ON NORMALIZED DATA-REUSING AND AFFINE-PROJECTIONS ALGORITHMS

M. L. R. de Campos,† *P. S. R. Diniz,*† *and J. A. Apolinário Jr.*‡

†COPPE/Univ. Federal do Rio de Janeiro
Programa de Engenharia Elétrica
P.O. Box 68504, Rio de Janeiro, RJ
21.945-970 Brazil
campos@lps.ufrj.br, diniz@lps.ufrj.br

‡Escuela Politécnica del Ejército
Facultad de Ingeniería Electrónica
Av El Progreso, s/n
Quito, Ecuador
apolin@espe.edu.ec

‡Instituto Militar de Engenharia
Depto. de Engenharia Elétrica
Praça Gal. Tibúrcio, 80
Praia Vermelha, Rio de Janeiro, RJ
22.290-270, Brazil

ABSTRACT

This paper addresses the comparison between data-reusing LMS algorithms and the Affine-Projections algorithm. The Normalized New Data-Reusing LMS (NNDR-LMS) algorithm, the Binormalized Data-Reusing LMS (BNDR-LMS) algorithm, and the Affine-Projections (AP) algorithm are briefly presented within a common framework and their relationships are clarified. Topics such as equivalence of algorithms, graphical representation, and computational complexity are discussed.

1. INTRODUCTION

In many adaptive-filtering applications conventional gradient-type algorithms such as the least mean squares (LMS) algorithm and the normalized LMS (NLMS) algorithm do not present the necessary convergence speed. On the other hand, for these applications Newton-type algorithms and the recursive least squares (RLS) algorithm may not be suitable for their exceeding computational complexity. Data-reusing LMS (DR-LMS) algorithms and the affine-projections (AP) algorithm may be compromise solutions for many among these applications.

New normalized data-reusing algorithms have been recently proposed [1]–[5] and analyzed [6]–[8] in the literature. Similarities with the orthogonal-projections algorithm [9] and the affine-projections algorithm [10][11] have been pointed out, but not investigated in detail to yield a complete understanding of their relationship. Although algorithm derivation and implementation is often different for DR-LMS algorithms and for the AP algorithm, their similarities certainly justify a thorough discussion and clarification under a unified approach. This paper provides a discussion of the normalized new data-reusing LMS (NNDR-LMS) algorithm [1][2], the binormalized data-reusing LMS (BNDR-LMS) algorithm, and the affine-projections (AP) algorithm. It is organized as follows. In Section 2 we briefly present the NNDR-LMS, BNDR-LMS, and the AP algorithms. In Section 3 we clarify their relationships. In section 4 we discuss computational complexity of the various implementations, and in Section 5 we draw conclusions.

2. THE ALGORITHMS

Data-reusing LMS adaptation algorithms, as opposed to the conventional LMS algorithm, reutilize information provided by the available (present or past) data in order to achieve better convergence rates. It has been shown [1][2] that reusing past data pairs, i.e., input-signal vector and desired signal, is often advantageous compared to repeatedly reusing the current version of the data pair. Based on this idea, unnormalized and normalized DR-LMS algorithms were proposed [1]. Extending this concept, the BNDR-LMS algorithm was derived which also utilizes current and past data pairs, but can achieve better convergence rates at least for high signal-to-noise ratios. Although the geometric interpretation of the BNDR-LMS algorithm can be very clarifying, its derivation was based on the LMS algorithm where constraints imposed by the normalization with respect to two data pairs were added. The AP algorithm can be viewed as a generalization to an arbitrary number of data pairs of the normalized data-reusing concept of the BNDR-LMS algorithm.

In this section the NNDR-LMS, the BNDR-LMS, and the AP algorithms are briefly presented as they were derived and proposed in the literature.

2.1. The NNDR-LMS Algorithm

The NNDR-LMS algorithm [1][2] innovates with respect to DR-LMS algorithms by utilizing data from previous iterations. The algorithm equations are summarized below.

If desired signal and input-signal vector at iteration k are denoted by $d(k)$ and $\mathbf{x}(k)$, respectively, for N data reuses the coefficients are updated as

$$\mathbf{w}_{i+1}(k) = \mathbf{w}_i(k) + \frac{e_i(k)}{\|\mathbf{x}(k-i)\|^2 + \epsilon}\mathbf{x}(k-i) \tag{1}$$

0-7803-5682-9/99/$10.00©1999 IEEE.

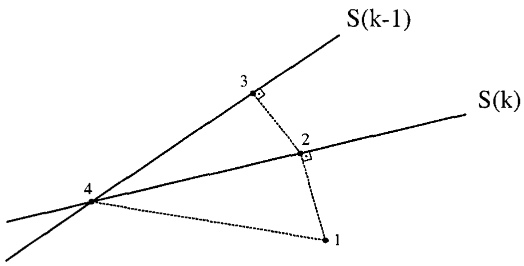

Figure 1: Coefficient vector update:

Position 1. $\mathbf{w}(k)$;
Position 2. $\mathbf{w}_{NLMS}(k+1)$ and first step towards $\mathbf{w}_{NNDR-LMS}(k+1)$;
Position 3. $\mathbf{w}_{NNDR-LMS}(k+1)$;
Position 4. $\mathbf{w}_{BNDR-LMS}(k+1)$ and $\mathbf{w}_{AP}(k+1)$.

for $i = 0, \ldots, N$, where

$$e_i(k) = d(k) - \mathbf{x}^T(k)\mathbf{w}_i(k) \quad (2)$$

$$\mathbf{w}_0(k) = \mathbf{w}_{NNDR-LMS}(k) \quad (3)$$

and

$$\mathbf{w}_{NNDR-LMS}(k+1) = \mathbf{w}_{N+1}(k) \quad (4)$$

For every intermediate update, i, the algorithm takes a normalized step towards the closest point belonging to the hyperplane defined by the data pair $[d(k-i),\ \mathbf{x}(k-i)]$ (see Fig.1). The algorithm is indeed a data-reusing algorithm, but normalization with respect to different hyperplanes implies nonorthogonal projections.

2.2. The BNDR-LMS Algorithm

The BNDR-LMS algorithm extends the idea of the NNDR-LMS algorithm for the particular case of two data reuses. The algorithm also applies normalization with respect to different hyperplanes defined by previous data pairs. However, the normalized update is taken towards the point belonging to the intersection of hyperplanes defined by the present and the previous data pairs. In this case, directions are given by orthogonal projections (see Fig.1).

The coefficients are updated as follows:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \mu\left[\frac{\lambda_1}{2}\mathbf{x}(k) + \frac{\lambda_2}{2}\mathbf{x}(k-1)\right] \quad (5)$$

where μ is the step-size,

$$\frac{\lambda_1}{2} = \frac{[d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k)]||\boldsymbol{x}(k-1)||^2}{||\boldsymbol{x}(k)||^2||\boldsymbol{x}(k-1)||^2 - [\boldsymbol{x}^T(k)\boldsymbol{x}(k-1)]^2} - \frac{[d(k-1) - \boldsymbol{x}^T(k-1)\boldsymbol{w}(k)]\boldsymbol{x}^T(k-1)\boldsymbol{x}(k)}{||\boldsymbol{x}(k)||^2||\boldsymbol{x}(k-1)||^2 - [\boldsymbol{x}^T(k)\boldsymbol{x}(k-1)]^2} \quad (6)$$

and

$$\frac{\lambda_2}{2} = \frac{[d(k-1) - \boldsymbol{x}^T(k-1)\boldsymbol{w}(k)]||\boldsymbol{x}(k)||^2}{||\boldsymbol{x}(k)||^2||\boldsymbol{x}(k-1)||^2 - [\boldsymbol{x}^T(k)\boldsymbol{x}(k-1)]^2} - \frac{[d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k)]\boldsymbol{x}^T(k-1)\boldsymbol{x}(k)}{||\boldsymbol{x}(k)||^2||\boldsymbol{x}(k-1)||^2 - [\boldsymbol{x}^T(k)\boldsymbol{x}(k-1)]^2} \quad (7)$$

If we acknowledge that optimality with respect to the previous data pair is carried on to the next iteration ($\mu = 1$), i.e., $\mathbf{x}^T(k-1)\mathbf{w}(k) = d(k-1)$, then a simplified version of the algorithm may result as follows:

$$\mathbf{w}(k+1) = \mathbf{w}(k) + \frac{\lambda_1'}{2}\mathbf{x}(k) + \frac{\lambda_2'}{2}\mathbf{x}(k-1) \quad (8)$$

where

$$\frac{\lambda_1'}{2} = \frac{[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]||\mathbf{x}(k-1)||^2}{||\mathbf{x}(k)||^2||\mathbf{x}(k-1)||^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2} \quad (9)$$

and

$$\frac{\lambda_2'}{2} = \frac{-[d(k) - \mathbf{x}^T(k)\mathbf{w}(k)]\mathbf{x}^T(k-1)\mathbf{x}(k)}{||\mathbf{x}(k)||^2||\mathbf{x}(k-1)||^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2} \quad (10)$$

2.3. The AP Algorithm

The AP algorithm relates to the NLMS and BNDR-LMS algorithms directly as it is a generalization for N data reuses of an algorithm that yields at every iteration k a solution that belongs to the intersection of hyperplanes defined by the present and all N previous data pairs. The coefficients are updated as follows:

$$\mathbf{w}(k+1) = \mathbf{w}(k-1) + \mu\mathbf{X}(k)\mathbf{t}(k) \quad (11)$$

where

$$\mathbf{t}(k) = \left[\mathbf{X}^T(k)\mathbf{X}(k) + \delta\mathbf{I}\right]^{-1}\mathbf{e}(k) \quad (12)$$

$$\mathbf{e}(k) = \mathbf{d}(k) - \mathbf{X}^T(k)\mathbf{w}(k) \quad (13)$$

and, for $L = N+1$ projections, the desired-signal vector and input-signal matrix are, respectively,

$$\mathbf{d}(k) = \begin{bmatrix} d(k) \\ d(k-1) \\ \vdots \\ d(k-L+1) \end{bmatrix} \quad (14)$$

and

$$\mathbf{X}(k) = \begin{bmatrix}\mathbf{x}(k) & \mathbf{x}(k-1) & \cdots & \mathbf{x}(k-L+1)\end{bmatrix} \quad (15)$$

with $\mathbf{x}(k)$ denoting the input-signal vector, i.e.,

$$\mathbf{x}(k) = \begin{bmatrix}x(k) & x(k-1) & \cdots & x(k-M+1)\end{bmatrix}^T \quad (16)$$

3. THE AP ALGORITHM FOR TWO PROJECTIONS

In this section the AP-algorithm update equations are rewritten for the particular case of $L = 2$. It is shown that the BNDR-LMS algorithm is the AP algorithm for the particular case of two projections and $\delta = 0$, as the NLMS algorithm is the AP algorithm for the particular case of one projection.

For $L = 2$, then (14) and (15) become, respectively,

$$\mathbf{d}(k) = \begin{bmatrix} d(k) \\ d(k-1) \end{bmatrix} \tag{17}$$

and

$$\mathbf{X}(k) = \begin{bmatrix} \mathbf{x}(k) & \mathbf{x}(k-1) \end{bmatrix} \tag{18}$$

Therefore, (13) is given by

$$\mathbf{e}(k) = \begin{bmatrix} e(k) \\ \varepsilon(k-1) \end{bmatrix} \tag{19}$$

where $\varepsilon(k-1)$ denotes the *a posteriori* error at iteration $k-1$, i.e.,

$$\varepsilon(k-1) = d(k-1) - \mathbf{x}^T(k-1)\mathbf{w}(k) \tag{20}$$

For $\delta = 0$, $\mathbf{t}(k)$ becomes

$$\begin{aligned} \mathbf{t}(k) &= \begin{bmatrix} \|\mathbf{x}(k)\|^2 & \mathbf{x}^T(k)\mathbf{x}(k-1) \\ \mathbf{x}^T(k)\mathbf{x}(k-1) & \|\mathbf{x}(k-1)\|^2 \end{bmatrix}^{-1} \mathbf{e}(k) \\ &= \frac{1}{\Delta(k)} \begin{bmatrix} \|\mathbf{x}(k-1)\|^2 & -\mathbf{x}^T(k)\mathbf{x}(k-1) \\ -\mathbf{x}^T(k)\mathbf{x}(k-1) & \|\mathbf{x}(k)\|^2 \end{bmatrix} \mathbf{e}(k) \\ &= \frac{1}{\Delta(k)} \begin{bmatrix} e(k)\|\mathbf{x}(k-1)\|^2 - \varepsilon(k-1)\mathbf{x}^T(k)\mathbf{x}(k-1) \\ \varepsilon(k-1)\|\mathbf{x}(k)\|^2 - e(k)\mathbf{x}^T(k)\mathbf{x}(k-1) \end{bmatrix} \end{aligned} \tag{21}$$

where

$$\Delta(k) = \|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2 - [\mathbf{x}^T(k)\mathbf{x}(k-1)]^2 \tag{22}$$

and (11)–(16) may be rewritten as (5)–(7).

Extending the results for the case $L = 3$ is trivial and would yield a trinormalized data-reusing LMS algorithm.

4. THE NNDR-LMS ALGORITHM FOR ONE DATA REUSE

In the previous section we showed that the AP-algorithm update equations can be rewritten such that the BNDR-LMS algorithm update equations result. In this section we show that the NNDR-LMS algorithm update equations can be rewritten in order to establish the relationship between the NNDR-LMS and AP algorithms.

The coefficient-update equations for the NNDR-LMS algorithm may be written in the same format as (11)–(16). For the particular case of $N = 1$, i.e., one data reuse, we have

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \mathbf{X}(k)\frac{1}{\|\mathbf{x}(k)\|^2\|\mathbf{x}(k-1)\|^2} \\ &\times \begin{bmatrix} \|\mathbf{x}(k-1)\|^2 & 0 \\ -\mathbf{x}^T(k)\mathbf{x}(k-1) & \|\mathbf{x}(k)\|^2 \end{bmatrix} \begin{bmatrix} e(k) \\ \varepsilon(k-1) \end{bmatrix} \end{aligned} \tag{23}$$

or, equivalently,

$$\begin{aligned} \mathbf{w}(k+1) &= \mathbf{w}(k) + \mathbf{X}(k) \\ &\times \begin{bmatrix} \|\mathbf{x}(k)\|^2 & 0 \\ \mathbf{x}^T(k)\mathbf{x}(k-1) & \|\mathbf{x}(k-1)\|^2 \end{bmatrix}^{-1} \begin{bmatrix} e(k) \\ \varepsilon(k-1) \end{bmatrix} \end{aligned} \tag{24}$$

Note that the equivalent to the projections matrix in this case is a matrix which is not symmetric, as in the AP algorithm. Extension to the case of $N = 2$, i.e., two data reuses, is trivial and yields the following update equation for the coefficient vector:

$$\begin{aligned} &\mathbf{w}(k+1) = \mathbf{w}(k) + \mathbf{X}(k) \\ &\times \begin{bmatrix} \|\mathbf{x}(k)\|^2 & 0 & 0 \\ \mathbf{x}^T(k)\mathbf{x}(k-1) & \|\mathbf{x}(k-1)\|^2 & 0 \\ \mathbf{x}^T(k)\mathbf{x}(k-2) & \mathbf{x}^T(k-1)\mathbf{x}(k-2) & \|\mathbf{x}(k-2)\|^2 \end{bmatrix}^{-1} \\ &\times \begin{bmatrix} e(k) \\ \varepsilon(k-1) \\ \varepsilon(k-2) \end{bmatrix} \end{aligned} \tag{25}$$

where

$$\varepsilon(k-2) = d(k-2) - \mathbf{x}^T(k-2)\mathbf{w}(k) \tag{26}$$

Note also that the projection matrix (25) that shall be inverted is regular with respect to order update, therefore it can be efficiently inverted if a large number of reuses is utilized. In that case, however, it may be more advantageous to employ different projection matrices, e.g., the tridiagonal correlation matrix shown below

$$\hat{R}(k) = \mathbf{M}^{-1} \begin{bmatrix} e(k) \\ \varepsilon(k-1) \\ \varepsilon(k-2) \end{bmatrix} \tag{27}$$

with M equal to

$$\begin{bmatrix} \|\mathbf{x}(k)\|^2 & \mathbf{x}^T(k)\mathbf{x}(k-1) & 0 \\ \mathbf{x}^T(k)\mathbf{x}(k-1) & \|\mathbf{x}(k-1)\|^2 & \mathbf{x}^T(k-1)\mathbf{x}(k-2) \\ 0 & \mathbf{x}^T(k-1)\mathbf{x}(k-2) & \|\mathbf{x}(k-2)\|^2 \end{bmatrix}$$

The advantages and implementation of the algorithm employing the projection matrix proposed in (27) are under investigation.

5. COMPUTATIONAL COMPLEXITY

As far as computational complexity is concerned, there are several possible implementations of the above-mentioned algorithms, especially if fast implementations and efficient inversion techniques of the projection matrix $\hat{R}(k)$ are considered. In Table I the computational complexity of the NNDR-LMS, BNDR-LMS, and AP algorithms is compared. In this case the AP algorithm was considered in its conventional form, but employing the Levinson method for matrix inversion.

TABLE I
COMPARISON OF COMPUTATIONAL COMPLEXITY

ALG.	MULT.
NNDR-LMS	6p
BNDR-LMS	6p+8
AP ($L = 2$)	4p+28

The computational complexity of the AP algorithm was presented in [11] as $2pL + K_{inv}L^2$. In order to obtain the above value K_{inv} was made equal to 7 (assuming the use of the generalized Levinson algorithm) and $L = 2$. In this same reference another efficient implementation of the AP algorithm was proposed.

6. CONCLUSIONS

The relationship between data-reusing LMS algorithms and the affine-projections algorithm was clarified. By means of a structured approach based on the AP algorithm equations, possible alternative implementation of the DR-LMS algorithms were suggested and new algorithms that are possibly faster than the NLMS algorithm and yet less complex than the NNDR-LMS and AP algorithms were also discussed. Computational complexity and different possible implementations of the algorithms were also addressed.

7. REFERENCES

[1] B. A. Schnaufer, *Practical Techniques for Rapid and Reliable Real-Time Adaptive Filtering*. PhD thesis, University of Illinois at Urbana-Champaign, Urbana-Champaign, USA, 1995.

[2] W. K. Jenkins, A. W. Hull, J. C. Strait, B. A. Schnaufer, and X. Li, *Advanced Concepts in Adaptive Signal Processing*. Norwell, MA: Kluwer Academic Publishers, 1996.

[3] J. A. Apolinário Jr., M. L. R. de Campos, and P. S. R. Diniz, "The binormalized data-reusing LMS algorithm," *submitted to IEEE Trans. on Signal Processing*, 1998.

[4] J. A. Apolinário Jr., M. L. R. de Campos, P. S. R. Diniz, and T. I. Laakso, "The binormalized data-reusing LMS algorithm with optimized step-size sequence," *Revista da Sociedade Brasileira de Telecomunicações,* special issue dedicated to the XV Simpósio Brasileiro de Telecomunicações.

[5] J. A. Apolinário Jr., M. L. R. de Campos, and P. S. R. Diniz, "The binormalized data-reusing LMS algorithm," in *Proc. XV Simpósio Brasileiro de Telecomunicações*, Recife, Brazil, pp. 77–80, 1997.

[6] J. A. Apolinário Jr., M. L. R. de Campos, and P. S. R. Diniz, "Convergence analysis of the binormalized data-reusing LMS algorithm," in *Proc. European Conference on Circuit Theory and Design*, Budapest, Hungary, pp. 972–977, 1997.

[7] M. L. R. de Campos, J. A. Apolinário Jr., and P. S. R. Diniz, "Mean-squared error analysis of the binormalized data-reusing LMS algorithm using a discrete-angular-distribution model for the input signal," in *Proc. International Conference of Acoustics, Speech, and Signal Processing*, Seattle, USA, pp. 1677–1680, May 1998.

[8] M. L. R. de Campos, J. A. Apolinário Jr., and P. S. R. Diniz, "Colored input-signal analysis of normalized data-reusing LMS algorithms," in *Proc. IEEE International Conference on Electronics, Circuits, and Systems*, Lisbon, 1998.

[9] G. C. Goodwin and S. K. Sin, *Adaptive Filtering Prediction and Control*. New Jersey: Prentice-Hall, Englewood-Cliffs, 1984.

[10] S. L. Gay, *Fast Projection Algorithms with Application to Voice Echo Cancellation*. PhD thesis, The State University of New Jersey, New Brunswick, New Jersey, USA, Oct. 1994.

[11] S. L. Gay and S. Tavathia, "The fast affine projection algorithm," in *Proc. International Conference of Acoustics, Speech, and Signal Processing*, Detroit, USA, pp. 3023–3026, 1995.