Journal of VLSI Signal Processing 30, 293–309, 2002

Constrained Adaptive Linear Multiuser Detection Schemes*

GEORGE V. MOUSTAKIDES
Institut National de Recherche en Informatique et en Automatique (INRIA), Rennes 35042, France;
Department of Computer Engineering and Informatics, University of Patras, Greece

Received September 1, 2000; Revised June 7, 2001

Abstract. By using a fair comparison method we show that contrary to the general belief the conventional LMS, when in training mode, does not necessarily outperform the popular blind LMS (BLMS). With the help of a constrained MMSE criterion we identify the correct trained version which is guaranteed to have uniformly superior performance over BLMS since it maximizes the SIR over an algorithmic class containing BLMS. Because the proposed optimum trained version requires knowledge of the amplitude of the user of interest we also present simple and efficient techniques that estimate the amplitude in question. The resulting algorithm in both modes, training and decision directed, is significantly superior to BLMS.

1. Introduction

Code division multiple access (CDMA) implemented with direct sequence (DS) spread spectrum signaling is a technology applied to a number of important applications nowadays such as mobile telephony, wireless networks and personal communications. In these systems, multiple access interference (MAI) becomes an intrinsic factor for severe performance degradation that necessitates the application of signal processing techniques to improve quality. Multiuser detection schemes developed over the last years, successfully mitigate MAI achieving at the same time significant capacity improvement for the corresponding CDMA systems. Due to their capability to combat MAI, multiuser detection schemes have attracted considerable attention and currently significant research is devoted in this direction [1].

Among multiuser detection schemes, the linear MMSE detector appears to be the most popular one [2–9]. This popularity mainly stems from its characteristic simplicity which is combined with excellent performance. Although the MMSE detector is not optimum from a minimum bit error rate (BER) point of view, it nevertheless optimizes a number of alternative criteria as asymptotic efficiency and near-far resistance [10, pages 195–202]. Despite its asymptotic optimality, the MMSE detector was recently found not to uniformly outperform, in the BER sense, the other two well known linear detectors, namely the conventional matched filter and the decorrelating detector [11]. This fact however does not attenuate the significance of this popular detection scheme since counterexamples seem to be possible only at extreme signaling conditions.

After its introduction [9], the usefulness of the MMSE detector was spurred with the appearance of a blind adaptive realization [12] that does not require training or knowledge of the interfering users' signature waveforms. Following this work a number of alternative blind techniques were proposed, differing mainly in the adaptive algorithm used to estimate the corresponding linear filter [3, 6, 8]. Among these versions the subspace based adaptations [6, 8], due to the special structure of the multiuser detection problem, tend to exhibit superior performance as compared to the corresponding classical adaptive schemes.

It is often stated in the literature that the most severe drawback of blind schemes is their inferior performance as compared to adaptations that use training

*This work was supported by a Collaborative Research Grant from the NATO International Scientific Exchange Programme.

[3, 12]. The first important result of this paper consists in showing that this widely accepted statement is in fact FALSE. In order for a trained algorithm to uniformly outperform its blind counterpart, as we will show, it is necessary to have also available the information of the amplitude of the user of interest. Unfortunately this type of a priori knowledge is difficult to obtain in practice, we therefore propose a simple adaptive algorithm to estimate it. This adaptation combined with the trained algorithm that estimates the linear filter, results in almost optimum performance.

The considerable difference in performance between blind and optimum trained schemes suggests that there is room for performance improvement of blind adaptations. The second main result of this work aims in this direction. Specifically we show that the decision directed version of our optimum trained scheme is extremely efficient with performance that follows closely the performance of its trained prototype. Since decision directed algorithms do not require training or knowledge of interfering users' signatures, they are obviously blind as well. Due to its excellent performance, the proposed decision directed scheme can consequently become a possible alternative to the existing popular blind algorithm of [12].

2. Signal Model and Background

Consider a K-user synchronous DS-CDMA system with identical chip waveforms and signaling antipodally through an additive white Gaussian noise (AWGN) channel. Although the signals appearing in CDMA systems are continuous in time, the system we are interested in, can be adequately modeled by an equivalent discrete time system [10]. Specifically no information is lost if we limit ourselves to the discrete time output of a chip matched filter applied to the received analog signal [10, page 310]. The resulting sequence can be represented as a collection of vectors $\mathbf{r}(n)$ of length N, with N denoting the common spreading factor of all signature waveforms. To be more precise, if σ^2 denotes the power of the AWGN we can then write[1]

$$\mathbf{r}(n) = \sum_{i=1}^{K} \mathbf{s}_i a_i b_i(n) + \sigma \mathbf{n}(n) \tag{1}$$

where $\mathbf{s}_i$ is a unit norm vector denoting the discrete time version of the signature of User-i, a_i the corresponding amplitude, $b_i(n)$ the n-th symbol of User-i and finally $\mathbf{n}(n)$ a white Gaussian noise vector with i.i.d. components of zero mean and unit variance that models the ambient noise. Using matrix notation, Eq. (1) can be transformed into

$$\mathbf{r}(n) = \mathbf{S}\mathbf{A}\mathbf{b}(n) + \sigma \mathbf{n}(n) \tag{2}$$

where $\mathbf{S} = [\mathbf{s}_1 \cdots \mathbf{s}_K]$, $\mathbf{A} = \operatorname{diag}\{a_1, \ldots, a_K\}$ and $\mathbf{b}(n) = [b_1(n) \cdots b_K(n)]^t$.

A linear detector estimates the i-th user's transmitted bits by taking the sign of the inner product of $\mathbf{r}(n)$ with a properly selected vector (filter) $\mathbf{c}$ of length N, specifically

$$\hat{b}_i(n) = \operatorname{sgn}\{\mathbf{c}^t \mathbf{r}(n)\}. \tag{3}$$

Three are the most well known linear detectors encountered in the literature, namely the conventional matched filter, the decorrelating detector (or decorrelator) and the MMSE detector which is also equivalent to the Minimum Output Energy (MOE) detector of [12]. Without loss of generality if we consider User-1 as the user of interest then the corresponding $\mathbf{c}$ filters for the three detectors take the form

$$\mathbf{c} = \begin{cases} \mathbf{s}_1 & \text{matched filter} \\ \mathbf{S}\mathbf{R}^{-1}\mathbf{e}_1 & \text{decorrelator} \\ \dfrac{\boldsymbol{\Sigma}_{\mathbf{r}}^{-1}\mathbf{s}_1}{\mathbf{s}_1^t \boldsymbol{\Sigma}_{\mathbf{r}}^{-1}\mathbf{s}_1} & \text{MMSE, MOE,} \end{cases} \tag{4}$$

where $\mathbf{R} = \mathbf{S}^t\mathbf{S}$ is the correlation matrix of the signature waveforms, $\mathbf{e}_1 = [10 \cdots 0]^t$ and $\boldsymbol{\Sigma}_{\mathbf{r}} = \mathbb{E}\{\mathbf{r}(n) \times \mathbf{r}^t(n)\} = \mathbf{S}\mathbf{A}^2\mathbf{S}^t + \sigma^2\mathbf{I}$ (with $\mathbf{I}$ the identity matrix) is the data covariance matrix. The three detectors are extensively analyzed in [10] and their relative performance considered in [11]. Notice that apart from the conventional matched filter the other two linear detectors require knowledge of all interfering users' signatures, while the MMSE requires additional knowledge of all user and noise powers.

3. A Constrained MMSE Criterion

The MMSE linear detector presented in the previous section can be obtained by minimizing the MSE between the output of the filter and the desired bit sequence, that is

$$\min_{\mathbf{c}} \mathbb{E}\big\{\big(b_1(n) - \mathbf{c}^t \mathbf{r}(n)\big)^2\big\}. \tag{5}$$

Minimizing this criterion yields the following optimum filter

$$\mathbf{c}_o = a_1 \boldsymbol{\Sigma}_{\mathbf{r}}^{-1} \mathbf{s}_1, \tag{6}$$

which is a scaled version of the filter introduced in (4). Although the two filters are not equal, they are equivalent because, when substituted in (3), produce exactly the same bit estimates. The MOE criterion on the other hand is defined as

$$\min_{\mathbf{c}} \mathbb{E}\{(\mathbf{c}^t \mathbf{r}(n))^2\}, \quad \text{subject to} \quad \mathbf{c}^t \mathbf{s}_1 = 1, \tag{7}$$

and its optimum filter is exactly the one presented in (4).

Even though the two criteria produce equivalent optimum filters, when these filters are estimated adaptively, the resulting schemes tend to differ considerably in nature and in performance. Specifically the MMSE criterion gives rise to adaptations requiring training [9] whereas the MOE results in the popular blind version of [12]. As it was stated in the introduction, contrary to the general belief, the trained version does not uniformly outperform the blind. From the analysis that follows it will become apparent that it is relatively easy to generate counterexamples (see for instance Fig. 1). Consequently this section will be devoted to the identification of the correct trained version that uniformly outperforms the blind. To achieve our goal we first need to introduce a modified MMSE criterion.

Since we detect the bit sequence $b_1(n)$ through relation (3) we can conclude that, from a detection point of view, any filter $\mathbf{c}$ is equivalent to its scaled version $\delta\mathbf{c}$ with $\delta > 0$ because

$$\hat{b}_1(n) = \operatorname{sgn}\{\mathbf{c}^t \mathbf{r}(n)\} = \operatorname{sgn}\{\delta \mathbf{c}^t \mathbf{r}(n)\}. \tag{8}$$

From the above we understand that, as far as detection is concerned, there is an ambiguity in $\mathbf{c}$ which can be eliminated by imposing a constraint on the filter. We intend to use the same constraint as the one introduced in [4], namely

$$\mathbf{c}^t \mathbf{s}_1 = 1. \tag{9}$$

With (9) we force our filter $\mathbf{c}$ to leave unchanged any information coming from the "direction of interest" $\mathbf{s}_1$. The criterion we now propose is the following

$$\min_{\mathbf{c}} \mathbb{E}\left\{\left(\alpha b_1(n) - \mathbf{c}^t \mathbf{r}(n)\right)^2\right\}, \quad \text{subject to } \mathbf{c}^t \mathbf{s}_1 = 1, \tag{10}$$

where α is a scalar parameter. In other words we are interested in minimizing the MSE between the output of the filter and a scaled version of the bit sequence. Notice that our criterion reduces to MOE when we select $\alpha = 0$.

Using Lagrange multipliers let us transform the above constrained problem into an equivalent unconstrained one. Define the function

$$\phi(\mathbf{c}) = \frac{1}{2}\left(\alpha b_1(n) - \mathbf{c}^t \mathbf{r}(n)\right)^2 - \lambda \mathbf{c}^t \mathbf{s}_1, \tag{11}$$

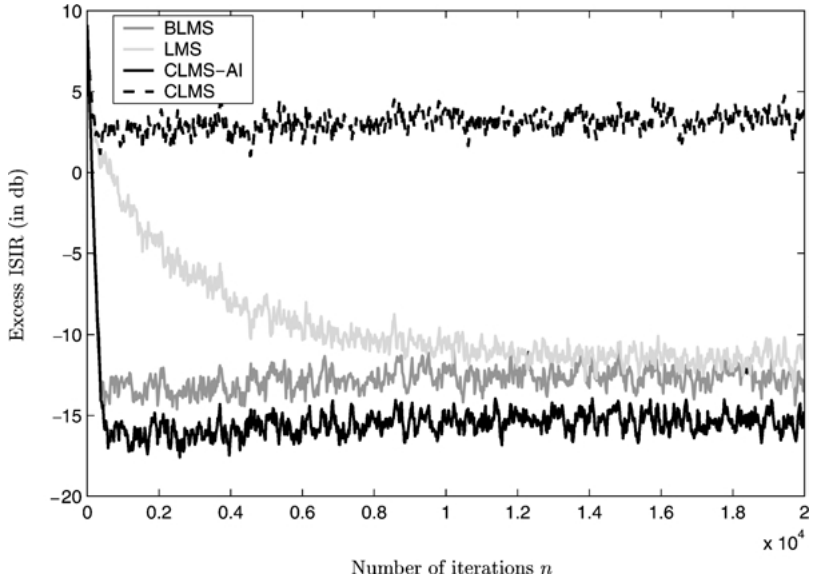

Figure 1. Performance of CLMS, CLMS-AI, BLMS and conventional LMS for $a_1 = 0.1$, $\sigma^2 = 0.01$.

then the solution to (10) can be obtained by solving

$$\min_{\mathbf{c}} \mathbb{E}\{\phi(\mathbf{c})\}, \tag{12}$$

with λ the necessary Lagrange multiplier. It is quite easy to verify that the optimum filter satisfies

$$\mathbf{c}_o = \frac{\boldsymbol{\Sigma}_{\mathbf{r}}^{-1}\mathbf{s}_1}{\mathbf{s}_1^t\boldsymbol{\Sigma}_{\mathbf{r}}^{-1}\mathbf{s}_1}, \tag{13}$$

which is an expression independent of the parameter α and equal to the MMSE and MOE optimum filter introduced in (4).

3.1. Constrained Adaptations

If the statistics of the processes involved in the minimization problem defined in (12) are not known, it is still possible to obtain the optimum solution using stochastic gradient techniques. A stochastic gradient algorithm that solves (12) can be defined by the following recursion [10, pages 306–308]

$$\mathbf{c}(n) = \mathbf{c}(n-1) - \mu\nabla_{\mathbf{c}}\phi(\mathbf{c}(n-1)), \tag{14}$$

where $\mu > 0$ a positive constant known as step size and $\phi(\mathbf{c})$ is defined in (11). Using (14) in (11) generates an LMS like adaptation of the form

$$\epsilon(n) = \alpha b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{15}$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu(\epsilon(n)\mathbf{r}(n) + \lambda\mathbf{s}_1), \tag{16}$$

that can be generalized to the following richer algorithmic class

$$\epsilon(n) = \alpha b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{17}$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu\mathbf{Q}(n)(\epsilon(n)\mathbf{r}(n) + \lambda\mathbf{s}_1), \tag{18}$$

with $\mathbf{Q}(n)$ a sequence of nonsingular matrices that can depend on the data. We can now identify the Lagrange multiplier λ by enforcing validity of the constraint at every time step n, that is, $\mathbf{s}_1^t\mathbf{c}(n) = \mathbf{s}_1^t\mathbf{c}(n-1) = \cdots = 1$. Indeed if we multiply (18) from the left by $\mathbf{s}_1^t$ and require $\mathbf{s}_1^t\mathbf{c}(n) = \mathbf{s}_1^t\mathbf{c}(n-1) = 1$ we obtain

$$\lambda = -\epsilon(n)\mathbf{Q}(n)\frac{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{r}(n)}{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{s}_1} \tag{19}$$

which if substituted in (18) yields

$$\epsilon(n) = \alpha b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{20}$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu\epsilon(n)\mathbf{Q}(n) \times \left(\mathbf{r}(n) - \frac{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{r}(n)}{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{s}_1}\mathbf{s}_1\right). \tag{21}$$

When $\mathbf{Q}(n) = \mathbf{I}$, the algorithm reduces to a *constrained LMS* version whereas $\mathbf{Q}(n) = \boldsymbol{\Sigma}_{\mathbf{r}}^{-1}(n)$, with $\boldsymbol{\Sigma}_{\mathbf{r}}(n) = (1-\mu)\boldsymbol{\Sigma}_{\mathbf{r}}(n-1) + \mu\mathbf{r}(n)\mathbf{r}^t(n)$ the (exponentially weighted) sample covariance matrix of the data, leads to a *constrained RLS* version.

3.2. Robust Constrained Adaptations

It is known that the recursion defined in (20), (21) exhibits instability under finite precision (nonrobustness) which eventually results in useless filter estimates [7, 10, page 320]. In order to correct this serious handicap, in [7] the original constrained minimization problem is transformed into an equivalent unconstrained one by enforcing (9) directly onto the filter elements. Then the application of (14) gives rise to robust adaptations. We would like here to propose an alternative method which achieves robustness by slightly modifying the recursion in (21).

If we make a first order perturbation analysis to identify the error accumulation mechanism in (21) we can show that, if (21) is stable under infinite precision then, the linear system that describes the error accumulation exhibits instability only along $\mathbf{s}_1$. By perturbing (21) we can write

$$\epsilon(n) = \alpha b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{22}$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu\epsilon(n)\mathbf{Q}(n) \times \left(\mathbf{r}(n) - \frac{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{r}(n)}{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{s}_1}\mathbf{s}_1\right) + \text{rounding errors} \tag{23}$$

with all operations performed now in infinite precision. In other words rounding errors are modeled as additive (white) noise. Multiplying the last equation from the left by $\mathbf{s}_1^t$ and subtracting from unity yields

$$1 - \mathbf{s}_1^t\mathbf{c}(n) = 1 - \mathbf{s}_1^t\mathbf{c}(n-1) + \text{rounding errors}. \tag{24}$$

The quantity $1 - \mathbf{s}_1^t\mathbf{c}(n)$ measures the accumulated finite precision error along direction $\mathbf{s}_1$ since it is zero under infinite precision (because of the constraint). The divergence of (24) towards infinity is of a random walk type because, for sufficiently high accuracy, rounding

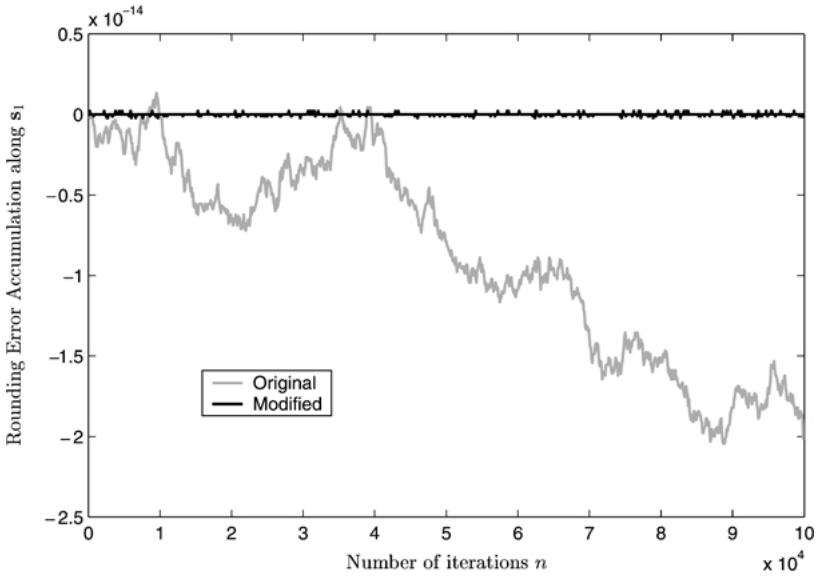

Figure 2. Rounding error accumulation along direction $\mathbf{s}_1$ for original and modified algorithm.

errors tend to be independent. Of course, due to the small variance of the rounding errors, the divergence is in fact be very slow.

Having identified the form of instability in our recursion we can proceed with the modification of the algorithm in order to correct its non-robustness. If we return to the identification of the Lagrange multiplier in (19), we recall that λ was computed by assuming $\mathbf{s}_1^t\mathbf{c}(n) = \mathbf{s}_1^t\mathbf{c}(n-1) = 1$. However, due to rounding errors, the second equality is clearly false. Taking this fact into account and recomputing the Lagrange multiplier we obtain

$$\lambda = -\epsilon(n)\mathbf{Q}(n)\frac{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{r}(n)}{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{s}_1} + \big(1 - \mathbf{s}_1^t\mathbf{c}(n-1)\big)\frac{1}{\mu\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{s}_1}. \tag{25}$$

Substituting in (18) results in the following modified version

$$\epsilon(n) = \alpha b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{26}$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu\epsilon(n)\mathbf{Q}(n) \times \left(\mathbf{r}(n) - \frac{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{r}(n)}{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{s}_1}\mathbf{s}_1\right) + \big(1 - \mathbf{s}_1^t\mathbf{c}(n-1)\big)\frac{\mathbf{Q}(n)\mathbf{s}_1}{\mathbf{s}_1^t\mathbf{Q}(n)\mathbf{s}_1}. \tag{27}$$

Using again first order perturbation analysis one can verify that $1 - \mathbf{s}_1^t\mathbf{c}(n)$ is no longer accumulating rounding errors therefore the proposed modification is robust.

Another notable property of the recursion in (27) is the fact that, if for some reason the product $\mathbf{s}_1^t\mathbf{c}(n-1)$ differs significantly from unity, (27) enforces validity of the constraint in a single step. In Fig. 2 we present a simulation of the constrained LMS algorithm (i.e. $\mathbf{Q}(n) = \mathbf{I},\ \alpha = 1$) with and without the modification. We observe that the error $1 - \mathbf{s}_1^t\mathbf{c}(n)$ of the original unmodified algorithm increases continuously whereas in the modified version it remains bounded. Although the instability appears to be extremely slow, we should bear in mind that the simulation was performed with Matlab's high accuracy computations. In a less accurate environment this instability would have been more pronounced.

3.3. *Algorithms of Interest*

From now on, for simplicity, we will limit our presentation to the LMS like algorithmic class corresponding to $\mathbf{Q}(n) = \mathbf{I}$; generalization to other $\mathbf{Q}(n)$ matrices is straightforward. Let us present the special form of the LMS like recursion. By substituting $\mathbf{Q}(n) = \mathbf{I}$ in (27) and recalling that $\|\mathbf{s}_1\| = 1$ we obtain

$$\epsilon(n) = \alpha b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{28}$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu\epsilon(n)\big(\mathbf{r}(n) - \mathbf{s}_1^t\mathbf{r}(n)\mathbf{s}_1\big) + \big(1 - \mathbf{s}_1^t\mathbf{c}(n-1)\big)\mathbf{s}_1, \qquad \mathbf{c}(0) = \mathbf{s}_1. \tag{29}$$

Notice that we initialize the algorithm with the matched filter. As we will see in the next section this form of

initialization, combined with the somewhat uncommon second order statistics of the data, turns out to be the reason for an unconventional behavior of this algorithmic class.

Let us now consider parameter α; we distinguish the following selections

$\alpha = 0$: This value generates the well known blind LMS (BLMS) of [12] (more precisely its robust version). Notice that BLMS requires the same amount of a priori information as the conventional matched filter, namely only the signature of the user of interest.

$\alpha = 1$: With this selection we generate a constrained version of LMS (CLMS). This algorithm must be distinguished from the conventional (unconstrained) LMS, used in the literature for the same problem [9], that satisfies the recursion

$$\epsilon(n) = b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{30}$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu\epsilon(n)\mathbf{r}(n), \quad \mathbf{c}(0) = \mathbf{s}_1. \tag{31}$$

which is known to be robust.

$\alpha = \alpha_1$: Here parameter α is equal to the amplitude of User-1 (user of interest). This gives rise to a constrained LMS with amplitude information (CLMS-AI). As we show in Section 5, CLMS-AI turns out to be optimum in a very well defined sense.

4. Performance Measure and Fair Comparisons

The most suitable measure of performance for the multiuser detection problem is definitely the BER. This quantity however suffers from serious mathematical intractability, therefore alternative measures have been proposed which, at least asymptotically, are equivalent to BER. One such possibility is the signal to interference ratio (SIR) that can also be used to obtain efficient approximations for BER [5]. To define the SIR let us recall that detection at time n is performed through (3), but with $\mathbf{c}$ replaced with the filter estimate at time $n-1$, that is,

$$\hat{b}_1(n) = \operatorname{sgn}\{\mathbf{c}^t(n-1)\mathbf{r}(n)\}. \tag{32}$$

We can then write

$$\mathbf{c}^t(n-1)\mathbf{r}(n) = a_1 b_1(n) + \mathbf{c}^t(n-1)\tilde{\mathbf{r}}(n) \tag{33}$$

$$\begin{aligned}\tilde{\mathbf{r}}(n) &= \sum_{i=2}^{K} a_i b_i(n)\mathbf{s}_i + \sigma\mathbf{n}(n)\\ &= \tilde{\mathbf{S}}\tilde{\mathbf{A}}\tilde{\mathbf{b}}(n) + \sigma\mathbf{n}(n),\end{aligned} \tag{34}$$

where $\tilde{\mathbf{r}}(n)$ denotes the interference plus noise part of the data, $\tilde{\mathbf{b}}(n) = [b_2(n)\cdots b_K(n)]^t$, $\tilde{\mathbf{S}} = [\mathbf{s}_2\cdots\mathbf{s}_K]$ and $\tilde{\mathbf{A}} = \operatorname{diag}\{a_2,\ldots,a_K\}$. For our problem it is more convenient to use the inverse SIR (ISIR) which is defined as

$$\operatorname{ISIR}(n) = a_1^{-2}\mathbb{E}\{[\mathbf{c}^t(n-1)\tilde{\mathbf{r}}(n)]^2\}. \tag{35}$$

To put this quantity under a more suitable form, let $\bar{\mathbf{c}}(n)$ denote the mean and $\boldsymbol{\Sigma}_{\mathbf{c}}(n)$ the covariance matrix of the filter estimates $\mathbf{c}(n)$, that is

$$\begin{aligned}\bar{\mathbf{c}}(n) &= \mathbb{E}\{\mathbf{c}(n)\},\\ \boldsymbol{\Sigma}_{\mathbf{c}}(n) &= \mathbb{E}\{[\mathbf{c}(n) - \bar{\mathbf{c}}(n)][\mathbf{c}(n) - \bar{\mathbf{c}}(n)]^t\}.\end{aligned} \tag{36}$$

Let also $\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}} = \mathbb{E}\{\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)\}$ denote the covariance matrix of the interference plus noise part of the data then, because of independence between $\tilde{\mathbf{r}}(n)$ and $\mathbf{c}(n-1)$ and because for any two matrices $\mathbf{D}$, $\mathbf{E}$ of the same dimensions we have $\operatorname{trace}\{\mathbf{D}^t\mathbf{E}\} = \operatorname{trace}\{\mathbf{E}\mathbf{D}^t\}$, we can write

$$\begin{aligned}\mathbb{E}\{(\mathbf{c}^t(n-1)\tilde{\mathbf{r}}(n))^2\} &= \mathbb{E}\{(\bar{\mathbf{c}}^t(n-1)\tilde{\mathbf{r}}(n))^2\}\\ &\quad + \mathbb{E}\{([\mathbf{c}(n-1)\\ &\quad - \bar{\mathbf{c}}(n-1)]^t\tilde{\mathbf{r}}(n))^2\}\\ &= \bar{\mathbf{c}}^t(n-1)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\bar{\mathbf{c}}(n-1)\\ &\quad + \operatorname{trace}\{\boldsymbol{\Sigma}_{\mathbf{c}}(n-1)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\},\end{aligned} \tag{37}$$

which leads to

$$\begin{aligned}\operatorname{ISIR}(n) = a_1^{-2}\big\{&\mathbf{c}_o^t\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}_o + \big[\bar{\mathbf{c}}^t(n-1)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\bar{c}(n-1)\\ &- \mathbf{c}_o^t\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}_o\big] + \operatorname{trace}\{\boldsymbol{\Sigma}_{\mathbf{c}}(n-1)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\}\big\}.\end{aligned} \tag{38}$$

We can now make the following observations from (38) for the three terms comprising ISIR.

- The first term is constant and common to all adaptive algorithms since it involves the optimum MMSE filter $\mathbf{c}_o$. This term would have been our ISIR had we available the statistics of the data.
- The next two terms are due to the adaptive algorithm. The second term involves the deterministic sequence of mean estimates $\bar{\mathbf{c}}(n)$, with the first element of this sequence being $\mathbf{s}_1$ (since $\mathbf{c}(0) = \mathbf{s}_1$) and the limit, for (asymptotically) unbiased estimators, being the optimum MMSE filter $\mathbf{c}_o$. Therefore the second term in (38) starts from an $O(1)$ (order of a constant) value and tends to zero as time progresses.

- The last term in (38) is due to the randomness of our estimates. This term is initially zero (since $\mathbf{c}(0) = \mathbf{s}_1$ is deterministic) and converges, at steady state, to an $O(\mu)$ value [13, pages 106–107]. Therefore this term is always small.

From the above we conclude that the second term is mainly responsible for the transient phase of the algorithm while the third for its steady state behavior.

Since the first term in (38) is common to all algorithms we will not consider it in our performance evaluation process. As our final performance measure we therefore propose the sum of the last two terms in (38) which are the terms directly related to the adaptive algorithm. Specifically we propose the following performance measure

$$\begin{aligned} J(n) &= a_1^{-2}\left\{\mathbb{E}\left\{\mathbf{c}^t(n)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}(n)\right\} - \mathbf{c}_o^t\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}_o\right\} \\ &= a_1^{-2}\left\{\left[\bar{\mathbf{c}}^t(n)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\bar{\mathbf{c}}(n) - \mathbf{c}_o^t\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}_o\right]\right. \\ &\quad \left. + \operatorname{trace}\{\boldsymbol{\Sigma}_{\mathbf{c}}(n)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\}\right\}. \end{aligned} \tag{39}$$

It is clear that $J(n)$ expresses excess ISIR due to adaptation.

4.1. Fair Comparisons of Adaptive Algorithms

A common mistake made in the literature when comparing constant step size adaptive algorithms consists in performing comparisons by selecting the same step size μ in all algorithms under consideration. As it is discussed in detail in [14, 15], this selection has no mathematical grounds and can often lead to erroneous conclusions. Since the step size μ affects both the convergence rate and the steady state behavior of the algorithm, its correct choice is crucial for the comparison process.

A fair comparison method proposed in [14] and extensively analyzed in [15] consists in selecting the step sizes in such a way that all algorithms attain the same steady state performance level. Once the selection of step sizes is completed, the algorithms can be ranked according to their convergence rate. Alternatively, we could select the step size in each algorithm so that all algorithms have the same convergence rate and then rank the algorithms according to their steady state performance. It is the latter method we find more appropriate for our problem; both approaches however are theoretically equivalent. A last point that needs to be said here is that, since convergence refers to the transient phase and this phase is primarily due to the mean filter estimates $\bar{\mathbf{c}}(n)$, it is through this process that the convergence rate will be defined.

5. Performance Analysis

In this section we will analyze the behavior of the algorithm defined by (28), (29). In particular we are interested in the mean trajectory and the second order statistics of the corresponding estimates. It turns out that for our analysis we can discard the last term in (29) introduced to correct the non-robustness problem. This is because stability (convergence) and robustness are two problems that are traditionally considered separately. Therefore for studying convergence, we will assume infinite precision, which results in the elimination of the last term in (29). Consequently the algorithm we intend to analyze is the following

$$\epsilon(n) = \alpha b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{40}$$

$$\begin{aligned} \mathbf{c}(n) &= \mathbf{c}(n-1) + \mu\epsilon(n)\left(\mathbf{r}(n) - \mathbf{s}_1^t\mathbf{r}(n)\mathbf{s}_1\right) \\ &= \mathbf{c}(n-1) + \mu\left(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\right)\epsilon(n)\mathbf{r}(n), \quad \mathbf{c}(0) = \mathbf{s}_1. \end{aligned} \tag{41}$$

We recall that $\alpha = 1$ corresponds to CLMS, $\alpha = 0$ to BLMS and $\alpha = a_1$ to CLMS-AI.

5.1. Qualitative Analysis of Trained Algorithms

We can now state our first theorem that provides the necessary statistics for the estimates $\mathbf{c}(n)$ of (40), (41).

Theorem 1. *The trajectory of the mean filter estimates of the algorithm in* (40), (41) *satisfies the recursion*

$$\bar{\mathbf{c}}(n) = \left(\mathbf{I} - \mu\left(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\right)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\right)\bar{\mathbf{c}}(n-1), \quad \bar{\mathbf{c}}(0) = \mathbf{s}_1. \tag{42}$$

The covariance matrix $\boldsymbol{\Sigma}_{\mathbf{c}}(n)$ of the filter estimates can be written as the sum

$$\boldsymbol{\Sigma}_{\mathbf{c}}(n) = \boldsymbol{\Sigma}_{\mathbf{x}}(n) + (\alpha - a_1)^2\boldsymbol{\Sigma}_{\mathbf{y}}(n) \tag{43}$$

where $\boldsymbol{\Sigma}_{\mathbf{x}}(n)$ and $\boldsymbol{\Sigma}_{\mathbf{y}}(n)$ are the covariance matrices of two vector processes $\mathbf{x}(n)$ and $\mathbf{y}(n)$ defined by the recursions

$$\begin{aligned} \mathbf{x}(n) &= \left(\mathbf{I} - \mu\left(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\right)\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)\right)\mathbf{x}(n-1) \\ &\quad - \mu\left(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\right)\left(\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n) - \boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\right)\bar{\mathbf{c}}(n-1), \\ &\qquad\qquad \mathbf{x}(0) = 0 \end{aligned} \tag{44}$$

$$\mathbf{y}(n) = \big(\mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)\big)\mathbf{y}(n-1) + \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)b_1(n)\tilde{\mathbf{r}}(n), \quad \mathbf{y}(0) = 0. \tag{45}$$

Proof: The proof is presented in the Appendix. □

Using Theorem 1 we can now make the following important remarks.

- Trajectory $\bar{\mathbf{c}}(n)$ of the mean filter estimates, i.e. recursion (42), *is independent of* α. In particular CLMS ($\alpha = 1$) has the same mean trajectory with BLMS ($\alpha = 0$). This equality does not for example apply when we compare BLMS to the conventional LMS defined in (30), (31). An important consequence of this property is the fact that if we like to compare algorithms corresponding to different α values with the fair method described in Subsection 4.1, it is sufficient to select the *same step size* μ. Indeed this selection guarantees exactly the same trajectory for the mean filter estimates and therefore the same convergence rate (remember that the transient phase is primarily due to mean filter estimates). Again we should stress that this statement is not true when comparing any member from our class to the conventional LMS, in this case we do need to select different step sizes.
- Our performance measure $J(n)$, using (43), can be written as

$$J(n) = a_1^{-2}\big\{\big[\bar{\mathbf{c}}^t(n)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\bar{\mathbf{c}}(n) - \mathbf{c}_o^t\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}_o\big] + \text{trace}\{\boldsymbol{\Sigma}_{\mathbf{x}}(n)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\} + (\alpha - \alpha_1)^2\text{trace}\{\boldsymbol{\Sigma}_{\mathbf{y}}(n)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\}\big\}. \tag{46}$$

  We note that the first two terms are independent of α and so is trace $\{\boldsymbol{\Sigma}_{\mathbf{y}}(n)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\}$ in the third term while parameter α appears only in this last term. Furthermore this term is nonnegative because

$$\text{trace}\{\boldsymbol{\Sigma}_{\mathbf{y}}(n)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\} = \text{trace}\big\{\boldsymbol{\Sigma}_{\mathbf{y}}^{1/2}(n)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\boldsymbol{\Sigma}_{\mathbf{y}}^{1/2}\big\} \geq 0, \tag{47}$$

  the last inequality being true because traces of nonnegative definite matrices are also nonnegative. Due to (47), from (46) we can now conclude that *the algorithm that has the uniformly* (*at all time instances*) *smallest excess ISIR corresponds to* $\alpha = a_1$, i.e. CLMS-AI. Because of its importance let us explicitly write the recursion for the optimum algorithm CLMS-AI, we have

$$\epsilon(n) = a_1 b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{48}$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu\epsilon(n)\big(\mathbf{r}(n) - \mathbf{s}_1^t\mathbf{r}(n)\mathbf{s}_1\big) + \big(1 - \mathbf{s}_1^t\mathbf{c}(n-1)\big)\mathbf{s}_1, \quad \mathbf{c}(0) = \mathbf{s}_1. \tag{49}$$

- Using (46) we can also compare algorithms corresponding to different values of α. In particular if we like to compare CLMS to BLMS we need to set respectively $\alpha = 1$ and $\alpha = 0$. We can then verify that CLMS is better than BLMS iff $(1-a_1)^2 \leq a_1^2$ or equivalently iff $a_1 \geq 0.5$. This means that although CLMS uses the exact bits in its adaptation, it does not necessarily perform better than BLMS which completely ignores bit information. This fact is also true when we compare the conventional LMS of (30), (31) with BLMS. Figure 1 depicts such an example. The previous conclusion is rather surprising because in the literature it is widely believed that LMS is uniformly better than its blind counterpart!

The results of Theorem 1 are exact (there is no approximation involved in any sense) and no additional assumptions were used apart the ones we initially made regarding the statistics of the data. In fact the previous remarks hold for every time instant, for every μ and even when the algorithms diverge. So far we were able to rank the algorithms of interest without quantifying their relative performance. This is the subject of our next subsection.

5.2. *Quantitative Analysis of Trained Algorithms*

In this subsection we are going to use results from Stochastic Approximation Theory pertinent to the analysis of constant step size adaptive algorithms [13, 16]. In order for this theory to be applicable we need to assume that the step size μ is small, i.e. $0 < \mu \ll 1$, which is usually the case in practice.

Our goal is to find expressions for the performance measure $J(n)$ at steady state; this of course presumes convergence of the algorithms of interest. Convergence is assured [13] for sufficiently small μ and because matrix $(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}$ has real nonnegative eigenvalues (it has the same eigenvalues with $\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}^{1/2}(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}^{1/2}$ since the two matrices are related through a similarity transformation). To estimate $J(\infty)$ we need to compute $\boldsymbol{\Sigma}_{\mathbf{x}}(\infty)$ and $\boldsymbol{\Sigma}_{\mathbf{y}}(\infty)$. Although it is possible to find exact expressions for both covariances the results turn out to be mathematically involved. To simplify our presentation and at the same time gain a realistic feeling of

the relative performance of the algorithms of interest we will make, as in [4], the assumption that $\mathbf{R} \approx \mathbf{I}$. In other words that the correlation matrix of the signature waveforms is close to the identity, or equivalently that the signature waveforms are almost orthogonal. We have now the second theorem that quantifies the performance of the algorithm in (40), (41).

Theorem 2. *Let* $\mathbf{x}(n)$, $\mathbf{y}(n)$ *be the processes defined in* (44), (45); *if* $0 < \mu \ll 1$ *then, to a first order approximation in* μ, *we have that*

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{x}}(n) \approx{}& \big(\mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big)\boldsymbol{\Sigma}_{\mathbf{x}}(n-1)\\ &\times \big(\mathbf{I} - \mu\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\big)\\ &+ \mu^2\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\mathbb{E}\big\{\big(\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n) - \boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big)\bar{\mathbf{c}}(n-1)\\ &\times \tilde{\mathbf{c}}^t(n-1)\big(\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n) - \boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big)^t\big\}\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\end{aligned} \tag{50}$$

$$\begin{aligned}\boldsymbol{\Sigma}_{\mathbf{y}}(n) \approx{}& \big(\mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big)\boldsymbol{\Sigma}_{\mathbf{y}}(n-1)\\ &\times \big(\mathbf{I} - \mu\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\big)\\ &+ \mu^2\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big),\end{aligned} \tag{51}$$

with initial conditions $\boldsymbol{\Sigma}_{\mathbf{x}}(0) = \boldsymbol{\Sigma}_{\mathbf{y}}(0) = 0$. *If we further assume that the correlation matrix of the signature waveforms satisfies* $\mathbf{R} \approx \mathbf{I}$ *then at steady state we can write*

$$\begin{aligned}J(\infty) \approx{}& \frac{\mu}{2}\left(\sum_{i=2}^{K} a_i^2 + (N-1)\sigma^2\right)\\ &\times \big(a_1^{-2}\sigma^2 + a_1^{-2}(\alpha - \alpha_1)^2\big)\end{aligned} \tag{52}$$

Proof: The proof is presented in the Appendix. □

Using (52) we can compute the relative performance of any two algorithms, at steady state, corresponding to different parameters α_1, α_2. This takes the following simple form

$$\boldsymbol{J}_{1,2} \approx \frac{\sigma^2 + (\alpha_1 - a_1)^2}{\sigma^2 + (\alpha_2 - a_1)^2}. \tag{53}$$

In particular if we consider the relative performance of BLMS with respect to the optimum CLMS-AI we obtain

$$J_{\mathrm{BLMS,CLMS-AI}} \approx 1 + \frac{a_1^2}{\sigma^2} = 1 + \mathrm{SNR}, \tag{54}$$

suggesting that the optimum algorithm can be significantly better than BLMS in high SNR channels. Figure 1 depicts a case where BLMS performs better than CLMS and the conventional LMS but is of course inferior to the optimum CLMS-AI. We used $a_1 = 0.1$ and $\sigma^2 = 0.01$. Although the conventional LMS does not have the same convergence rate as the remaining algorithms it is safe to conclude that it is inferior to BLMS. This is because, at the same time, it has a smaller convergence rate and a larger steady state excess ISIR. Consequently, attempting to make its rate equal to the rate of BLMS (for fair comparison), requires increase of its step size which will further increase its steady state performance. One can also verify that the relative performance of CLMS, BLMS, CLMS-AI, is very closely predicted by (53).

5.3. *Modes of Convergence*

A convergence characteristic that can be observed from Fig. 1 is the fact that the three algorithms from our class (CLMS, BLMS, CLMS-AI) exhibit two different modes of convergence, namely a fast mode during the initial transient phase and a subsequent slow drift toward inferior performance values. This behavior, particularly apparent when the number of users K is significantly smaller than the spreading factor N, is very uncommon in adaptive algorithms.

To understand why the algorithms behave in such a way, we recall from (39) that the excess ISIR is the sum of two components, the first due to mean filter estimates and the second to the covariance of the estimates. From Eq. (42) we have that the exponential convergence rate of the mean estimates is equal to (minus) the logarithm of the largest (in amplitude) eigenvalue of the matrix $\mathbf{I} - (\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}$ which, for $K < N$, is equal to $-\log(1 - \mu\sigma^2) \approx \mu\sigma^2$. This rate however is significantly slower than the one observed in Fig. 1 during the initial transient phase. After careful consideration of Eq. (42) one can show by induction that if the mean filter, at some instant, lies in the signal subspace generated by the signatures, it remains in this subspace afterwards (i.e. the mean filter is a linear combination of the signatures). This is exactly what happens in our algorithm since the filter estimate is initialized in the signal subspace ($\mathbf{c}(0) = \mathbf{s}_1$). Therefore the rate of convergence becomes $\mu(\sigma^2 + \lambda_{\min})$ with $\lambda_{\min}$ denoting the smallest nonzero eigenvalue of the rank $K-1$ matrix $(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t)\tilde{\mathbf{S}}\tilde{A}^2\tilde{S}^t$. We thus conclude that the first term in the performance measure starts from an $O(1)$ value and with an exponential rate equal to $2\mu(\sigma^2 + \lambda_{\min})$ (which is usually significantly larger than $2\mu\sigma^2$) converges to zero.

From Eqs. (50), (51) we can conclude that, because of the ambient noise, the covariance matrix is not limited into the signal subspace and has components lying in the complementary noise subspace (for example $\mathbf{\Sigma}_{\mathbf{y}}(n)$ contains a term equal to $\mu^2\sigma^2\mathbf{I}$). Therefore the covariance has a part that is slowly converging with a rate equal to $2\mu\sigma^2$. This means that the second term of our measure, since its initial value is zero ($\mathbf{\Sigma}_{\mathbf{c}}(0) = 0$), increases slowly from zero to its $O(\mu)$ steady state value estimated by (52).

It is now possible to understand the behavior of the algorithms in Fig. 1. During the transient phase the leading term of our measure is the first one due to the mean estimates, which converges very quickly to zero. The second term, due to the covariance, during the transient phase has negligible values (since it starts from zero and increases very slowly). After some point however the second term becomes the leading one and this is why we observe this second slowly increasing mode.

Notice that for fast convergence it is necessary for the mean estimates to lie completely in the signal subspace, because otherwise if there is a component in the noise subspace this part will exhibit slow convergence towards zero. When initializing the algorithm with $\mathbf{s}_1$, the mean estimates do lie in the signal subspace. This is also the case if we have an increase in the number of users. When however we have a reduction in the number of users then *this property is no longer true*. This is because the component of the filter corresponding to the signatures of the users that departed lies now inside the noise subspace and is therefore slowly converging to zero. We will be able to observe this mode of behavior in Section 6 where we present our simulations.

5.4. Decision Directed Version

It is interesting at this point to introduce the decision directed version of the optimum algorithm CLMS-AI. From (48), (49), by replacing $b_1(n)$ with the estimate $\hat{b}_1(n) = \operatorname{sgn}\{\mathbf{c}^t(n-1)\mathbf{r}(n)\}$ we obtain

$$\epsilon(n) = a_1 \operatorname{sgn}\{\mathbf{c}^t(n-1)\mathbf{r}(n)\} - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{55}$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \mu\epsilon(n)\big(\mathbf{r}(n) - \mathbf{s}_1^t\mathbf{r}(n)\mathbf{s}_1\big) + \big(1 - \mathbf{s}_1^t\mathbf{c}(n-1)\big)\mathbf{s}_1, \quad \mathbf{c}(0) = \mathbf{s}_1. \tag{56}$$

Let us call this algorithm decision directed constrained LMS with amplitude information (DD-CLMS-AI). We should note that we consider the recursion to be applied to the data from the first time instant and not after we have obtained satisfactory filter estimates using some other scheme (as is usually the case in practice). Due to the nonlinear function $\operatorname{sgn}\{\cdot\}$ involved in its definition, the analysis of DD-CLMS-AI is not as simple as CLMS-AI, neither our results can be of the same generality. In order to obtain closed form expressions for the mean field and the second order statistics we will make the simplifying assumption that the interference plus noise part of the data can be adequately modeled as a Gaussian process. Signaling conditions that ensure efficiency of this approximation are given in [5] (we basically need validity of the Central Limit Theorem).

Theorem 3. *If interference plus noise* $\tilde{\mathbf{r}}(n)$ *is Gaussian* $\mathcal{N}(0, \mathbf{\Sigma}_{\tilde{\mathbf{r}}})$ *then, for sufficiently small* μ *and to a first order approximation in* μ*, the trajectory of the mean filter estimates of the algorithm defined by* (55), (56) *satisfies*

$$\bar{\mathbf{c}}(n) = \left(\mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\mathbf{\Sigma}_{\tilde{\mathbf{r}}}\rho\left(\frac{a_1}{\big\|\mathbf{\Sigma}_{\tilde{\mathbf{r}}}^{1/2}\bar{\mathbf{c}}(n-1)\big\|}\right)\right) \times \bar{\mathbf{c}}(n-1), \quad \bar{\mathbf{c}}(0) = \mathbf{s}_1 \tag{57}$$

where scalar function $\rho(x)$ *is defined as*

$$\rho(x) = 1 - \sqrt{\frac{2}{\pi}}\, x e^{-\frac{x^2}{2}}. \tag{58}$$

If we further assume that $\mathbf{R} \approx \mathbf{I}$*, then the excess ISIR at steady state takes the form*

$$J_{dd}(\infty) \approx \frac{\mu}{2}\left(\sum_{i=0}^{K} a_i^2 + (N-1)\sigma^2\right) \times \left(a_1^{-2}\sigma^2 + 4Q\left(\frac{a_1}{\sigma}\right) - \frac{4\sigma}{\sqrt{2\pi}a_1}e^{-a_1^2/2\sigma^2}\right)\rho^{-1}\left(\frac{a_1}{\sigma}\right) \tag{59}$$

where $Q(x)$ *is the complementary Gaussian cumulative distribution function.*

Proof: The proof is presented in the Appendix. □

From our last theorem we can draw the following conclusions.

- Comparing the trajectory of the mean estimates in (57) with the corresponding in (42) we observe that the difference exists only in the scalar

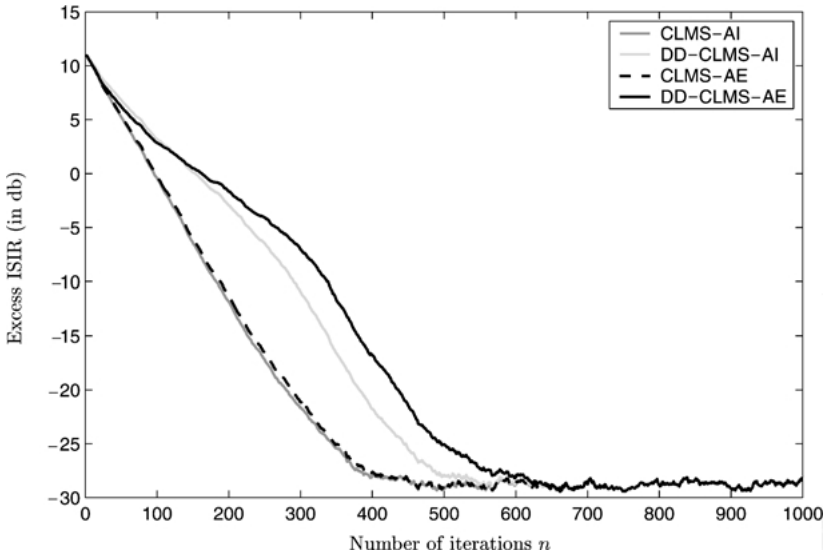

Figure 3. Typical relative performance of CLMS-AI, DD-CLMS-AI, CLMS-AE and DD-CLMS-AE.

quantity $\rho(\cdot)$. Although the adaptation in (57) is nonlinear it has invariant eigenspaces which coincide with the eigenspaces of the adaptation in (42). This facilitates the convergence analysis of the decision directed version considerably. Notice that $0.706 \approx 1 - \sqrt{2/\pi}\, e^{-1} \leq \rho(x) \leq 1$, which guarantees convergence of the mean field at a rate at least 0.706 times the rate of the optimum CLMS-AI. Furthermore as the algorithm converges, quantity $a_1/\|\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}^{1/2}\tilde{\mathbf{c}}(n-1)\|$ approaches its limit which, for $\mathbf{R} \approx \mathbf{I}$, is approximately equal to the square root of the SNR i.e. $\frac{a_1}{\sigma}$. It turns out that even for moderate SNR values the corresponding value of $\rho(\cdot)$ is very close to unity. For example for SNR $= 10$ db, $\rho(\sqrt{10}) = 0.983$; while for 20 db, $\rho(10) = 1 - 1.54 \times 10^{-21}$. This suggests that initially DD-CLMS-AI has a smaller convergence rate than CLMS-AI however, as time progresses, its rate approaches that of the optimum algorithm.

- Using (52) with $\alpha = a_1$ and (59) we can compare the steady state performance of CLMS-AI and DD-CLMS-AI. If we form the ratio $J_{dd}(\infty)/J(\infty)$ then this quantity is close to unity even for very low SNR. For example for SNR $= 0$ db it is equal to 1.2920; for 10 db, 1.0146; whereas for 20 db it is equal to unity (with Maltab's accuracy). This clearly suggests that DD-CLMS-AI has steady performance which is extremely close to the optimum, since even for SNR $= 0$ db it differs from the optimum slightly more than 1 db.

Figure 3 presents a typical example of the relative performance of DD-CLMS-AI and CLMS-AI. We can see that initially the decision directed version has a nonlinear behavior, however after some point the two curves tend to be parallel, meaning that the two convergence rates approach each other. At steady state on the other hand we have that the two algorithms are practically indistinguishable, as was predicted by our analysis.

5.5. *Amplitude Estimation Algorithms*

So far we have seen that the optimum trained algorithm in our LMS like class is CLMS-AI, i.e. the constrained LMS with amplitude information. Furthermore its decision directed version DD-CLMS-AI was seen to be equally efficient having slightly inferior performance. As it was stated in our introduction, it is unrealistic to assume knowledge of the amplitude of the user of interest even in training mode, we therefore need a means to acquire this information. It is this point we like to answer here by proposing a simple yet efficient adaptive estimation scheme that is consistent with the algorithmic class introduced in Subsection 3.3.

We can verify that for any filter $\mathbf{c}(n)$ that satisfies the constraint $\mathbf{c}^t(n)\mathbf{s}_1 = 1$ we have

$$a_1 = \mathbb{E}\big\{\mathbf{c}^t(n-1)\mathbf{r}(n)b_1(n)\big\}. \tag{60}$$

If we approximate expectation with sample mean, this suggests the following simple adaptation

$$\hat{a}_1(n) = (1-\nu)\hat{a}_1(n-1) + \nu \mathbf{c}^t(n-1)\mathbf{r}(n)b_1(n), \quad \hat{a}_1(0) = 0. \quad (61)$$

For $\nu = 1/n$ the above algorithm computes the sample mean of the quantities $\mathbf{c}^t(n-1)\mathbf{r}(n)b_1(n)$. However if we allow ν to be a positive constant smaller than unity the algorithm will also be able to adapt to slow variations in the amplitude a_1. The corresponding decision directed version satisfies

$$\hat{a}_1(n) = (1-\nu)\hat{a}_1(n-1) + \nu |\mathbf{c}^t(n-1)\mathbf{r}(n)|, \quad \hat{a}_1(0) = 0. \quad (62)$$

For $\mathbf{c}(n)$ we can of course use the filter estimates from the corresponding trained or decision directed filter estimation algorithm.

Estimates $\hat{a}_1(n)$ can now be used to supply the necessary amplitude information to CLMS-AI and DD-CLMS-AI. The resulting algorithms will be called respectively constrained LMS with amplitude estimation (CLMS-AE) and decision directed constrained LMS with amplitude estimation (DD-CLMS-AE). Figure 3 presents a typical example of the relative performance of CLMS-AI and CLMS-AE as well as their decision directed versions. As far as CLMS-AE is concerned we can see that it is extremely close to the optimum CLMS-AI, whereas DD-CLMS-AE is slightly inferior to DD-CLMS-AI. The relative performance however is very much dependent on the signaling conditions, as we will soon find out in the simulations section.

5.6. *Subspace Based Adaptations*

For our algorithmic class it is also possible to define subspace based adaptations similar to [6, 8]. To obtain the corresponding algorithms it is sufficient to replace the data vector $\mathbf{r}(n)$ with its projection onto the signal subspace. This will affect the convergence rate of the covariance matrix $\mathbf{\Sigma}_{\mathbf{c}}(n)$ and partly the convergence of the mean filter estimates. In both cases the rate will be equal to the slowest mode defined on the signal subspace. This means that the subspace based algorithms will not exhibit any annoying slow performance degradation, as the one observed in Fig. 1; furthermore the mean filter estimates will converge fast even after a reduction in the number of users. Finally the steady state excess ISIR will be smaller since the term $(N-1)\sigma^2$ in (52) will become $(K-1)\sigma^2$.

The above improved characteristics are obtained at the expense of an increased complexity [8] which needs to be paid in order to track the signal subspace and detect changes in its dimension (i.e. changes in the number of users).

6. Simulations

In this section we present a number of simulations to compare the relative performance of DD-CLMS-AE, CLMS-AE and BLMS. A common feature of all three algorithms is the fact that they are *scale invariant*. By this we mean that if we scale our data $\mathbf{r}(n)$ by a factor c then the three algorithms yield exactly the same estimates as with the initial unscaled data provided that we divide μ with c^2 (this property is not true for LMS or CLMS). We can therefore, without loss of generality, fix $a_1 = 1$.

In order to obtain algorithms with steady state performance that does not change every time there is a variation in the number of users or in user and noise powers, it is convenient to employ a normalized version of μ. We therefore propose to replace μ by $\mu = \bar{\mu}/\beta(n)$ where

$$\begin{aligned} \beta(n) &= (1-\kappa)\beta(n-1) + \kappa \|\mathbf{r}(n)\|^2, \\ \beta(1) &= \|\mathbf{r}(1)\|^2, \end{aligned} \quad (63)$$

is an estimate of the data power with κ a constant satisfying $1 > \kappa > 0$. The final form of the three algorithms we intend to test is the following

BLMS

$$\epsilon(n) = -\mathbf{c}^t(n-1)\mathbf{r}(n) \quad (64)$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \bar{\mu}\frac{\epsilon(n)}{\beta(n)}\big(\mathbf{r}(n) - \mathbf{s}_1^t\mathbf{r}(n)\mathbf{s}_1\big) + \big(1 - \mathbf{s}_1^t\mathbf{c}(n-1)\big)\mathbf{s}_1, \quad \mathbf{c}(0) = \mathbf{s}_1. \quad (65)$$

CLMS-AE

$$\hat{a}_1(n) = (1-\nu)\hat{a}_1(n-1) + \nu\mathbf{c}^t(n-1)\mathbf{r}(n)b_1(n), \quad \hat{a}_1(0) = 0 \quad (66)$$

$$\epsilon(n) = \hat{a}_1(n)b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n) \quad (67)$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \bar{\mu}\frac{\epsilon(n)}{\beta(n)}\big(\mathbf{r}(n) - \mathbf{s}_1^t\mathbf{r}(n)\mathbf{s}_1\big) + \big(1 - \mathbf{s}_1^t\mathbf{c}(n-1)\big)\mathbf{s}_1, \quad \mathbf{c}(0) = \mathbf{s}_1. \quad (68)$$

DD-CLMS-AE

$$\hat{a}_1(n) = (1-\nu)\hat{a}_1(n-1) + \nu|\mathbf{c}^t(n-1)\mathbf{r}(n)|, \quad \hat{a}_1(0) = 0 \quad (69)$$

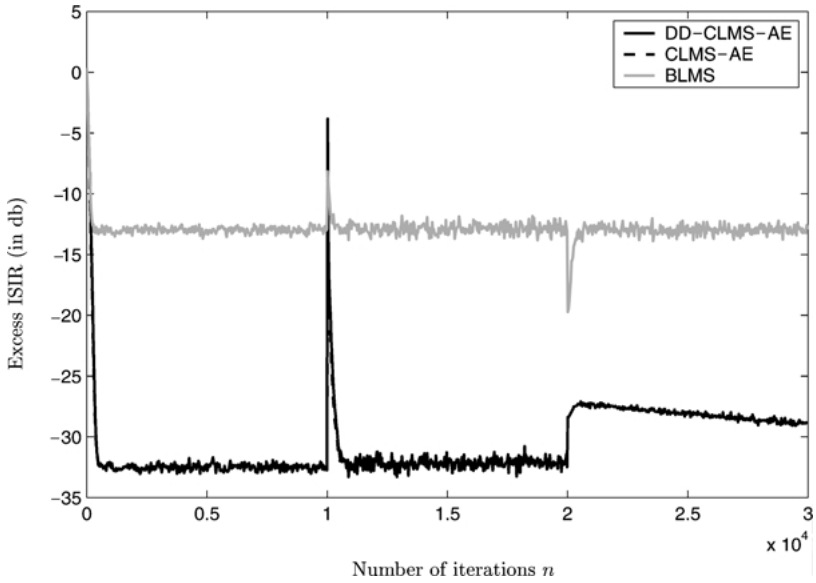

Figure 4. Relative performance of DD-CLMS-AE, CLMS-AE, BLMS. Initially there are six 10 db interferers; at $n = 10000$ a 20 db intereferer enters the channel and at $n = 20000$ it exits along with three 10 db intereferers.

$$\epsilon(n) = \hat{a}_1(n)\operatorname{sgn}\{\mathbf{c}^t(n-1)\mathbf{r}(n)\} - \mathbf{c}^t(n-1)\mathbf{r}(n) \tag{70}$$

$$\mathbf{c}(n) = \mathbf{c}(n-1) + \bar{\mu}\frac{\epsilon(n)}{\beta(n)}\big(\mathbf{r}(n) - \mathbf{s}_1^t\mathbf{r}(n)\mathbf{s}_1\big) + \big(1 - \mathbf{s}_1^t\mathbf{c}(n-1)\big)\mathbf{s}_1, \quad \mathbf{c}(0) = \mathbf{s}_1. \tag{71}$$

It should be noted that DD-CLMS-AE, like BLMS, is blind since it requires the same a priori information (i.e. $\mathbf{s}_1$) and uses the same data (i.e. $\mathbf{r}(n)$). It is exactly this version we propose as an alternative to BLMS. Comparing also the computational complexity of the two algorithms, we realize that DD-CLMS-AE requires only an additional constant number of scalar operations as compared to BLMS. It is therefore equally computationally efficient.

For our simulations we used a spreading factor $N = 128$, with signature waveforms generated randomly but then kept constant during the whole simulation set. For the user of interest we selected $a_1 = 1$, while for noise power $\sigma^2 = 0.01$ (SNR of 20 db). Step sizes were selected as follows: $\bar{\mu} = 0.1$, $\nu = \kappa = 0.01$. Our performance measure from (39) takes the form $J(n) = E\{\mathbf{c}^t(n)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}(n)\} - \mathbf{c}_o^t\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}_o$. Expectations were estimated by taking the sample mean of 100 independent runs, while $\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}$ and $\mathbf{c}_o$ were computed using the exact signatures and user and noise powers. Since CLMS-AE has almost optimum performance it was regarded as a point of reference and BLMS and DD-CLMS-AE were compared against it.

In the first example, we initially have the user of interest with six interfering users of power 10 db; at time 10000 a 20 db interferer enters the channel, and at time 20000 this interferer exits along with three more 10 db interfering users. This example was selected in order to observe the behavior of the algorithms under signaling condition that do not favor validity of the Gaussian assumption of the signal plus noise data part (because of small K). Figure 4 depicts the outcome of our simulation. We can see that DD-CLMS-AE follows very closely CLMS-AE. Notice the slow convergence, after the second change, that was predicted by our analysis for the case of users exiting the channel.

The second example consists in comparing the behavior of the algorithms when power control is employed (i.e. all $a_i = 1$). Initially we have thirty users; at time 10000 the number of users increases to thirty five while at 20000 it is reduced to twenty five. Notice from Fig. 5 that the decision directed version follows again very closely CLMS-AE.

The third and final example contains rather extreme signaling conditions. We start with thirty users, only now the interferers have power equal to 10 db. At time 10000 five 20 db interferers enter, while at time 20000 all five 20 db along with five 10 db interferes exit the channel. In Fig. 6 we can observe that the initial nonlinear behavior of DD-CLMS-AE is now more

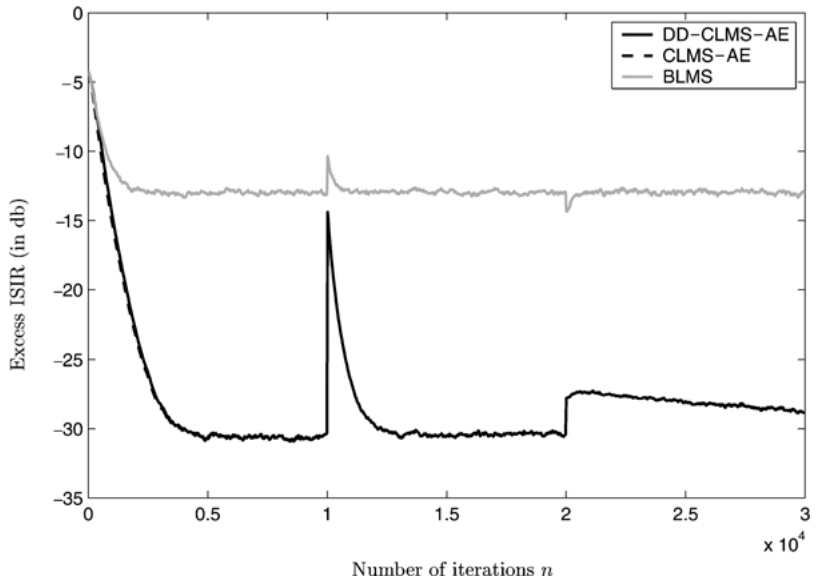

Figure 5. Relative performance of DD-CLMS-AE, CLMS-AE, BLMS under perfect power control. Initially there are 30 users; at $n = 10000$ five additional intereferers enter the channel and at $n = 20000$ ten interferers exit the channel.

pronounced. However the algorithm, very quickly, establishes a convergence rate that is similar to CLMS-AE. The steady state behavior of the two algorithms on the other hand is indistinguishable.

In all three examples we observe that the steady state excess ISIR for BLMS is approximately equal to 13 db whereas that of DD-CLMS-AE, to 30 db. To translate these numbers into actual BER we use the estimates $\text{BER} \approx Q(\sqrt{\text{SIR}})$ [5] and $\mathbf{c}_o^t \boldsymbol{\Sigma}_{\tilde{\mathbf{r}}} \mathbf{c}_o \approx \sigma^2$. We then obtain that the BER of BLMS is of the order of 10^{-5} whereas that of DD-CLMS-AE is 10^{-20}.

Finally we should mention that under extreme interference conditions it is possible that DD-CLMS-AE exhibits divergence. This is because the algorithm is

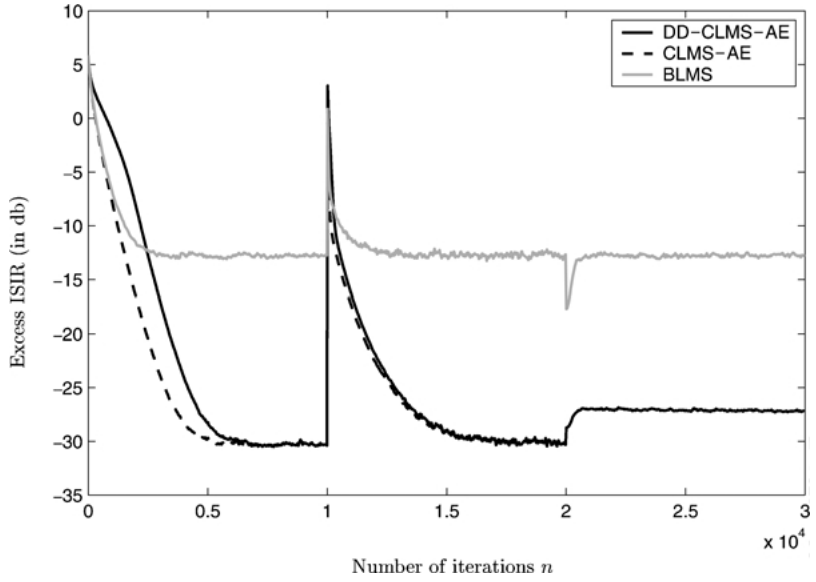

Figure 6. Relative performance of DD-CLMS-AE, CLMS-AE, BLMS. Initially there are twenty nine 10 db interferers; at $n = 10000$ five 20 db intereferers enter the channel while at $n = 20000$ all five 20 db along with five 10 db interferers exit the channel.

unable to obtain, quickly enough, satisfactory estimates of the amplitude a_1 and therefore provides erroneous amplitude estimates to the filter estimation part resulting in divergence. In such cases it is advisable to run BLMS in parallel with DD-CLMS-AE and use its filter estimates in (69) to estimate the amplitude a_1 (BLMS tends to be more robust to extreme signaling conditions). Once convergence has been established we can use the filter estimates of DD-CLMS-AE in (69) and discard completely BLMS.

7. Conclusion

We presented a constrained class of adaptive linear multiuser detection algorithms that constitutes an extension to the popular blind LMS algorithm of [12]. Applying a detailed analysis to the proposed class we showed that the conventional LMS and its constrained version, under training, do not necessarily outperform the blind LMS. In order for this property to be true it is necessary to incorporate the information of amplitude of the user of interest in the trained algorithm. Simple and efficient adaptations that estimate the required amplitude were proposed which, when combined with the filter estimation algorithm result in both modes, trained and decision directed, in nearly optimum performance. Since the proposed decision directed version is also blind, it is clear that, it could constitute a serious alternative to the popular blind LMS.

Appendix

Proof of Theorem 1: By using the constraint $\mathbf{c}^t(n-1)\mathbf{s}_1 = 1$ and the fact that $(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t)\mathbf{s}_1 = 0$ we can write (40), (41) as follows

$$\begin{aligned}\epsilon(n) &= \alpha b_1(n) - \mathbf{c}^t(n-1)\mathbf{r}(n)\\ &= (\alpha - a_1)b_1(n) - \mathbf{c}^t(n-1)\tilde{\mathbf{r}}(n) \qquad (72)\\ \mathbf{c}(n) &= \mathbf{c}(n-1) + \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\epsilon(n)\mathbf{r}(n)\\ &= \mathbf{c}(n-1) + \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\epsilon(n)\tilde{\mathbf{r}}(n), \quad \mathbf{c}(0) = \mathbf{s}_1, \qquad (73)\end{aligned}$$

Taking expectation in (73) and using the independence between data $\mathbf{r}(n)$ and $\mathbf{c}(n-1)$ as well as independence between data bits and noise, we can easily show (42).

To show (43), notice first that by subtracting (42) from (41) and combining (44) with (45), we can show that $\mathbf{c}(n) - \bar{\mathbf{c}}(n)$ satisfies exactly the same recursion as $\mathbf{x}(n) + (\alpha - a_1)\mathbf{y}(n)$, therefore $\mathbf{c}(n) - \bar{\mathbf{c}}(n) = \mathbf{x}(n) + (\alpha - a_1)\mathbf{y}(n)$. Now notice that if $\mathrm{col}\{\boldsymbol{\Sigma}_{\mathbf{c}}(n)\}$ denotes the column vector that we obtain by stacking the columns of $\boldsymbol{\Sigma}_{\mathbf{c}}(n)$ one after the other then

$$\begin{aligned}\mathrm{col}\{\boldsymbol{\Sigma}_{\mathbf{c}}(n)\} &= \mathbb{E}\{(\mathbf{c}(n) - \bar{\mathbf{c}}(n)) \otimes (\mathbf{c}(n) - \bar{\mathbf{c}}(n))\}\\ &= \mathbb{E}\{(\mathbf{x}(n) + (\alpha - a_1)\mathbf{y}(n))\\ &\quad \otimes (\mathbf{x}(n) + (\alpha - a_1)\mathbf{y}(n))\} \qquad (74)\end{aligned}$$

where "$\otimes$" denotes Kronecker product. Using the independence of the processes involved in the recursions and induction we can show that vector processes $\mathbf{x}(n)$, $\mathbf{y}(n)$ are zero mean and uncorrelated. This in turn can help us to prove that $\mathrm{col}\{\boldsymbol{\Sigma}_{\mathbf{c}}(n)\}$ satisfies exactly the same recursion as $\mathrm{col}\{\boldsymbol{\Sigma}_{\mathbf{x}}(n)\} + (\alpha - a_1)^2\mathrm{col}\{\boldsymbol{\Sigma}_{\mathbf{x}}(n)\}$. This concludes the proof. □

Proof of Theorem 2: The easiest way to show (50), (51) is to consider the column version of the corresponding covariance matrices. Let us for simplicity show only (51), in the same way we can show (50). We have

$$\begin{aligned}\mathrm{col}\{\boldsymbol{\Sigma}_{\mathbf{y}}(n)\} &= \mathbb{E}\Big\{\big(\mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)\big)\\ &\quad \otimes \big(\mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)\big)\Big\}\\ &\quad \times \mathrm{col}\{\boldsymbol{\Sigma}_{\mathbf{y}}(n-1)\} + \mu^2\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\\ &\quad \otimes \big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\mathrm{col}\{\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\} \qquad (75)\end{aligned}$$

Now notice that, to a first order approximation in μ, we can write

$$\begin{aligned}&\mathbb{E}\Big\{\big(\mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)\big)\\ &\quad \otimes \big(\mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)\big)\Big\}\\ &= \mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}} \otimes \mathbf{I} - \mu\mathbf{I}\\ &\quad \otimes \big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}} + O(\mu^2)\\ &\approx \mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}} \otimes \mathbf{I} - \mu\mathbf{I} \otimes \big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\\ &\quad + \mu^2\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}} \otimes + \big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\\ &= \big(\mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big) \otimes \big(\mathbf{I} - \mu\big(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big). \qquad (76)\end{aligned}$$

Since $O(\mu^2)$ terms do not contribute in approximations up to order μ we replaced in one of the previous equations the $O(\mu^2)$ term with another, suitable to our goal, $O(\mu^2)$ expression. Substituting the above into (75) yields the column version of (51).

To show (52) we have from (46) that, at steady state

$$\begin{aligned}J(\infty) = a_1^{-2}\big\{&\mathrm{trace}\{\boldsymbol{\Sigma}_{\mathbf{x}}(\infty)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\}\\ &+ (\alpha - a_1)^2\mathrm{trace}\{\boldsymbol{\Sigma}_{\mathbf{y}}(\infty)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\}\big\}. \qquad (77)\end{aligned}$$

Consider now (50) at steady state, we have the following Lyapunov equation that defines $\boldsymbol{\Sigma}_{\mathbf{x}}(\infty)$ to a first order approximation in μ

$$\begin{aligned}\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\boldsymbol{\Sigma}_{\mathbf{x}}(\infty)+\boldsymbol{\Sigma}_{\mathbf{x}}(\infty)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\\ \approx \mu\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\mathbb{E}\big\{\big(\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)-\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big)\\ \times\mathbf{c}_o\mathbf{c}_o^t\big(\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)-\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big)^t\big\}\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big).\end{aligned}\tag{78}$$

From (50) we can show by induction that $\boldsymbol{\Sigma}_{\mathbf{x}}(n)\mathbf{s}_1=0$ which suggests that

$$\begin{aligned}&\operatorname{trace}\big\{\boldsymbol{\Sigma}_{\mathbf{x}}(\infty)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\big\}\\ &\quad=\operatorname{trace}\big\{\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\mathbf{x}}(\infty)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big\}=\operatorname{trace}\{\boldsymbol{\Sigma}_{\mathbf{x}}(\infty)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\}.\end{aligned}\tag{79}$$

With the assumption $\mathbf{R}\approx\mathbf{I}$ we conclude that $\mathbf{c}_o\approx\mathbf{s}_1$ which means that

$$\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}_o\approx\sigma^2\mathbf{s}_1,\quad \tilde{\mathbf{r}}^t(n)\mathbf{c}_o\approx\sigma\,\mathbf{n}^t(n)\mathbf{s}_1.\tag{80}$$

Taking traces in (78), using (79), (80) and the fact that $(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t)\mathbf{s}_1=0$, yields

$$\begin{aligned}\operatorname{trace}\{\boldsymbol{\Sigma}_{\mathbf{x}}(\infty)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\}&\approx\frac{\mu}{2}\operatorname{trace}\big\{\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\big(\sigma^2\tilde{\mathbf{S}}\tilde{\mathbf{A}}^2\tilde{\mathbf{S}}^t\\ &\quad+\sigma^4\mathbb{E}\big\{\mathbf{n}(n)\mathbf{n}^t(n)\big(\mathbf{n}^t(n)\mathbf{s}_1\big)^2\big\}\big)\\ &\quad\times\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\big\}\\ &=\frac{\mu}{2}\operatorname{trace}\big\{\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\big(\sigma^2\tilde{\mathbf{S}}\tilde{\mathbf{A}}^2\tilde{\mathbf{S}}^t+\sigma^4\mathbf{I}\big)\\ &\quad\times\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\big\}\\ &=\frac{\mu}{2}\left(\sigma^2\sum_{i=2}^{K}a_i^2+\sigma^4(N-1)\right).\end{aligned}\tag{81}$$

In a similar way we can show that

$$\operatorname{trace}\{\boldsymbol{\Sigma}_{\mathbf{y}}(\infty)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\}\approx\frac{\mu}{2}\left(\sum_{i=2}^{K}a_i^2+\sigma^2(N-1)\right).\tag{82}$$

Substituting in (77) yields the desired result. □

Proof of Theorem 3: To show the theorem, as in the trained case, we will disregard the last part of the recursion introduced to correct the non-robustness problem. Since we will use results from Stochastic Approximation Theory contained in [13, Pages 104–108], let us define certain quantities according to the notation used in this reference

$$H(\mathbf{c},\mathbf{r}(n))=\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\big(a_1\operatorname{sgn}\{\mathbf{c}^t\mathbf{r}(n)\}-\mathbf{c}^t\mathbf{r}(n)\big)\mathbf{r}(n)\tag{83}$$

$$h(\mathbf{c})=\mathbb{E}_{\mathbf{r}}\{H(\mathbf{c},\mathbf{r}(n))\}\tag{84}$$

$$R(\mathbf{c})=\mathbb{E}_{\mathbf{r}}\{H(\mathbf{c},\mathbf{r}(n))H^t(\mathbf{c},\mathbf{r}(n))\}.\tag{85}$$

where $\mathbf{c}$ is a deterministic vector and $\mathbb{E}_{\mathbf{r}}\{\cdot\}$ denotes expectation with respect to the data vector $\mathbf{r}(n)$.

In order to proceed we need a number of identities that we present without proof. Let z be a random variable and $\mathbf{z}$ a random vector that are both zero mean and jointly Gaussian and γ a constant, then

$$\mathbb{E}\{\operatorname{sgn}\{\gamma+z\}\mathbf{z}\}=\frac{\mathbb{E}\{z\mathbf{z}\}}{\sigma_z}\sqrt{\frac{2}{\pi}}\,e^{-\gamma^2/2\sigma_z^2}\tag{86}$$

$$\begin{aligned}\mathbb{E}\{|\gamma+z|\mathbf{z}\mathbf{z}^t\}&=\omega\mathbb{E}\{z\mathbf{z}\}\mathbb{E}\{z\mathbf{z}^t\}\\ &\quad+\left\{\gamma\left(1-2Q\left(\frac{\gamma}{\sigma_z}\right)\right)\right.\\ &\quad\left.+\sigma_z\sqrt{\frac{2}{\pi}}e^{-\gamma^2/2\sigma_z^2}\right\}\boldsymbol{\Sigma}_{\mathbf{z}}\end{aligned}\tag{87}$$

$$\mathbb{E}\{z^2\mathbf{z}\mathbf{z}^t\}=\sigma_z^2\boldsymbol{\Sigma}_{\mathbf{z}}+2\mathbb{E}\{z\mathbf{z}\}\mathbb{E}\{z\mathbf{z}^t\},\tag{88}$$

where $\sigma_z^2=\mathbb{E}\{z^2\}$, $\boldsymbol{\Sigma}_{\mathbf{z}}=\mathbb{E}\{\mathbf{z}\mathbf{z}^t\}$ and ω is a scalar quantity whose exact form is unimportant. Using (86) we can now compute $h(\mathbf{c})$ and we obtain

$$h(\mathbf{c})=-\rho\left(\frac{a_1}{\sqrt{\mathbf{c}^t\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}}}\right)\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{c}.\tag{89}$$

From [13] we then have that the mean trajectory satisfies the recursion

$$\bar{\mathbf{c}}(n)=\bar{\mathbf{c}}(n-1)+\mu h(\bar{\mathbf{c}}(n-1))\tag{90}$$

which is exactly (57).

To compute the steady state covariance matrix we need to find $R(\mathbf{c}_o)$. Again using the identities (86), (87), (88), assuming that $\mathbf{R}\approx\mathbf{I}$, which yields $\mathbf{c}_o\approx\mathbf{s}_1$ and $\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{s}_1\approx\sigma^2\mathbf{s}_1$, we have that

$$\begin{aligned}R(\mathbf{c}_o)&\approx\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\big(2a_1^2\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}-2a_1\mathbb{E}\big\{\big|a_1b_1(n)\\ &\quad+\mathbf{s}_1^t\tilde{\mathbf{r}}(n)\big|\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)\big\}\\ &\quad+\mathbb{E}\big\{\big(\mathbf{s}_1^t\tilde{\mathbf{r}}(n)\big)^2\tilde{\mathbf{r}}(n)\tilde{\mathbf{r}}^t(n)\big\}\big)\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\\ &=\left(\sigma^2+4Q\left(\frac{a_1}{\sigma}\right)-\frac{4}{\sqrt{2\pi}}a_1\sigma\,e^{-a_1^2/2\sigma^2}\right)\\ &\quad\times\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\big(\mathbf{I}-\mathbf{s}_1\mathbf{s}_1^t\big)\end{aligned}\tag{91}$$

Furthermore we need to compute

$$h_{\mathbf{c}}(\mathbf{c}_o) \approx \nabla_{\mathbf{c}} h(\mathbf{c})|_{\mathbf{c}=\mathbf{s}_1} = -\left(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\right) \times \left\{\rho\left(\frac{a_1}{\sigma}\right)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}} + \boldsymbol{\Sigma}_{\tilde{\mathbf{r}}}\mathbf{s}_1\left(\nabla_{\mathbf{c}}\rho(\cdot)|_{\mathbf{c}=\mathbf{s}_1}\right)^t\right\}$$
$$= -\left(\mathbf{I} - \mathbf{s}_1\mathbf{s}_1^t\right)\rho\left(\frac{a_1}{\sigma}\right)\boldsymbol{\Sigma}_{\tilde{\mathbf{r}}} \quad (92)$$

Substituting in the Lyapunov equation that determines the steady state covariance matrix of the estimates [13, page 107]

$$h_{\mathbf{c}}(\mathbf{c}_o)\boldsymbol{\Sigma}_{\mathbf{c}}(\infty) + \boldsymbol{\Sigma}_{\mathbf{c}}(\infty)j_{\mathbf{c}}^t(\mathbf{c}_o) + \mu R(\mathbf{c}_o) = 0, \quad (93)$$

and taking traces yields (59). □

Note

1. With lower case letters we denote scalars, with bold face lowercase, vectors and with boldface upper case, matrices.

References

1. M. Honig and H.V. Poor, "Adaptive Interference Suppression," in *Wireless Communications: Signal Processing Perspectives*, H.V. Poor and G.W. Wornell (Eds.), Upper Saddle River, NJ: Prentice Hall, 1998, Ch. 2, pp. 64–128.
2. U. Madhow and M. Honig, "MMSE Interference Suppression for Direct Sequence Spread Spectrum CDMA," *IEEE Trans. Commun.*, vol. COM-42, Dec. 1994, pp. 3178–3188.
3. U. Madhow, "Blind Adaptive Interference Suppression for Direct-Sequence CDMA," in *Proceedings of IEEE*, Oct. 1998, vol. 86, no. 10, pp. 2049–2069.
4. S.L. Miller, "An Adaptive Direct Sequence Code Division Multiple Access Receiver for Multiuser Interference Rejection," *IEEE Trans. Commun.*, vol. COM-33, 1995, pp. 1746–1755.
5. H.V. Poor and S. Verdú, "Probability of Error in MMSE Multiuser Detection," *IEEE Trans. Inf. Theory*, vol. 43, no. 2, May 1997, pp. 858–871.
6. S. Roy, "Subspace Blind Adaptive Detection for Multiuser CDMA," *IEEE Trans. Commun.*, vol. 48, no. 1, Jan. 2000, pp. 169–175.
7. J.B. Schodorf and D.B. Douglas, "A Constraint Optimization Approach to Multiuser Detection," *IEEE Trans. Sign. Process.*, vol. SP-45, no. 1, Jan. 1997, pp. 258–262.
8. X. Wang and H.V. Poor, "Blind Multiuser Detection: A Subspace Approach," *IEEE Trans. Inform. Theory*, vol. IT-44, no. 2, March 1998, pp. 667–690.
9. Z. Xie, R.T. Short, and C.K. Rushforth, "A Family of Suboptimum Detectors for Coherent Multiuser Communications," *IEEE J. Selec. Areas Commun.*, vol. 8, May 1990, pp. 683–690.
10. S. Verdú, *Multiuser Detection*, New York: Cambridge University Press, 1998.
11. G.V. Moustakides and H.V. Poor, "On the Relative Error Probabilities of Linear Multiuser Detectors," *IEEE Trans. Inform. Theory*, vol. 47, no. 1, Jan. 2001, pp. 450–456.
12. M. Honig, U. Madhow, and S. Verdú, "Blind Multiuser Detection," *IEEE Trans. Inform. Theory*, vol. IT-41, July 1995, pp. 944–960.
13. A. Benveniste, M. Métivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximation*. Berlin: Springer, 1990.
14. J.A. Bucklew, T.G. Kurtz, and W.A. Sethares, "Weak Convergence and Local Stability Properties of Fixed Step Size Recursive Algorithms," *IEEE Trans. Inform. Theory*, vol. IT-39, no. 3, May 1993, pp. 966–978.
15. G.V. Moustakides, "Locally Optimum Signal Processing Adaptive Estimation Algorithms," *IEEE Trans. Signal Process.*, vol. SP-46, no. 12, Dec. 1998, pp. 3315–3325.
16. V. Solo and X. Kong, *Adaptive Signal Processing Algorithms, Stability and Performance*, Englewood Cliffs, NJ: Prentice Hall, 1995.
17. S. Haykin, *Adaptive Algorithms*, 3rd edn., Upper Saddle River, NJ: Prentice Hall, 1997.
18. R. Lupas and S. Verdú, "Linear Multiuser Detectors for Synchronous Code-Division Multiple-Access Channels," *IEEE Trans. Inform. Theory*, vol. IT-35, Jan. 1989, pp. 123–136.

George V. Moustakides was born in Greece, in 1955. He received the diploma in Electrical Engineering from the National Technical University of Athens, Greece, in 1979, the M.Sc in Systems Engineering from the Moore School of Electrical Engineering, University of Pennsylvania, Philadelphia, in 1980, and the Ph.D. in Electrical Engineering and Computer Science from Princeton University, Princeton NJ, in 1983. From 1983 to 1986 he held a research position at INRIA, Rennes, France and from 1987 to 1990 a research position at the Computer Technology Institute of Patras, Patras, Greece. From 1991 to 1996, he was Associate Professor at the Computer Engineering and Informatics department (CEID) of the University of Patras and in 1996 he became a Professor at the same department. Since 2001 he is with INRIA, Rennes, France as research director of the Communication Signal Processing group and part time professor at CEID. His interests include optical communications, multiuser detection, adaptive algorithms and sequential change detection.
moustaki@ceid.upatras.gr