CONSTRAINED NORMALIZED ADAPTIVE FILTERS FOR CDMA MOBILE COMMUNICATIONS

J.A. Apolinário Jr.[1,2,3*], *S. Werner*[1†], *P.S.R. Diniz*[2] and *T.I. Laakso*[1]
[1] Helsinki University of Technology, Helsinki, FINLAND
e-mails: Stefan.Werner@hut.fi and Timo.Laakso@hut.fi
[2] COPPE/Universidade Federal do Rio de Janeiro, Rio de Janeiro, BRAZIL
e-mail: diniz@lps.ufrj.br
[3] Instituto Militar de Engenharia, Rio de Janeiro, BRAZIL
e-mail: apolin@aquarius.ime.eb.br

ABSTRACT

This work presents an extension of the classical LMS-based Frost algorithm to include both the Normalized LMS (NLMS) and the Binormalized Data-Reusing LMS (BNDR-LMS) algorithms. Two simple versions of these algorithms are derived for the Frost structure. These new algorithms were applied to a DS-CDMA mobile receiver. The results showed a considerable speed up of the convergence rate compared to the LMS Frost.

1 INTRODUCTION

A constrained adaptive filter has several fields of applications such as antenna array processing and interference cancellation in direct-sequence code-division multiple access (DS-CDMA) mobile communication systems. The two traditional methods used in these applications are the so called Frost [1] approach and the general sidelobe canceler [2] (GSC) approach. The Frost scheme is probably the most widely used technique due to the simplicity of the LMS algorithm. Nevertheless, the main drawback of the LMS algorithm is also present in Frost scheme; that is, its performance depends strongly on the eigenvalue spread of the input-signal autocorrelation matrix. An alternative approach is the use of fast least-squares techniques as proposed in [3]. Since the Frost algorithm turns out to be the projection of the conventional LMS result onto a constrained hyperplane, a natural step would be to use the traditional normalized LMS-like algorithms [4, 8] followed by a projection just like in the LMS Frost case. This approach results in a superior convergence rate compared to the LMS Frost algorithm when the input signals are strongly correlated. This intuitive approach, however, lacks an optimization criterion and performs worse than the corresponding GSC structure in some cases.

It was observed in our experiments that the LMS Frost and the LMS GSC schemes give identical results [2]. This work first shows an alternative way of deriving the conventional NLMS and BNDR-LMS algorithms. The same optimization approach is then used to derive the constrained versions of these algorithms for the Frost structure.

This paper is organized as follows. Section 2 presents an alternative derivation of the conventional NLMS and BNDR-LMS algorithms. In Section 3 the constrained normalized algorithms are derived based on the approach used in the previous section. Section 4 shows some simulation results in a typical field of application, followed by conclusions.

2 DERIVATION OF THE NLMS AND THE BNDR-LMS ALGORITHMS

In this section we show an alternative way of deriving the NLMS and the BNDR-LMS algorithms. Let us start with the normalized LMS. Suppose we have an LMS-like algorithm that updates the coefficient vector according to the following expression.

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu_k \boldsymbol{x}(k) \quad (1)$$

where $\boldsymbol{w}(k)$ is the coefficient vector (of size $(N+1)\times 1$ where N is the order of the adaptive filter) at instant k, $\boldsymbol{x}(k)$ is the input signal vector and μ_k is the variable step-size (or convergence factor) which must be chosen with the objective of achieving a faster convergence. The strategy used here is to reduce the instantaneous squared error as much as possible since this is a good and simple estimate of the mean squared error (MSE) [4]. Since the instantaneous error is given by $e(k) = d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k)$[1], the instantaneous squared error at instant k after the *updating* of the coefficient vector can be written as

$$\begin{aligned} e'^2(k) &= (d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k+1))^2 \\ &= (d(k) - \boldsymbol{x}^T(k)(\boldsymbol{w}(k) + \mu_k \boldsymbol{x}(k)))^2 \end{aligned} \quad (2)$$

where the prime ($\prime$) indicates the *a posteriori* error. In order to increase the convergence rate rate by choosing an appropriate step-size, we take the partial derivative

*This work was partially supported by CAPES, Brazil.
†This work is part of a research project of the Institute of Radio Communication (IRC) funded by Technology Development Center (TEKES), NOKIA Research Center, Telecom Finland and the Helsinki Telephone Company, Finland.

[1] $d(k)$ is the desired or reference signal

of $e'^2(k)$ with respect to μ_k and make it equal to zero, obtaining

$$\mu_k = \frac{d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k)}{\boldsymbol{x}^T(k)\boldsymbol{x}(k)} \tag{3}$$

which corresponds, as expected, to the traditional normalized LMS algorithm.

In the BNDR-LMS algorithm, we update the coefficient vector by adding the input-signal vectors $\boldsymbol{x}(k)$ and $\boldsymbol{x}(k-1)$ weighted by two step-sizes, μ_{1k} and μ_{2k}, respectively.

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu_{1k}\boldsymbol{x}(k) + \mu_{2k}\boldsymbol{x}(k-1) \tag{4}$$

In this case, we minimize a cost function $F(k)$ which corresponds to the instantaneous squared error at instant k plus the instantaneous squared error at instant $k-1$ calculated with the coefficient vector of instant k or $F(k) = (d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k))^2 + (d(k-1) - \boldsymbol{x}^T(k-1)\boldsymbol{w}(k))^2$. We next define the $F'(k)$ as $F(k)$ calculated with the updated coefficient vector.

$$\begin{aligned} F'(k) &= (d(k) - \boldsymbol{x}^T(k)(\boldsymbol{w}(k) + \mu_{1k}\boldsymbol{x}(k) + \mu_{2k}\boldsymbol{x}(k-1)))^2 \\ &+(d(k-1) - \boldsymbol{x}^T(k-1)(\boldsymbol{w}(k) + \mu_{1k}\boldsymbol{x}(k) + \mu_{2k}\boldsymbol{x}(k-1)))^2 \end{aligned} \tag{5}$$

In the next step, we take the partial derivatives of $F'(k)$ with respect to μ_{1k} and μ_{2k} and make them equal to zero. After some algebraic manipulations we obtain

$$\begin{aligned} e_1 &= d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k) \\ e_2 &= d(k-1) - \boldsymbol{x}^T(k-1)\boldsymbol{w}(k) \\ den &= \boldsymbol{x}^T(k)\boldsymbol{x}(k)\boldsymbol{x}^T(k-1)\boldsymbol{x}(k-1) - (\boldsymbol{x}^T(k-1)\boldsymbol{x}(k))^2 \\ \mu_{1k} &= \frac{e_1\boldsymbol{x}^T(k-1)\boldsymbol{x}(k-1) - e_2\boldsymbol{x}^T(k-1)\boldsymbol{x}(k)}{den} \\ \mu_{2k} &= \frac{e_2\boldsymbol{x}^T(k)\boldsymbol{x}(k) - e_1\boldsymbol{x}^T(k-1)\boldsymbol{x}(k)}{den} \end{aligned} \tag{6}$$

which together with (4) correspond to the binormalized data-reusing LMS algorithm of [8].

3 THE CONSTRAINED ALGORITHMS

In linearly constrained adaptive filtering, the J constraints are represented by the following linear system.

$$\boldsymbol{C}^T\boldsymbol{w}(k) = \boldsymbol{f} \tag{7}$$

where $\boldsymbol{C}$ is a $(N+1)\times J$ matrix containing the constraint vectors, and $\boldsymbol{f}$ is a vector of J elements containing the constraint values. One single constraint means that $\boldsymbol{C}$ is a vector and $\boldsymbol{f}$ is a scalar.

In the LMS case (Frost structure), the resulting algorithm is given by the projection of the coefficient vector ($\boldsymbol{w}(k+1)$ unconstrained) onto the hyperplane defined by (7). The constrained coefficient vector is obtained by first projecting the unconstrained solution onto the homogeneous hyperplane $\boldsymbol{C}^T\boldsymbol{w}(k) = \boldsymbol{0}$ with the help of the projection matrix $\boldsymbol{P} = \boldsymbol{I} - \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T$. Finally, the resulting vector is moved back to the constraint hyperplane by adding the vector $\boldsymbol{F} = \boldsymbol{C}(\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{f}$.

$$\begin{aligned} \boldsymbol{w}(k+1) &= \boldsymbol{P}\boldsymbol{w}_{LMS}(k+1) + \boldsymbol{F} \\ &= \boldsymbol{P}[\boldsymbol{w}(k) + \mu e(k)\boldsymbol{x}(k)] + \boldsymbol{F} \end{aligned} \tag{8}$$

where $e(k) = d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k)$.

Our approach here for both the NLMS and the BNDR-LMS algorithms is the projection of the unconstrained solution followed by the optimization of the step-size(s) similar to what was done in the previous section. Let us start with the NLMS algorithm.

$$\begin{aligned} \boldsymbol{w}(k+1) &= \boldsymbol{P}\boldsymbol{w}_{NLMS}(k+1) + \boldsymbol{F} \\ &= \boldsymbol{P}[\boldsymbol{w}(k) + \mu_k\boldsymbol{x}(k)] + \boldsymbol{F} \end{aligned} \tag{9}$$

Remembering that $\boldsymbol{w}(k)$ was forced to satisfy the constraint in (7) which means that $\boldsymbol{P}\boldsymbol{w}(k) + \boldsymbol{F} = \boldsymbol{w}(k)$, it follows that (9) can be written as

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu_k\boldsymbol{P}\boldsymbol{x}(k) \tag{10}$$

Now comparing (1) and (10) we can see that they are formally equivalent if we substitute the input vector by a rotated version $\boldsymbol{x}'(k) = \boldsymbol{P}\boldsymbol{x}(k)$. Moreover, recalling that $\boldsymbol{P}^2 = \boldsymbol{P}$, it follows that

$$\begin{aligned} e(k) &= d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k) \\ \boldsymbol{w}(k+1) &= \boldsymbol{P}[\boldsymbol{w}(k) + \frac{e(k)\boldsymbol{x}(k)}{\boldsymbol{x}^T(k)\boldsymbol{P}\boldsymbol{x}(k)}] + \boldsymbol{F} \end{aligned} \tag{11}$$

which correspond to the constrained NLMS algorithm.

The same approach can be applied to the BNDR-LMS if we make

$$\begin{aligned} \boldsymbol{w}(k+1) &= \boldsymbol{P}\boldsymbol{w}_{BNDR-LMS}(k+1) + \boldsymbol{F} \\ &= \boldsymbol{P}[\boldsymbol{w}(k) + \mu_{1k}\boldsymbol{x}(k) + \mu_{2k}\boldsymbol{x}(k-1)] + \boldsymbol{F} \\ &= \boldsymbol{w}(k) + \mu_{1k}\boldsymbol{P}\boldsymbol{x}(k) + \mu_{2k}\boldsymbol{P}\boldsymbol{x}(k-1) \end{aligned} \tag{12}$$

and compare with (4). The equations of the constrained BNDR-LMS algorithm are obtained as

$$\begin{aligned} e_1 &= d(k) - \boldsymbol{x}^T(k)\boldsymbol{w}(k) \\ e_2 &= d(k-1) - \boldsymbol{x}^T(k-1)\boldsymbol{w}(k) \\ den &= \boldsymbol{x}^T(k)\boldsymbol{P}\boldsymbol{x}(k)\boldsymbol{x}^T(k-1)\boldsymbol{P}\boldsymbol{x}(k-1) - (\boldsymbol{x}^T(k-1)\boldsymbol{P}\boldsymbol{x}(k))^2 \\ \mu_{1k} &= \frac{e_1\boldsymbol{x}^T(k-1)\boldsymbol{P}\boldsymbol{x}(k-1) - e_2\boldsymbol{x}^T(k-1)\boldsymbol{P}\boldsymbol{x}(k)}{den} \\ \mu_{2k} &= \frac{e_2\boldsymbol{x}^T(k)\boldsymbol{P}\boldsymbol{x}(k) - e_1\boldsymbol{x}^T(k-1)\boldsymbol{P}\boldsymbol{x}(k)}{den} \\ \boldsymbol{w}(k+1) &= \boldsymbol{P}[\boldsymbol{w}(k) + \mu_{1k}\boldsymbol{x}(k) + \mu_{2k}\boldsymbol{x}(k-1)] + \boldsymbol{F} \end{aligned} \tag{13}$$

It is worth mentioning that these two constrained algorithms give identical results when compared to the NLMS and BNDR-LMS algorithms used in the GSC structure. It is also interesting to remark that (11) and (13) can be simplified by admitting that $\boldsymbol{P}\boldsymbol{w}(k) + \boldsymbol{F} = \boldsymbol{w}(k)$. This simplification, however, can only be used in an infinite precision environment. In a finite precision environment, round-off errors will make the solution drift away from the constraint hyper plane. To illustrate this we plot the deviation of the constraint as function of the number of iterations in a finite precision environment. Figure 1 shows the deviation from the constraint using 8 digit fixed-point arithmetic with and without the above simplification. Without the above simplification, the result is very close to zero and does not appear in the figure since it is located along the horizontal axis. As can be seen from the figure, the accumulation of round-off errors increases the deviation with time.

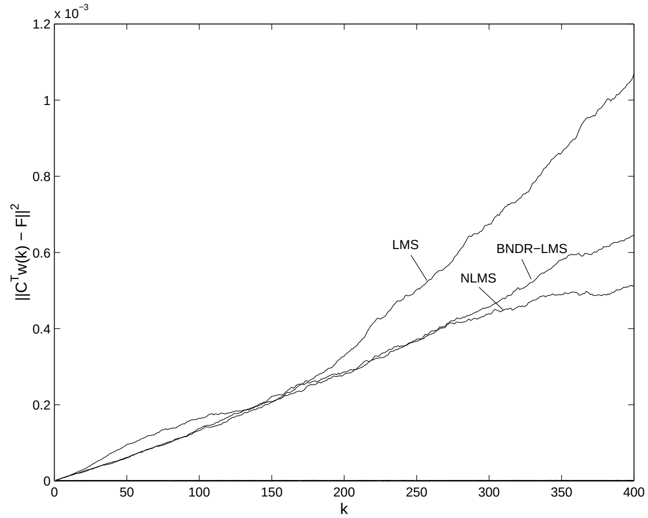

Figure 1: Deviation from constraint as function of the number of iterations curves for the NLMS algorithm.

4 SIMULATION RESULTS

In this section, we describe two experiments carried out to test the constrained algorithms derived in this paper.

4.1 Example 1

We first consider an example where the received signal consists of three sinusoids in white noise. The example is taken from [3] and the received signal is

$$\begin{aligned} u(n) &= \sin(0.3n\pi) + \sin(0.325n\pi) \\ &\quad + \sin(0.7n\pi) + r(n) \end{aligned} \tag{14}$$

where $r(n)$ is white noise with power such that the SNR is 40 dB.

The filter is constrained to pass components at frequencies $0.1\text{rad}/s$ and $0.25\text{rad}/s$ undistorted. This results in a constraint matrix given by

$$\boldsymbol{C}^T = \begin{bmatrix} 1 & \cos(0.2\pi) & \dots & \cos[(N-1)0.2\pi] \\ 1 & \cos(0.5\pi) & \dots & \cos[(N-1)0.5\pi] \\ 1 & \sin(0.2\pi) & \dots & \sin[(N-1)0.2\pi] \\ 1 & \sin(0.5\pi) & \dots & \sin[(N-1)0.5\pi] \end{bmatrix} \tag{15}$$

and

$$\boldsymbol{F}^T = [1\,1\,0\,0] \tag{16}$$

The optimum filter coefficient vector is given by

$$\boldsymbol{w}_{opt} = \begin{bmatrix} -0.4132 \\ 0.2964 \\ -1.0324 \\ -0.2535 \\ -0.5921 \\ -0.7046 \\ -0.6467 \\ -0.8854 \\ 0.2681 \\ -0.7307 \\ -0.0580 \end{bmatrix} \tag{17}$$

In this example, we compared the constrained BNDR-LMS algorithm with the constrained NLMS algorithm, and the algorithm proposed by Frost (LMS). Constant step-sizes chosen to yield the fastest convergence were used in all algorithms. For both the BNDR-LMS and the NLMS algorithms, the step-size was set to one, and for the Frost's algorithm we set $\mu = 0.1$.

The L_2 norm of the coefficient-error vectors of all algorithms are depicted in Figure 2 as the result of an averaging over 500 trials. We can clearly verify the superior performance of the BNDR-LMS algorithm as compared to the other algorithms.

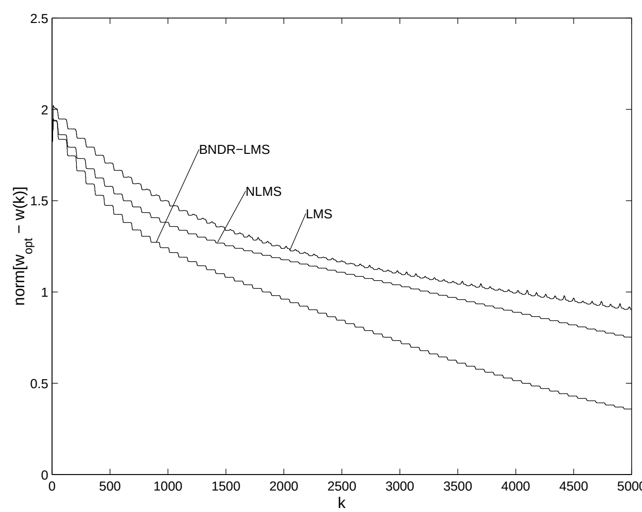

Figure 2: Coefficient-vector deviation for the first experiment.

4.2 Example 2

We next apply the constrained adaptive algorithms to the case of a DS-CDMA downlink transmission system. The received signal for a system with K simultaneous users can be written as

$$\boldsymbol{x}(k) = \sum_{i=1}^{K} A_i b_i(k) \boldsymbol{s}_i + \boldsymbol{n}(k) \tag{18}$$

where A_i is the amplitude of user i, $\boldsymbol{s}_i$ is the signature sequence of the ith user, and $b_i(k) \in \{\pm 1\}$ is the transmitted bit of the ith user. At the mobile receiver we are only interested in detecting one user (here assumed to be $i = 1$). One way of constructing the receiver coefficients is to minimize the mean variance under the constraint that the desired user's code sequence can pass with unity response. The problem is then to find a coefficient vector $\boldsymbol{w}(k)$ that solves

$$\min_{\boldsymbol{w}(k)} \|\boldsymbol{x}^T(k)\boldsymbol{w}(k)\|^2 \quad \text{subject to} \quad \boldsymbol{s}_1^T \boldsymbol{w}(k) = 1 \qquad (19)$$

where, using the notation of the previous section, we see that the reference signal $d(k) = 0$, $\boldsymbol{C} = \boldsymbol{s}_1$ and $\boldsymbol{f} = 1$. The system used in our experiment consists of $K = 5$ users whose spreading sequences were Gold codes of length 7 [6]. The signal-to-noise ratio (SNR) for the desired user was set to $8dB$ and the interfering users power was set to $10dB$ stronger than the desired user power, i.e., $10log\left(\frac{P_i}{P_1}\right) = 10$.

In the simulation, we have used time varying step-size in order to have a fast convergence rate and also a small misadjustment. The choice of the step-sizes corresponds to the optimal sequences presented in [7, 8].

Figure 3 shows the learning curves for the LMS, NLMS and BNDR-LMS algorithms (average of 500 runs). The step-size for the LMS algorithm was chosen to be $\mu = 1.10^{-3}$ so that its misadjustment is comparable with the two other algorithms where the adaptive step-size where used. As can be seen from the figure, the performance of the normalized algorithms is superior to the LMS algorithm in terms of convergence rate. Probably due to the input signal which in this example is not correlated enough, the BNDR-LMS algorithm could not have the best performance and the NLMS algorithm is to be preferred in these cases. This assertion is supported by our experience that even the RLS algorithm has not shown much better performance than the NLMS algorithm in this particular example. It is worth mentioning that the GSC structure was also simulated and presented identical learning curves.

5 CONCLUSIONS

This paper introduced the constrained NLMS and BNDR-LMS algorithms using the so-called Frost structure. A straightforward method of obtaining the normalized algorithms was presented and it was shown that this method is also valid for the constrained normalized algorithms. The resulting constrained normalized algorithms using the Frost structure presented identical results then the unconstrained counterparts using the GSC structure. The algorithms were applied to CDMA mobile reception and the simulation results showed a fast convergence rate as well as a small misadjustment when a time-varying step-size is used.

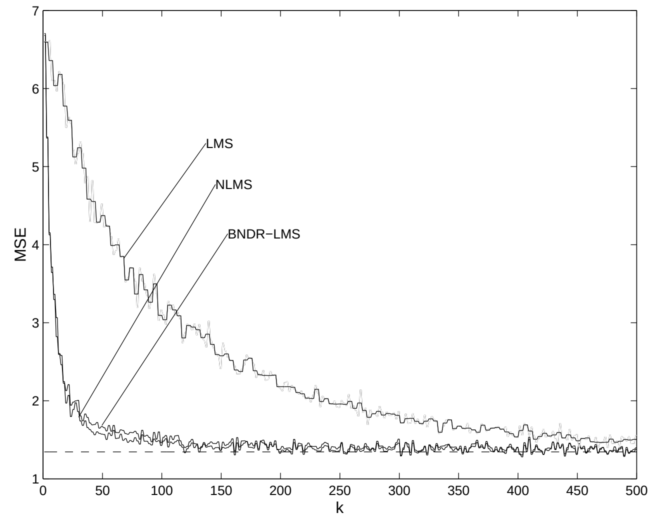

Figure 3: Learning curves of the constrained algorithms.

References

[1] O. L. Frost, III, " An algorithm for linearly constrained adaptive array processing," *Proc. IEEE*, vol. 60, no. 8, pp. 926-935, Aug. 1972.

[2] L. J. Griffiths and C. W. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Trans. on Ant. and Propagation*, vol. AP-30, no. 1, pp. 27-34, Jan. 1982.

[3] L. S. Resende, J. M. T. Romano, and M. G. Bellanger, "A fast least-squares algorithm for linearly constrained adaptive filtering," *IEEE Trans. on Signal Proc.*, vol. 44, pp. 1168-1174, May 1996.

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

[5] J. A. Apolinário Jr., M. L. R. de Campos and P. S. R. Diniz, "Convergence analysis of the Binormalized Data-Reusing LMS algorithm," Proc. of the *ECCTD* , Budapest, Hungary, 1997.

[6] C. Schlegel, S. Roy, P. D. Alexander and Z.-J. Xiang, "Multiuser projection receivers," *IEEE J. Sel. Areas Comms.*, vol. 14, no. 8, Oct. 1996.

[7] 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.

[8] J. A. Apolinário Jr., P. S. R. Diniz, T. I. Laakso and M. L. R. de Campos, "Step-size optimization of the BNDR-LMS algorithm," *EUSIPCO* , Island of Rhodes, Greece, 1998.