NEW INSIGHTS INTO THE WEIGHT BEHAVIOUR OF THE AFFINE PROJECTION ALGORITHM

Márcio H. Costa[1], *Sérgio J. M. de Almeida*[2], *José C. M. Bermudez*[1] *and Rodrigo B. Barcelos*[1]

[1]Departamento de Engenharia Elétrica, Universidade Federal de Santa Catarina
Centro Tecnológico, Cidade Universitária, 88040-900, Florianópolis-SC, Brazil
[2]Centro Politécnico, Universidade Católica de Pelotas
96010-000, Pelotas-RS, Brazil
E-mail: costa@eel.ufsc.br; smelo@ucpel.tche.br; j.bermudez@ieee.org; borgesrb@gmail.com

ABSTRACT

This paper presents a new analytical model for the mean weight behaviour of the Affine Projection adaptive algorithm for autoregressive inputs and unity step-size. Deterministic recursive equations are derived which consider an initial transient phase not accounted for in previous analyses. This phase occurs at the very beginning of the adaptation process, and is due to the arbitrary initialization of the coefficients. It is a deterministic process that prevents the correct orthogonalization of the coefficient vector in relation to the subspace defined by the past input vectors. Monte Carlo simulations show improvements in the accuracy of the presented model when compared with a previously developed one.

1. INTRODUCTION

Adaptive filters are widely used in many important real-time applications such as echo cancellation, active noise control and hearing-aids. Although popular for its low computational cost, the LMS family suffers from slow convergence for highly correlated input signals. The Affine Projection (AP) algorithm proposed by Ozeki and Umeda in 1984 [1] applies weight vector updates that are orthogonal to the last *P* input vectors. This strategy tends to reduce the input signal correlation in time, speeding up convergence [2] and making the algorithm attractive for applications with highly correlated input signals. The improved transient performance comes at the cost of a larger computational complexity and a higher noise floor, as compared to the normalized LMS algorithm (NLMS). As the impact of the extra computational cost consistently decreases with advances in the semiconductor industry, the AP algorithm and its fast versions [3]-[4] become more employed in practical systems.

Although much has already been studied about the AP behaviour, its complete understanding still represents a challenge. This is because the embedded underdetermined least squares estimation process significantly complicates any statistical analysis.

A statistical analysis of the AP algorithm was presented in [5] for autoregressive (AR) inputs and unity step size (fastest convergence). Deterministic recursive equations were derived for modelling the mean and mean-square behaviour of the AP for a large number of adaptive taps. The examples in [5] showed good predictions of the AP behaviour as compared to Monte Carlo simulations for all analysed cases. However, subsequent extensive use of the theory developed in [5] has revealed model inaccuracies at the initial algorithm iterations for certain parameter settings. A typical example is the processing of high order AR input signals combined with a weight vector initialization very far from the optimum. These initial errors were then propagated in time due to the recursive nature of the adaptive learning process.

This work presents an extension to the analytical model in [5] for the mean weight behaviour of the AP algorithm. Improved model accuracy is obtained through the addition of a new term to the equations in [5]. This new term estimates the effect of the transient error caused by an arbitrary initialization of the weight vector on the orthogonal projection of the weight update onto the subspace defined by the previous *P* input vectors. The extended model accurately predicts the algorithm mean behaviour even in the situation found problematic for the model in [5]. The contributions of this work can be used to improve not only existing theoretical models, but also robustness and stability analyses such as those in [6].

The paper is organized as follows. Section 2 introduces the input signal model and the notation used. Section 3 reviews the AP algorithm. Section 4 briefly presents the theoretical model derived in [5]. Section 5 derives the new analytical model. Section 6 presents Monte Carlo simulations to validate the developed model. Finally, Section 7 presents the main conclusions of this work. In this paper scalars are denoted by plain lowercase or uppercase letters, vectors are denoted by lowercase boldface letters and matrices by uppercase boldface letters. The letter *n* denotes discrete time.

2. THE INPUT SIGNAL MODEL

In this work the input signal $u(n)$ is assumed to be a zero-mean wide-sense stationary AR process of order *P*. It can be described by

$$u(n) = \sum_{i=1}^{P} a_i u(n-i) + z(n) \tag{1}$$

where a_i are the AR coefficients and $z(n)$ is a wide-sense stationary white noise process with variance σ_z^2. AR processes are mathematically tractable and can be used to model input signals for many practical applications. A set of N consecutive samples of (1) can be described by the following matrix notation

$$\mathbf{u}(n)=\mathbf{U}(n)\mathbf{a}+\mathbf{z}(n) \tag{2}$$

where $\mathbf{u}(n) = [u(n)\ u(n\text{-}1)\ \ldots\ u(n\text{-}N\text{+}1)]^T$ is the input regressor with autocorrelation matrix $\mathbf{R_u} = E\{\mathbf{u}(n)\mathbf{u}^T(n)\}$, $\mathbf{a} = [a_1\ a_2\ \ldots\ a_P]^T$, $\mathbf{z}(n) = [z(n)\ z(n\text{-}1)\ \ldots\ z(n\text{-}N\text{+}1)]^T$ and $\mathbf{U}(n) = [\mathbf{u}(n\text{-}1)\ \mathbf{u}(n\text{-}2)\ \ldots\ \mathbf{u}(n\text{-}P)]^T$. The adaptive system attempts to estimate a desired signal $d(n)$ that is modelled by

$$d(n) = \mathbf{w}^{\mathbf{o}T}\mathbf{u}(n) + r(n) \tag{3}$$

where the N-length vector $\mathbf{w^o} = [w^o_0\ w^o_1\ w^o_2 \ldots w^o_{N\text{-}1}]^T$ models the impulse response of the unknown system (plant) and $r(n)$ is an independent, identically distributed, zero-mean noise with variance σ_r^2.

3. THE AP ADAPTIVE ALGORITHM

The AP algorithm can be formulated as the solution of an underdetermined least squares problem subject to multiple constraints [2]. The optimization problem can be stated as the minimization of the Euclidian norm of $\Delta\mathbf{w}(n) = \mathbf{w}(n\text{+}1)–\mathbf{w}(n)$ (minimum disturbance principle), where $\mathbf{w}(n) = [w_0(n)\ w_1(n\text{-}1)\ \ldots\ w_{N\text{-}1}(n)]^T$ is the adaptive weight vector, subjected to the set of constraints given by

$$\begin{aligned}\mathbf{u}^T(n)\mathbf{w}(n+1) &= d(n)\\ \mathbf{u}^T(n-1)\mathbf{w}(n+1) &= d(n-1)\\ &\vdots\\ \mathbf{u}^T(n-P)\mathbf{w}(n+1) &= d(n-P).\end{aligned} \tag{4}$$

The solution of this optimization problem using the method of *Lagrange multipliers* leads to the AP weight update equation [1]

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\mu\mathbf{U}_\mathbf{u}(n)\left[\mathbf{U}_\mathbf{u}^T(n)\mathbf{U}_\mathbf{u}(n)\right]^{-1}\mathbf{e}_\mathbf{e}(n) \tag{5}$$

where $\mathbf{U_u}(n) = [\mathbf{u}(n)\ \mathbf{U}(n)]$, and $\mathbf{e_e}(n) = [e_e(n)\ e_e(n\text{-}1)\ \ldots\ e_e(n\text{-}P)]^T$ is the error vector with entries $e_e(n\text{-}k) = d(n\text{-}k)\text{-}\mathbf{u}^T(n\text{-}k)\mathbf{w}(n)$ and $e_e(n\text{-}k) = e(n\text{-}k)$, the instantaneous error signal. The step-size μ is equal to one in the solution of the optimization problem and is introduced in (5) only to permit some control over the algorithm convergence.

For $u(n)$ an AR process as in (1) and for μ =1 (maximum convergence speed), it was shown in [2] that the weight vector updates occur in the direction of a vector $\boldsymbol{\phi}(n)$ given by

$$\boldsymbol{\phi}(n)=\mathbf{u}(n)-\mathbf{U}(n)\hat{\mathbf{a}}(n) \tag{6}$$

where $\hat{\mathbf{a}}(n)$ is the least squares (LS) estimate of the AR coefficient vector $\mathbf{a}$, and is given by

$$\hat{\mathbf{a}}(n)=\left[\mathbf{U}^T(n)\mathbf{U}(n)\right]^{-1}\mathbf{U}^T(n)\mathbf{u}(n). \tag{7}$$

As a result the weight update equation can be written as

$$\mathbf{w}(n+1)=\mathbf{w}(n)+\frac{\boldsymbol{\phi}(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}e(n) \tag{8}$$

Defining the weight error vector $\mathbf{v}(n) = \mathbf{w}(n)\text{-}\mathbf{w^o}$, the scalar error signal $e(n)$ is given by

$$e(n)=r(n)-\mathbf{v}^T(n)\mathbf{u}(n) \tag{9}$$

The AP algorithm order is the number P+1 of input vectors used to represent $\boldsymbol{\phi}(n)$.

4. PREVIOUS ANALYSISIS

The works in [2] and [5] provided important insights into the AP behaviour. Some of these results are briefly revisited here, as they constitute the basis of this work.

4.1 Weight Behaviour

Equation (12) of [2] introduced the following set of deterministic relations for the AP:

$$\begin{aligned}\boldsymbol{\phi}^T(n)\mathbf{v}(n+1)&=\boldsymbol{\phi}^T(n)\mathbf{v}(n)-\mathbf{u}^T(n)\mathbf{v}(n)+r(n)\\ \mathbf{u}^T(n)\mathbf{v}(n+1)&=r(n)\\ \mathbf{U}^T(n)\mathbf{v}(n+1)&=\mathbf{U}^T(n)\mathbf{v}(n)\end{aligned} \tag{10}$$

Equations (10) result in

$$\mathbf{U}^T(n)\mathbf{v}(n+1)=\mathbf{r}(n-1) \tag{11}$$

where $\mathbf{r}(n\text{-}1) = [r(n\text{-}1)\ r(n\text{-}2)\ \ldots\ r(n\text{-}P)]^T$. Using then equations (6), (9), and (10) in (8) resulted in

$$\mathbf{v}(n+1)=\mathbf{v}(n)-\frac{\boldsymbol{\phi}(n)\boldsymbol{\phi}^T(n)\mathbf{v}(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}+\frac{\boldsymbol{\phi}(n)r_a(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)} \tag{12}$$

where $r_a(n)=r(n)-\sum_{i=1}^{P}\hat{a}_i(n)r(n-i)$ is the filtered noise sequence [2].

4.2 Mean Weight Behaviour

Equation (12) was the starting point of the mean weight analysis provided in [5] where the following statistical assumptions were used:

- A1: The number of adaptive filter weights is large enough so that $N >> P$.
- A2: The statistical dependence between $\mathbf{z}(n)$ and $\mathbf{U}(n)$ can be neglected for $N >> P$.
- A3: Vectors $\boldsymbol{\phi}(n)$ and $\mathbf{w}(n)$ are statistically independent and $\boldsymbol{\phi}(n)$ is orthogonal to the columns of $\mathbf{U}(n)$.
- A4: $\boldsymbol{\phi}(n)$ is a zero-mean Gaussian random vector.

Using A3 and noting that $E\{\boldsymbol{\phi}(n)r_a(n)\} = \mathbf{0}$ due to the zero mean of $r(n)$, the expected value of (12) was found in [5] to be given by

$$E\{\mathbf{v}(n+1)\}=E\{\mathbf{v}(n)\}-E\left\{\frac{\boldsymbol{\phi}(n)\boldsymbol{\phi}^T(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}\right\}E\{\mathbf{v}(n)\} \tag{13}$$

Moreover, using the following results from [5]

$$E\left\{\left[\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)\right]^{-1}\right\}=\frac{N}{G(G-2)\sigma_z^2} \tag{14}$$

$$E\{\boldsymbol{\phi}(n)\boldsymbol{\phi}^T(n)\}=(G/N)\sigma_z^2\mathbf{I}. \tag{15}$$

where $G = N–P$, (13) leads to

$$E\{\mathbf{v}(n+1)\}=\left(1-\frac{1}{G-2}\right)E\{\mathbf{v}(n)\}. \tag{16}$$

Equation (16) was the deterministic recursion derived in [5] for predicting the mean weight error behaviour.

5. NEW WEIGHT BEHAVIOUR ANALYSIS

This section presents a new analytical model for the mean weight behaviour of the AP adaptive algorithm in order to improve the results presented by [2] and [5].

5.1 Initialization error

In [5], equations (10) and (11) presented here were assumed to hold for all stages of the adaptation process, resulting in accurate predictions for the AP mean and mean square behaviour in all cases studied. However, equation (11) is not always valid. There is a transient phase during the initial algorithm iterations that is a function of the distance between the initial weight vector and the optimum solution. This can be easily verified by noting that $\mathbf{U}^T(n)\mathbf{v}(n+1)|_{n=-1} = \mathbf{U}^T(-1)[\mathbf{w}(0)-\mathbf{w}^o]$ (before the first weight update) is not necessarily equal to $\mathbf{r}(n-2)$, where $\mathbf{w}(0)$ is the initial weight vector. Thus, an initialization not satisfying equation (11) will result in an error in the expected orthogonalization of the weight vector with respect to the subspace defined by the past input vectors. As the adaptation process evolves, the r.h.s. of equation (11) is successively (term by term) satisfied. However, the influence of an initial error will still persist during all the transient phase due to the infinite memory of equation (8).

To avoid the transient error, the initialization vector $\mathbf{w}(0)$ should satisfy the following equation

$$\begin{bmatrix} u(n-1) & \dots & u(n-N) \\ u(n-2) & \dots & u(n-N-1) \\ \dots & \dots & \dots \\ u(n-P) & \dots & u(n-N-P+1) \end{bmatrix} \begin{bmatrix} w_0(0)-w_0^o \\ w_1(0)-w_1^o \\ \dots \\ w_{N-1}(0)-w_{N-1}^o \end{bmatrix} = \begin{bmatrix} r(n-1) \\ r(n-2) \\ \dots \\ r(n-P) \end{bmatrix} \quad (17)$$

Since the optimum solution is not known, an arbitrary $\mathbf{w}(0)$ will not satisfy (17) in general. Hence, the transient algorithm behaviour may be significantly different from that predicted by the models revisited in Section 4. The model accuracy will depend on the magnitude of the error in estimating the delayed noise vector $\mathbf{r}(n-1)$ and on the memory of the adaptive filter. We have verified that this modelling inaccuracy has more significant impact when three conditions are simultaneously presented: high AP order, large number of coefficients and initialization far from the optimum solution.

5.2 Proposed solution

The proposed solution to the initialization problem consists in adding a correction factor to equation (11) that incorporates the effect of the incorrect coefficient vector orthogonalization at the beginning of the adaptation process. Equation (11) is modified to

$$\mathbf{U}^T(n)\mathbf{v}(n+1) = \mathbf{r}(n-1) + \mathbf{q}(n) \quad (18)$$

where $\mathbf{q}(n) = [q_1(n) \ q_2(n) \ \dots \ q_P(n)]^T$. The entries of $\mathbf{q}(n)$ should disappear sequentially at each algorithm iteration. One possible way to obtain this effect is to define

$$q_i(n) = \left\{\mathbf{u}^T(n-i)\mathbf{v}(0) - r(n-i)\right\} u_{-1}(i-n) \quad (19)$$

where $u_{-1}(k)$ is the step function (1 if $k \geq 0$ and 0 for others values of k). As a result, $\mathbf{q}(n)$ can be represented in a vector form as

$$\mathbf{q}(n) = \mathbf{Q}(-n+1)\left[\mathbf{U}^T(n)\mathbf{v}(0) - \mathbf{r}(n-1)\right] \quad (20)$$

where the $P \times P$ matrix $\mathbf{Q}(-n+1)$ is

$$\mathbf{Q}(-n+1) = \begin{bmatrix} u_{-1}(-n+1) & 0 & \dots & 0 \\ 0 & u_{-1}(-n+2) & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & u_{-1}(-n+P) \end{bmatrix}. \quad (21)$$

Using (18) and (20) in (8) as in [2], we obtain

$$\begin{aligned} \mathbf{v}(n+1) = \mathbf{v}(n) &- \frac{\boldsymbol{\phi}(n)\boldsymbol{\phi}^T(n)\mathbf{v}(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)} + \frac{\boldsymbol{\phi}(n)r_a(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)} \\ &- \frac{\boldsymbol{\phi}(n)\hat{\mathbf{a}}^T(n)\mathbf{Q}(-n+1)\mathbf{U}^T(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}\mathbf{v}(0) \\ &+ \frac{\boldsymbol{\phi}(n)\hat{\mathbf{a}}^T(n)\mathbf{Q}(-n+1)\mathbf{r}(n-1)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}. \end{aligned} \quad (22)$$

Note the appearance of two new terms in the r.h.s. of equation (22) in comparison with (12). It is clear from (21) that all $q_i(n)$ will vanish after P iterations. Thus, both equations (12) and (22) will produce the same steady-state result.

5.3 Mean weight behaviour

Taking the expected value of (22) yields

$$\begin{aligned} E\{\mathbf{v}(n+1)\} = E\{\mathbf{v}(n)\} &- E\left\{\frac{\boldsymbol{\phi}(n)\boldsymbol{\phi}^T(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}\mathbf{v}(n)\right\} \\ &+ E\left\{\frac{\boldsymbol{\phi}(n)r_a(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}\right\} \\ &- E\left\{\frac{\boldsymbol{\phi}(n)\hat{\mathbf{a}}^T(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}\mathbf{Q}(-n+1)\mathbf{U}^T(n)\right\}\mathbf{v}(0) \\ &+ E\left\{\frac{\boldsymbol{\phi}(n)\hat{\mathbf{a}}^T(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}\mathbf{Q}(-n+1)\mathbf{r}(n-1)\right\}. \end{aligned} \quad (23)$$

Assuming that the measuring noise and the input signal are independent of each other, the 3rd and the 5th terms on the r.h.s. of (23) are null due to the zero mean of $r(n)$, leading to

$$\begin{aligned} E\{\mathbf{v}(n+1)\} = E\{\mathbf{v}(n)\} &- E\left\{\frac{\boldsymbol{\phi}(n)\boldsymbol{\phi}^T(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}\mathbf{v}(n)\right\} \\ &- E\left\{\frac{\boldsymbol{\phi}(n)\hat{\mathbf{a}}^T(n)}{\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)}\mathbf{Q}(-n+1)\mathbf{U}^T(n)\right\}\mathbf{v}(0) \end{aligned} \quad (24)$$

In the same way as in [5], numerator and denominator of each term in (24) can be assumed weakly correlated for large values of N. For ergodic inputs, this assumption is equivalent to apply the *averaging principle* [7]. As a result we have

$$\begin{aligned} E\{\mathbf{v}(n+1)\} &= E\{\mathbf{v}(n)\} \\ &- E\left\{\left[\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)\right]^{-1}\right\} E\left\{\boldsymbol{\phi}(n)\boldsymbol{\phi}^T(n)\mathbf{v}(n)\right\} \\ &- E\left\{\left[\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)\right]^{-1}\right\} \\ &\quad \times E\left\{\boldsymbol{\phi}(n)\hat{\mathbf{a}}^T(n)\mathbf{Q}(-n+1)\mathbf{U}^T(n)\right\}\mathbf{v}(0). \end{aligned} \quad (25)$$

Under assumption A3 (Section 4.2), (25) becomes

$$E\{\mathbf{v}(n+1)\} = E\{\mathbf{v}(n)\}$$
$$- E\left\{\left[\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)\right]^{-1}\right\} E\left\{\boldsymbol{\phi}(n)\boldsymbol{\phi}^T(n)\right\} E\{\mathbf{v}(n)\} \quad (26)$$
$$- E\left\{\left[\boldsymbol{\phi}^T(n)\boldsymbol{\phi}(n)\right]^{-1}\right\}\mathbf{T}(n)\mathbf{v}(0)$$

where

$$\mathbf{T}(n) = E\left\{\mathbf{u}_\perp(n)\mathbf{u}_{\|\mathbf{q}}^T(n)\right\} \quad (27)$$

with

$$\mathbf{u}_\perp(n) = \mathbf{u}(n) - \mathbf{U}(n)\left[\mathbf{U}^T(n)\mathbf{U}(n)\right]^{-1}\mathbf{U}^T(n)\mathbf{u}(n) \quad (28)$$

$$\mathbf{u}_{\|\mathbf{q}}(n) = \mathbf{U}(n)\mathbf{Q}^T(-n+1)\left[\mathbf{U}^T(n)\mathbf{U}(n)\right]^{-1}\mathbf{U}^T(n)\mathbf{u}(n) \quad (29)$$

$\mathbf{T}(n)$ depends only on the statistics of the input signal and deterministically varies with time as $\mathbf{Q}(-n+1)$ changes. $\mathbf{T}(n)$ can be directly estimated from the statistics of the input signal, assuming short-term stationarity. Finally, using (14) and (15) in (26) results, after some algebra, in

$$E\{\mathbf{v}(n+1)\} = \left(1 - \frac{1}{G-2}\right)E\{\mathbf{v}(n)\} - \frac{N}{G(G-2)\sigma_z^2}\mathbf{T}(n)\mathbf{v}(0). \quad (30)$$

Equation (30) provides a new analytical model for the mean weight behaviour of the AP algorithm. Note that (30) agrees with (16) for $n > P$, after that the extra transient term vanishes. Both (16) and (30) result in unbiased estimations of the optimal solution in steady-state.

5.4 Understanding T(n)

The input vector $\mathbf{u}(n)$ can be expressed as

$$\mathbf{u}(n) = \mathbf{u}_\perp(n) + \mathbf{u}_\|(n) \quad (31)$$

where $\mathbf{u}_\|(n)$ is the projection of $\mathbf{u}(n)$ into the subspace spanned by the columns of $\mathbf{U}(n)$ and $\mathbf{u}_\perp(n)$ is the projection of the input vector onto the orthogonal complement of the subspace defined by the columns of $\mathbf{U}(n)$. They can be written as:

$$\mathbf{u}_\|(n) = \mathbf{P}_\|(n)\mathbf{u}(n) = \mathbf{U}(n)\left[\mathbf{U}^T(n)\mathbf{U}(n)\right]^{-1}\mathbf{U}^T(n)\mathbf{u}(n) \quad (32)$$

and

$$\mathbf{u}_\perp(n) = \mathbf{P}_\perp(n)\mathbf{u}(n) = \left[\mathbf{I} - \mathbf{P}_\|(n)\right]\mathbf{u}(n) = \mathbf{u}(n) - \mathbf{U}(n)\left[\mathbf{U}^T(n)\mathbf{U}(n)\right]^{-1}\mathbf{U}^T(n)\mathbf{u}(n) \quad (33)$$

where $\mathbf{P}_\|(n)$ is the projection matrix onto the subspace spanned by the columns of $\mathbf{U}(n)$ and $\mathbf{P}_\perp(n)$ is the projection matrix onto its orthogonal complement.

Using the interpretations given by (32) and (33), it can be verified that for $n = 0$ $\mathbf{T}(n)$ corresponds to the cross-correlation matrix between the projections of the input vector into the subspace of the columns of $\mathbf{U}(n)$ and into its orthogonal complementary subspace. As time evolves, the elements of the main diagonal of $\mathbf{Q}(-n+1)$ turn to zero, resulting in a reduction in the dimensionality of the subspace of $\mathbf{u}_\|(n)$. After P iterations it turns into the null space and $\mathbf{T}(n)$ turns to a $N \times N$ null matrix.

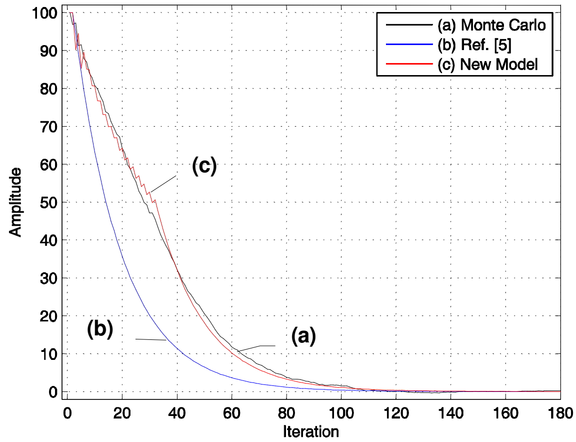

Figure 1 – Mean weight behaviour of the fifth coefficient $E\{v_5(n)\}$ (linear scale). (a) Monte Carlo simulation (black); (b) Model in [5] (blue); and (c) proposed model (red).

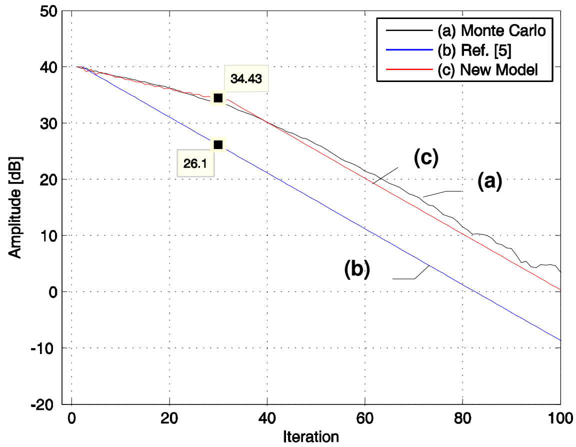

Figure 2 – Mean weight behaviour of the fifth coefficient $E\{v_5(n)\}$ (logarithmic scale). (a) Monte Carlo simulation (black); (b) Model in [5] (blue); and (c) proposed model (red).

6. SIMULATION

This section compares the accuracy of the proposed model with that of the model in [5]. Several Monte Carlo simulations were carried out in order to verify the accuracy of the analytical model given by equation (30). However, only one representative example is provided here due to space limitations.

The used parameters, chosen in order to emphasize the initialization effect, were $\sigma_z^2 = 1$, $\sigma_r^2 = 10^{-6}$, $P = 30$, $N = 50$. The input signal was generated by an AR(31) model with coefficients $a_i = (-0.99)^i$ for $i = 1, \ldots, P$. The unknown system $\mathbf{w^o}$ is a Hanning window of length N, $\mathbf{w}(0) = [100\ 100\ \ldots\ 100]^T$ and the results shown were averaged over 3000 runs. $\mathbf{T}(n)$ has been numerically estimated.

Figures 1 and 2 show the mean weight behaviour of the fifth weight as predicted by the model in [5], by the proposed model and obtained from Monte Carlo simulations in linear and logarithmic scales, respectively. Note that the new model provides a better prediction of the mean weight behaviour in the transient stage as compared to the model in [5]. Figure 2 displays an improvement of approximately 8,3 dB after P iterations.

Figures 3 and 4 show the evolution of the Euclidean distance between the weight error vectors predicted by each of the two models and the result of the Monte Carlo simulation $\sqrt{\sum_{k=1}^{N}[E\{v_{\text{model}_k}(n)\} - E\{v_{\text{simulated}_k}(n)\}]^2}$ in linear and logarithmic scale. These results corroborate the theoretically expected improvement in the predictions of the mean weight behaviour.

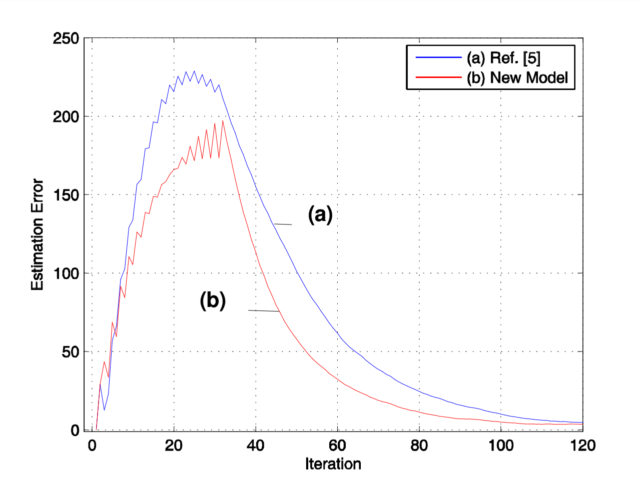

Figure 3 – Euclidean distance between the weight error vectors predicted by each of the two models and the result of the Monte Carlo simulation (linear scale). (a) Model in [5] (blue); (b) proposed model (red).

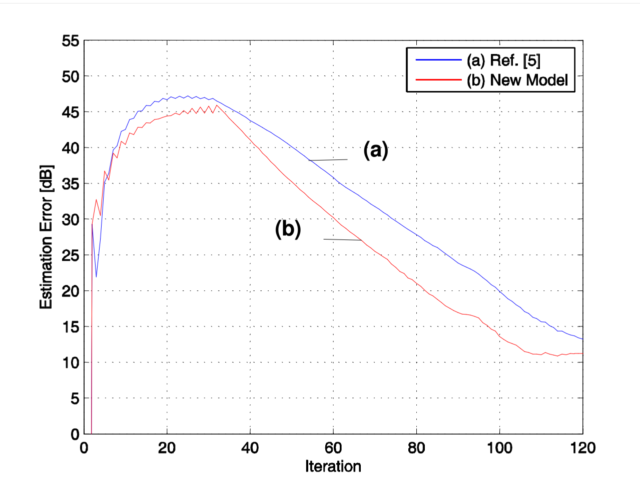

Figure 4 – Euclidean distance between the weight error vectors predicted by each of the two models and the result of the Monte Carlo simulation (logarithmic scale). (a) Model in [5] (blue); (b) proposed model (red).

7. CONCLUSIONS

This paper presented a new analytical model for predicting the mean weight behaviour of the AP algorithm for AR inputs and unity step size. This analysis focuses on modelling the initial transient phase of the AP algorithm convergence due to errors in the orthogonalization process associated with an arbitrary initialization of the adaptive coefficients. The developed model extends the model presented in [5] with the inclusion of a vanishing transient term. Monte Carlo simulations and comparisons with the model presented in [5] have shown an improvement in the predictions. The contribution made in this work can be used to improve not only previously developed theoretical models but, as well, robustness and stability analyses of AP versions.

ACKNOWLEDGMENTS

The authors would like to thank CNPq (Brazilian Ministry of Science and Technology) for the financial support (grants 303803/2009-6 and 305377/2009-4) and FAPERGS/CNPq for the partial financial support (PRONEM/FAPERGS/CNPq-2011).

REFERENCES

[1] K. Ozeki and T. Umeda, “An adaptive filtering algorithm using orthogonal projection to an affine subspace and its properties,” *Electron. Commun. Jpn.*, vol 67-A, no.5, pp. 19-27, Feb. 1984.

[2] M. Rupp, “A family of filter algorithms with decorrelating properties,” *IEEE Transactions on Signal Processing*, vol. 46, no. 3, pp. 771–775, Mar. 1998.

[3] S.L. Gay, “The fast affine projection algorithm,” *IEEE Int. Conf. Acoust., Speech, Signal Process.*, vol. 5, pp. 3023–3026, May 1995.

[4] M. Bouchard, “Multichannel affine and fast affine projection algorithms for active noise control and acoustic equalization systems,” *IEEE Transactions on Signal Processing*, vol. 11, no. 1, pp. 54–60, Jan. 2003.

[5] S.J.M. Almeida, J.C.M. Bermudez, N.J. Bershad, and M.H. Costa, “A statistical analysis of the Affine Projection algorithm for unity step size and autoregressive inputs,” *IEEE Transactions on Circuits and Systems – I*, vol. 52, no. 7, pp. 1394–1405, Jul. 2005.

[6] M. Rupp, “Pseudo-affine projection algorithms revisited: robustness and stability analysis,” *IEEE Transactions on Signal Processing*, vol. 59, no. 5, pp. 2017–2023, May 2011.

[7] J. E. Mazo, “On the independence theory of equalizer convergence,” *Bell Syst. Tech. Journal*, vol. 58, pp. 963-993, May-June 1979.