A Statistical Analysis of the Affine Projection Algorithm for Unity Step Size and Autoregressive Inputs

Sérgio J. M. de Almeida, José Carlos M. Bermudez, *Senior Member, IEEE*, Neil J. Bershad, *Fellow, IEEE*, and Márcio Holsbach Costa, *Member, IEEE*

***Abstract*—This paper presents a new statistical analysis of the affine projection (AP) algorithm. An analytical model is derived for autoregressive (AR) inputs for unity step size (fastest convergence). Deterministic recursive equations are derived for the mean AP weight and mean-square error for large values of N (the number of adaptive taps). The value of N is also assumed large compared to the algorithm order (number of input vectors used to determine the weight update direction). The model predictions display better agreement with Monte Carlo simulations in both transient and steady-state than models previously presented in the literature. The model's accuracy is sufficient for most practical design purposes.**

***Index Terms*—Adaptive filters, adaptive signal processing, statistical analysis, system identification.**

I. Introduction

THE least mean squares (LMS) adaptive algorithm and the normalized least mean squares (NLMS) algorithm are among the most often used algorithms in adaptive signal processing applications. However, their convergence rates are significantly reduced for nonwhite (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. More recent works have studied the AP algorithm as a link between NLMS and recursive least squares (RLS) algorithms, and fast versions have been proposed [3]–[6]. The AP algorithm updates the adaptive filter weights in directions that are orthogonal to the last P input vectors. This update rule whitens an autoregressive (AR) (P) input and increases convergence speed [7]. Thus, AP may be a better algorithm choice than LMS or NLMS for applications with highly correlated input signals [8]. However, the AP algorithm is computationally more complex and also results in a slightly higher noise floor. The complexity cost has decreased in importance as more advanced semiconductor elements have been developed. Thus, complex signal processing algorithms have become feasible for applications such as echo cancellation, channel equalization, and noise cancellation.

This feasibility has enhanced the interest in analyzing the stochastic behavior of the AP algorithm. However, quantitative statistical analysis is extremely difficult because of the underdetermined least squares solution embedded in the algorithm. Reference [8] has presented a quantitative analysis of the AP algorithm. The analysis is based upon an independent input signal model originally proposed in [9] for the analysis of the NLMS algorithm. However, the independent signal model cannot handle the pre-whitening properties of the AP algorithm. Reference [10] presented a quantitative analysis for AR Gaussian inputs. This analysis follows the work in [7] 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 [11] presented a new statistical analysis for the behavior of the AP algorithm for Gaussian 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]. The "independence assumption" has been used to analyze successfully many adaptive algorithms, including the AP algorithm [1], [10]. More recently, [12] 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 [12] 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 [12] is still an open issue.

This paper follows the work in [11], but is not limited to Gaussian inputs[1]. The input signal is modeled as an AR process. This is the most employed model for representing signals in practical applications. The algorithm step size is unity. Though

Manuscript received April 8, 2004; revised December 2, 2004. This work was supported in part by CNPq under Grant 308 095/2003-0. This paper was recommended by Associate Editor H. Lev-Ari.

S. J. M. de Almeida is with the Escola de Engenharia e Arquitetura, Universidade Católica de Pelotas, 96010-000, Pelotas, Brazil (e-mail: smelo@eel.ufsc.br).

J. C. M. Bermudez is with the Department of Electrical Engineering, Federal University of Santa Catarina, 88040-900, Florianópolis, Brazil (e-mail: j.bermudez@ieee.org).

N. J. Bershad is with the Department of Electrical Engineering and Computer Science, University of California Irvine, Irvine, CA 92697 USA (e-mail: bershad@ece.uci.edu).

M. H. Costa was with Grupo de Engenharia Biomédica, Escola de Engenharia e Arquitetura, Universidade Católica de Pelotas, 96010-000, Pelotas, Brazil. He is now with Department of Electrical Engineering, Federal University of Santa Catarina, 88040-900, Florianópolis, Brazil (e-mail: m.costa@ieee.org).

Digital Object Identifier 10.1109/TCSI.2005.851720

[1]Initial results on this work have been presented in [15].

nonunity step sizes are employed in many practical situations, the unity step size leads to a scalar error signal, and permits a more detailed analysis than for arbitrary step sizes [7], [10], [11] when the error is a vector. For instance, the analysis in [12] holds for any step size, but the derived model requires numerical estimations of complex input statistics. Unity step size yields the fastest convergence of the AP algorithm. Hence, understanding the algorithm's performance in this case is of great interest in its own right. Specializing the analysis for AR inputs and unity step size allows the derivation of a complete analytical model which is directly applicable to practical designs and provides important insights into the AP algorithm behavior. Results in [10] and new views of the input signals are used to determine statistical properties of the decorrelated input signal. New statistical assumptions and approximations are used to evaluate the mean AP weight vector and the mean-square error (MSE) behaviors. A simple analytical model is derived for a large number of taps and for algorithm orders much smaller than the number of taps. The new model requires no numerical estimations for most practical applications and is able to predict the AP algorithm statistical behavior with sufficient accuracy for most practical design purposes. A closed-form expression is determined for the MSE behavior. Steady-state results are obtained as a limit of the recursive transient model. Monte Carlo simulations show good agreement with the theoretical predictions for design purposes, during both the adaptation phase and in steady-state.

Scalars are denoted by plain lowercase or uppercase letters, such as $x(n)$ or N. Vectors are all column vectors and denoted by lowercase boldface letters, such as $\mathbf{u}(n)$. Matrices are denoted by bold capital letters, such as $\mathbf{R}$. Sequences are denoted by curl brackets around the variable name, such as $\{r(n)\}$. The superscript T denotes vector or matrix transposition. The letter n denotes time or iteration number.

II. 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 process described in (2). Thus

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

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

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

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

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

III. Affine Projection Algorithm

The AP algorithm can be formulated as the solution of an underdetermined least squares problem subject to multiple constraints [1], [7]. The optimization problem can be stated as the minimization of the Euclidian norm of $\Delta\mathbf{w}(n) = \mathbf{w}(n+1) - \mathbf{w}(n)$, where $\mathbf{w}(n) = [w_0(n)\ w_1(n)\ \dots\ w_{N-1}(n)]^T$ is the adaptive weight vector, subject to the set of constraints

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

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

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

where $\mathbf{w}(n) = [w_0(n)\ w_1(n)\ \dots\ w_{N-1}(n)]^T$ is the adaptive weight vector, $\mathbf{U}_{\mathbf{u}}(n) = [\mathbf{u}(n)\ \mathbf{U}_{\mathbf{u}}]$ and $\mathbf{e}(n) = [e(n)\ e(n-1)\ \cdots\ e(n-P)]^T$ is the estimation error vector. 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 of the algorithm convergence behavior by the designer. Thus, using $\mu \neq 1$ leads the algorithm to perform nonorthogonal projections of the weight error vector $\mathbf{w}(n) - \mathbf{w^{o}}$ onto the subspace spanned by the columns of $\mathbf{U}_{\mathbf{u}}(n)$.

Expressing the set of constraints and the error vector in vector form yields

$$\mathbf{d}(n) - \mathbf{U}_{\mathbf{u}}^T(n)\mathbf{w}(n+1) = \mathbf{0} \tag{6}$$

$$\begin{aligned} \mathbf{e}(n) &= \mathbf{d}(n) - \mathbf{U}_{\mathbf{u}}^T(n)\mathbf{w}(n) \\ &= \begin{bmatrix} d(n) \\ d(n-1) \\ \vdots \\ d(n-P) \end{bmatrix} - \begin{bmatrix} \mathbf{u}^T(n) \\ \mathbf{U}(n) \end{bmatrix}\mathbf{w}(n) \end{aligned} \tag{7}$$

where $\mathbf{0}$ is the $(P+1)\times 1$ vector of zeros.

Using the first row of (6) at sample instants $n-1,\ldots,n-P$ it can be easily verified that rows 2 through $P+1$ of (7) are equal to zero. Thus, for $\mu = 1$ the error vector is equal to $\mathbf{e}(n) = [e(n)\,0\,\ldots,0]^T$,[2] and the error becomes a scalar [16, ch. 2].

When the input sequence $\{u(n)\}$ is an AR process as in (2) and $\mu = 1$ (maximum convergence speed), it was demonstrated in [7] that the AP algorithm updates the adaptive weight vector in the direction of a vector $\phi(n)$ given by

$$\mathbf{\Phi}(n) = \mathbf{u}(n) - \mathbf{U}(n)\hat{\mathbf{a}}(n) \tag{8}$$

which is orthogonal to the past P input vectors $\mathbf{u}(n-1),\ldots,\mathbf{u}(n-P)$, and that the weight update equation can be written as[3]

$$\mathbf{w}(n+1) = \mathbf{w}(n) + \frac{\mathbf{\Phi}(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}e(n) \tag{9}$$

where the scalar error signal $e(n)$ is given by

$$e(n) = d(n) - \mathbf{w}^T(n)\mathbf{u}(n) = \mathbf{w}^{o^T}\mathbf{u}(n) + r(n) - \mathbf{w}^T(n)\mathbf{u}(n). \tag{10}$$

The AP algorithm order is the number $(P+1)$ of input vectors used to represent $\mathbf{\Phi}(n)$ [number of input vectors in $\mathbf{U}_{\mathbf{u}}(n)$ in (5)].

IV. Vector and Statistical Properties of Vector $\mathbf{\Phi}(n)$

The AP algorithm convergence properties depend upon the correlation matrix $\mathbf{R}_{\phi\phi} = E\{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\}$. The structure and the properties of $\mathbf{R}_{\phi\phi}$ in turn depend upon the vector and statistical properties of $\mathbf{\Phi}(n)$. The following analysis uses assumptions similar to the "independence assumption" [1], which has been used to analyze many stochastic algorithms.

Assumption A1*:* The adaptive algorithm order is assumed sufficient to model the input AR process since $\mathbf{U}(n)$ is used in both (3) and (8).

Assumption A2*:* The statistical dependence between $\mathbf{z}(n)$ and $\mathbf{U}(n)$ can be neglected.

This assumption is more realistic for $N \gg P$ and is justified as follows. Equation (3) shows an algebraic dependence between $\mathbf{z}(n)$ and vectors $\mathbf{u}(n-1),\ldots,\mathbf{u}(n-P)$. Also, $\mathbf{z}(n)$ is of dimension N. Consider $\mathbf{P}_U(n) = \mathbf{U}(n)[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\mathbf{U}^T(n)$, the projection matrix onto the subspace spanned by the columns of $\mathbf{U}(n)$, and $\mathbf{P}_\perp(n) = \mathbf{I} - \mathbf{P}_U(n)$, the projection matrix onto the orthogonal complement subspace. Then, $\mathbf{z}(n)$ can be decomposed as $\mathbf{z}_U(n) + \mathbf{z}_\perp(n)$, where $\mathbf{z}_U(n) = P_U(n)\mathbf{z}(n)$ and $\mathbf{z}_\perp(n) = \mathbf{P}_\perp(n)\mathbf{z}(n)$. Only $\mathbf{z}_U(n)$ is algebraically dependent upon $\mathbf{U}(n)$. Moreover, since $\{z(n)\}$ is white, the average energy of $\mathbf{z}(n)$ is equally distributed among its N dimensions. Thus, only the energy in $\mathbf{z}_U(n)$ creates a dependence between $\mathbf{z}(n)$ and $\mathbf{U}(n)$. This dependence can be neglected if $N \gg P$, which is usually the case in systems with long impulse responses, since P tends to be limited by algorithm complexity considerations.

[2]Only for $\mu = 1$ the optimization constraints can be described by (6).

[3]Note that (9) corresponds to the update equation of the NLMS algorithm with unity step size and input vector $\mathbf{\Phi}(n)$.

Assumption A3*:* The vectors $\mathbf{\Phi}(n)$ and $\mathbf{w}(n)$ are statistically independent.

This assumption is similar to the "independence assumption" as applied to delay line adaptive filters with white inputs. $\mathbf{\Phi}(n)$ can be interpreted as a vector whose elements are estimates of the white noise sequence $\{z(n)\}$.

This property is verified in the following and the form of the correlation matrix $\mathbf{R}_{\phi\phi} = E\{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\}$ is determined. Substituting (4) in (8), and using (3) in the resulting expression yields

$$\mathbf{\Phi}(n) = [\mathbf{I} - \mathbf{P}_U(n)]\mathbf{z}(n) = \mathbf{P}_\perp(n)\mathbf{z}(n) = \mathbf{z}_\perp(n). \tag{11}$$

Equation (11) shows that $\mathbf{\Phi}(n)$ is orthogonal to the columns of $\mathbf{U}(n)$. This is a vector orthogonality and is valid for every n and for any realization of the input processes.

To determine the statistical properties of $\mathbf{\Phi}(n)$, note that $\mathbf{z}_\perp(n)$ is a vector with power only in $(N-P)$ dimensions of the N-dimensional space. The vector $\mathbf{z}_U(n)$ contributes the power in the remaining P dimensions. Consider a given iteration (a fixed value for n). The dimensions excited by $\mathbf{z}_\perp(n)$ are different in general for each sample function of the adaptive process because of the randomness of $u(n)$. This is equivalent to each dimension being excited on average for each run (for any given n) with $(N-P)/N$ of the power in $\mathbf{z}(n)$. Thus, using (11) and distributing the power equally in each dimension, the autocorrelation matrix of $\mathbf{\Phi}(n)$ can be written as

$$\begin{aligned}\mathbf{R}_{\phi\phi} &= E\{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\} \\ &= E\{\mathbf{z}_\perp(n)\mathbf{z}_\perp^T(n)\} \\ &= \sigma_\phi^2\mathbf{I} \\ &= \left(\frac{N-P}{N}\right)\sigma_z^2\mathbf{I}.\end{aligned} \tag{12}$$

Equations (9) and (12) show that the AP algorithm with sufficient order (P greater than or equal to the order of the input process) and N taps has a transient behavior similar to the NLMS algorithm with $N-P$ taps and a white input.

Assumption A4*:* $\mathbf{\Phi}(n)$ is a zero mean Gaussian random vector.

V. Mean Weight Behavior

Defining the weight error vector, $\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}^{\mathbf{o}}$ and using (10), (9) can be written as

$$\mathbf{v}(n+1) = \mathbf{v}(n) - \frac{\mathbf{\Phi}(n)\mathbf{u}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\mathbf{v}(n) + \frac{\mathbf{\Phi}(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}r(n). \tag{13}$$

Pre-multiplying (13) by $\mathbf{\Phi}^T(n), \mathbf{u}^T(n)$, and $\mathbf{U}^T(n)$, using (8) as $\mathbf{u}^T(n) = \mathbf{\Phi}^T(n) + \hat{\mathbf{a}}^T(n)\mathbf{U}^T(n)$, and using the property demonstrated in (11) that $\mathbf{U}^T(n)\mathbf{\Phi}(n) = \mathbf{0}$, yields

$$\mathbf{\Phi}^T(n)\mathbf{v}(n+1) = \mathbf{\Phi}^T(n)\mathbf{v}(n) - \mathbf{u}^T(n)\mathbf{v}(n) + r(n) \tag{14}$$

$$\mathbf{u}^T(n)\mathbf{v}(n+1) = r(n) \tag{15}$$

$$\mathbf{U}^T(n)\mathbf{v}(n+1) = \mathbf{U}^T(n)\mathbf{v}(n). \tag{16}$$

These properties were derived in [7] with sign changes due to a different definition of the weight error vector $\mathbf{v}(n)$[4]. Properties (15) and (16) yield

$$\mathbf{U}^T(n)\mathbf{v}(n+1) = \mathbf{r}(n-1) \tag{17}$$

where $\mathbf{r}(n-1) = [r(n-1)\dots r(n-P)]^T$.

Using (16), (17) and again $\mathbf{u}^T(n)$ as a function of $\mathbf{\Phi}(n)$ and $\mathbf{U}(n)$ in (13) leads to

$$\mathbf{v}(n+1) = \mathbf{v}(n) - \frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\mathbf{v}(n) + \frac{\mathbf{\Phi}(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}r_a(n) \tag{18}$$

where $\{r_a(n)\}$ is the filtered noise sequence [7]

$$r_a(n) = r(n) - \sum_{i=1}^{P}\hat{a}_i(n)r(n-i). \tag{19}$$

Under Assumption A3[5] and noting that $E\{\mathbf{\Phi}(n)r_a(n)\} = \mathbf{0}$ because $r(n)$ is zero mean and independent of any other signal, the expected value of (18) yields

$$E\{\mathbf{v}(n+1)\} = E\{\mathbf{v}(n)\} - E\left\{\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\}E\{\mathbf{v}(n)\}. \tag{20}$$

Each element of the expectation in the r.h.s. of (20) has a numerator given by $\phi(n-i)\phi(n-j)$ and a denominator given by $\sum_{k=0}^{N-1}\phi^2(n-k)$. The components of $\mathbf{\Phi}(n)$ in the numerator affect only two out of N terms in the denominator. Hence, numerator and denominator can be assumed weakly correlated for large values of N. For ergodic inputs, this assumption is equivalent to apply the averaging principle [13] as $\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)$ tends to be slowly time-varying when compared to $\phi(n-i)\phi(n-j)$ for large values of N. Hence, the following approximation is used:

$$E\{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^{-1}\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\} \approx E\{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^{-1}\}R_{\phi\phi} \tag{21}$$

with $\mathbf{R}_{\phi\phi}$ given by (12).

$E\{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^{-1}\}$ is evaluated assuming that $\mathbf{\Phi}(n)$ is a jointly distributed Gaussian vector with negligible statistical dependence between its components (estimates of a white sequence). Thus, $y = \mathbf{\Phi}^T(n)\mathbf{\Phi}(n)$ has a chi-square distribution with $G = N - P$ degrees of freedom. The value of G follows from the statistical properties of $\mathbf{\Phi}(n)$ determined in the previous section. Thus, [14]

$$f_y(y) = \frac{1}{2^{G/2}\sigma_\phi^G\Gamma\left(\frac{G}{2}\right)}y^{(G/2)-1}e^{-y/2\sigma_\phi^2}u(y) \tag{22}$$

[4]The weight error vector is defined as $\mathbf{w}^o - \mathbf{w}(n)$ in [7].

[5]Note that Assumption A3 also implies independence of $\mathbf{\Phi}(n)$ and $\mathbf{v}(n)$ since $\mathbf{v}(n) = \mathbf{w}(n) - \mathbf{w}^o$.

and direct integration leads to

$$E\{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^{-1}]\} = \frac{1}{\sigma_\phi^2(G-2)} \tag{23}$$

where $\sigma_\phi^2 = (N-P)/N\sigma_z^2$. Using (23) in (20) leads to

$$\begin{aligned} E\{\mathbf{v}(n+1)\} &= \left[\mathbf{I} - \frac{1}{\sigma_\phi^2(G-2)}\mathbf{R}_{\phi\phi}\right]E\{\mathbf{v}(n)\} \\ &= \left(1 - \frac{1}{G-2}\right)E\{\mathbf{v}(n)\} \end{aligned} \tag{24}$$

where the rightmost expression was obtained using $\mathbf{R}_{\phi\phi} = \sigma_\phi^2\mathbf{I}$ as determined in (12). Equation (24) is the recursion for the mean weight error vector. Note that (24) 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

$$\boxed{G = N - P \geq 3}. \tag{25}$$

The condition in (25) is a stability bound for the mean weight vector. It establishes an upper bound for P for a given number of adaptive filter coefficients. This condition is verified in Fig. 1 for a system identification problem with randomly selected coefficients. The mean weight and the MSE behaviors are shown for $N = 64$ and $P = 61$ [Fig. 1(a) and (c)], within the stability region, and for $N = 64$ and $P = 63$ [Fig. 1(b) and (d)], which violates (25). Fig. 1(a) and (b) shows the ensemble average (100 runs) of the squared norm of the mean weight vector. Fig. 1(c) and (d) shows the ensemble average (100 runs) of the MSE. The plots clearly show that the algorithm becomes unstable in the second case, confirming (25). Fortunately, this stability condition does not represent a major concern for most applications, since usually $N \gg P$ from practical design considerations.

VI. Mean-Square Error Behavior

A convenient expression for the MSE can be determined as follows. Express (10) as a function of $\mathbf{v}(n)$. Then, replace $\mathbf{u}^T(n)\mathbf{v}(n)$ using (14) and $\mathbf{\Phi}(n)$ using (8). Next, use (15) and (16) to express $\mathbf{\Phi}^T(n)\mathbf{v}(n+1)$ as a function of $r(n)$ and $\mathbf{r}(n-1)$. Finally, use (19) and square the resulting expression to obtain

$$\begin{aligned} E\{e^2(n)\} &= E\{[r_a(n) - \mathbf{\Phi}^T(n)\mathbf{v}(n)]^2\} \\ &= E\left\{r_a^2(n)\right\} - 2E\{\mathbf{\Phi}^T(n)\mathbf{v}(n)r_a(n)\} \\ &\quad + E\{\mathbf{v}^T(n)\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\mathbf{v}(n)\}. \end{aligned} \tag{26}$$

The second term on the r.h.s. of (26) can be neglected in comparison with the other two terms as explained next. Assuming that the algorithm has sufficient order (greater than P), in a

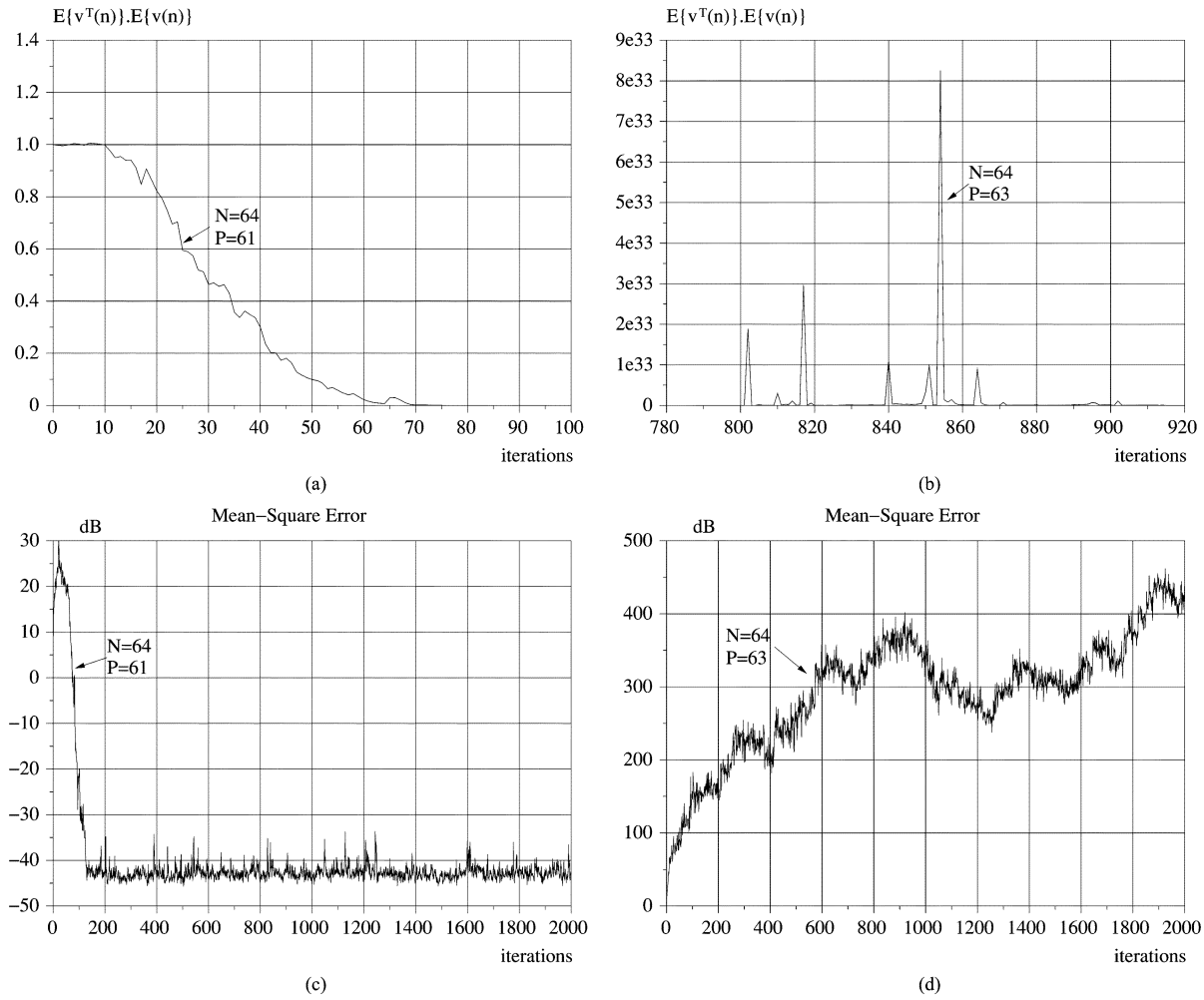

Fig. 1. Verification of stability condition. (a) $E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}$ for $N = 64$ and $P = 61$, stable. (b) $E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}$ for $N = 64$ and $P = 63$, unstable. (c) MSE for case (a). (d) MSE for case (b).

few iterations $\hat{\mathbf{a}}(n)$ tends to the vector $\mathbf{a}$ plus a small fluctuation which is correlated with $u(n)$[6]. Thus, $r_a(n)$ can be considered weakly correlated with $u(n)$. Since $\mathbf{\Phi}(n)$ is also assumed statistically independent of $\mathbf{v}(n)$, then $\mathbf{\Phi}(n)$ is weakly correlated with $\mathbf{v}(n)r_a(n)$ and the expected value can be approximated by

$$E\{\mathbf{\Phi}^T(n)\mathbf{v}(n)r_a(n)\} \approx E\{\mathbf{\Phi}^T(n)\}E\{\mathbf{v}(n)r_a(n)\} = 0. \tag{27}$$

The expectation is zero because $\mathbf{\Phi}(n)$ is zero-mean, since $u(n)$ is zero-mean. Thus, after simple manipulation, (26) can be written as

$$E\{e^2(n)\} = E\left\{r_a^2(n)\right\} + \mathrm{tr}[\mathbf{R}_{\phi\phi}\mathbf{K}(n)] \tag{28}$$

where $\mathbf{K}(n) = E\{\mathbf{v}(n)\mathbf{v}^T(n)\}$ is the correlation matrix of the weight error vector.

[6]This is the same reasoning that leads to Assumption A2 and to the whiteness of $\phi(n)$. The fluctuation in $\hat{\mathbf{a}}(n)$ is the term that leads to the vector $\mathbf{z}_U(n)$, which can be neglected for $N \gg P$.

The expected value of $r_a^2(n)$ can be expanded using (19) as

$$E\left\{r_a^2(n)\right\} = E\{r^2(n)\} - 2\sum_{i=1}^{P} E\{\hat{a}_i(n)r(n)r(n-i)\} + \sum_{i=1}^{P}\sum_{j=1}^{P} E\{\hat{a}_i(n)\hat{a}_j(n)r(n-i)r(n-j)\}. \tag{29}$$

Since $r(n)$ is white ($E\{r(n-i)r(n-j)\} = 0$ for $i \neq j$) and independent of $\hat{\mathbf{a}}(n)$, (29) becomes

$$E\left\{r_a^2(n)\right\} = E\{r^2(n)\} + \sum_{i=1}^{P} E\left\{\hat{a}_i^2(n)\right\} E\{r^2(n-i)\} = (1 + E\{\hat{\mathbf{a}}^T(n)\hat{\mathbf{a}}(n)\})\sigma_r^2. \tag{30}$$

Substituting (3) in (4), using Assumption A2 and the fact that $\mathbf{z}(n)$ is zero-mean yields, after some algebra,

$$E\{\hat{\mathbf{a}}^T(n)\hat{\mathbf{a}}(n)\} = \mathbf{a}^T\mathbf{a} + \sigma_z^2(\mathrm{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}]). \tag{31}$$

Using (31) in (30), gives

$$E\left\{r_a^2(n)\right\} = \left(1 + \mathbf{a}^T\mathbf{a} + \sigma_z^2\,\mathrm{tr}\left[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}\right]\right)\sigma_r^2. \tag{32}$$

Finally, substituting (32) in (28) yields

$$E\{e^2(n)\} = \left(1 + \mathbf{a}^T\mathbf{a} + \sigma_z^2 \operatorname{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}]\right)\sigma_r^2 + \operatorname{tr}[\mathbf{R}_{\phi\phi}\mathbf{K}(n)]. \quad (33)$$

The first term in (33) is a function of the input statistics. $\mathbf{K}(n)$ in the second term needs to be evaluated.

Post-multiplying (18) by its transpose and taking the expected value yields

$$\begin{aligned}\mathbf{K}(n+1) = \mathbf{K}(n) &- E\left\{\mathbf{v}(n)\mathbf{v}^T(n)\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} \\ &+ E\left\{\mathbf{v}(n)\frac{r_a(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} \\ &- E\left\{\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\mathbf{v}(n)\mathbf{v}^T(n)\right\} \\ &+ E\left\{\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\mathbf{v}(n)\mathbf{v}^T(n)\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} \\ &- E\left\{\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\mathbf{v}(n)\frac{r_a(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} \\ &+ E\left\{\frac{\mathbf{\Phi}(n)r_a(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\mathbf{v}^T(n)\right\} \\ &- E\left\{\frac{\mathbf{\Phi}(n)r_a(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\mathbf{v}^T(n)\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} \\ &+ E\left\{\frac{\mathbf{\Phi}(n)r_a(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\frac{r_a(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\}. \quad (34)\end{aligned}$$

The second and fourth terms on the r.h.s. of (34) are easily determined using Assumption A3, (21) and (23) as $\mathbf{K}(n)\mathbf{R}_{\phi\phi}/[\sigma_\phi^2(G-2)]$ and $\mathbf{R}_{\phi\phi}\mathbf{K}(n)/[\sigma_\phi^2(G-2)]$, respectively. Assuming $\mathbf{v}(n)r_a(n)$ uncorrelated with $\mathbf{\Phi}(n)$, the third term on the r.h.s. of (34) becomes

$$E\left\{\mathbf{v}(n)\frac{r_a(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} = E\{\mathbf{v}(n)r_a(n)\} \times E\{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^{-1}\mathbf{\Phi}^T(n)\}. \quad (35)$$

The second expected value in (35) is then determined using the same assumption used in (20)–(23)

$$E\{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^{-1}\mathbf{\Phi}^T(n)\} \approx E\{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^{-1}\} \times E\{\mathbf{\Phi}^T(n)\} = 0 \quad (36)$$

since $E\{\mathbf{\Phi}(n)\} = 0$. The same result is obtained for the seventh term of (34).

The sixth term of (34) can be evaluated as

$$E\left\{\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\mathbf{v}(n)\frac{r_a(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} = E\{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^{-2}\} \times E\{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\mathbf{v}(n)r_a(n)\mathbf{\Phi}^T(n)\}. \quad (37)$$

Since $\mathbf{\Phi}(n)$ is assumed zero-mean Gaussian (Assumption A4), third moments of its elements are equal to zero. Assuming also that $\mathbf{\Phi}(n)$ is uncorrelated with $\mathbf{v}(n)r_a(n)$, straightforward calculation shows that the second expectation in (37) equals zero. The same conclusion also holds for the eighth term in (34).

The evaluation of the fifth term on the right-hand side (r.h.s.) of (34) involves higher order statistics of the variables. Thus, it is not simple to infer any correlation property as done in the second moment calculations. To proceed with the analysis, two approximations have been tried: 1) the same approximation used in (21), in which $E\{1/[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^2\}$ multiplies the expectation of the numerator; 2) the assumption that numerator and denominator can be approximated by their mean values. It has been verified through exhaustive simulations that the latter approach leads to better matching between the model and Monte Carlo simulations. Thus

$$E\left\{\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\mathbf{v}(n)\mathbf{v}^T(n)\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} \approx \frac{1}{E\{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^2\}}E\{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n) \times \mathbf{v}(n)\mathbf{v}^T(n)\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\}. \quad (38)$$

The first expectation in (38) is evaluated by direct integration using (22)

$$\frac{1}{E\{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^2\}} = \frac{1}{\sigma_\phi^4(G^2+2G)}. \quad (39)$$

The second expectation is evaluated in the Appendix, and is given by

$$E\{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\} = \sigma_\phi^2\left[\frac{G}{N}\operatorname{tr}[\mathbf{K}(n)] + \left(1-\frac{G}{N}\right)E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}\right]\mathbf{R}_{\phi\phi}. \quad (40)$$

Using (39) and (40) in (38) yields

$$E\left\{\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\mathbf{v}(n)\mathbf{v}^T(n)\frac{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} \approx \frac{1}{\sigma_\phi^2(G^2+2G)} \times \left[\frac{G}{N}\operatorname{tr}[\mathbf{K}(n)] + \left(1-\frac{G}{N}\right)E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}\right]\mathbf{R}_{\phi\phi}. \quad (41)$$

Finally, the last expectation in (34) is determined using again the assumptions that $r_a(n)$ is uncorrelated with $\mathbf{\Phi}(n)$ and that $\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)$ is uncorrelated with $\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)$. Then

$$E\left\{\frac{\mathbf{\Phi}(n)r_a(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\frac{r_a(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} \approx E\left\{r_a^2(n)\right\} \times E\left\{\frac{1}{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^2}\right\}E\{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\}. \quad (42)$$

The first expected value in (42) is given by (32). The second expectation is evaluated by direct integration using (22)

$$E\left\{\frac{1}{[\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)]^2}\right\} = \frac{1}{\sigma_\phi^4(G-2)(G-4)} \quad (43)$$

and the third expectation is simply $\mathbf{R}_{\phi\phi}$. Thus

$$E\left\{\frac{\mathbf{\Phi}(n)r_a(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\frac{r_a(n)\mathbf{\Phi}^T(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)}\right\} \approx \left(1 + \mathbf{a}^T\mathbf{a} + \sigma_z^2\operatorname{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}]\right) \times \frac{\sigma_r^2}{\sigma_\phi^4(G-2)(G-4)}\mathbf{R}_{\phi\phi}. \quad (44)$$

Using these results in (34) yields a recursion for $\mathbf{K}(n)$

$$\begin{aligned}\mathbf{K}(n+1) = \mathbf{K}(n) &- \frac{1}{\sigma_\phi^2(G-2)}[\mathbf{K}(n)\mathbf{R}_{\phi\phi} + \mathbf{R}_{\phi\phi}\mathbf{K}(n)] \\ &+ \frac{1}{\sigma_\phi^2(G^2+2G)}\left[\frac{G}{N}\operatorname{tr}[\mathbf{K}(n)] + \left(1-\frac{G}{N}\right)\right. \\ &\left.\times\, E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}\right]\mathbf{R}_{\phi\phi} \\ &+ \left(1+\mathbf{a}^T\mathbf{a}+\sigma_z^2\operatorname{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}]\right) \\ &\times \frac{\sigma_r^2}{\sigma_\phi^4(G-2)(G-4)}\mathbf{R}_{\phi\phi}. \end{aligned} \tag{45}$$

Expressions (45) and (24), with $\mathbf{R}_{\phi\phi}$ given by (12), can be substituted in (33) to recursively determine the MSE behavior. The matrix $E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}$ must be numerically estimated from the input signal $u(n)$. However, assuming for simplicity that all elements of $\mathbf{U}(n)$ have similar powers, it is easy to show that $\sigma_z^2\operatorname{tr}[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}$ is approximately given by $(P/N)(\sigma_z^2/\sigma_u^2)$, which can be neglected when compared to 1 for $N \gg P$. In this case, $\sigma_z^2\operatorname{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}]$ can be neglected in both (45) and (33). The resulting model does not require any matrix estimation and is as accurate as the complete model for most practical purposes.

Since $\mathbf{R}_{\phi\phi} = \sigma_\phi^2\mathbf{I}$, the MSE expression (33) is a function of $\operatorname{tr}[\mathbf{K}(n)]$. Defining the scalars

$$\alpha = \frac{2}{G-2} \tag{46}$$

$$\beta = \frac{G}{N(G^2+2G)} \tag{47}$$

$$\gamma = \frac{P}{N(G^2+2G)} \tag{48}$$

$$\begin{aligned}\delta = &\left(1+\mathbf{a}^T\mathbf{a}+\sigma_z^2\operatorname{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}]\right) \\ &\times \frac{\sigma_r^2}{\sigma_\phi^2(G-2)(G-4)}\end{aligned} \tag{49}$$

and using $\mathbf{R}_{\phi\phi} = \sigma_\phi^2\mathbf{I}$, (45) can be written as

$$\begin{aligned}\mathbf{K}(n+1) = \mathbf{K}(n) - \alpha\mathbf{K}(n) + \beta\operatorname{tr}[\mathbf{K}(n)]\mathbf{I} \\ + \gamma E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}\mathbf{I} + \delta\mathbf{I}.\end{aligned} \tag{50}$$

Taking the trace of (50) and using the closed form solution of (24) as a function of α and $\mathbf{v}(0)$ (a deterministic quantity) yields

$$\begin{aligned}\operatorname{tr}[\mathbf{K}(n+1)] = (1-\alpha+N\beta)\operatorname{tr}[\mathbf{K}(n)] \\ + N\gamma\left(1-\frac{\alpha}{2}\right)^{2n}\mathbf{v}^T(0)\mathbf{v}(0) + N\delta.\end{aligned} \tag{51}$$

As $\mathbf{v}^T(0)\mathbf{v}(0) = \operatorname{tr}[\mathbf{K}(0)]$, the solution of (51) can be determined in closed form as

$$\begin{aligned}\operatorname{tr}[\mathbf{K}(n)] = &\left\{(1-\alpha+N\beta)^n + N\gamma\sum_{k=0}^{n-1}(1-\alpha+N\beta)^k\right. \\ &\left.\times\left(1-\frac{\alpha}{2}\right)^{2(n-k-1)}\right\}\operatorname{tr}[\mathbf{K}(0)] + N\delta\sum_{k=0}^{n-1}(1-\alpha+N\beta)^k.\end{aligned} \tag{52}$$

Using (52) in (33) with $\mathbf{R}_{\phi\phi} = \sigma_\phi^2\mathbf{I}$ yields a closed form expression for the MSE.

VII. 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 (24). Thus, taking the $\lim_{n\to\infty}$ of (50) yields

$$\mathbf{K}_\infty = \frac{1}{\alpha}\left(\beta\operatorname{tr}[\mathbf{K}_\infty] + \delta\right)\mathbf{I}. \tag{53}$$

Equation (53) clearly shows that $\mathbf{K}_\infty$ is a multiple of the identity matrix. Taking the trace of (53) yields $\operatorname{tr}[\mathbf{K}_\infty] = N\delta/(\alpha - N\beta)$ and thus

$$\begin{aligned}\mathbf{K}_\infty = &\left(1+\mathbf{a}^T\mathbf{a}+\sigma_z^2\operatorname{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}]\right) \\ &\times\frac{(G+2)\sigma_r^2}{(G-4)(G+6)\sigma_\phi^2}\mathbf{I}.\end{aligned} \tag{54}$$

Using (54) and $\mathbf{R}_{\phi\phi} = \sigma_\phi^2\mathbf{I}$ in (33) gives the expression for the steady-state MSE

$$\begin{aligned}\xi &= \lim_{n\to\infty} E\{e^2(n)\} \\ &= \left(1+\mathbf{a}^T\mathbf{a}+\sigma_z^2\operatorname{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}]\right) \\ &\quad\times\left(1+\frac{N(G+2)}{(G-4)(G+6)}\right)\sigma_r^2.\end{aligned} \tag{55}$$

Equation (55) provides an expression for the steady-state MSE of the AP algorithm. Note that the multiplier $(1+(N(G+2))/((G-4)(G+6)))$ 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 $P \ll N$ in practical designs. If $P \ll N$ and $N \gg 6$, the steady-state MSE reduces to $2(1+\mathbf{a}^T\mathbf{a}+\sigma_z^2\operatorname{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}])\sigma_r^2$. The factor $\mathbf{a}^T\mathbf{a}$ may represent a significant increase in steady-state MSE in comparison with simpler algorithms such as NLMS. Using the arguments immediately after (45), the term $\sigma_z^2\operatorname{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}]$ can be neglected in (55) 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 [7] that the AP algorithm leads to an increase in the noise floor σ_r^2 by an extra term $\mathbf{a}^T\mathbf{a}\sigma_r^2$. Moreover, since $\xi_{\min} = \sigma_r^2$, the AP algorithm adds at least 3 dB to this noise floor, due to the rightmost multiplier in (55). This conclusion is in agreement with the result reported in [12, eq.(22)].

VIII. Simulation Results

The analytical model comprised of (24), (33) and (45) has been tested in several different situations. This section presents a sample of these results to illustrate the accuracy of the new model. These results are representative of the results obtained in all tested cases. σ_z^2 is adjusted so that $\sigma_\phi^2 = 1$ in all the examples. The signal-to-noise ratio is given by $\mathrm{SNR} = 10\log_{10}(\sigma_\phi^2/\sigma_r^2)$ dB. The term $\sigma_z^2\operatorname{tr}[E\{[\mathbf{U}^T(n)\mathbf{U}(n)]^{-1}\}]$ has been neglected in (45) and (33) in all the examples. Also, 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. 2. The derived model requires large N and $N \gg P$. Our simulation results have shown that the model provides good predictions of the

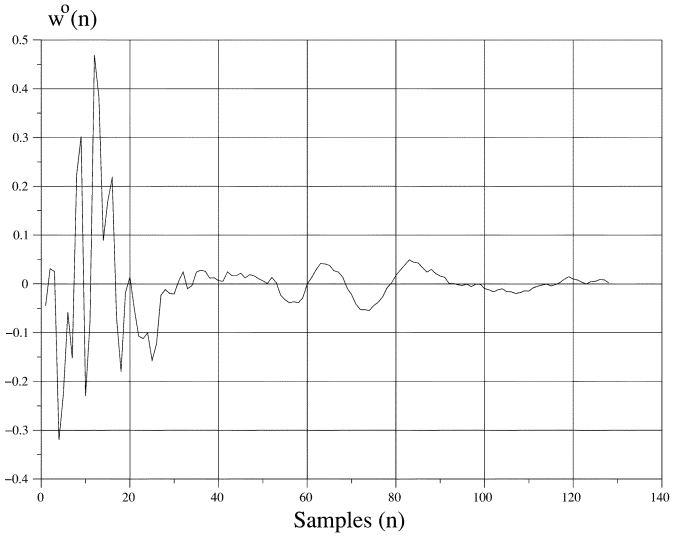

Fig. 2. Impulse response $\mathbf{w^o}$ used in all examples. In each example, the first N samples of this response are the elements of $\mathbf{w^o}$.

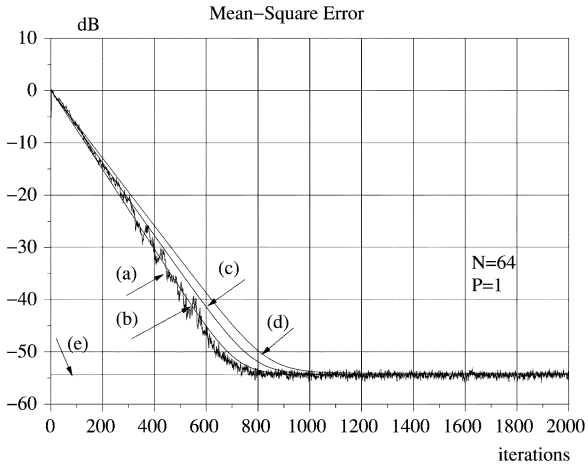

Fig. 3. MSE: Input AR(1) Gaussian, $a_1 = -0.9$, algorithm AP(2), $N = 64, \mathrm{SNR} = 60$ dB. (a) Monte Carlo simulations (200 runs). (b) New model. (c) Model in [10]. (d) Model in [11]. (e) Steady-state MSE predicted by (55).

algorithm behavior for N greater than 40. This is a reasonable value for most practical applications of the AP algorithm[7]. Also, $P \ll N$ is a desirable practical condition for manageable computational complexity and low steady-state MSE. The example parameters reflect these conditions. These examples show that this paper has advanced the stochastic analysis of the AP algorithm to an extent which is sufficient for many practical applications. However, the present model does not predict the finer details of actual performance as is most obvious from Figs. 7 and 10, shown later.

A. Example 1

The input is a Gaussian AR(1) process given by $u(n) = -0.9u(n-1)+z(n)$, with $z(n)$ white and Gaussian. Figs. 3–6 show, respectively, the MSE behavior for $N = 64$ and $P = 1, N = 64$ and $P = 4, N = 128$ and $P = 1$, and $N = 128$ and $P = 8$, all for $\mathrm{SNR} = 60$ dB. Fig. 7 shows the results corresponding to Fig. 5 for $\mathrm{SNR} = 30$ dB. The figures show the

[7]Typical values for acoustic impulse responses of a car cabin and and office have 256 and 1024 samples, respectively [17].

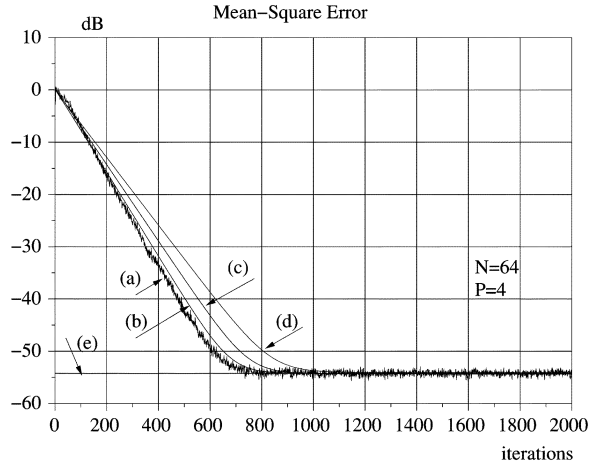

Fig. 4. MSE: Input AR(1) Gaussian, $a_1 = -0.9$, algorithm AP(5), $N = 64, \mathrm{SNR} = 60$ dB. (a) Monte Carlo simulations (200 runs). (b) New model. (c) Model in [10]. (d) Model in [11]. (e) Steady-state MSE predicted by (55).

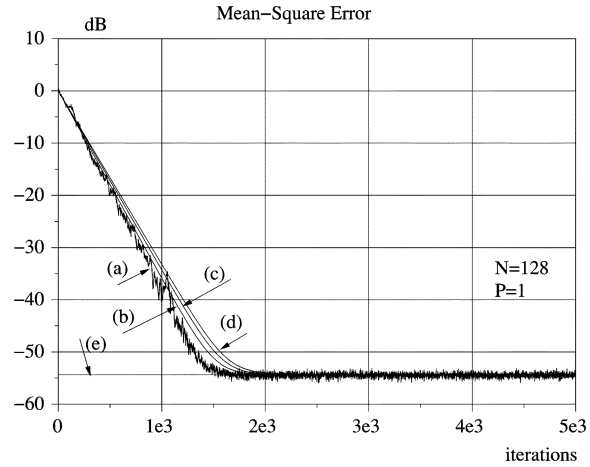

Fig. 5. MSE: Input AR(1) Gaussian, $a_1 = -0.9$, algorithm AP(2), $N = 128, \mathrm{SNR} = 60$ dB. (a) Monte Carlo simulations (200 runs). (b) New model. (c) Model in [10]. (d) Model in [11]. (e) Steady-state MSE predicted by (55).

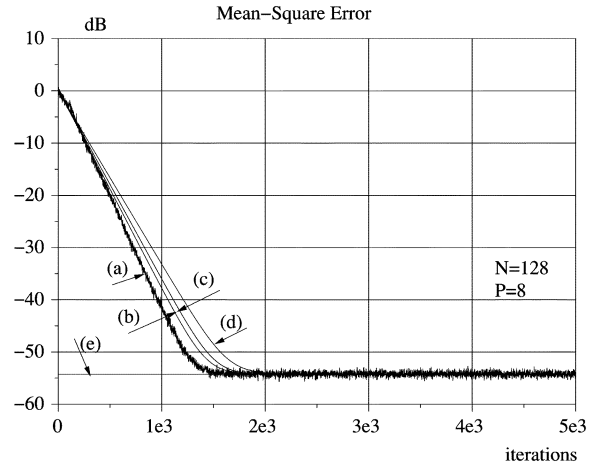

Fig. 6. MSE: Input AR(1) Gaussian, $a_1 = -0.9$, algorithm AP(9), $N = 128, \mathrm{SNR} = 60$ dB. (a) Monte Carlo simulations (200 runs). (b) New model. (c) Model in [10]. (d) Model in [11]. (e) Steady-state MSE predicted by (55).

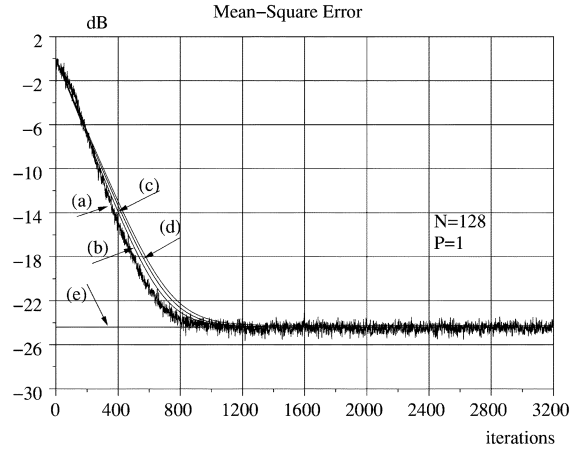

Fig. 7. MSE: Input AR(1) Gaussian, $a_1 = -0.9$, algorithm AP(2), $N = 128, \mathrm{SNR} = 30$ dB. (a) Monte Carlo simulations (200 runs). (b) New model. (c) Model in [10]. (d) Model in [11]. (e) Steady-state MSE predicted by (55).

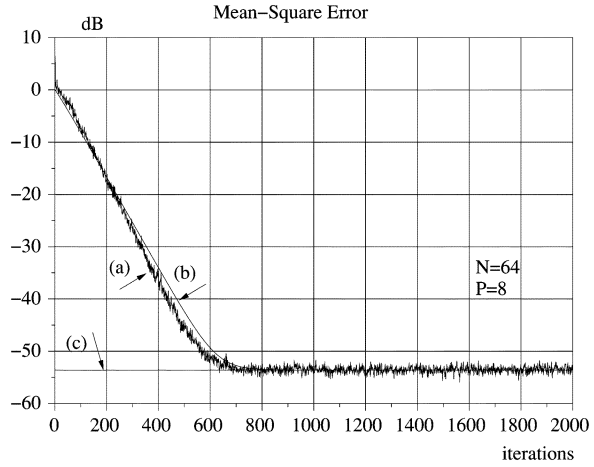

Fig. 8. MSE: Input AR(2) Gaussian, $\mathbf{a} = [0.95 \ -0.35]^T$, algorithm AP(9), $N = 64, \mathrm{SNR} = 60$ dB. (a) Monte Carlo simulations (200 runs). (b) New model. (c) Steady-state MSE predicted by (55).

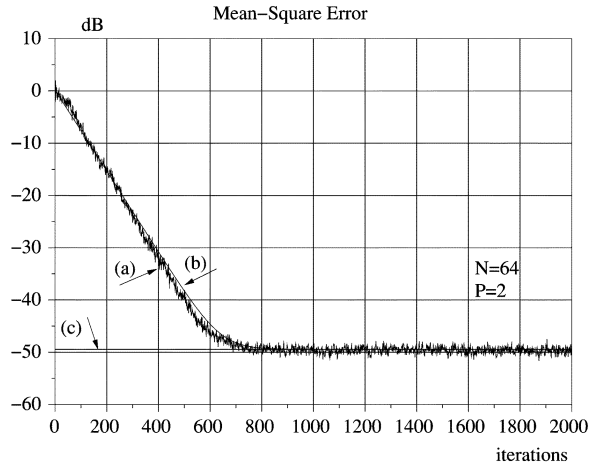

Fig. 9. MSE: Input AR(2) Gaussian, $\mathbf{a} = [-1.9114 \ -0.95]^T$, algorithm AP(3), $N = 64, \mathrm{SNR} = 60$ dB. (a) Monte Carlo simulations (200 runs). (b) New model. (c) Steady-state MSE predicted by (55).

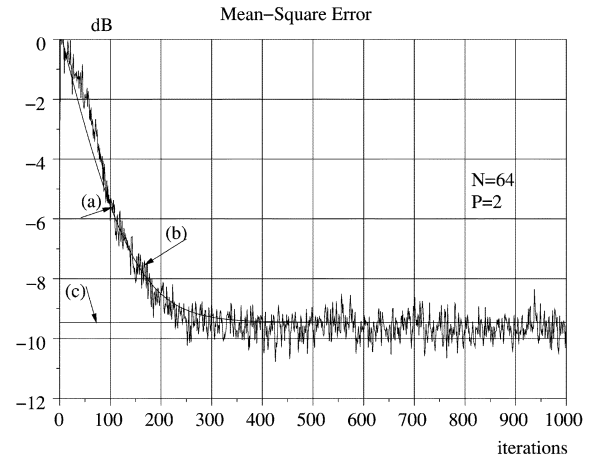

Fig. 10. MSE: Input AR(2) Gaussian, $\mathbf{a} = [-1.9114 \ -0.95]^T$, algorithm AP(3), $N = 64, \mathrm{SNR} = 20$ dB. (a) Monte Carlo simulations (200 runs). (b) New model. (c) Steady-state MSE predicted by (55).

Monte Carlo simulation results (average of 200 realizations) and the MSE behavior predicted by the new model. The MSE behavior predicted by the models in [10] and [11] are also shown in the figures. Note that the new model provides the best prediction of the algorithm behavior. This has been true in all cases tested. The new model behavior predictions are clearly accurate for most practical purposes. The match between theory and simulation is excellent during steady-state and good during the initial transient, where an S-type performance curve can be verified. The theoretical model underestimates the MSE during the initial transient and overestimates the MSE during the later transient. Thus, the model provides a theoretical prediction that should be sufficient for most design purposes. These plots confirm the practical knowledge that AP convergence performance does not improve in proportion to P beyond the input signal's order[8]. For instance, compare Figs. 3 and 4. Similar convergence rates occur. $P = 1$ is sufficient for $\phi(n)$ to be an excellent prediction of the white noise $z(n)$. Thus, the algorithm is already working with a white input. The effect of increasing P decreases as P is increased. Comparison of Figs. 3 and 5 shows, as expected, that convergence time increases with increasing N.

B. Example 2

Consider an AR(2) input signal given by $u(n) = 0.95u(n-1) - 0.35u(n-2) + z(n)$ with $z(n)$ zero-mean white Gaussian, which corresponds to a relatively wide-band signal (pole radius equal to 0.59). $\mathrm{SNR} = 60$ dB, $N = 64$ and $P = 2$. The simulation and theoretical predictions are shown in Fig. 8. Simulation and theory are in good agreement, especially during the initial transient and in steady-state. Note the small increase in steady-state MSE as compared to Fig. 4. This is because the example in Fig. 4 has somewhat smaller values for $\mathbf{a}^T\mathbf{a}$ and for G.

C. Example 3

The input is a narrow-band AR(2) signal (pole radius equal to 0.975) given by $u(n) = -1.9114u(n-1) - 0.95u(n-2) + z(n)$ with $z(n)$ zero-mean white Gaussian. $N = 64$ and $P = 2$. Simulation and theoretical prediction are shown in Fig. 9 for $\mathrm{SNR} = 60$ dB and in Fig. 10 for $\mathrm{SNR} = 20$ dB. The theoretical

[8]The input whitening properties of the algorithm disappear when the number of past input vectors used in the AP algorithm is smaller than the order of the input process. This case is not considered in this analysis.

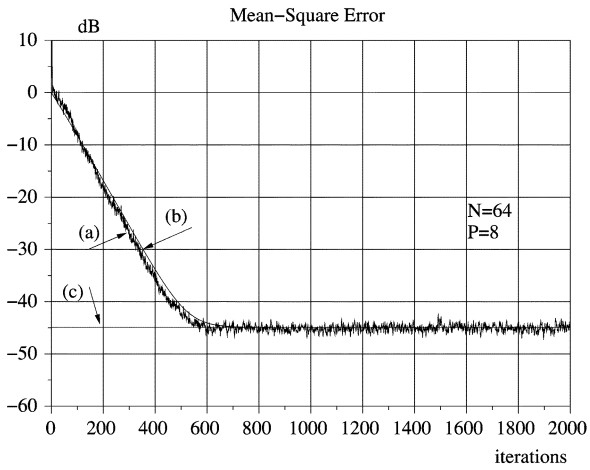

Fig. 11. MSE: Input AR(3) Gaussian, $\mathbf{a} = [2.614\ -2.5221\ 0.88155]^T$, algorithm AP(9), $N = 64, \mathrm{SNR} = 60$ dB. (a) Monte Carlo simulations (200 runs). (b) New model. (c) Steady-state MSE predicted by (55).

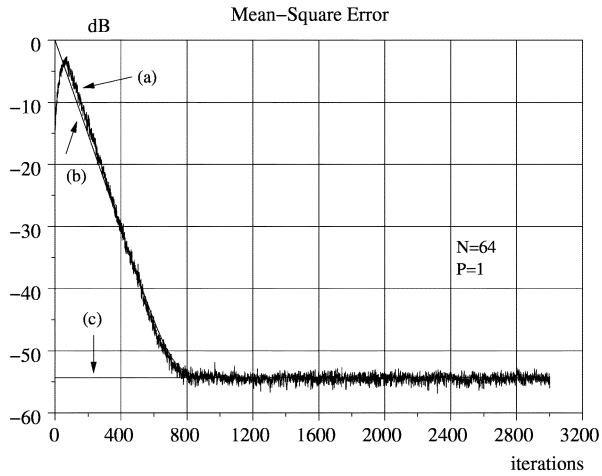

Fig. 12. MSE: Non-Gaussian correlated input generated from $z(n)$ zero mean and uniform, $a_1 = -0.9$, algorithm AP(2), $N = 64, \mathrm{SNR} = 60$ dB. (a) Monte Carlo simulations (100 runs). (b) Behavior predicted by the new model. (c) Steady-state MSE predicted by (55).

model predicts the algorithm behavior very well for the 60 dB SNR case. Compare Figs. 4 and 9 and note in both cases that the algorithm has approximately the same initial convergence rate ($\approx$15 dB after 200 iterations). However, the AR(2) case has a significantly larger steady-state MSE due to the larger value of $\mathbf{a}^T\mathbf{a}$. The slightly larger value of G (62 against 60 in Fig. 4) is not enough to compensate for the detrimental effect of the noise filtering. Fig. 10 displays the S shape performance behavior for lower SNR (20 dB) when the theory is less accurate during the transient phase as discussed above.

D. Example 4

The input is a narrow-band AR(3) signal (pole radii equal to 0.9 and 0.99) given by $u(n) = 2.614u(n-1) - 2.5221u(n-2) + 0.881\,55u(n-3) + z(n)$ with $z(n)$ zero-mean white Gaussian. $\mathrm{SNR} = 60$ dB, $N = 64$ and $P = 8$. The simulations and theoretical prediction are shown in Fig. 11. The theoretical model predictions are again in good agreement with the observed algorithm behavior. It is clear the increase in steady-state MSE is due to the larger value of $\mathbf{a}^T\mathbf{a}$ (compare with Fig. 4).

E. Example 5

This example considers non-Gaussian correlated input sequences. Two different cases are presented. First, the input AR(1) process is given by $u(n) = -0.9u(n-1) + z(n)$ with $z(n)$ zero mean uniform with variance $\sigma_z^2 = N/(N-P) = 1.02$ so that $\sigma_\phi^2 = 1$. $\mathrm{SNR} = 60$ dB, $N = 64$ and $P = 1$. Fig. 12 shows the Monte Carlo simulations of the MSE and the behavior predicted by the new model. The steady-state MSE predicted by (55) is also indicated in the figure. There is a very good match between simulation and theory in both steady-state and transient adaptation phases.

The second case presented uses an AR(1) input process given by $u(n) = -0.7u(n-1) + z(n)$ with $z(n)$ obtained from a squared Gaussian sequence with mean adjusted to zero and variance $\sigma_z^2 = N/(N-P) = 1.04$ so that $\sigma_\phi^2 = 1$. $\mathrm{SNR} = 60$ dB, $N = 64$ and $P = 4$. Fig. 13 shows the Monte Carlo simulations of the MSE and the behavior predicted by the new model.

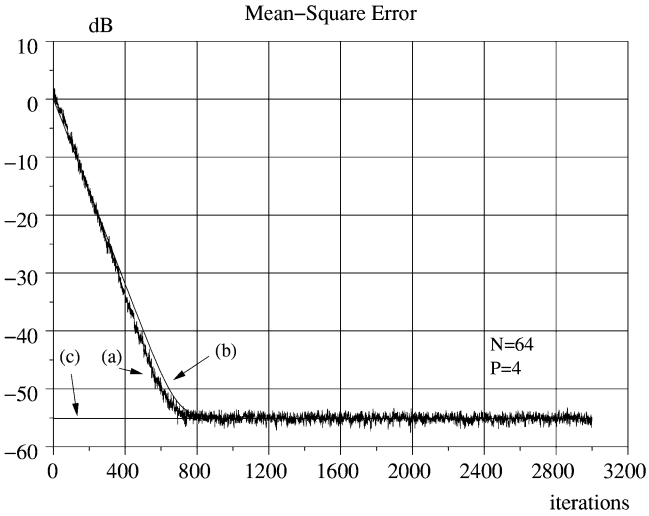

Fig. 13. MSE: Non-Gaussian correlated input generated from $z(n)$ squared Gaussian with mean reduced to zero, $a_1 = -0.7$, algorithm AP(5), $N = 64$, $\mathrm{SNR} = 60$ dB. (a) Monte Carlo simulations (100 runs). (b) Behavior predicted by the new model. (c) Steady-state MSE predicted by (55).

The steady-state MSE predicted by (55) is also indicated in the figure. As in the first case, a very good match can be verified between simulation and theory in steady-state and during the transient adaptation phase.

IX. Conclusion

This paper presents a new analytical model for predicting the stochastic behavior of the AP algorithm for AR inputs and for unity step size (fastest convergence). The analysis is based upon both the vector and statistical properties of the algorithm variables. The analytical results provide new insights into the algorithm behavior. It was shown that the transient behavior of the AP algorithm with N taps and an AR input is similar to that of the NLMS algorithm with $N - P$ taps and a white input. A theoretical stability limit was determined as a function of the difference between the number of adaptive taps and the memory of the AP algorithm $(N - P)$. The steady-state MSE was also

shown to depend on the difference $N-P$. The new model results also agree with the result in [7] for $N \gg P$: the noise floor is increased by the factor $1+\mathbf{a}^T\mathbf{a}$, due the pre-whitening behavior of the AP algorithm. For most practical applications, the new analytical model does not require numerical estimation of the input statistics. Monte Carlo simulations have shown that the theoretical model is accurate for design purposes in both the transient and steady-state.

APPENDIX
EVALUATION OF (40)

The second expectation in (38) can be written as

$$E\{\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\mathbf{v}(n)\mathbf{v}^T(n)\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\} = E\{\mathrm{tr}\,[\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\mathbf{v}(n)\mathbf{v}^T(n)]\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\}. \quad (56)$$

The (i,j)th element of (56) is given by

$$E\{\mathrm{tr}\,[\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\mathbf{v}(n)\mathbf{v}^T(n)]\phi_i(n)\phi_j(n)\} = E\left\{\left[\sum_{\ell=0}^{N-1}\sum_{k=0}^{N-1}\phi_\ell(n)\phi_k(n)v_k(n)v_\ell(n)\right]\phi_i(n)\phi_j(n)\right\}. \quad (57)$$

Now, $\phi_i(n)\phi_j(n)$ can be assumed weakly correlated with the double summation in (57) for large N, since each pair $(\phi_i(n),\phi_j(n))$ affects at most $4N-2$ out of the N^2 terms in the double summation. Using this property and Assumption A3 yields

$$E\{\mathrm{tr}\,[\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\mathbf{v}(n)\mathbf{v}^T(n)]\phi_i(n)\phi_j(n)\} = E\left\{\sum_{\ell=0}^{N-1}\sum_{k=0}^{N-1}\phi_\ell(n)\phi_k(n)E\{v_k(n)v_\ell(n)\}\right\} \times E\{\phi_i(n)\phi_j(n)\}. \quad (58)$$

Separating the terms with $i=j$ and $i\neq j$, the r.h.s. of (58) is written as

$$E\left\{\sum_{k=0}^{N-1}\phi_k^2(n)E\left\{v_k^2(n)\right\}\right\}E\{\phi_i(n)\phi_j(n)\} + E\left\{\sum_{k=0}^{N-1}\sum_{\substack{\ell=0\\ \ell\neq k}}^{N-1}\phi_\ell(n)\phi_k(n)E\{v_k(n)v_\ell(n)\}\right\} \times E\{\phi_i(n)\phi_j(n)\}. \quad (59)$$

Since from (12) $E\{\phi_i(n)\phi_j(n)\}=0$ for $i\neq k$, (58) becomes

$$E\{\mathrm{tr}\,[\mathbf{\Phi}(n)\mathbf{\Phi}^T(n)\mathbf{v}(n)\mathbf{v}^T(n)]\phi_i(n)\phi_j(n)\} = E\left\{\sum_{k=0}^{N-1}\phi_k^2(n)E\left\{v_k^2(n)\right\}\right\}E\{\phi_i(n)\phi_j(n)\}. \quad (60)$$

Since the distribution of $v(n)$ is unknown, an approximation must be determined for $E\{v_k^2(n)\}$. Assuming equal fluctuations about the mean for all components $v_k(n), k=0,\ldots,N-1$ (same variances), $E\{v_k^2(n)\}$ can be approximated by

$$E\left\{v_k^2(n)\right\} = E^2\{v_k(n)\} + \sigma^2_{v_k(n)} \approx E^2\{v_k(n)\} + \frac{\mathrm{tr}\,[\mathbf{C}(n)]}{N} \quad (61)$$

where $\mathbf{C}(n)$ is the covariance matrix of $\mathbf{v}(n)$, given by

$$\begin{aligned}\mathbf{C}(n) &= E\{(\mathbf{v}(n)-E\{\mathbf{v}(n)\})(\mathbf{v}^T(n)-E\{\mathbf{v}^T(n)\})\}\\ &= E\{\mathbf{v}(n)\mathbf{v}^T(n)\} - E\{\mathbf{v}(n)\}E\{\mathbf{v}^T(n)\}\\ &= \mathbf{K}(n) - E\{\mathbf{v}(n)\}E\{\mathbf{v}^T(n)\}.\end{aligned} \quad (62)$$

Substituting this result in (61) yields

$$E\left\{v_k^2(n)\right\} = \frac{1}{N}[\mathrm{tr}\,[\mathbf{K}(n)] - E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}] + E^2\{v_k(n)\}. \quad (63)$$

Using (63), the r.h.s. of (60) becomes

$$\begin{aligned}&E\left\{\sum_{k=0}^{N-1}\phi_k^2(n)E\left\{v_k^2(n)\right\}\right\}E\{\phi_i(n)\phi_j(n)\}\\ &= E\left\{\frac{1}{N}[\mathrm{tr}\,[\mathbf{K}(n)] - E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}]\sum_{k=0}^{N-1}\phi_k^2(n) + \sum_{k=0}^{N-1}\phi_k^2(n)E^2\{v_k(n)\}\right\}E\{\phi_i(n)\phi_j(n)\}\\ &= \left\{\frac{1}{N}[\mathrm{tr}\,[\mathbf{K}(n)] - E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}]E\{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)\} + E\left\{\sum_{k=0}^{N-1}\phi_k^2(n)E^2\{v_k(n)\}\right\}\right\}E\{\phi_i(n)\phi_j(n)\}.\end{aligned} \quad (64)$$

Since $\phi(n)$ is stationary, $E\{\phi_k^2(n)\}=\sigma_\phi^2$ for all $k=0,\ldots,N-1$. Thus, it can be easily verified that

$$E\left\{\sum_{k=0}^{N-1}\phi_k^2(n)E^2\{v_k(n)\}\right\} = \sigma_\phi^2 E\{\mathbf{v}^T(n)\}E\{\mathbf{v}(n)\}. \quad (65)$$

The expected value of $\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)$ is determined using (22):

$$E\{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)\} = \int_{-\infty}^{\infty} f_y(y)\,dy = G\sigma_\phi^2. \quad (66)$$

Using (65) and (66) in (64), inserting the result in (60) and noting that $E\{\phi_i(n)\phi_j(n)\}$ is the (i,j)th element of $\mathbf{R}_{\phi\phi}$ leads to the result shown in (40).

REFERENCES

[1] S. Haykin, *Adaptive Filter Theory*, 2nd ed. Englewood Cliffs, NJ: Prentice-Hall, 1991.

[2] 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.

[3] D. T. M. Slock, "The block underdetermined covariance (BUC) fast transversal filter (FTF) algorithm for adaptive filtering," in *Proc. 26th Asilomar Conf. Signals, Systems, Computers*, Oct. 1992, pp. 550–554.

[4] ——, "Underdetermined growing and sliding covariance fast transversal filter RLS algorithms," in *Proc. EUSIPCO'92*, 1992, pp. 1169–1172.
[5] B. Baykal and A. G. Constantinides, "Underdetermined-order recursive least-squares adaptive filtering: The concept and algorithms," *IEEE Trans. Signal Process.*, vol. 45, no. 2, pp. 346–362, Feb. 1997.
[6] S. L. Gay and S. Tavathia, "The fast affine projection algorithm," in *Proc. Int. Conf. Acoustics, Speech, Signal Processing*, vol. 5, May 2003, pp. 3023–3026.
[7] M. Rupp, "A family of filter algorithms with decorrelating properties," *IEEE Trans. Signal Process.*, vol. 46, no. 3, pp. 771–775, Mar. 1998.
[8] S. G. Sankaran and A. A. (Louis) Beex, "Convergence behavior of affine projection algorithms," *IEEE Trans. Signal Process.*, vol. 48, no. 4, pp. 1086–1096, Apr. 2000.
[9] D. T. M. Slock, "On the convergence behavior of the LMS and the normalized LMS algorithms," *IEEE Trans. Signal Process.*, vol. 41, no. 9, pp. 2811–2825, Sep. 1993.
[10] N. J. Bershad, D. Linebarger, and S. McLaughlin, "A stochastic analysis of the affine projection algorithm for gaussian autoregressive inputs," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 6, May 2001, pp. 3837–3840.
[11] S. J. M. Almeida and J. C. M. Bermudez, "A new stochastic analysis of the affine projection algorithm for gaussian inputs and large number of coefficients," in *Proc. Int. Telecommunications Symp. ITS'02*, Natal, Brazil, Sep. 2002.
[12] H. C. Shin and A. H. Sayed, "Mean-square performance of a family of affine projection algorithms," *IEEE Trans. Signal Process.*, vol. 52, no. 1, pp. 90–102, Jan. 2004.
[13] C. Samson and V. U. Reddy, "Fixed point error analysis of the normalized ladder algorithms," *IEEE Trans. Acoust., Speech, Signal Process.*, vol. ASSP-31, no. 5, pp. 1177–1191, Oct. 1983.
[14] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 1st ed., 1965.
[15] 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," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, vol. 6, Apr. 2003, pp. 313–316.
[16] S. L. Gay and J. Benesty, Eds., *Acoustic Signal Processing for Telecommunications*. Norwell, MA: Kluwer, 2000.
[17] C. Breining, P. Dreiseitel, E. Hänsler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt, and J. Tilp, "Acoustic echo control: An application of very-high-order adaptive filters," *IEEE Signal Proces. Mag.*, vol. 16, no. 4, pp. 42–69, Jul. 1999.

Sérgio J. M. de Almeida was born in Pernambuco, Brazil, in 1962. He received the diploma in electrical engineering from the University of Pernambuco, Pernambuco, Brazil, the M.Sc. degree in electrical engineering from the Federal University of Paraíba, Paraíba, Brazil, and the Ph.D.degree in electrical engineering from Federal University of Santa Catarina, Florianópolis, Brazil, in 1988, 1991, and 2004, respectively.

Since 1995, he has been a Professor in the Department of Electrical Engineering of the Catholic University of Pelotas, Pelotas, Brazil. His research interests include adaptive algorithms and adaptive filtering.

José Carlos M. Bermudez (M'85–SM'02) received the B.E.E. degree from Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, the M.Sc. degree in electrical engineering from COPPE/UFRJ, and the Ph.D. degree in electrical engineering from Concordia University, Montreal, Canada, in 1978, 1981, 1985,respectively.

He joined the Department of Electrical Engineering, Federal University of Santa Catarina (UFSC), Florianópolis, Brazil, in 1985. He is currently a Professor of electrical engineering. In the winter of 1992, he was a Visiting Researcher with the Department of Electrical Engineering, Concordia University. In 1994, he was a Visiting Researcher with the Department of Electrical Engineering and Computer Science, University of California, Irvine (UCI). His research interest have involved analog signal processing using continuous-time and sampled-data systems. His recent research interests are in digital signal processing, including linear and nonlinear adaptive filtering, active noise and vibration control, echo cancellation, image processing, and speech processing.

Prof. Bermudez served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING in the area of adaptive filtering from 1994 to 1996 and from 1999 to 2001. He was a member of the Signal Processing Theory and Methods Technical Committee of the IEEE Signal Processing Society from 1998 to 2004.

Neil J. Bershad (S'60–M'62–SM'81–F'88) received the B.E.E. degree from Rensselaer Polytechnic Institute, Troy, NY, the M.S. degree in electrical engineering from the University of Southern California, Los Angeles, CA, and the Ph.D. degree in electrical engineering from Rensselaer Polytechnic Institute in 1958, 1960, and 1962, respectively.

He joined the Faculty of the Henry Samueli School of Engineering, University of California, Irvine in 1966, and is now an Emeritus Professor of Electrical Engineering and Computer Science. His research interests have involved stochastic systems modeling and analysis. His recent interests are in the area of stochastic analysis of adaptive filters. He has published a significant number of papers on the analysis of the stochastic behavior of various configurations of the LMS adaptive filter. His present research interests include the statistical learning behavior of adaptive filter structures for nonlinear signal processing, neural networks when viewed as nonlinear adaptive filters and active acoustic noise cancellation.

Dr. Bershad has served as an Associate Editor of the IEEE TRANSACTIONS ON COMMUNICATIONS in the area of phase-locked loops and synchronization. More recently, he was an Associate Editor of the IEEE TRANSACTIONS ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING in the area of adaptive filtering.

Márcio Holsbach Costa (M'95) was born in Pelotas, Brazil, in 1970. He received the B.E.E. degree from Universidade Federal do Rio Grande do Sul (UFRGS), Porto Alegre, Brazil, the M.Sc. degree in biomedical engineering from Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, and the Doctoral degree in electrical engineering from Universidade Federal de Santa Catarina (UFSC), Florianópolis, Brazil, in 1991, 1994, and 2001, respectively.

From 1994 to 2004, he was with the Department of Electrical Engineering, Biomedical Engineering Group, Universidade Católica de Pelotas. He is currently an Associate Professor in the Department of Electrical Engineering, Universidade Federal de Santa Catarina. His present research interests are in biomedical signal processing, linear and nonlinear adaptive filters, adaptive inverse control and active noise and vibration control.