CONVERGENCE ANALYSIS OF THE BINORMALIZED DATA-REUSING LMS ALGORITHM

J. A. Apolinário Jr.[1,2] M. L. R. de Campos[1] P. S. R. Diniz[2]

[1] *Instituto Militar de Engenharia, Rio de Janeiro, RJ, Brazil, campos@aquarius.ime.eb.br*
[2] *COPPE/UFRJ, Rio de Janeiro, RJ, Brazil, apolin@coe.ufrj.br and diniz@coe.ufrj.br*

Abstract — A binormalized data-reusing least mean squares (BNDR-LMS) adaptive filtering algorithm has been recently proposed and has presented good results, particularly when the input signal is highly correlated. Simulations have shown that with a computational complexity only slightly higher than that of other normalized data-reusing LMS algorithms, great improvement in convergence speed can be achieved for a similar excess of mean squared error after convergence. In this paper, the BNDR-LMS algorithm is analyzed and conditions for convergence of the mean of the coefficient vector are established. A comparative study is carried out where the figures of merit of interest are convergence speed, excess of mean squared error, and computational complexity. Simulations are presented where the algorithm performance is confronted with the performance of other normalized data-reusing and the normalized LMS algorithms. Conclusions stating advantages of using the BNDR-LMS algorithm are also provided.

I. INTRODUCTION

The least mean squares (LMS) algorithm is very popular and has been widely used due to its simplicity. Its convergence speed, however, is highly dependent on the eigenvalue spread of the input-signal autocorrelation matrix (conditioning number) [1, 2]. Alternative schemes which try to improve this performance at the cost of minimum additional computational complexity have been proposed and extensively discussed in the past [1, 3, 4].

The data-reusing LMS (DR-LMS) algorithm [3], which uses current desired and input signals repeatedly within each iteration is one among such schemes. It can be easily shown that in the limit of infinite data reuses per iteration the DR-LMS and the normalized LMS (NLMS) algorithms yield the same solution [5]. Performance can be further improved with the recently proposed normalized and unnormalized new data-reusing LMS (NNDR-LMS and UNDR-LMS) algorithms [6].

These algorithms reuse the data pair, namely desired and input signals, from previous iterations as well.

In reference [6], a graphical description of NNDR-LMS and UNDR-LMS algorithms was presented and it was shown that this new class of data-reusing algorithms had prospective better performance than the NLMS algorithm in terms of convergence rate. The geometric description also showed why improvement is achieved when the number of reuses is increased. The new binormalized data-reusing LMS (BNDR-LMS) algorithm introduced in [7] employs normalization on two orthogonal directions obtained from consecutive data pairs within each iteration.

In all simulations carried out with colored input signals, the new algorithm presented faster convergence than all other algorithms mentioned above (case of two data pairs).

This paper is organized as follows. Section 2 summarizes the LMS-like algorithms as well as a graphical illustration of their coefficient updating. Section 3 presents the BNDR-LMS algorithm. In Section 4 the convergence analysis is addressed. Section 5 contains the simulation results and Section 6 draws some conclusions.

II. LMS, DR-LMS, NLMS AND NDR-LMS ALGORITHMS

For the LMS algorithm, the coefficient vector $\boldsymbol{w}$ is updated in the opposite direction of the gradient vector ($\nabla_w[\cdot]$) obtained from instantaneous squared output error, i.e.,

$$\boldsymbol{w}_{LMS}(k+1) = \boldsymbol{w}_{LMS}(k) - \mu\nabla_w[e^2(k)] \qquad (1)$$

where

$$e(k) = d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}_{LMS}(k) \qquad (2)$$

is the output error, $d(k)$ is the desired signal, $\boldsymbol{x}(k)$ is the input-signal vector containing the $N+1$ most recent input-signal samples, and μ is the step size. The coefficient-updating equation is

$$\boldsymbol{w}_{LMS}(k+1) = \boldsymbol{w}_{LMS}(k) + \mu e(k)\boldsymbol{x}(k) \qquad (3)$$

For the DR-LMS with L data reuses, the coefficients are updated as

$$\boldsymbol{w}_{i+1}(k) = \boldsymbol{w}_i(k) + \mu e_i(k)\boldsymbol{x}(k) \quad \textbf{(4)}$$

for $i = 0, \ldots, L$; where

$$e_i(k) = d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}_i(k), \quad \textbf{(5)}$$

$$\boldsymbol{w}_0(k) = \boldsymbol{w}_{DR-LMS}(k), \quad \textbf{(6)}$$

and

$$\boldsymbol{w}_{DR-LMS}(k+1) = \boldsymbol{w}_{L+1}(k). \quad \textbf{(7)}$$

Note that if $L = 0$ these equations correspond to the LMS algorithm.

The NLMS algorithm normalizes the step-size such that the relation $\boldsymbol{x}^T(k)\boldsymbol{w}_{NLMS}(k+1) = d(k)$ is always satisfied, i.e.,

$$\boldsymbol{w}_{NLMS}(k+1) = \boldsymbol{w}_{NLMS}(k) + \frac{e(k)}{\boldsymbol{x}^T(k)\boldsymbol{x}(k) + \epsilon}\boldsymbol{x}(k) \quad \textbf{(8)}$$

where ϵ is a very small number used to avoid division by zero.

The NNDR-LMS algorithm is specified by the following relations

$$\boldsymbol{w}_{i+1}(k) = \boldsymbol{w}_i(k) + \frac{e_i(k)}{\boldsymbol{x}^T(k-i)\boldsymbol{x}(k-i) + \epsilon}\boldsymbol{x}(k-i) \quad \textbf{(9)}$$

for $i = 0, \ldots, L$; where

$$e_i(k) = d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}_i(k), \quad \textbf{(10)}$$

$$\boldsymbol{w}_0(k) = \boldsymbol{w}_{NNDR-LMS}(k), \quad \textbf{(11)}$$

and

$$\boldsymbol{w}_{NNDR-LMS}(k+1) = \boldsymbol{w}_{L+1}(k). \quad \textbf{(12)}$$

Fig. 1 illustrates geometrically the updating of the coefficient vector for a two-dimensional problem for all algorithms discussed above, starting with an arbitrary $\boldsymbol{w}(k)$. Since we are interested in comparing algorithms of similar complexity, it was considered the case of one unique reuse, i.e., $L = 1$.

Let $\mathcal{S}(k)$ denote the hyperplane which contains all vectors $\boldsymbol{w}$ such that $\boldsymbol{x}^T(k)\boldsymbol{w} = d(k)$. In a noise-free perfect-modeling situation, $\mathcal{S}(k)$ contains the optimal coefficient vector, $\boldsymbol{w}_o$. Furthermore, it can be easily shown that $\boldsymbol{x}(k)$ and, consequently, $\nabla_w[e^2(k)]$ are orthogonal to the hyperplane $\mathcal{S}(k)$.

The solution given by the DR-LMS algorithm, $\boldsymbol{w}_{DR-LMS}(k+1)$, iteratively approaches $\mathcal{S}(k)$ by following the direction given by $\boldsymbol{x}(k)$ (see 3 in Fig. 1). This solution would reach $\mathcal{S}(k)$ in the limit, as the number of data reuses goes to infinity [6]. The NLMS algorithm performs a line search to yield the solution $\boldsymbol{w}_{NLMS}(k+1) \in \mathcal{S}(k)$ in a single step (see 4 in Fig. 1).

The algorithms presented in [6] use more than one hyperplane, i.e., use previous data pairs $(\boldsymbol{x}, d)$, in order to produce a solution $\boldsymbol{w}(k+1)$ (see 5 and 6 in Fig. 1) that is closer to $\boldsymbol{w}_o$ than the solution obtained with only the current data pair $(\boldsymbol{x}(k), d(k))$. For a noise-free perfect-modeling situation, $\boldsymbol{w}_o$ is at the intersection of $N+1$ hyperplanes constructed with linearly independent input-signal vectors. In this case, the orthogonal-projections algorithm [8] yields the solution $\boldsymbol{w}_o$ in $N+1$ iterations. This algorithm may be viewed as a normalized data-reusing orthogonal algorithm which utilizes $N+1$ data pairs $(\boldsymbol{x}, d)$.

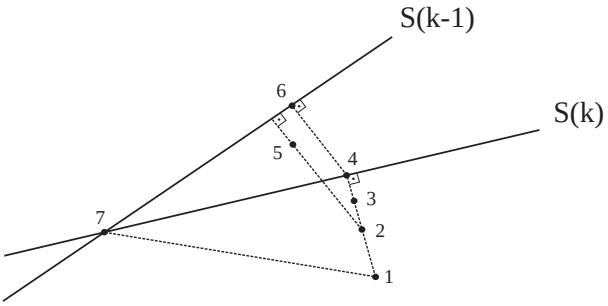

Figure 1: Updating the coefficient vector:

1. $\boldsymbol{w}(k)$;
2. $\boldsymbol{w}_{LMS}(k+1)$, first step of $\boldsymbol{w}_{DR-LMS}(k+1)$ and $\boldsymbol{w}_{UNDR-LMS}(k+1)$;
3. $\boldsymbol{w}_{DR-LMS}(k+1)$;
4. $\boldsymbol{w}_{NLMS}(k+1)$ and first step of $\boldsymbol{w}_{NNDR-LMS}(k+1)$;
5. $\boldsymbol{w}_{UNDR-LMS}(k+1)$;
6. $\boldsymbol{w}_{NNDR-LMS}(k+1)$;
7. $\boldsymbol{w}_{BNDR-LMS}(k+1)$.

In the next section, the new binormalized data-reusing LMS algorithm will be described. This algorithm combines data reusing, orthogonal projections of two consecutive gradient directions, and normalization in order to achieve faster convergence when compared to other LMS-like algorithms. At each iteration, the BNDR-LMS algorithm yields the solution $\boldsymbol{w}(k+1)$ which is at the intersection of hyperplanes $\mathcal{S}(k)$ and $\mathcal{S}(k-1)$ and at a minimum distance from $\boldsymbol{w}(k)$ (see 7 in Fig. 1). The algorithm can also be viewed as a simplified version of the orthogonal projection algorithm which utilizes just two previous consecutive directions.

III. THE BNDR-LMS ALGORITHM

In order to state the problem, we recall that the solution which belongs to $\mathcal{S}(k)$ and $\mathcal{S}(k-1)$ at a minimum

distance from $\boldsymbol{w}(k)$ is the one that solves

$$\min_{\boldsymbol{w}(k+1)} \|\boldsymbol{w}(k+1) - \boldsymbol{w}(k)\|^2 \quad (13)$$

subjected to

$$\boldsymbol{x}^T(k)\boldsymbol{w}(k+1) = d(k) \quad (14)$$

and

$$\boldsymbol{x}^T(k-1)\boldsymbol{w}(k+1) = d(k-1) \quad (15)$$

The functional to be minimized is, therefore,

$$\begin{aligned} f[\boldsymbol{w}(k+1)] = &[\boldsymbol{w}(k+1) - \boldsymbol{w}(k)]^T[\boldsymbol{w}(k+1) - \boldsymbol{w}(k)] \\ &+\lambda_1[\boldsymbol{x}^T(k)\boldsymbol{w}(k+1) - d(k)] \\ &+\lambda_2[\boldsymbol{x}^T(k-1)\boldsymbol{w}(k+1) - d(k-1)] \end{aligned} \quad (16)$$

which, for linearly independent input-signal vectors $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)$, has the unique solution

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + (-\lambda_1/2)\boldsymbol{x}(k) + (-\lambda_2/2)\boldsymbol{x}(k-1) \quad (17)$$

where

$$-\lambda_1/2 = \frac{num1}{den} \quad (18)$$

and

$$-\lambda_2/2 = \frac{num2}{den} \quad (19)$$

with:

$$\begin{aligned} num1 = &[d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k)]\boldsymbol{x}^T(k-1)\boldsymbol{x}(k-1) \\ &-[d(k-1) - \boldsymbol{x}^T(k-1)\boldsymbol{w}(k)]\boldsymbol{x}^T(k)\boldsymbol{x}(k-1) \end{aligned} \quad (20)$$

$$\begin{aligned} num2 = &[d(k-1) - \boldsymbol{x}^T(k-1)\boldsymbol{w}(k)]\boldsymbol{x}^T(k)\boldsymbol{x}(k) \\ &-[d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k)]\boldsymbol{x}^T(k-1)\boldsymbol{x}(k) \end{aligned} \quad (21)$$

$$\begin{aligned} den = &\boldsymbol{x}^T(k)\boldsymbol{x}(k)\boldsymbol{x}^T(k-1)\boldsymbol{x}(k-1) \\ &-[\boldsymbol{x}^T(k)\boldsymbol{x}(k-1)]^2 \end{aligned} \quad (22)$$

The BNDR-LMS algorithm is described by equations (17) to (22). This algorithm can be alternatively derived from a purely geometric reasoning. The first step is to reach a preliminary solution, $\boldsymbol{w}_1(k)$, which belongs to $\mathcal{S}(k)$ and is at a minimum distance from $\boldsymbol{w}(k)$. This is achieved by the NLMS algorithm starting from $\boldsymbol{w}(k)$, i.e.,

$$\boldsymbol{w}_1(k) = \boldsymbol{w}(k) + \frac{e(k)}{\boldsymbol{x}^T(k)\boldsymbol{x}(k)}\boldsymbol{x}(k) \quad (23)$$

In the second step, $\boldsymbol{w}_1(k)$ is updated in a direction orthogonal to the previous one, therefore belonging to $\mathcal{S}(k)$, until the intersection with $\mathcal{S}(k-1)$ is reached. This is achieved by the NLMS algorithm starting from $\boldsymbol{w}_1(k)$ and following the direction $\boldsymbol{x}_1^{\perp}(k)$ which is the projection of $\boldsymbol{x}(k-1)$ onto $\mathcal{S}(k)$.

$$\boldsymbol{w}(k+1) = \boldsymbol{w}_1(k) + \frac{e_1(k)}{\boldsymbol{x}_1^{\perp T}(k)\boldsymbol{x}_1^{\perp}(k)}\boldsymbol{x}_1^{\perp}(k) \quad (24)$$

where

$$\boldsymbol{x}_1^{\perp}(k) = \left[\mathbf{I} - \frac{\boldsymbol{x}(k)\boldsymbol{x}^T(k)}{\boldsymbol{x}^T(k)\boldsymbol{x}(k)}\right]\boldsymbol{x}(k-1) \quad (25)$$

and

$$e_1(k) = d(k-1) - \boldsymbol{x}^T(k-1)\boldsymbol{w}_1(k) \quad (26)$$

The use of $\boldsymbol{x}_1^{\perp}(k)$ obtained from $\boldsymbol{x}(k-1)$ assures that the minimum-distance path is chosen. Note that the requirement of linear independence of consecutive input-signal vectors $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)$, necessary to ensure existence and uniqueness of the solution, is also manifested here.

As will be seen in the simulation results (see Table 2) and is expected for the class of normalized algorithms, the excess of the mean-square error (MSE) for the BNDR-LMS algorithm as in equations (17) to (22) is close to the variance of the observation noise (supposing no modeling error). Therefore, in order to control this excess of MSE a step-size μ may be introduced. Although the maximum convergence rate is obtained with $\mu = 1$, the use of a smaller value for the step-size may be required in applications where measurement error is too high.

Recalling (25), we see that if $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)$ are linearly dependent, we cannot find $\boldsymbol{x}_1^{\perp}(k) \in \mathcal{S}(k)$. This situation will be avoided with a simple "if" test. Let us imagine that $\mathcal{S}(k)$ is parallel to $\mathcal{S}(k-1)$ in Fig. 1 which is equivalent to the rare situation where $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)$ are linearly dependent. Equation (23) can be used without any problem and $\boldsymbol{w}(k+1)$ can be made equal to $\boldsymbol{w}_1(k)$. This way we are taking an optimal step onto $\mathcal{S}(k)$ and indeed the BNDR-LMS algorithm corresponds — in this very specific case where $\boldsymbol{x}(k)$ is parallel to $\boldsymbol{x}(k-1)$ — to the NLMS algorithm. It is also correct to say that for the case where the step-size is one, $\mu = 1$, $\boldsymbol{w}(k)$ is already on the hyperplane $\mathcal{S}(k-1)$ and nothing is left to be done, i.e., $\boldsymbol{w}(k+1) = \boldsymbol{w}_1(k) = \boldsymbol{w}(k)$.

The BNDR-LMS algorithm is summarized in Table 1. A very small positive number ϵ (we used $\sigma_x^2/10^{10}$ in our experiments) is present in the algorithm to prevent division by zero in cases where the signal input vector is zero.

Table 1: The Binormalized Data-Reusing LMS Algorithm.

BNDR-LMS

ϵ = small value
for each k
{ $\boldsymbol{x}_1 = \boldsymbol{x}(k)$
$\boldsymbol{x}_2 = \boldsymbol{x}(k-1)$
$d_1 = d(k)$
$d_2 = d(k-1)$
$a = \boldsymbol{x}_1^T \boldsymbol{x}_2$
$b = \boldsymbol{x}_1^T \boldsymbol{x}_1$
$c = \boldsymbol{x}_2^T \boldsymbol{x}_2$
$d = \boldsymbol{x}_1^T \boldsymbol{w}(k)$
if $a^2 == bc$
{ $\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu(d_1 - d)\boldsymbol{x}_1/(b+\epsilon)$
}
else
{ $e = \boldsymbol{x}_2^T \boldsymbol{w}(k)$
$den = bc - a^2$
$A = (d_1 c + ea - dc - d_2 a)/den$
$B = (d_2 b + da - eb - d_1 a)/den$
$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu(A\boldsymbol{x}_1 + B\boldsymbol{x}_2)$
}
}

IV. CONVERGENCE ANALYSIS OF THE COEFFICIENT VECTOR

In this section, we assume that an unknown FIR filter with coefficient vector given by $\boldsymbol{w}_o$ is to be identified by an adaptive filter of same order, employing the BNDR-LMS algorithm. It is also assumed that the input signal and measurement noise are independent and identically distributed zero mean white noise with variances σ_x^2 and σ_n^2, respectively.

We are interested in analyzing the convergence behavior of the coefficient vector in terms of a step-size μ. Let $\Delta\boldsymbol{w}(k) = \boldsymbol{w}(k) - \boldsymbol{w}_0$ be the error in the adaptive filter coefficients as related to the ideal coefficient vector. For the BNDR-LMS algorithm as described in (17)–(19), $\Delta\boldsymbol{w}(k+1)$ is given by

$$\Delta\boldsymbol{w}(k+1) = \Delta\boldsymbol{w}(k) + \mu(num1 + num2)/den \quad \textbf{(27)}$$

Replacing *num1*, *num2* and *den* in the above equation by (20) to (22), and recalling that in our system identification problem $d(k)$ is given by $\boldsymbol{x}^T(k)\boldsymbol{w}_0 + n(k)$, it is easy to find that

$$\Delta\boldsymbol{w}(k+1) = [\boldsymbol{I} + \mu\ term1]\Delta\boldsymbol{w}(k) + \mu\ term2 \quad \textbf{(28)}$$

where

$$\begin{aligned} term1 \;=\; & -\boldsymbol{x}(k)\boldsymbol{x}^T(k)\boldsymbol{x}^T(k-1)\boldsymbol{x}(k-1)/den \\ & +\boldsymbol{x}(k)\boldsymbol{x}^T(k-1)\boldsymbol{x}^T(k)\boldsymbol{x}(k-1)/den \\ & -\boldsymbol{x}(k-1)\boldsymbol{x}^T(k-1)\boldsymbol{x}^T(k)\boldsymbol{x}(k)/den \\ & +\boldsymbol{x}(k-1)\boldsymbol{x}^T(k)\boldsymbol{x}^T(k-1)\boldsymbol{x}(k)/den \end{aligned} \quad \textbf{(29)}$$

and

$$\begin{aligned} term2 \;=\; & n(k)\boldsymbol{x}^T(k-1)\boldsymbol{x}(k-1)\boldsymbol{x}(k)/den \\ & -n(k-1)\boldsymbol{x}^T(k)\boldsymbol{x}(k-1)\boldsymbol{x}(k)/den \\ & +n(k-1)\boldsymbol{x}^T(k)\boldsymbol{x}(k)\boldsymbol{x}(k-1)/den \\ & -n(k)\boldsymbol{x}^T(k-1)\boldsymbol{x}(k)\boldsymbol{x}(k-1)/den \end{aligned} \quad \textbf{(30)}$$

The expected value of this error vector is given by

$$E[\Delta\boldsymbol{w}(k+1)] = E[\Delta\boldsymbol{w}(k)] + \mu E[term1 + term2] \quad \textbf{(31)}$$

Since $n(k)$ and $x(k)$ are samples from independent random processes, then $E[term2] = 0$. Moreover, if we replace (29) in (31), we will see that $\mu\ E[term1]$ has four expressions which will be named $exp.1$ to $exp.4$. In order to simplify these expressions, the following relations will be used:

$$E[\boldsymbol{x}^T(k-i)\boldsymbol{x}(k-i)] = (N+1)\sigma_x^2 \quad \textbf{(32)}$$

$$E[(\boldsymbol{x}^T(k)\boldsymbol{x}(k-1))^2] = (N+1)(\sigma_x^2)^2 \quad \textbf{(33)}$$

$$\begin{aligned} \boldsymbol{x}^T(k-1)\Delta\boldsymbol{w}(k) \;=\; & (1-\mu)\boldsymbol{x}^T(k-1)\Delta\boldsymbol{w}(k-1) \\ & +\mu n(k-1) \end{aligned} \quad \textbf{(34)}$$

The expressions above can be easily obtained with the use of the independence assumption [1] and the BNDR-LMS algorithm equations.

If we assume that $\boldsymbol{R} = \sigma_x^2\boldsymbol{I}$, $\Delta\boldsymbol{w}(k)$ is statistically independent of $\boldsymbol{x}(k)\boldsymbol{x}^T(k)$, and that $E[\boldsymbol{v}/s] \approx E[\boldsymbol{v}]/E[s]$ for large values of N, where $\boldsymbol{v}$ is a vector and s is a scalar, the expressions $exp.1$ to $exp.4$ can be written as

$$\begin{aligned} exp.1 \;=\; & -\mu E\left[\frac{\boldsymbol{x}(k)\boldsymbol{x}^T(k)\boldsymbol{x}^T(k-1)\boldsymbol{x}(k-1)\Delta w(k)}{den}\right] \\ \approx\; & -\frac{\mu E[\boldsymbol{x}(k)\boldsymbol{x}^T(k)\Delta\boldsymbol{w}(k)]}{E[\boldsymbol{x}^T(k)\boldsymbol{x}(k) - \frac{(\boldsymbol{x}^T(k)\boldsymbol{x}(k-1))^2}{\boldsymbol{x}^T(k-1)\boldsymbol{x}(k-1)}]} \\ \approx\; & -\frac{\mu E[\Delta\boldsymbol{w}(k)]}{N} \end{aligned} \quad \textbf{(35)}$$

$$exp.2 = \mu E\left[\frac{\boldsymbol{x}(k)\boldsymbol{x}^T(k-1)\boldsymbol{x}^T(k)\boldsymbol{x}(k-1)\Delta\boldsymbol{w}(k)}{den}\right]$$

$$= \mu(1-\mu)E[\frac{\boldsymbol{x}(k)\boldsymbol{x}^T(k)\boldsymbol{x}(k-1)\boldsymbol{x}^T(k-1)\Delta\boldsymbol{w}(k-1)}{den}]$$

$$+\mu^2 E[\boldsymbol{x}(k)\boldsymbol{x}^T(k)\boldsymbol{x}(k-1)n(k-1)/den]$$

$$\approx \frac{\mu(1-\mu)\boldsymbol{R}^2 E[\Delta\boldsymbol{w}(k-1)]}{(N+1)^2(\sigma_x^2)^2-(N+1)(\sigma_x^2)^2}$$

$$= \frac{\mu(1-\mu)E[\Delta\boldsymbol{w}(k-1)]}{N(N+1)} \quad \textbf{(36)}$$

$$exp.3 = -\mu E\left[\frac{\boldsymbol{x}(k-1)\boldsymbol{x}^T(k-1)\boldsymbol{x}^T(k)\boldsymbol{x}(k)\Delta\boldsymbol{w}(k)}{den}\right]$$

$$= -\mu E\left[\frac{\boldsymbol{x}^T(k)\boldsymbol{x}(k)\boldsymbol{x}(k-1)\boldsymbol{x}^T(k-1)\Delta\boldsymbol{w}(k)}{den}\right]$$

$$\approx -\frac{\mu(1-\mu)(N+1)\sigma_x^2\boldsymbol{R}E[\Delta\boldsymbol{w}(k-1)]}{N(N+1)(\sigma_x^2)^2}$$

$$= -\frac{\mu(1-\mu)E[\Delta\boldsymbol{w}(k-1)]}{N} \quad \textbf{(37)}$$

Finally, the last expression is obtained with the already used assumptions as well as by making use of the fact that $E\left[E\left[g(\boldsymbol{x},\boldsymbol{y}) \mid \boldsymbol{x}\right]\right] = E\left[g(\boldsymbol{x},\boldsymbol{y})\right]$.

$$exp.4 = \mu E\left[\frac{\boldsymbol{x}(k-1)\boldsymbol{x}^T(k)\boldsymbol{x}^T(k-1)\boldsymbol{x}(k)\Delta\boldsymbol{w}(k)}{den}\right]$$

$$= \mu E\left[\frac{\boldsymbol{x}(k-1)\boldsymbol{x}^T(k-1)\boldsymbol{x}(k)\boldsymbol{x}^T(k)\Delta\boldsymbol{w}(k)}{den}\right]$$

$$\approx \frac{\mu E[\boldsymbol{x}(k-1)\boldsymbol{x}^T(k-1)\sigma_x^2\boldsymbol{I}\Delta\boldsymbol{w}(k)]}{E[den]}$$

$$= \frac{\mu(1-\mu)\boldsymbol{R}E[\Delta\boldsymbol{w}(k)]}{N(N+1)\sigma_x^2}$$

$$= \frac{\mu(1-\mu)E[\Delta\boldsymbol{w}(k-1)]}{N(N+1)} \quad \textbf{(38)}$$

Substituting (35)–(38) in (31),

$$E[\Delta\boldsymbol{w}(k+1)] = aE[\Delta\boldsymbol{w}(k)] + bE[\Delta\boldsymbol{w}(k-1)] \quad \textbf{(39)}$$

where $a = \frac{N-\mu}{N}$ and $b = -\frac{\mu(N-1)(1-\mu)}{N(N+1)}$.

Each component of the above vector will have a difference equation of the type $f(k) = a\ f(k-1) + b\ f(k-2) + c\ \delta(k)$ with $c = f(0)$. The system (or transfer) function is obtained by taking the $\mathcal{Z}$ transform of this difference equation.

$$F(z) = \frac{c\ z^2}{z^2 - a\ z - b} \quad \textbf{(40)}$$

By using the final value theorem[1], we can see that $f(\infty)$ tends to zero if $F(z)$ is stable. The condition of convergence is then obtained for values of the step-size μ such that the poles of $F(z)$ in (40) fall inside the unit circle. Once this is achieved, the convergence of the algorithm in the mean is guaranteed. We know that $\mu = 0$ indicates that the initialization vector $\boldsymbol{w}(0)$ will be propagated. We are then interested in finding a μ_{max} such that $F(z)$ remains stable. The poles of $F(z)$ are

$$z_{\text{poles}} = \frac{a \pm \sqrt{a^2+4b}}{2} \quad \textbf{(41)}$$

Since the term inside the square root is always positive, we have two real poles $z_1 = \frac{a+\sqrt{a^2+4b}}{2}$ and $z_2 = \frac{a-\sqrt{a^2+4b}}{2}$. Moreover, it is possible to state that $z_1 < 1$ and $z_2 > -1$ if $a^2 + 4b < (2-a)^2$ since a is always positive and smaller than one. Replacing a and b by their expressions we obtain $\mu((N-1)\mu - 2N) < 0$. Based on the assumption of a large value of N, we can state from the expression above the condition of convergence as

$$0 < \mu < 2 \quad \textbf{(42)}$$

V. SIMULATION RESULTS

In order to test the BNDR-LMS algorithm, simulations were carried out for a system identification problem. The system order was $N = 10$, the input signal was correlated noise with a conditioning number around 55 and the input signal to measurement noise ratio SNR was 150dB. The learning curves (MSE in dB) for the NLMS, the NNDR-LMS (one reuse) and the BNDR-LMS algorithms are depicted in Fig. 2, corresponding to an average of 200 realizations. It is worth mentioning that in all experiments, the step-size μ was set to 1 in order to achieve the fastest convergence rate of the BNDR-LMS algorithm.

In this example we can clearly verify the superior performance of the BNDR-LMS algorithm in terms of speed of convergence when compared to the NLMS and the NNDR-LMS algorithms (with one single reuse). Simulations for the conventional LMS algorithm and for the DR-LMS algorithm were also carried out for the same setup, but their performances were, as expected, inferior than that of the NLMS algorithm and the results were omitted from Fig. 2.

In order to test the performance of the algorithms in terms of mean-square error after convergence, we measured the excess of MSE (MSE - MSE_{min}) in dB. The MSE_{min} is the variance of the measurement noise, set equal to 10^{-6} in this experiment. The results

[1] $f(\infty) = \lim_{z\to 1}(z-1).F(z)$

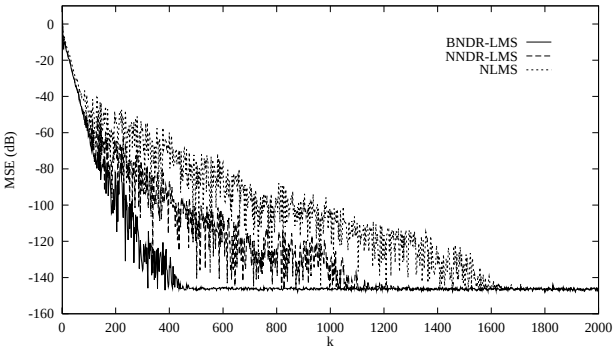

Figure 2: Learning curves of the following algorithms: NLMS, NNDR-LMS and BNDR-LMS.

are summarized in Table 2 where we can also observe the excess of MSE in dB for a nonstationary environment. In this case, measurement noise was set to zero and the system (plant) coefficients varied according to $\boldsymbol{w}(k) = \boldsymbol{w}(k-1) + \boldsymbol{v}$, where $\boldsymbol{v}$ was a random vector with zero mean and variance equal to 10^{-6}. As we can see from Table 2, in both stationary and nonstationary environments, the BNDR-LMS algorithm performed closely to NLMS and NNDR-LMS algorithms.

Table 2: Excess Mean-Square Error.

Algorithm Type	$(\text{MSE} - \text{MSE}_{min})_{dB}$	
	Stationary	Nonstationary
NLMS	-59.09	-39.15
NNDR-LMS	-59.40	-39.42
BNDR-LMS	-58.60	-39.45

In terms of computational complexity, Table 3 shows the comparisons among these three algorithms. Note that $p = N + 1$ is the number of coefficients.

Table 3: Comparison of computational complexity.

ALG.	ADD	MULT.	DIV.
NLMS	3p-1	3p	1
NNDR-LMS	6p-2	6p	2
BNDR-LMS	7p+3	7p+2	2

VI. CONCLUSIONS

This paper presented the analysis of the convergence in the mean of the BNDR-LMS algorithm and showed that the coefficient vector tends in average to the optimal coefficient vector provided that the step-size is between 0 and 2. This algorithm is also known to have faster convergence than a number of other LMS-like algorithms when the input signal is highly correlated. A geometric interpretation of the algorithm was also provided showing that the coefficients are updated in two normalized steps following orthogonal directions. The relationship between the BNDR-LMS algorithm and the orthogonal-projections algorithm was clarified. Simulations carried out in a system identification application showed that the BNDR-LMS algorithm compared favorably with other LMS-like algorithms in terms of speed of convergence. Moreover, the more correlated is the input signal, the better is the performance of the new algorithm. This improvement is more clearly observed in cases of a very small measurement noise such as the example depicted in Fig. 2.

References

[1] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*. Kluwer Academic Press, Boston, MA, USA, 1997.

[2] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, Englewood Cliffs, NJ, USA, 2nd edition, 1991.

[3] S. Roy and J. J. Shynk, "Analysis of the data-reusing LMS algorithm," *32nd Midwest Symposium Circuits and Systems*, pp. 1127-1130, Urbana-Champaign, IL, USA, 1990.

[4] D. Slock, "On the convergence behavior of the LMS and the normalized LMS algorithms," *IEEE Transactions on Signal Processing*, vol. 41, pp. 2811-2825, Sept. 1993.

[5] W. K. Jenkins, A. W. Hull, J. C. Strait, B. A. Schnaufer and X. Li, *Advanced Concepts in Adaptive Signal Processing*. Kluwer Academic Press, Boston, MA, USA, 1996.

[6] B. A. Schnaufer, *Practical Techniques for Rapid and Reliable Real-Time Adaptive Filtering*, Ph.D. Thesis, Urbana-Champaign, IL, USA, 1995.

[7] J. A. Apolinário Jr., M. L. R. de Campos and P. S. R. Diniz, "The Binormalized Data-Reusing LMS Algorithm," to be presented at the *XV Simpósio Brasileiro de Telecomunicações*, Recife, Brazil, 1997.

[8] G. C. Goodwin and K. S. Sin, *Adaptive Filtering Prediction and Control*. Prentice-Hall, Englewood Cliffs, NJ, USA, 1984.