New Results on the Performance of the Affine Projection Algorithm

Sérgio J. M. de Almeida*, José Carlos M. Bermudez† and Neil J. Bershad‡

Abstract— **This paper present new results and insights on the performance of the Affine Projection algorithm, based in the stochastic model derived in [7]. The new results provide a condition for convergence of the mean weight vector, a simplified closed form model for the MSE behavior and a model for the steady-state algorithm performance. Simulation results illustrate the accuracy of the new results.**

I. Introduction

The least mean squares (LMS) adaptive algorithm and its normalized version (NLMS) are among the most often used algorithms in adaptive signal processing applications. However, their convergence rates are significantly reduced for non-white (highly correlated) inputs [1]. Acoustic echo cancellation is one important application with such input signal characteristics. The Affine Projection (AP) algorithm was proposed by Ozeki and Umeda in 1984 [2] as a solution to this problem. The AP algorithm updates the adaptive filter weights in directions that are orthogonal to the last P input vectors. This update rule whitens an AR(P) input and increases convergence speed [3]. Thus, AP is a better algorithm choice than LMS or NLMS for applications with highly correlated input signals [4]. It has been increasingly employed in applications such as echo cancellation, channel equalization and noise cancellation.

In spite of the increased interest, quantitative statistical analysis of the AP is extremely difficult because of the underdetermined least squares solution embedded in the algorithm. Reference [4] has presented a quantitative analysis of the AP algorithm. The analysis is based upon an independent input signal model originally proposed in [5] for the analysis of the NLMS algorithm. However, the independent signal model cannot handle the pre-whitening properties of the AP algorithm. Reference [6] presented a quantitative analysis for autoregressive (AR) Gaussian inputs. This analysis follows the work in [3] for obtaining the solution of a recursion for the weight error vector variances. The solution uses previous results for the NLMS algorithm with white inputs. More recently [7] presented a new statistical analysis for the behavior of the AP algorithm for AR inputs. Analytical difficulties are avoided for the case of a large number of adaptive taps compared to the AP algorithm order. This case allows an assumption similar to the "independence assumption" [1]. More recently, [8] presented a unified analysis of the transient behavior of a class of AP algorithms. The analysis is based on energy conservation arguments. The results obtained in [8] are quite general in that they do not assume a model for the input data and are valid for any adaptation step size. The resulting expressions are in terms of the statistics of the input data. Such a general model may be used for the derivation of more specialized models in the future. However, direct use in system design requires the numerical estimation of the specific input statistics. The derivation of completely analytical models for special cases of interest from the results in [8] is still an open issue.

This paper follows the work in [7]. Starting from the results derived in [7], new results and insights are obtained on the performance of the AP algorithm for AR inputs. A new stability condition is determined and a simplified analytical model is derived for the time evolution of the mean-square error (MSE). The new results lead to a closed form (non-recursive) expression for the MSE behavior. Finally, an analytical expression is obtained for the steady-state MSE behavior. Simulation results are presented which illustrate the accuracy of the derived models.

The following notation is used in this paper. Scalars are denoted by regular lowercase letters; regular uppercase letters are used for integers such as vector or matrix dimensions; vectors are all column vectors and are denoted by lowercase boldface letters. Matrices are denoted by uppercase boldface letters. The matrix dimensions are either clear from the context or explicitly given in the text.

II. The Signal Models

The adaptive system attempts to estimate a desired signal $d(n)$ which is linearly related to the input signal $u(n)$ by the model

$$d(n) = \mathbf{w^o}^T \mathbf{u}(n) + r(n) \tag{1}$$

where $\mathbf{w^o} = [w_0^o \; w_1^o \; \dots \; w_{N-1}^o]^T$ is the vector of the model parameters and the random sequence $\{r(n)\}$ is independent, identically distributed (i.i.d.), zero-mean with variance σ_r^2, and statistically independent of the random input sequence $\{u(n)\}$. $r(n)$ accounts for measurement noise and modeling errors in (1).

The input sequence $\{u(n)\}$ is assumed to be a zero-mean wide sense stationary AR process of order P and can be used to model input signals for many practical applications. Thus, $\{u(n)\}$ is described by

$$u(n) = \sum_{i=1}^{P} a_i u(n-i) + z(n) \tag{2}$$

where the sequence $\{z(n)\}$ is drawn from a wide sense stationary white process with variance σ_z^2.

A set of N consecutive samples of $\{u(n)\}$ can be collected in a vector equation. Let $\mathbf{u}(n)$ be a vector of N samples of the AR

* Escola de Engenharia e Arquitetura, Universidade Católica de Pelotas, Pelotas-RS, Brazil, e-mail: smelo@eel.ufsc.br.

† Depto. de Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil, e-mail: j.bermudez@ieee.org.

‡ Dept. of Electrical Engineering and Computer Science, University of California, Irvine, USA, e-mail: bershad@ece.uci.edu.

This work was supported in part by CNPq Grants Nos. 308095/2003-0 and 472762/2003-6.

process described in (2). Thus,

$$\mathbf{u}(n) = \sum_{i=1}^{P} a_i \mathbf{u}(n-i) + \mathbf{z}(n) = \mathbf{U}(n)\ \mathbf{a} + \mathbf{z}(n) \qquad (3)$$

where the matrix $\mathbf{U}(n) = [\mathbf{u}(n-1) \ldots \mathbf{u}(n-P)]$ is a collection of P past input vectors $\mathbf{u}(n-k) = [u(n-k)\ \ldots\ u(n-k-N+1)]^T$ and $\mathbf{z}(n) = [z(n)\ \ldots\ z(n-N+1)]^T$.

The least squares estimate of the parameter vector $\mathbf{a}$ is given by:

$$\hat{\mathbf{a}}(n) = [\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\mathbf{U}^T(n)\mathbf{u}(n) \qquad (4)$$

where $\mathbf{U}^T(n)\mathbf{U}(n)$ is assumed of rank P.

III. The Affine Projection Algorithm

The weight update equation of the AP algorithm with unity step size (maximum convergence speed) can be written as [3][1]:

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mathbf{\Phi}(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\, e(n) \qquad (5)$$

where the error signal $e(n)$ (a scalar only for unity step size) is given by

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{u}(n) = \mathbf{w}^{o^T}\mathbf{u}(n) + r(n) - \mathbf{w}^T(n)\mathbf{u}(n) \qquad (6)$$

where $\mathbf{w}(n) = [w_0(n)\ w_1(n)\ \ldots\ w_{N-1}(n)]^T$ is the adaptive weight vector. The vector $\mathbf{\Phi}(n)$ defines the direction of update, and is given by:

$$\mathbf{\Phi}(n) = \mathbf{u}(n) - \mathbf{U}(n)\,\hat{\mathbf{a}}(n). \qquad (7)$$

The order P of the AP algorithm is given by the number of past input vectors used to estimate $\hat{\mathbf{a}}(n)$ in (4). It is assumed that the order of the adaptive algorithm is sufficient to model the input AR process. Thus, the same matrix $\mathbf{U}(n)$ is used in (3), (4) and (7).

IV. Statistical Assumptions and Properties

The statistical analysis of the AP algorithm behavior requires the use of statistical assumptions to overcome mathematical difficulties. The analysis presented in [7] uses the following assumptions:

Assumption A1: The statistical dependence between $\mathbf{z}(n)$ and $\mathbf{U}(n)$ can be neglected;

Assumption A2: $\mathbf{\Phi}(n)$ and the weight vector $\mathbf{w}(n)$ are statistically independent;

Assumption A3: $\mathbf{\Phi}(n)$ is a zero mean Gaussian random vector.

These three assumptions are justified in [7]. Another important property derived in [7] is the form of the correlation matrix of the direction vector $\mathbf{\Phi}(n)$. It was shown in [7] that

$$\mathbf{R}_{\phi\phi} = E\{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\} = \sigma_\phi^2 \mathbf{I} = \left(\frac{N-P}{N}\right)\sigma_z^2 \mathbf{I} \qquad (8)$$

[1] Note that (5) corresponds to the update equation of the NLMS algorithm with unity step size and input $\phi(n)$.

V. New Stability Property

Defining the weight error vector, $\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}^{\mathbf{o}}$ and using (6), (5) can be written as

$$\mathbf{v}(n+1) = \mathbf{v}(n) - \frac{\mathbf{\Phi}(n)\mathbf{u}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\,\mathbf{v}(n) + \frac{\mathbf{\Phi}(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\, r(n). \qquad (9)$$

or, equivalently, as [3]

$$\mathbf{v}(n+1) = \mathbf{v}(n) - \frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\,\mathbf{v}(n) + \frac{\mathbf{\Phi}(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\, r_a(n) \qquad (10)$$

where $\{r_a(n)\}$ is the filtered noise sequence

$$r_a(n) = r(n) - \sum_{i=1}^{P} \hat{a}_i(n) r(n-i). \qquad (11)$$

Taking the expected value of (10), it was shown in [7] that the mean weight error vector behavior can be modeled by the recursive equation

$$E\{\mathbf{v}(n+1)\} = \left(\mathbf{I} - \frac{1}{\sigma_\phi^2(G-2)}\mathbf{R}_{\phi\phi}\right) E\{\mathbf{v}(n)\} \qquad (12)$$

where $G = N - P$. Now, using (8), (12) becomes

$$E\{\mathbf{v}(n+1)\} = \left(1 - \frac{1}{G-2}\right) E\{\mathbf{v}(n)\} \qquad (13)$$

Eq. (12) is the recursion for the mean weight error vector. Note that (12) establishes a convergence condition as a function of $G = N - P$. The mean weight vector will converge to zero if $|1 - 1/(G-2)| < 1$, which leads to the condition $G > 5/2$. Since G is an integer, convergence of the mean weight vector requires

$$G = N - P \geq 3. \qquad (14)$$

Figures 1 to 4 verify the stability condition (14) for a system identification problem with randomly selected coefficients. The mean weight and the MSE behaviors are shown for N=64 and P=61 (within the stability region) in Figs. 1 and 2. Figs. 3 and 4 show the mean weight and the MSE for N=64 and P=63, a case that violates (14). Figs. 1 and 3 show the squared norm of the mean weight error vector. The four figures correspond to averages of 100 runs. These plots clearly show that the algorithm becomes unstable when (14) is not satisfied.

VI. Mean Square Error Behavior

The expression for the mean square error (MSE) of the AP algorithm (5) is given by [7]:

$$E\{e^2(n)\} = \left(1 + \mathbf{a}^T\mathbf{a} + \sigma_z^2 \mathrm{tr}\left[E\left\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\right\}\right]\right)\sigma_r^2 + \mathrm{tr}[\mathbf{R}_{\phi\phi}\mathbf{K}(n)]. \qquad (15)$$

where $\mathbf{K}(n) = E\{\mathbf{v}(n)\mathbf{v}^T(n)\}$ is the correlation matrix of the weight error vector and tr$[\cdot]$ stands for the trace of a matrix. The

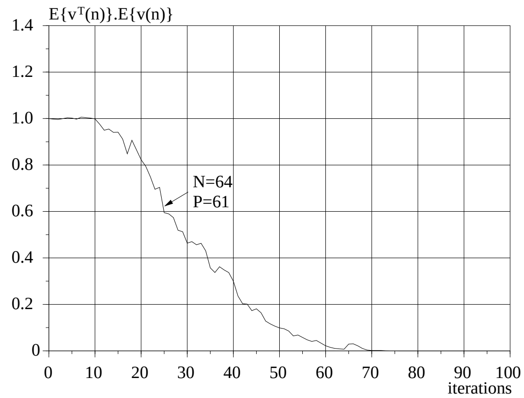

Fig. 1. $E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}$ for N=64 and P=61, stable.

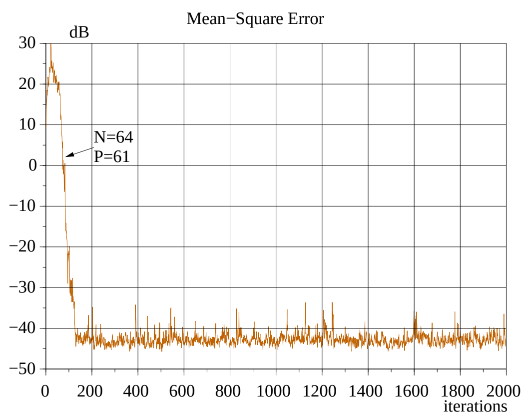

Fig. 2. MSE for N=64 and P=61, stable.

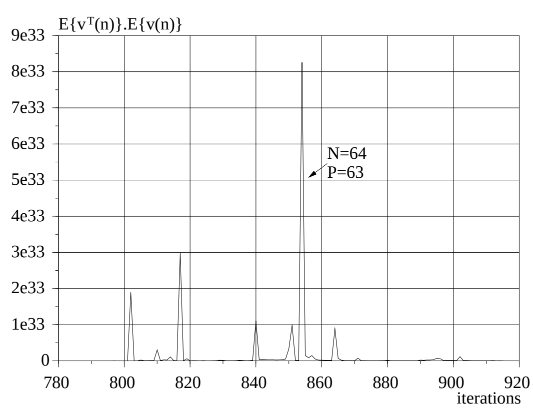

Fig. 3. $E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}$ for N=64 and P=63, unstable.

analysis in [7] has led to the following recursive equation for the

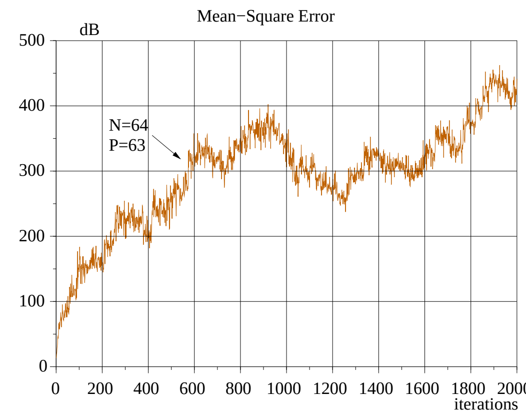

Fig. 4. MSE for N=64 and P=63, unstable.

behavior of $\mathbf{K}(n)$

$$\begin{aligned}\mathbf{K}(n+1) &= \mathbf{K}(n) - \frac{1}{\sigma_\phi^2(G-2)}[\mathbf{K}(n)\mathbf{R}_{\phi\phi} + \mathbf{R}_{\phi\phi}\mathbf{K}(n)] \\ &+ \frac{\mathbf{R}_{\phi\phi}}{\sigma_\phi^2(G^2+2G)} \\ &\times \left[\frac{G}{N}\,\mathrm{tr}[\mathbf{K}(n)] + \left(1-\frac{G}{N}\right)E\{\mathbf{v}^T(n)\}\;E\{\mathbf{v}(n)\}\right] \\ &+ \left(1 + \mathbf{a}^T\mathbf{a} + \sigma_z^2\,\mathrm{tr}\left[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}\right]\right) \\ &\times \frac{\sigma_r^2\mathbf{R}_{\phi\phi}.}{\sigma_\phi^4(G-2)(G-4)}.\end{aligned} \tag{16}$$

Though (16) provides an excellent prediction of the behavior of $\mathbf{K}(n)$, it is often more complex than necessary for the implementation of the analytical model. Furthermore, it does not provide any analytical insight on the steady-state algorithm behavior. Using again (8), simpler expressions and useful insights can be derived from (15) and (16).

A. *Transient Behavior*

Using $\mathbf{R}_{\phi\phi} = \sigma_\phi^2\mathbf{I}$ from (8), the last term of (15) reduces to $\sigma_\phi^2\mathrm{tr}[\mathbf{K}(n)]$. Thus, the MSE is a function of $\mathrm{tr}\mathbf{K}(n)$. Defining the scalars

$$\alpha = \frac{2}{G-2} \tag{17}$$

$$\beta = \frac{G}{N(G^2+2G)} \tag{18}$$

$$\gamma = \frac{P}{N(G^2+2G)} \tag{19}$$

$$\delta = \left(1 + \mathbf{a}^T\mathbf{a} + \sigma_z^2\mathrm{tr}\left[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}\right]\right) \tag{20}$$

$$\times \frac{\sigma_r^2}{\sigma_\phi^2(G-2)(G-4)} \tag{21}$$

and using $\mathbf{R}_{\phi\phi} = \sigma_\phi^2\mathbf{I}$, (16) can be written as

$$\begin{aligned}\mathbf{K}(n+1) = \mathbf{K}(n) - \alpha\mathbf{K}(n) + \beta\mathrm{tr}[\mathbf{K}(n)]\mathbf{I} \\ + \gamma E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}\mathbf{I} + \delta\mathbf{I}.\end{aligned} \tag{22}$$

Taking the trace of (22) and using the closed form solution of (12)

$$\begin{aligned} E\{\mathbf{v}(n)\} &= \left(1 - \frac{1}{G-2}\right)\mathbf{v}(0) \\ &= \left(1 - \frac{\alpha}{2}\right)^n \mathbf{v}(0) \end{aligned} \tag{23}$$

where $\mathbf{v}(0)$ is a deterministic quantity, yields

$$\begin{aligned} \text{tr}[\mathbf{K}(n+1)] &= (1-\alpha+N\beta)\text{tr}[\mathbf{K}(n)] \\ &+ N\gamma\left(1-\frac{\alpha}{2}\right)^{2n}\mathbf{v}^T(0)\mathbf{v}(0) + N\delta. \end{aligned} \tag{24}$$

As $\mathbf{v}^T(0)\mathbf{v}(0) = \text{tr}[\mathbf{K}(0)]$, the solution of (24) can be determined in closed form as

$$\begin{aligned} \text{tr}[\mathbf{K}(n)] = &\Bigg\{(1-\alpha+N\beta)^n \\ &+ N\gamma\sum_{k=0}^{n-1}(1-\alpha+N\beta)^k\left(1-\frac{\alpha}{2}\right)^{2(n-k-1)}\Bigg\} \\ &\times \text{tr}[\mathbf{K}(0)] + N\delta\sum_{k=0}^{n-1}(1-\alpha+N\beta)^k. \end{aligned} \tag{25}$$

Using (25) in (15) with $\mathbf{R}_{\phi\phi} = \sigma_\phi^2\mathbf{I}$ yields a closed form expression for the MSE.

B. *Steady-State Behavior*

Assuming convergence, the algorithm steady-state behavior can be determined as the limit as $n \to \infty$ of the analytical model. As $n \to \infty$, it can be written that $\mathbf{K}(n+1) = \mathbf{K}(n) = \mathbf{K}_\infty$. Also, $\lim_{n\to\infty} E\{\mathbf{v}(n)\} = 0$ from (12). Thus, taking the $\lim_{n\to\infty}$ of (22) yields

$$\mathbf{K}_\infty = \frac{1}{\alpha}\Big(\beta\text{tr}[\mathbf{K}_\infty] + \delta\Big)\,\mathbf{I}. \tag{26}$$

Eq. (26) clearly shows that $\mathbf{K}_\infty$ is a multiple of the identity matrix, and thus diagonal. Taking the trace of (26) yields $\text{tr}[\mathbf{K}_\infty] = N\delta/(\alpha - N\beta)$. Using this expression again in (26) leads, after some algebraic manipulation, to

$$\begin{aligned} \mathbf{K}_\infty = &\left(1 + \mathbf{a}^T\mathbf{a} + \sigma_z^2\text{tr}\Big[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}\Big]\right) \\ &\times \frac{(G+2)\sigma_r^2}{(G-4)(G+6)\sigma_\phi^2}\,\mathbf{I}. \end{aligned} \tag{27}$$

Using (27) and $\mathbf{R}_{\phi\phi} = \sigma_\phi^2\mathbf{I}$ in (15) gives the expression for the steady-state MSE

$$\begin{aligned} \xi = \lim_{n\to\infty} E\{e^2(n)\} = \\ &\left(1 + \mathbf{a}^T\mathbf{a} + \sigma_z^2\text{tr}\Big[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}\Big]\right) \\ &\times\left(1 + \frac{N(G+2)}{(G-4)(G+6)}\right)\sigma_r^2. \end{aligned} \tag{28}$$

Eq. (28) provides an expression for the steady-state MSE of the AP algorithm. Note that the multiplier $\left(1 + \frac{N(G+2)}{(G-4)(G+6)}\right)$ is reduced as $G = N - P$ increases. Thus, increasing $N - P$ reduces the steady-state MSE. This is another good reason (besides computational complexity) to use $N \gg P$ in practical designs. If $N \gg P$ and $N \gg 6$, the steady-state MSE reduces to $2\big(1 + \mathbf{a}^T\mathbf{a} + \sigma_z^2\text{tr}\big[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}\big]\big)\sigma_r^2$. The factor of 2 represents a significant increase in steady-state MSE in comparison with simpler algorithms such as NLMS. Using the same arguments provided below (16), the term $\sigma_z^2\text{tr}\big[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}\big]$ can be neglected in (28) when compared to 1, leading to $\xi \approx 2(1 + \mathbf{a}^T\mathbf{a})\sigma_r^2$. This latter result agrees with the conclusion in [3] that the AP algorithm leads to an increase in the noise floor σ_r^2 by an extra term $\mathbf{a}^T\mathbf{a}\;\sigma_r^2$. It has not been shown previously that at least 3dB is added to this increased noise floor, due to the rightmost multiplier in (28).

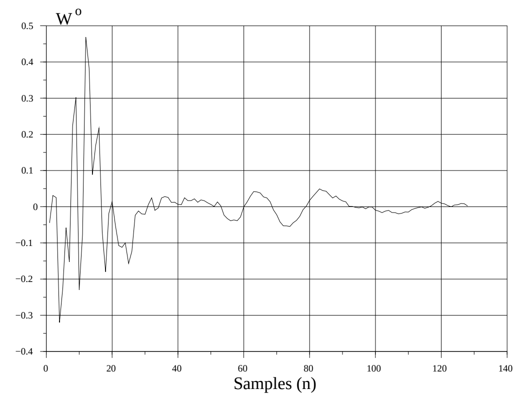

Fig. 5. Acoustic impulse response used in exemples.

VII. Simulations

This section presents simulations to verify the accuracy of the analytical models given by equations (15), (25) and (28). Several simulations have been realized using the derived models. The examples presented here are representative of the results obtained. In all cases, the term $\sigma_z^2\,\text{tr}\big[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}\big]$ has been neglected in (15), (20) and (28). In the examples, AR(P) means an autoregressive process of order P, and AP($P+1$) means the AP algorithm of order $P+1$ (using P input vectors in $\mathbf{U}(n)$). The signal-to-noise ratio of the adaptive system is defined as $SNR = 10\log(\sigma_\phi^2/\sigma_r^2)$dB, with σ_ϕ^2 obtained from (8). $SNR = 60$dB has been used in all examples. The ideal response $\mathbf{w^o}$ in each example corresponds to the first N samples of the measured acoustic response of a room shown in Fig. 5. This vector is normalized so that $\mathbf{w}^{o^T}\mathbf{w}^o = 1$.

Figs. 6, 7 and 8 show the MSE behavior obtained from Monte Carlo simulations of the algorithm (200 runs) and the behavior predicted by the models (15), (25) and (28). The ragged curves correspond to the actual algorithm behavior. The smooth curves show the MSE behavior predicted by (15) and (25). The horizontal lines (indicated as curves (a) in the plots) correspond to the steady-state MSE predicted by (28). Note that the analytical model predicts very accurately the algorithm behavior for all practical design purposes.

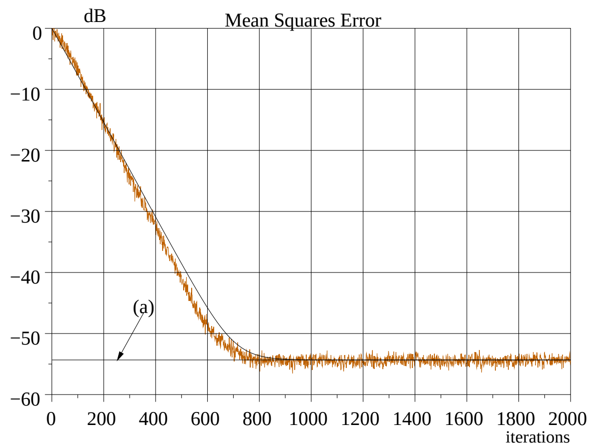

Fig. 6. MSE - AR(1), $a_1 = -0.9$, algorithm AP(3), $N = 64$. Monte Carlo Simulations (200 runs)

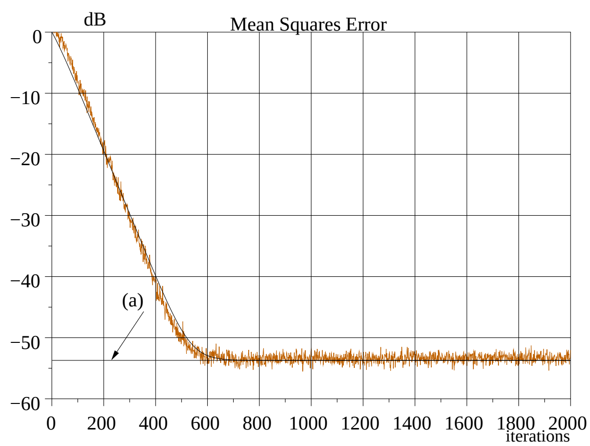

Fig. 7. MSE - AR(1), $a_1 = -0.9$, algorithm AP(16), $N = 64$. Monte Carlo Simulations (200 runs)

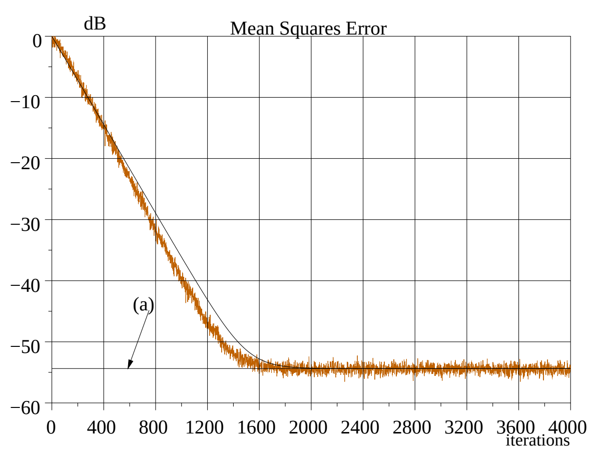

Fig. 8. MSE - AR(1), $a_1 = -0.9$, algorithm AP(3), $N = 128$. Monte Carlo Simulations (200 runs)

VIII. Conclusions

This paper presented new results and insights on the performance of the Affine Projection algorithm, based on the analytical model derived in [7]. A new mean weight error vector stability condition was derived and verified through simulations. A closed form expression was obtained for the transient behavior of the second order moments of the weight error vector. This latter result led to an expression for the steady-state MSE, which provided new interesting insights on the algorithm behavior. Monte Carlo simulations illustrated the accuracy of the new theoretical results.

References

[1] S. Haykin, *Adaptive Filter Theory* second edition, Prentice-Hall, 1991.

[2] K. Ozeki, T. Umeda, "An adaptive filtering algorithm using orthogonal projection to an affine subspace and its properties," *Electronics and Communications in Japan*, vol.67-A, No.5, pp. 19-27, February 1984.

[3] M. Rupp, "A family of filter algorithms with decorrelating properties," *IEEE Trans. on Signal Processing*, vol.46 , No.3, pp.771-775, March 1998.

[4] S. G. Sankaran, A. A. (Louis) Beex, "Convergence behavior of affine projection algorithms," *IEEE Trans. on Signal Processing*, vol.48, No.4, pp.1086-1096.

[5] D. T. M. Slock, "On the convergence behavior of the LMS and the normalized LMS algorithms," *IEEE Trans. on Signal Processing*, vol.41 , No. 9, pp. 2811-2825, September 1993.

[6] N.J. Bershad, D. Linebarger, S. McLaughlin, "A stochastic analysis of the affine projection algorithm for gaussian autoregressive inputs," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol.6, pp.3837-3840, 2001.

[7] S. J. M. Almeida, J. C. M. Bermudez, N. J. Bershad and M. H. Costa, "A stochastic model for the convergence behavior of the affine projection algorithm for Gaussian inputs," *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. VI, pp. 313-316, 2003.

[8] H. C. Shin and A. H. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Trans. on Signal Processing*, vol. 52, No. 1, pp. 90-102, January 2004.