \(G(n-1)\) is an \(L \times L\) diagonal matrix which assigns an individual step-size to each filter coefficient, is the error vector and \(I\) is the \(p \times p\) identity matrix. In the case of the improved PAPA (IPAPA) [11], the diagonal elements of \(G(n-1)\), denoted in the following by \(g(n-1)\), with \(0 \cdot \ast \cdot L - 1\), are evaluated as
\[
g_l(n-1) = \frac{1 - \alpha}{2L} + \frac{1 + \alpha}{2} \frac{1}{\sum_{i=0}^{L-1} [\hat{h}_i(n-1)]^2} \xi
\]
where \(-1 \cdot \ast < 1\) and the small positive constant avoids division by zero. Good choices for the parameter \(\alpha\) are 0 or \(-0.5\). Let us denote \([12]\)
\[
P(n) = G(n-1)X(n)
\]
\[
= \left[ g(n-1) \odot x(n) \ldots g(n-1) \odot x(n-p+1) \right],
\]
where \(g(n-1)\) is a vector containing the diagonal elements of \(G(n-1)\); the operator \(\odot\) denotes the Hadamard product, i.e., \(a \odot b = [a(1)b(1), a(2)b(2), \ldots, a(L)b(L)]^T\), where \(a\) and \(b\) are two vectors of length \(L\). The other equations of IPAPA are
\[
S(n) = \delta I_p + \hat{X}^T(n)P(n)
\]
\[
\hat{h}(n) = \hat{h}(n-1) + \mu P(n)S^{-1}(n)e(n)
\]
Following a similar idea of that used to derive the pseudo affine projection algorithm [7], \(P(n)\) is approximated with
\[
P'(n) = \left[ g(n-1) \odot x(n) \ldots g(n-p) \odot x(n-p+1) \right],
\]
where \(g(n-k)\) are the vectors containing the diagonal elements of the matrices \(G(n-k)\), with \(k = 1, 2, \ldots, p\) [12]. The computational complexity is lower as compared to (5), because (8) can be written as
\[
P'(n) = \left[ g(n-1) \odot x(n) \quad P'_{-1}(n-1) \right]
\]
where the matrix
\[
P'_{-1}(n-1) = \left[ g(n-2) \odot x(n-1) \ldots g(n-p) \odot x(n-p+1) \right]
\]
contains the first \(p - 1\) columns of \(P'(n-1)\). The MIPAPA equations are [12]:
\[
S_1(n) = \delta I_p + \hat{X}^T(n)P'(n)
\]
\[
\hat{h}(n) = \hat{h}(n-1) + \mu P'(n)S_1^{-1}(n)e(n)
\]
An efficient implementation of MIPAPA was proposed in [13]. Only the first row and column of \(S_1(n)\) are computed, and the bottom-right \((p-1) \times (p-1)\) submatrix of \(S_1(n)\) is replaced with the top-left \((p-1) \times (p-1)\) submatrix of \(S_1(n-1)\) [13]. The first column is given by
\[
X^T(n) \left[ g(n-1) \odot x(n) \right],
\]
while the first row is computed as \(x^T(n)P'(n)\). 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 [13]. This new matrix, \(S_2(n)\), is obtained by updating both its first row and its first column with
\[
X^T(n) \left[ g(n-1) \odot x(n) \right] = X^T(n)P'_{-1}(n) + \delta
\]
where \(P'_{-1}(n)\) denotes the first column of \(P'(n)\). The bottom-right \((p-1) \times (p-1)\) submatrix of \(S_2(n)\) is replaced with the top-left \((p-1) \times (p-1)\) submatrix of \(S_2(n-1)\). The approximated MIPAPA (AMIPAPA) filter coefficients are given by
\[
\hat{h}(n) = \hat{h}(n-1) + \mu P'(n)S_2^{-1}(n)e(n)
\]
\[
3. LOW COMPLEXITY PROPORIONATE-TYPE AP ALGORITHMS
\]
The proposed algorithms are obtained from AMIPAPA. Firstly important numerical complexity reduction is obtained if \(e(n) = d(n) - x^T(n)\hat{h}(n-1)\) is approximated as in the original fast affine projection algorithm [7]
\[
e(n) = \left[ e(n); (1 - \mu) \overline{e}(n-1) \right]
\]
where \(\overline{e}(n-1)\) represents the first \(p-1\) elements of \(e(n-1)\).
The new algorithm using (14) instead of (2) is called simplified AMIPAPA (SAMIPAPA).

The logarithmic proportionate updating scheme [4] can be easily incorporated in SAMIPAPA by modifying the proportionate coefficients of (4) as follows:
\[
g_l(n-1) = \frac{1 - \alpha}{2L} + \frac{1 + \alpha}{2} \frac{F(\hat{h}_i(n-1))}{\sum_{i=0}^{L-1} F(\hat{h}_i(n-1)) + \xi}
\]
where
\[
F(\hat{h}_i(n-1)) = \ln(1 + \mu_{\log} \hat{h}_i(n-1))
\]
Typically, a value of \(\mu_{\log} = 1000\) is used. The new algorithm is called the \(\mu\)-law SAMIPAPA (MSAMIPAPA).
4. NUMERICAL COMPLEXITY COMPARISON

The numerical complexity of the proposed algorithms in terms of multiplications is the following [14]:

\[ C_{\text{MIPAPA}} = L(4p+1) + p + P_m, \]
\[ C_{\text{AMIPAPA}} = L(3p+2) + p + P_m, \]
\[ C_{\text{SAMIPAPA/MSAMIPAPA}} = L(2p+3) + 2p + P_m. \]

The notation \( P_m = O(p^3) \) indicates the numerical complexity in terms of multiplications associated with solving the implicit linear systems of equations using the \( \text{LDL}^T \) method [15]. Fig. 1 shows the numerical complexity comparison of the investigated algorithms as a function of \( L \). It can be noticed that the number of multiplications varies linearly with the filter length for all the considered algorithms. MSAMIPAPA requires additional \( L \) logarithmic functions and \( L \) additions per iteration in comparison with SAMIPAPA. It can be easily seen that SAMIPAPA/MSAMIPAPA are the least complex algorithms in terms of multiplications. The numerical complexity savings are important. For example, in case of \( L = 512, p = 8 \), the MIPAPA requires 17044 multiplications, SAIPAPA needs 13460 multiplications, while SAMIPAPA/MSAMIPAPA needs only 9884; therefore SAMIPAPA/MSAMIPAPA are about 25% less complex than AMIPAPA and about 40% less complex than MIPAPA in terms of multiplications.

Fig. 1. Numerical complexity of the considered algorithms in terms of multiplications for \( p = 8 \) for a variable \( L \).

5. SIMULATION RESULTS

Simulations were performed in the context of network echo cancellation. The first impulse response from ITU-T G168 Recommendation [16] is padded with zeros in order to have 512 coefficients. 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. The adaptive filter has 512 taps. The input signal is either a white Gaussian noise or a speech signal. The output of the echo path is corrupted by an independent white Gaussian noise with a signal-to-noise ratio \( \text{SNR} = 30 \, \text{dB} \). The performance measure is the normalized misalignment (in dB), defined as \( 20 \log_{10}(||h - \hat{h}(n)||^2/||h||^2) \), where \( h \) denotes the true impulse response of the echo path. In the simulations with white noise, the performance curves are averaged over 10 independent trials. It was shown in [13] that MIPAPA and AMIPAPA have virtually the same performance. Therefore only AMIPAPA curves are shown and compared with those of the proposed algorithms. For the simulations with white noise as input, the step-sizes are chosen in order to have the same steady-state misalignment error for all the algorithms (\( \mu_{\text{AMIPAPA}} = \mu_{\text{SAMIPAPA}} = 0.2 \), \( \mu_{\text{MSAMIPAPA}} = 0.35 \)). The regularization constant is \( \delta = 0.01 \), \( p = 8 \) and \( \alpha = 0 \).

Fig. 2. Misalignment of the AMIPAPA, SAMIPAPA and MSAMIPAPA. The input signal is white signal, \( p = 8, L = 512, \text{SNR} = 30 \, \text{dB}, \) echo path changes after 10000 samples.

It can be seen from Fig. 2 that, for white signals, AMIPAPA and SAMIPAPA have the same performance. The MSAMIPAPA has the worst convergence speed and tracking abilities among the considered algorithms.

For the next simulation, a speech sequence was used as input, \( \text{SNR} = 30 \, \text{dB}, \) echo path changes after 0.5 seconds and \( p = 8 \). The same parameters as above are used. Fig. 3 shows the better performance of both SAMIPAPA and AMIPAPA, as compared to the MSAMIPAPA. Fig. 3 shows that, in case of a speech signal input, the approximation used by SAMIPAPA lead to a slightly reduced performance (around 1 dB) in comparison with AMIPAPA. However, SAMIPAPA offers a better performance/complexity ratio than AMIPAPA, due to its reduced numerical complexity by about 25% (as shown in Section 4). Fig. 3 also confirms that MSAMIPAPA has the worst performance among the considered algorithms. Although the use of the logarithmic proportionate scheme in [14] led to an improved performance of MMIPAPA over MIPAPA, its use in MSAMIPAPA...
reduce the performances if compared with SAMIPAPA. Therefore, the approximation (14) of (2) annihilates the supplementary proportionate effect of the logarithmic functions observed in [4] and [14].

Similar results were obtained when using a colored noise as an input signal, for different filter lengths, and projection orders.

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

6. CONCLUSION

In this paper, two low complexity proportionate-type AP algorithms are proposed. If compared with AMIPAPA, SAMIPAPA offers slightly smaller performances, but with about 25% less numerical complexity. Future work will be focused in designing variable step size and variable projection order versions of SAMIPAPA. These versions are expected to have a better behaviour in case of near-end adverse conditions such as variable background noise or double-talk.

7. REFERENCES


