16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

A LOW COST COUPLED LMS ADAPTIVE ALGORITHM WITH AFFINE PROJECTION PERFORMANCE FOR AUTORREGRESSIVE INPUTS: MEAN WEIGHT ANALYSIS

Márcio H. Costa, Sérgio J.M. Almeida, and Guilherme H. Costa

Depto de Engenharia Elétrica, Universidade Federal de Santa Catarina, Florianópolis, Brazil
Escola de Engenharia e Arquitetura, Universidade Católica de Pelotas, Pelotas, Brazil
Depto de Engenharia Mecânica, Universidade de Caxias do Sul, Caxias do Sul, Brazil
e-mail: costa@eel.ufsc.br; smelo@ucpel.tche.br; ghcosta@ucs.br

ABSTRACT

This work presents a low cost fast convergence adaptive algorithm for autoregressive (AR) input signals. It basically consists on an NLMS adaptive algorithm coupled with a linear adaptive predictor. Mean weight theoretical analysis shows that this adaptive structure, under AR conditions, provides a performance quite similar to the well known Affine Projection algorithm. Theoretical results are supported by Monte Carlo simulations showing a very good matching for considerably correlated signals. This adaptive algorithm could be a good choice in control and filtering applications that require high adaptation speed and low computational cost.

1. INTRODUCTION

The Normalized Least Mean Square (NLMS) is a widely used adaptive algorithm in practical applications due to its robustness and ease of implementation. However, its major drawback is its slow convergence speed for correlated input signals. The Affine Projection (AP) algorithm was originally proposed by Ozeki and Umeda in 1984 [1] as a solution for this problem. The AP algorithm updates the adaptive filter weights in directions that are orthogonal to the last P input vectors. It has been shown that the AP algorithm converges much faster than NLMS for correlated inputs. The price for such performance is a computational complexity of $2NP+k_{inv}P^2$ multiply and accumulate operations, where N is the number of adaptive coefficients, P is the AP order and k_{inv} is about 7 [2]. As a result, despite its good properties, real-time applications of the AP algorithm in multichannel and long-filter problems are still restricted.

Different approaches have been used to improve the convergence speed of adaptive filters while keeping a low computational cost. Fast and efficient versions of the AP algorithm have been proposed. However, such fast algorithms have a higher complexity than the NLMS and suffer from instability problems associated with the matrix inversion process [2],[3]. In [4], Mboup et al. presented an adaptive structure where a linear adaptive predictor filters the input and error signals used in the LMS update equation. The coupled adaptive filter showed faster speed when compared with the conventional LMS. Afterwards, [5] presented a similar structure where the predictor is replaced by a copy of the adaptive filter coefficients. As a result, this algorithm presents a faster speed with a small additional complexity (50% greater). However such algorithms are capable of fastening the LMS velocity, the desired AP convergence characteristics and NLMS low cost are not simultaneously obtained.

This work demonstrates, through analytical and simulation results, that for medium to high correlated autoregressive (AR) input signals the performance of the AP algorithm can be achieved by using a very simple structure with a very low extra computational complexity when compared to the conventional NLMS. This algorithm consists on an adaptive linear predictor that decorrelates the input regressor of the NLMS adaptive filter.

A mean weight theoretical analysis is provided in order to show that this algorithm has approximately the same first order behaviour as the AP algorithm. Simulation results corroborate the analysis providing the evidence that both algorithms also present the same excess mean square error. The main limitation of this algorithm is the use of very high correlated input signals. In such situation the algorithm may not converge. Finally, simulations with a real acoustic signal show its usefulness in real conditions.

2. 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 process with variance r_z. 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 $\mathbf{u}(n)=[u(n)\ u(n-1)\ ...\ u(n-N+1)]^T$, $\mathbf{a}=[a_1\ a_2\ \dots\ a_P]^T$, $\mathbf{z}(n)=[z(n)\ z(n-1)\ ...\ z(n-N+1)]^T$, and $\mathbf{U}(n)=[\mathbf{u}(n-1)\ \mathbf{u}(n-2)\ \dots\ \mathbf{u}(n-P)]$.

3. THE AP ADAPTIVE ALGORITHM

The weight error update equation of the unit step-size AP algorithm (resulting in a scalar error) subjected to an AR input can be written as [6]

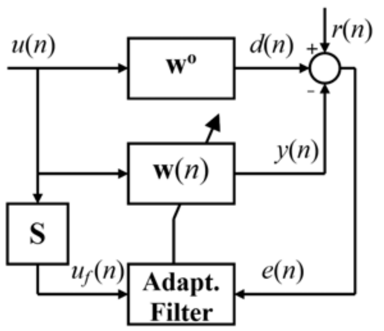

Figure 1 – Block diagram of the proposed algorithm.

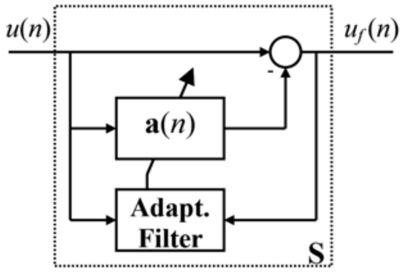

Figure 2 – Block diagram of the linear predictor.

$$\mathbf{v}_{AP}(n+1) = \mathbf{v}_{AP}(n) + \frac{\mathbf{\Phi}(n)}{\mathbf{\Phi}^T(n)\mathbf{\Phi}(n)} e_{AP}(n) \tag{3}$$

where $\mathbf{v}_{\text{AP}}(n)=\mathbf{w}_{\text{AP}}(n)-\mathbf{w}^{\mathbf{o}}$, $\mathbf{w}_{AP}(n)=[w_{AP0}\ w_{AP1}\ ...\ w_{APN-1}]^T$ is the adaptive coefficient vector, $\mathbf{w}^{\mathbf{o}}$ is the plant to be identified, $e_{AP}(n)=-\mathbf{v}_{AP}^T(n)\mathbf{u}(n)+r(n)$, $r(n)$ is the additive noise,

$$\mathbf{\Phi}(n) = \mathbf{u}(n) - \mathbf{U}(n)\mathbf{a}_{AP}(n) \tag{4}$$

and $\mathbf{a}_{AP}(n)$ is the least squares estimate of the AR coefficients:

$$\mathbf{a}_{AP}(n) = \left[\mathbf{U}^T(n)\mathbf{U}(n)\right]^{-1}\mathbf{U}^T(n)\mathbf{u}(n) \tag{5}$$

where $\mathbf{U}^T(n)\mathbf{U}(n)$ is assumed of rank P; and $\mathbf{a}_{AP}(n)=[a_{AP1}(n)\ a_{AP2}(n)\ \dots\ a_{APP}(n)]^T$.

It was demonstrated in [6] that the mean weight error behaviour of the AP can be analytically described by:

$$E\{\mathbf{v}_{\text{AP}}(n+1)\} \cong \left(1 - \frac{1}{N-P-2}\right) E\{\mathbf{v}_{\text{AP}}(n)\} \tag{6}$$

where $E\{\cdot\}$ corresponds to expectation.

4. NLMS PLUS A LINEAR PREDICTOR

The proposed adaptive structure consists on including an adaptive linear predictor (**S**) to decorrelate the NLMS input signal. This structure is shown in Fig.1. Fig.2 details the structure of the considered predictor **S**.

From the block diagram showed in Fig. 1, the proposed weight error update equation is given by

$$\mathbf{v}(n+1) = \mathbf{v}(n) + \mu \frac{\mathbf{u}_f(n)}{\mathbf{u}_f^T(n)\mathbf{u}_f(n)} e(n) \tag{7}$$

where

$$e(n) = -\mathbf{v}^T(n)\mathbf{u}(n) + r(n) \tag{8}$$

$\mathbf{v}(n)=\mathbf{w}(n)-\mathbf{w}^{\mathbf{o}}$, $\mathbf{u}_f(n)=[u_f(n)\ u_f(n-1)\ ...\ u_f(n-N+1)]^T$ and $u_f(n)$ is the output of the adaptive predictor at iteration n.

Assuming an AR input, an LMS based update strategy, a sufficiently small step-size and enough time for achieving steady-state, the predictor coefficients $\mathbf{a}(n)$ tend to the AR parameters $\mathbf{a}$, in such way that $lim_{n\to\infty}E\{\mathbf{a}(n)\}\cong \mathbf{a}$ [7]. In such case $u_f(n) \cong z(n)$.

5. MEAN WEIGHT BEHAVIOUR

This section presents a theoretical analysis of the mean weight behaviour of the proposed algorithm in order to demonstrate its similarity with the analytical model of the AP algorithm presented in [6]. The study of the statistical properties of the weight vector requires simplifications in order to make the problem mathematically tractable. These simplifications are well established in the adaptive filter analysis area and, due to their complexity and length, proofs will be presented or referenced as needed.

For simplicity we assume that there is enough time for the adaptive predictor to achieve steady-state condition in order that $u_f(n) \cong z(n)$ [7]. Considering this, using (8) in (7) and taking the expectation we obtain

$$\begin{aligned} E\{\mathbf{v}(n+1)\} &= E\{\mathbf{v}(n)\} \\ &+ \mu E\left\{\frac{r(n)\mathbf{z}(n)}{\mathbf{z}^T(n)\mathbf{z}(n)}\right\} - \mu E\left\{\frac{\mathbf{u}^T(n)\mathbf{v}(n)\mathbf{z}(n)}{\mathbf{z}^T(n)\mathbf{z}(n)}\right\} \end{aligned} \tag{9}$$

Assuming independence between the input signal and the additive noise [7] we have

$$E\left\{\frac{\mathbf{z}(n)r(n)}{\mathbf{z}^T(n)\mathbf{z}(n)}\right\} = E\left\{\frac{\mathbf{z}(n)}{\mathbf{z}^T(n)\mathbf{z}(n)}\right\} E\{r(n)\} = \mathbf{0} \tag{10}$$

Using (2) and (10) in (9)

$$\begin{aligned} E\{\mathbf{v}(n+1)\} &= E\{\mathbf{v}(n)\} - \mu E\left\{\frac{\mathbf{z}(n)\mathbf{z}^T(n)\mathbf{v}(n)}{\mathbf{z}^T(n)\mathbf{z}(n)}\right\} \\ &- \mu E\left\{\frac{\mathbf{a}^T\mathbf{U}^T(n)\mathbf{v}(n)\mathbf{z}(n)}{\mathbf{z}^T(n)\mathbf{z}(n)}\right\} \end{aligned} \tag{11}$$

Assuming a large number of coefficients we can use the averaging principle [8], and approximate (11) by

$$\begin{aligned} E\{\mathbf{v}(n+1)\} &\cong E\{\mathbf{v}(n)\} \\ &-\mu E\left\{\left[\mathbf{z}^T(n)\mathbf{z}(n)\right]^{-1}\right\} E\left\{\mathbf{z}(n)\mathbf{z}^T(n)\mathbf{v}(n)\right\} \\ &-\mu E\left\{\left[\mathbf{z}^T(n)\mathbf{z}(n)\right]^{-1}\right\} E\left\{\mathbf{a}^T\mathbf{U}^T(n)\mathbf{v}(n)\mathbf{z}(n)\right\} \end{aligned} \tag{12}$$

In appendix A, assuming $\mathbf{v}(n)$ has a Gaussian probability density function [9], it is demonstrated that

$$E\left\{\mathbf{a}^T\mathbf{U}^T(n)\mathbf{v}(n)\mathbf{z}(n)\right\} = \mathbf{0} \tag{13}$$

Using (13) in (12) we obtain

$$\begin{aligned} E\{\mathbf{v}(n+1)\} &\cong E\{\mathbf{v}(n)\} \\ &-\mu E\left\{\left[\mathbf{z}^T(n)\mathbf{z}(n)\right]^{-1}\right\} E\left\{\mathbf{z}(n)\mathbf{z}^T(n)\mathbf{v}(n)\right\} \end{aligned} \tag{14}$$

The second expected value on the right hand side of (14) was already evaluated in [10]. Differently of [10], in this case $\mathbf{z}^T(n)\mathbf{z}(n)$ has truly a chi-square distribution with N degrees of freedom since $z(n)$ is white. As a result:

$$E\left\{\left[\mathbf{z}^T(n)\mathbf{z}(n)\right]^{-1}\right\} = \frac{1}{(N-2)r_z} \quad (15)$$

The last expectation in (14) can be obtained through the use of the independence theory [7]

$$\begin{aligned} E\{\mathbf{z}(n)\mathbf{z}^T(n)\mathbf{v}(n)\} &\cong E\{\mathbf{z}(n)\mathbf{z}^T(n)\}E\{\mathbf{v}(n)\} \\ &= r_z E\{\mathbf{v}(n)\} \end{aligned} \quad (16)$$

Using (15) and (16) in (14) we finally obtain a deterministic recursive equation for describing the mean weight behaviour of the new algorithm:

$$E\{\mathbf{v}(n+1)\} \cong \left(1-\frac{\mu}{N-2}\right)E\{\mathbf{v}(n)\} \quad (17)$$

6. COUPLED NLMS VERSUS AP ALGORITHMS

The similarity between the AP and the coupled NLMS algorithms can be verified comparing Eq. (17) with the AP mean weight model, given in Eq. (6):

$$\begin{cases} E\{\mathbf{v}_{AP}(n+1)\} \cong \left(1-\frac{1}{(N-P-2)}\right)E\{\mathbf{v}_{AP}(n)\} \\ E\{\mathbf{v}(n+1)\} \cong \left(1-\frac{\mu}{N-2}\right)E\{\mathbf{v}(n)\} \end{cases} \quad (18)$$

Eq. (18) indicates that, for the same initialization, both algorithms must show approximately the same mean weight behaviour if the step-size of the proposed algorithm is equated to:

$$\mu = \frac{(N-2)}{(N-P-2)} \quad (19)$$

for $N>>P$ then $\mu\cong 1$.

7. SIMULATIONS

This section presents comparisons among the proposed algorithm, the AP and the conventional NLMS algorithms. Three examples are provided. The first two are didactical situations where the input signals are generated by first and second order AR processes. In both cases, the plant impulse response is a 100 tap normalized Hanning window ($\mathbf{w}^{oT}\mathbf{w}^{o}=1$). In the third example we consider a real acoustic impulse response and an input signal (certainly not corresponding to an AR process) sampled from a real IBM-PC cooler. The common design parameters for all simulations are: 5000 predictor iterations before running the adaptive algorithms; predictor step-size $\mu_P=0.01$; $\mathbf{w}(0)=\mathbf{0}$; regularization factor $\varepsilon=10^{-4}$ [11]; additive noise with power $r_r=10^{-6}$.

- **Example 1 – AR(1):** in this example we consider a correlated input signal generated by a first order AR model given by $u(n)=a_1u(n-1)+z(n)$, with $a_1=0.9$ and $r_z=0.19$. The step-size of the proposed algorithm is evaluated from (19) (which leads to $\mu=1.0101$) and the step-size of the conventional NLMS is set to $\mu_{NLMS}=1$. The AP and the predictor orders are equal to the autoregressive order. Monte Carlo (MC) simulations consider 500 runs. Fig.3 shows the mean weight behaviour and Fig.4 the excess mean square error (EMSE) considering the three algorithms.
- **Example 2 – AR(2):** in this example the input signal is generated by a second order AR process given by $u(n)=0.2u(n-1)-0.85u(n-2)+z(n)$ (poles radii equal to 0.922) with $r_z=0.2722$. All other simulation conditions are the same as in the Example 1. Fig.5 shows the mean weight behaviour and Fig.6 the excess mean square error.
- **Example 3 – Acoustic noise:** in this example the input signal is sampled (at 44.1kHz) from a real IBM-PC cooler acoustic noise. The plant impulse response corresponds to the first 128 samples of a real acoustic response of a room. The adaptive filter uses 128 coefficients; $\mu=\mu_{NLMS}=1$; predictor and AP orders are set to 1; 30 runs were performed in the MC simulations. Fig.7 shows the mean weight behaviour and Fig.8 the excess mean square error. Only one in each fifty samples is plotted in order to obtain smoother curves.

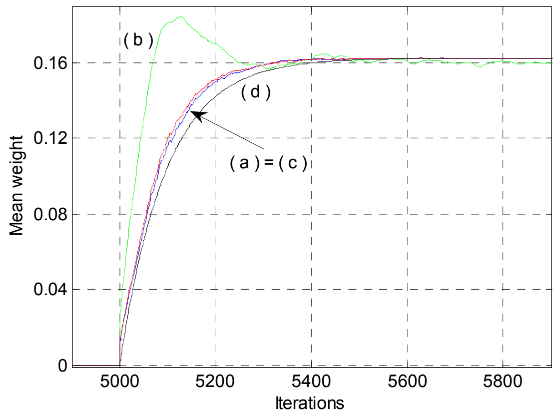

Figure 3 – Example 1. Mean weight behaviour of the 50th coefficient. (a) AP (red); (b) conventional NLMS (green); (c) NLMS + predictor (blue); and (d) analytical model (black).

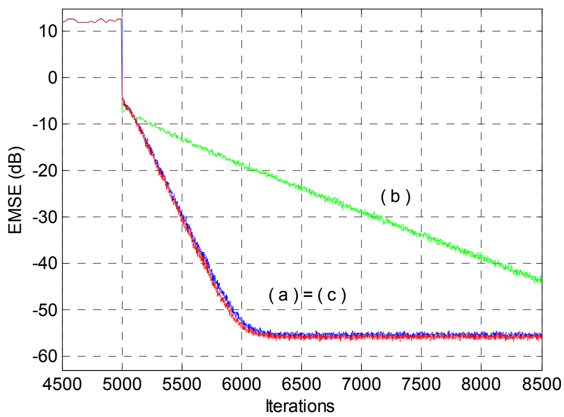

Figure 4 – Example 1. Excess mean square error. (a) AP (red); (b) conventional NLMS (green); and (c) NLMS + predictor (blue).

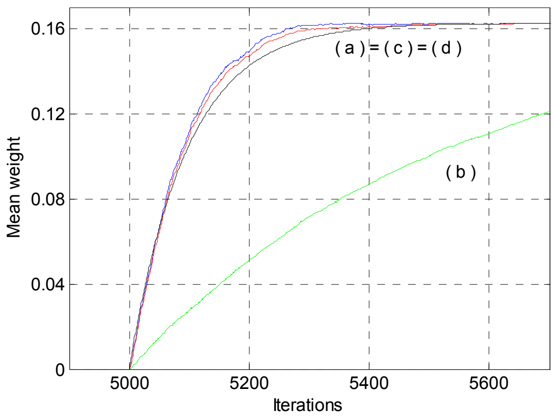

Figure 5 – Example 2. Mean weight behaviour of the 50th coefficient. (a) AP (red); (b) conventional NLMS (green); (c) NLMS + predictor (blue); and (c) analytical model (black).

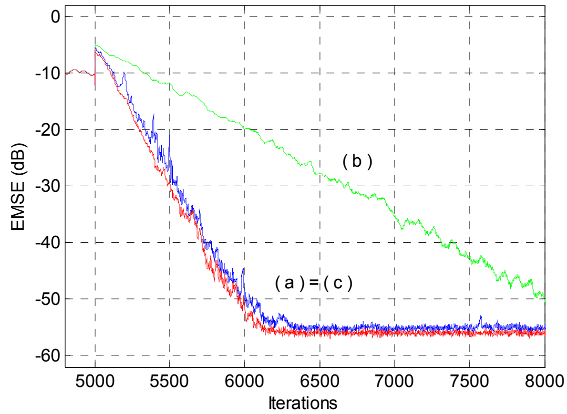

Figure 6 – Example 2. Excess mean square error. (a) AP (red); (b) conventional NLMS (green); and (c) NLMS + predictor (blue).

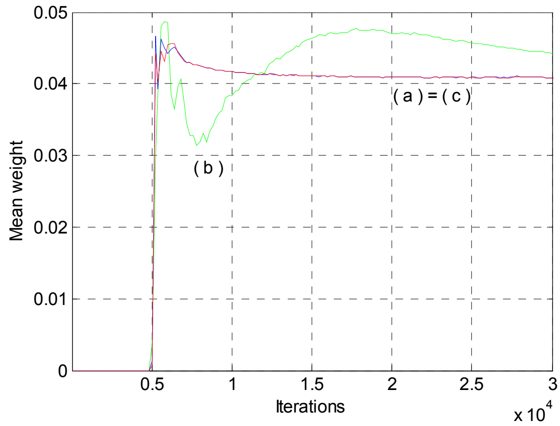

Figure 7 – Example 3. Mean weight behaviour of the 64th coefficient. (a) AP (red); (b) conventional NLMS (green); and (c) NLMS + predictor (blue).

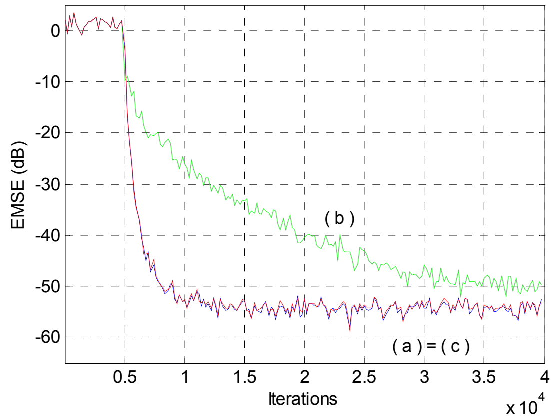

Figure 8 – Example 3. Excess mean square error. (a) AP (red); (b) conventional NLMS (green); and (c) NLMS + predictor (blue).

8. DISCUSSION

All simulation and analytical curves show a quite good matching. The behaviour of the conventional NLMS was presented only in order to show the gain of performance (due to the whiten characteristics) of the analysed algorithms. The mean weight curves illustrate the similarity between the AP and the proposed algorithm behaviours, as well as the validity of the theoretical analysis. The MC simulations show good matching even in near real conditions, when non-AR input signals are considered (Example 3). The excess mean square error simulations corroborate, under all assessed conditions, to the equivalence between the proposed structure and the AP algorithm.

The proposed algorithm has a computational load of N+2P+19 multiply and accumulate (MAC) operations [12]. This amount is of the same order of the conventional NLMS (N+18) and much less than the AP algorithm (2NP+7P^2). Table 1 presents the main steps to implement the proposed algorithm.

In spite of the good results of the proposed algorithm it was verified that it shows stability problems for AR processes with pole radius above 0.95 and highly correlated non-AR signals. In such case fast versions of the AP algorithm should be used. As a result, before practical application it is necessary a careful investigation about the input signal characteristics.

A theoretical analysis of the excess mean square error behaviour will be a topic of a future work.

Table 1: Proposed algorithm

Equation (i=1,…P) (j=1,…N)	**Comment**
$\mathbf{u}(n)=\left[u(n)\ \ \ldots\ \ u(n-N+1)\right]^T$	Input vector
$u_f(n)=u(n)-\sum_{i=1}^{P} a_i u(n-i)$	Filtered input
$\mathbf{u}_f(n)=\left[u_f(n)\ \ \ldots\ \ u_f(n-N+1)\right]^T$	Filtered input vector
$a_i(n+1)=a_i(n)+\mu_P u_f(n)u(n-i)$	Predictor update equation
$e(n)=d(n)-\sum_{i=0}^{N-1} w_i(n)u(n-i)$	Error signal
$w_j(n+1)=w_j(n)+\dfrac{\mu e(n)}{\sum_{k=1}^{N-1} u_f^2(n-k)+\varepsilon}u_f(n-j)$	NLMS update equation

9. CONCLUSION

This work presented the mean weight statistical analysis of the NLMS adaptive algorithm when the input regressor is filtered by a linear adaptive predictor. The obtained theoretical model, valid only for autoregressive signals, shows that its performance is similar to the Affine Projection algorithm, for a properly chosen step-size. Monte Carlo simulations show a very good match between both algorithms even for non-autoregressive inputs. This adaptive structure can be a good choice in control and filtering applications that require high adaptation speed and low computational cost.

ACKNOWLEDGMENTS

This work was supported by the Brazilian Ministry of Science and Technology (CNPq) under grant 301866/2005-8.

APPENDIX A: Evaluation of $E\{\mathbf{a}^T\mathbf{U}^T(n)\mathbf{v}(n)\mathbf{z}(n)\}$

Assuming three Gaussian random vectors $\mathbf{y}_1=\mathbf{a}^T\mathbf{U}^T(n)$, $\mathbf{y}_2=\mathbf{z}(n)$ and $\mathbf{y}_3=\mathbf{v}(n)$ [9] they can be expanded as an orthonormal series [13] given by

$$\begin{cases} \mathbf{y}_1 = \mathbf{A}_1\mathbf{w}_1 \\ \mathbf{y}_2 = \mathbf{B}_2\mathbf{w}_2 \\ \mathbf{y}_3 = \mathbf{C}_1\mathbf{w}_1 + \mathbf{C}_2\mathbf{w}_2 + \mathbf{C}_3\mathbf{w}_3 + \mathbf{c}_0 \end{cases} \tag{A1}$$

where $\mathbf{w}_i=[\ w_{i-1}\ w_{i-2}\ \ldots\ w_{i-N}\ ]^T$ for i=1,2; $E\{w_{l-i}w_{k-j}\}|_{i\neq j,l\neq k}=0$; $E\{w_{j-i}^2\}=1$; $\mathbf{c}_0$ is a mean value vector and

$$\mathbf{A}_1 = \begin{bmatrix} a_{1-1,1} & 0 & 0 & 0 \\ a_{1-2,1} & a_{1-2,2} & 0 & 0 \\ a_{1-3,1} & a_{1-3,2} & a_{1-3,3} & 0 \\ a_{1-4,1} & a_{1-4,2} & a_{1-4,3} & a_{1-4,4} \end{bmatrix}$$
$$\mathbf{B}_2 = \begin{bmatrix} b_{2-1,1} & 0 & 0 & 0 \\ b_{2-2,1} & b_{2-2,2} & 0 & 0 \\ b_{2-3,1} & b_{2-3,2} & b_{2-3,3} & 0 \\ b_{2-4,1} & b_{2,4,3} & b_{2,4,3} & b_{2-4,4} \end{bmatrix} \tag{A2}$$

Premultiplying $\mathbf{y}_3$ by $\mathbf{y}_1^T$ and postmultiplying the result by $\mathbf{y}_2$ we obtain

$$\begin{aligned} \mathbf{y}_1^T\mathbf{y}_3\mathbf{y}_2 = & \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\mathbf{A}_1^T\mathbf{C}_1\right)_{i,j} w_{1-i}w_{1-j}\mathbf{B}_2\mathbf{w}_2 \\ & +\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\mathbf{A}_1^T\mathbf{C}_2\right)_{i,j} w_{1-i}w_{2-j}\mathbf{B}_2\mathbf{w}_2 \\ & +\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\mathbf{A}_1^T\mathbf{C}_3\right)_{i,j} w_{1-i}w_{3-j}\mathbf{B}_2\mathbf{w}_2 \\ & +\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\mathbf{A}_1^T\right)_{i,j} c_{0-j}w_{1-i}\mathbf{B}_2\mathbf{w}_2 \end{aligned} \tag{A3}$$

Taking the expected value of (A3), using the independence between the orthonormal basis ($E\{w_{l-i}w_{k-j}\}|_{i\neq j,l\neq k}=0$) and using (A1) we come to (13).

REFERENCES

[1] K. Ozeki and T. Umeda, "An adaptive filtering algorithm using orthogonal projection to an affine subspace and its properties," *Electron. and Communic. in Japan*, vol. 67-A(5), pp.19-27, Feb. 1984.

[2] S. L. Gay and S. Tavathia, "The fast affine projection algorithm, " in *Proc. of Icassp* 2002, pp. 3023-3026.

[3] H. Ding, "A stable fast affine projection adaptation algorithm suitable for low-cost processors," in *Proc. of Icassp* 2000, pp. 360-363.

[4] M. Mboupp, M. Bonnet and N. Bershad, "LMS coupled adaptive prediction and system identification: a statistical model and transient mean analysis," *IEEE Trans. Signal Proc.*, vol. 42(10), pp. 2607-2615, Oct. 1994.

[5] S. C. Douglas, A. Cichocki and S. Amari, "Self-whitening algorithms for adaptive equalization and deconvolution," *IEEE Trans. Signal Proc.*, vol. 47(4), Apr. 1999.

[6] S. J. M. Almeida, J. C. M. Bermudez, N. J. Bershad and M. H. Costa, "Statistical analysis of the affine projection algorithm for unity size and autoregressive input," *IEEE Trans. Circ. and Syst. - I*, vol. 52, pp. 1394-1405, 2005.

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

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

[9] N. J. Bershad and Z. Q. Lian, "On the probability density function of the LMS adaptive filter weights," *Acoustics, Speech, and Signal Proc.*, vol. 37(1), pp. 43-56, 1989.

[10] M. H. Costa and J. C. M. Bermudez, "An improved model for the normalized LMS algorithm with Gaussian inputs and large number of coefficients," in *Proc. of Icassp* 2002, pp. 1-4.

[11] J. C. M. Bermudez and M.H. Costa, "A statistical analysis of the epsilon-NLMS and NLMS algorithms for correlated Gaussian Signals," *Revista da Sociedade Brasileira de Telecomunicações*, vol. 20(2), pp. 7-13, 2005.

[12] Y. Lu, R. Fowler, W. Tian, "Enhancing echo cancellation via estimation of delay", *IEEE Trans. Signal Proc.*, vol.53(11), pp. 4159-4168, 2005.

[13] N. J. Bershad, P. Celka and J. M. Vesin, "Stochastic analysis of gradient adaptive identification of nonlinear systems with memory for Gaussian data and noisy input and output measurements," *IEEE Trans. Signal Proc.*, vol. 47(3), pp. 675-689, 1999.