Blind Array Channel Division Multiple Access (AChDMA) for Mobile Communications

Victor A. N. Barroso, *Member, IEEE*, José M. F. Moura, *Fellow, IEEE*, and João Xavier, *Student Member, IEEE*

***Abstract*—The paper introduces array channel division multiple access (AChDMA), which is a new blind algorithm for advanced SDMA in mobile communications systems. As an SDMA technique, AChDMA increases the system capacity by improving its time and frequency reuse. Being a blind algorithm, it requires no training sequences, previously known directions of arrival, or user codes.**

AChDMA separates the moving sources by tracking their multipath configuration and resolving their distinct generalized steering vectors. It maximizes a finite mixture log-likelihood function, combining an efficient initialization procedure with an EM-based algorithm that provides fast convergence to the global maximum. AChDMA reconstructs the mobile data sequences using only internal variables of the EM algorithm. These characteristics and its parallel structure make AChDMA suitable for real-time mobile communications. We test AChDMA with synthetic data in a number of different scenarios, illustrating the ability of the blind algorithm to separate and track in time the moving sources, and showing its good performance in a variety of practical situations.

I. Introduction

THE RAPID evolution of DSP technology has enlarged the application of software radio architectures to advanced communication systems [12], namely, in the design of adaptive base station arrays. Array processing techniques have the ability to cancel undesirable interferences, thus reducing the frequency reuse distance and improving the spectral efficiency [31] and system capacity [5], [25] of cellular radio systems [14], [24], [29]. Array processing can alleviate the impact of multipath that typically affects both radio and underwater acoustic communications [3], [6], [7]. Several references have addressed this problem in code division multiple access (CDMA) and space division multiple access (SDMA) systems, [4], [8], [13], [21]. These CDMA schemes use the prior knowledge of the users's codes as reference signals. In addition, CDMA is robust to multipath effects, exhibiting higher capacity than existing frequency division multiple access (FDMA) and time division multiple access (TDMA) systems. This is achieved at the cost of higher receiver complexity, e.g., requiring code synchronization.

Manuscript received December 1, 1995; revised September 26, 1997. This work was supported in part by FEDER and PRAXIS XXI under Contract PRAXIS/3/3.1/TPR/23/94, by Darpa under AFOSR Grant F49620-96-1-0436, and by NATO Collaborative Research Grant CRG971184. The associate editor coordinating the review of this paper and approving it for publication was Dr. Guanghan Xu.

V. A. N. Barroso and J. Xavier are with Instituto Superior Técnico, Instituto de Sistemas e Robótica, Lisbon, Portugal.

J. M. F. Moura is with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, 15213 USA.

Publisher Item Identifier S 1053-587X(98)01990-4.

As described in the literature, SDMA improves system capacity, in particular, when combined with TDMA [11], [16], [22]. There are basically two classical approaches in SDMA: multibeam antenna based SDMA and pure SDMA based on adaptive beamforming capable of tracking the mobile sources. In this case, to acquire the channels, previous estimates of the directions of arrival (DOA's) are used (DOA-based SDMA) or known training sequences are transmitted. More recently, several blind methods have been proposed to implement advanced SDMA systems [1], [17], [18], [26], [30].

Rather than combating multipath as an hindrance, in this paper, we develop a new blind algorithm for multiple access systems that exploits the distinct multipath configuration of each mobile user. We call it array channel division multiple access (AChDMA). AChDMA makes full use of the spatial diversity (array) available to the receiver (as SDMA systems do) and of the temporal diversity (multipath) present in the channel. We will show that AChDMA performs well with nonstationary channels of mobile users, outperforming other blind SDMA methods.

AChDMA exploits the fact that each mobile receiver geometry gives rise to a distinct multipath configuration. This configuration is captured by the source receiver generalized steering vector. In the last decade, a similar situation has been explored in underwater acoustics localization, where combining coherently the energy in the multiple paths leads to significant gains in the localization ability of sonars. This is usually referred to as matched field processing, see [2], [20], and [23] and references therein. Likewise, by resolving the multipath configuration, i.e., the generalized steering vectors, AChDMA enables mobiles to transmit with the same code simultaneously in time and using the same frequency carrier. The array receiver that we develop controls, at the physical layer, distinct nonstationary virtual channels defined on the same cell. This is done by blind separation of the different propagation channels that result from the multipath structure corresponding to each mobile.

The two major tasks and algorithms in AChDMA are

1) blind estimation of the multiuser array/channel transfer function;
2) optimal reconstruction of the data streams transmitted by the mobile sources.

The blind estimation of the array channel transfer function is achieved by maximizing a finite mixture log-likelihood model for the array data. AChDMA combines an initialization procedure with an EM-type algorithm to optimize efficiently this log likelihood. It converges fast to its global maximum. The

convergence properties of AChDMA compare favorably with those of the ILSP/ILSE method proposed in [26]. Moreover, our algorithm has the ability to track in real time the array channels.

The optimal reconstruction of the data stream is cast as a multiple hypotheses test. We interpret the resulting solution as a multichannel beamforming operation based on the minimum noise power distortionless response criterion.

The paper is organized as follows. Section II establishes the array data model, focusing on the particular structure of the array/channel transfer function and pointing out the cluster structure of the array data. Section III addresses the problem of data reconstruction based on the ML estimates of the array channels: Section III-A presents the EM algorithm that computes the ML estimates of the array channels. Section III-B develops the initialization procedure. Section III-C reconstructs the multiple sources data symbols and interprets this as a beamforming operation. Section III-D discusses system implementation issues for AChDMA. In Section IV, we evaluate the performance of the AChDMA scheme. We illustrate with synthetic data that AChDMA tracks properly the time varying array channels, and we compute the average probability of error for several scenarios, comparing these average values with their theoretical values. Finally, Section V summarizes the results and concludes the paper.

II. Array Data Model

Consider P independent binary sources, where each one generates the baseband signal

$$s_p(t) = \sum_{k=-\infty}^{+\infty} b_p(k) u(t - kT) \tag{1}$$

where

$u(\cdot)$	unit energy pulse;
T	baud period;
$\{b_p(k)\}_{k=-\infty}^{+\infty}$	binary sequence of independent and equally like ± 1 symbols generated by the source p.

This signal modulates a radio carrier of frequency ω_0. The resulting binary phase shift keying (BPSK) waveform is received by a uniform linear array of N identical omnidirectional sensors. At the array sensor n, the received source signal p has the complex envelope

$$\begin{aligned}\overline{s}_{p_n}(t) = \sum_{m=1}^{M_p} \alpha_{mp} e^{-j\omega_0 \tau_{mp}} \\ s_p(t - \tau_{mp} - (n-1)\Delta_{mp}) e^{-j(n-1)\omega_0 \Delta_{mp}}\end{aligned} \tag{2}$$

where

M_p	number of propagation paths;
α_{mp}	corresponding attenuations;
τ_{mp}	corresponding travel time path delays;
Δ_{mp}	intersensor propagation delay

$$\Delta_{mp} = \frac{d}{c} \sin(\theta_{mp}).$$

Here, θ_{mp} is the angle of arrival of ray m for source p, d is the sensor separation, and c is the propagation velocity. Using the fact that the transmitted signal is narrowband and assuming a microcell-based system where the largest propagation delay is small when compared with T, we write

$$\overline{s}_{p_n}(t) = \sum_{m=1}^{M_p} \gamma_{mp} a_n(\theta_{mp}) s_p(t)$$

where $a_n(\theta_{mp}) = e^{-j(n-1)\omega_0 \Delta_{mp}}$, and $\gamma_{mp} = \alpha_{mp} e^{-j\omega_0 \tau_{mp}}$. We represent by $\boldsymbol{a}(\theta_{mp})$ the array steering vector for the angle of arrival θ_{mp} with nth element $a_n(\theta_{mp})$.

From (2), the array/channel transfer function for the pth source, i.e., the transfer function between the source p and the receiving array of N sensors, is given by the N-dimensional complex vector

$$\boldsymbol{h}_p = \sum_{m=1}^{M_p} \gamma_{mp} \boldsymbol{a}(\theta_{mp}). \tag{3}$$

Assuming ideal matched filtering and ideal sampling of the array data and using (1)–(3), we can write the output of the array with N sensors as the N-dimensional complex vector

$$\boldsymbol{z}(k) = \boldsymbol{H}\boldsymbol{s}(k) + \boldsymbol{w}(k) \tag{4}$$

where

$$\boldsymbol{H} = [\boldsymbol{h}_1, \cdots, \boldsymbol{h}_p] \tag{5}$$

is an $(N \times P)$ complex matrix representing the channel transfer function for the multisource (P sources)/array (N sensors) configuration, and

$$\boldsymbol{s}(k) = [b_1(k), \cdots, b_P(k)]^T \tag{6}$$

is the P-dimensional vector collecting the binary symbols transmitted by each source and that are to be reconstructed. In (6), $(\cdot)^T$ denotes vector or matrix transpose. The noise vector $\boldsymbol{w}(k)$ is modeled as a complex zero mean white Gaussian process with covariance matrix $\sigma^2 \boldsymbol{I}_N, \boldsymbol{I}_N$ being the N-dimensional identity matrix.

In the following subsection, we emphasize the specific structure of the array data.

A. Geometric Structure of the Data Space

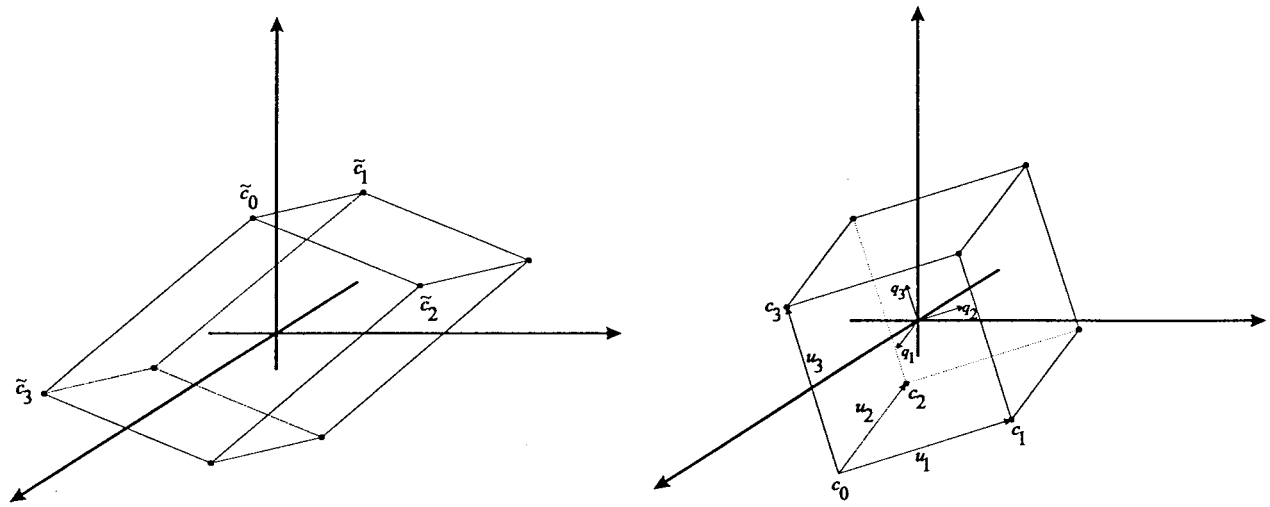
From (6), we conclude that the source vector $\boldsymbol{s}(k)$ takes values on a finite alphabet $\mathcal{A}$ with cardinality $C_{\mathcal{A}} = 2^P$. All the symbols $\boldsymbol{s}_i \in \mathcal{A}$ occur with equal probability $C_{\mathcal{A}}^{-1}$. Each $\boldsymbol{s}_i$ is a vertex of an hypercube in a $C_{\mathcal{A}}$-dimensional space. Adjacent vertices differ in a single component (± 1) and are a distance of two apart. For each snapshot k, $\boldsymbol{s}(k) = \boldsymbol{s}_i$ for a particular $\boldsymbol{s}_i \in \mathcal{A}$, and (4) takes the form

$$\boldsymbol{z}(k) = \boldsymbol{H}\boldsymbol{s}_i + \boldsymbol{w}(k), \qquad \boldsymbol{s}_i \in \mathcal{A}. \tag{7}$$

For low noise, the vector $\boldsymbol{z}(k)$ is close to $\boldsymbol{H}\boldsymbol{s}_i$. With K snapshots, the array data $\{\boldsymbol{z}(k)\}_K$ is grouped in clusters with centers

$$\check{\boldsymbol{c}}_i = \boldsymbol{H}\boldsymbol{s}_i, \qquad \boldsymbol{s}_i \in \mathcal{A}. \tag{8}$$

In array data space, the center $\check{\boldsymbol{c}}_i$ of each of these clusters is one of the $C_{\mathcal{A}}$ vertices of a hyperparallelipiped as represented in



(a) (b)

Fig. 1. (a) Array data geometry. (b) Signal space geometry.

Fig. 1(a). This hyperparallelipiped results from the geometric transformation performed by the channel transfer function $\boldsymbol{H}$ on the P-dimensional hypercube corresponding to the alphabet $\mathcal{A}$ of the sources. As observed before, this hypercube in symbol space also has $C_{\mathcal{A}}$ vertices, where the distance between any two adjacent vertices is equal to 2. However, the transformation $\boldsymbol{H}$ is not isometric; therefore, the centers of adjacent clusters in array data space are no longer equidistant. Because we do not have any prior knowledge about the parameters that determine the geometry of this parallelipiped in array data space, from the point of view of channel identification, it is important to preprocess the array data and map it into a different data set whose geometry reflects better the geometry of the symbol space. We find this transformation now.

We first compute the covariance $\boldsymbol{R}$ of the array data. Since the binary vectors $\boldsymbol{s}_i$ are zero mean, independent, and identically distributed, from (6) and (7), it follows that

$$\boldsymbol{R} = \boldsymbol{H}\boldsymbol{H}^H + \sigma^2 \boldsymbol{I}_N \tag{9}$$

where $(\cdot)^H$ denotes transpose conjugate.

We next find the spectral decomposition of $\boldsymbol{R}$. For $N > P$ and assuming that $\boldsymbol{H}$ has rank P

$$\boldsymbol{R} = [\boldsymbol{V}\boldsymbol{U}] \begin{bmatrix} \boldsymbol{\Lambda}^2 + \sigma^2 \boldsymbol{I}_P & \boldsymbol{0} \\ \boldsymbol{0} & \sigma^2 \boldsymbol{I}_{N-P} \end{bmatrix} \begin{bmatrix} \boldsymbol{V}^H \\ \boldsymbol{U}^H \end{bmatrix} \tag{10}$$

where $\boldsymbol{\Lambda}^2$ is a P-dimensional diagonal matrix whose elements are the nonzero eigenvalues of $\boldsymbol{H}\boldsymbol{H}^H$, and $[\boldsymbol{V}\boldsymbol{U}]$ is a unitary matrix formed by the eigenvectors of $\boldsymbol{R}$. The columns of $\boldsymbol{V}$ and $\boldsymbol{U}$ span the signal subspace and the noise subspace, respectively. Under the above conditions, it readily follows from (10), as is well known in the array processing literature, that $N = P + 1$ is the smallest value of N for which it is possible to estimate the noise variance σ^2. Increasing the number N of array elements with respect to the number P of independent sources results in improved estimates of the noise variance. Other side effects of this increase will be discussed later.

We are now ready to determine the transformation we are looking for. Clearly, in the noiseless case, the matrix

$$\hat{\boldsymbol{H}}_0 = \boldsymbol{V}\boldsymbol{\Lambda} \tag{11}$$

equals $\boldsymbol{H}$ up to a $(P \times P)$ unitary matrix $\boldsymbol{Q}$:

$$\boldsymbol{H} = \hat{\boldsymbol{H}}_0 \boldsymbol{Q}. \tag{12}$$

While $\hat{\boldsymbol{H}}_0$ can be computed from the array data using the sample covariance matrix as an estimate for $\boldsymbol{R}$, the matrix $\boldsymbol{Q}$ is unknown. We transform the array data according to

$$\boldsymbol{x}(k) = \hat{\boldsymbol{H}}_0^+ \boldsymbol{z}(k) \tag{13}$$

where $(\cdot)^+$ computes the pseudo-inverse of a matrix. Using (7) in (13) and evaluating the pseudo-inverse $\hat{\boldsymbol{H}}_0^+$ from (12), it is easy to verify that, for the noiseless case, the centers of the clusters given by (8) in array data space are mapped in the transformed data space to

$$\boldsymbol{c}_i = \boldsymbol{Q}\boldsymbol{s}_i, \qquad \boldsymbol{s}_i \in \mathcal{A}. \tag{14}$$

From (13) and (14), we deduce important properties for our model. Since $\boldsymbol{Q}$ is a unitary matrix, the centers $\boldsymbol{c}_i$ given by (14) in the transformed data space are, again, the vertices of a hypercube. This hypercube is a rotation by $\boldsymbol{Q}$ of the original hypercube defined by the alphabet of the sources [see Fig. 1(b)]. It follows that, in the transformed data space, the distance between any two adjacent centers $\boldsymbol{c}_i$ equals 2. As we will show later, this is important because to identify $\boldsymbol{Q}$ and, thus, $\boldsymbol{H}$ through (12), we just need to estimate the orientation of the hypercube in Fig. 1(b). Besides this, the transformation (13) eliminates the noise components in the orthogonal space of $\boldsymbol{H}$. This means that the transformed data $\boldsymbol{x}(k)$ has a higher SNR when compared with that of the original data $\boldsymbol{z}(k)$.

These properties of the data play a critical role in the derivation of the algorithm that estimates $\boldsymbol{H}$; see Section III. Beyond that, the clustered structure of the array data shows that given $\boldsymbol{H}$, the reconstruction of $\boldsymbol{s}(k)$ is formulated as a $C_{\mathcal{A}}$-ary detection problem of *known symbols in white Gaussian noise*. We address this problem first in the next section.

III. AChDMA: Multichannel Blind Estimation

To reconstruct the symbol sequence, AChDMA has to detect the symbols $\{\boldsymbol{s}(k)\}$ transmitted in K snapshots of the array data $\{\boldsymbol{z}(k)\}$. The vector sequence $\{\boldsymbol{s}(k)\}$ is reconstructed by solving the multiple hypotheses test [28]

$$H_i\colon \boldsymbol{z}(k) = \boldsymbol{H}\boldsymbol{s}_i + \boldsymbol{w}(k), \quad \boldsymbol{s}_i \in \mathcal{A} \qquad k = 1, \cdots, K. \tag{15}$$

The vector $\boldsymbol{z}(k)$ represents the array data at time instant k, given that $\boldsymbol{s}_i \in \mathcal{A}$ was transmitted.

The channel transfer function $\boldsymbol{H}$ is unknown in the mobile radio communication environments of interest. We adopt a generalized maximum-likelihood (GML) approach, whereby we first estimate the channel transfer function $\boldsymbol{H}$ and then solve the multiple hypotheses test problem. The conditional probability density function (pdf) of the array data $\boldsymbol{z}(k)$ at time instant k parameterized by the channel transfer function $\boldsymbol{H}$ and given that $\boldsymbol{s}(k) = \boldsymbol{s}_i \in \mathcal{A}$ was transmitted is

$$p(\boldsymbol{z}(k)|\boldsymbol{H}, \boldsymbol{s}_i) = (\pi\sigma^2)^{-N} \exp\left(-\frac{\|\boldsymbol{z}(k) - \boldsymbol{H}\boldsymbol{s}_i\|^2}{\sigma^2}\right). \tag{16}$$

Assuming that the transmitted symbols are statistically independent, the pdf of the single observation at time k conditioned on the channel transfer function is

$$\sum_{i=1}^{C_{\mathcal{A}}} p(\boldsymbol{z}(k)|\boldsymbol{H}, \boldsymbol{s}_i) P(\boldsymbol{s}(k) = \boldsymbol{s}_i) \tag{17}$$

where $P(\boldsymbol{s}(k) = \boldsymbol{s}_i)$ is the probability of occurrence of symbol $\boldsymbol{s}_i$. This is a mixture model. With white Gaussian noise measurements, the joint log-likelihood function (LLF) is the Kth-order product of the pdfs (17). The GML detector maximizes this LLF over all possible $\boldsymbol{H}$ and $\{\boldsymbol{s}(k)\}$. We consider in detail in Sections III-A and III-B the ML estimation of $\boldsymbol{H}$, and then, in Section III-C, we consider the reconstruction of the symbol sequences. In Section III-D, we make some additional comments regarding system aspect implementations of the AChDMA.

A. Multichannel Blind Estimation: EM Algorithm

Due to the structure of the array data model, it is easy to show from (16) and (17) that, given K snapshots of the array data, the ML estimate of $\boldsymbol{H}$ is

$$\hat{\boldsymbol{H}} = \arg\max_{\boldsymbol{H}} \sum_{k=1}^{K} \ln\left(\sum_{i=1}^{C_{\mathcal{A}}} \text{Gauss}_i(\boldsymbol{H}, \boldsymbol{z}(k))\right) \tag{18}$$

$$\text{Gauss}_i(\boldsymbol{H}, \boldsymbol{z}(k)) = \exp\left(-\frac{\|\boldsymbol{z}(k) - \boldsymbol{H}\boldsymbol{s}_i\|^2}{\sigma^2}\right). \tag{19}$$

The log-likelihood function to be maximized in (18) is a finite mixture probabilistic model. This mixture model is a consequence of the clustered structure of the data $\boldsymbol{z}(k)$. The strong nonlinearity of the log-likelihood precludes the analytical solution of the optimization in (18). Rather than performing the optimization over $\boldsymbol{H}$ directly, we could parameterize $\boldsymbol{H}$ in terms of the channel parameters, e.g., the path angles of arrival. However, this would lead to a multidimensional search over the parameter space, which is infeasible from a practical point of view. To overcome these difficulties, we develop an expectation-maximization (EM) [27] based algorithm to obtain the estimate $\hat{\boldsymbol{H}}$ of the channel $\boldsymbol{H}$. This results in an iterative algorithm that is known to converge to the true ML estimate.

1) EM Algorithm: We apply the EM algorithm to the estimation of the array/channel matrix transfer function $\boldsymbol{H}$. This is a blind procedure in the sense that it does not assume any prior knowledge about the channel parameters.

Using the array data model described in Section II, it is shown in Appendix A that the EM approach yields the iteration

$$\boldsymbol{H}_{l+1} = \left(\sum_{k=1}^{K} \boldsymbol{z}(k) \frac{\sum_{i=1}^{C_{\mathcal{A}}} \text{Gauss}_i(\boldsymbol{H}_l, \boldsymbol{z}(k)) s_i^H}{\sum_{i=1}^{C_{\mathcal{A}}} \text{Gauss}_i(\boldsymbol{H}_l, \boldsymbol{z}(k))}\right) \cdot \left(\sum_{k=1}^{K} \frac{\sum_{i=1}^{C_{\mathcal{A}}} \text{Gauss}_i(\boldsymbol{H}_l, \boldsymbol{z}(k)) s_i s_i^H}{\sum_{i=1}^{C_{\mathcal{A}}} \text{Gauss}_i(\boldsymbol{H}_l, \boldsymbol{z}(k))}\right)^{-1}. \tag{20}$$

To run this iterative algorithm, we must be aware of two important aspects.

- In general, this algorithm has a very slow convergence rate.
- Due to strong nonlinearities, it can easily get trapped into a local maximum.

These two aspects reflect the high sensitivity of the EM algorithm with respect to the initial estimate. Hence, special care has to be taken to determine a suitable initial estimate $\boldsymbol{H}_0$. In the next subsection, we develop our initialization procedure with which the EM algorithm converges in just a few iterations to the global maximum $\hat{\boldsymbol{H}}$ defined in (18) of the log-likelihood function.

B. Initialization of the EM Algorithm

We now develop the initialization procedure of the AChDMA algorithm. This specifies the initial condition for the EM algorithm of Section III-A. As it will be clarified in Section III-D, there are two modes for the operation of AChDMA, each corresponding to a different initialization. The first mode is the global initialization mode. This is the mode that AChDMA enters at startup and when there are major changes in the propagation environment (e.g., mobiles entering or leaving). After the global initialization mode, AChDMA switches to the tracking mode. In the tracking mode, the EM algorithm is restarted with the last available estimate of the channel transfer function; see Section III-D. We now explain the global initialization procedure.

1) Global Initialization: We use the geometric properties of the transformed data set defined in Section II-A. We compute the initial condition $\boldsymbol{H}_0$ in the global initialization mode by using (12), which is herein repeated:

$$\boldsymbol{H}_0 = \hat{\boldsymbol{H}}_0 \boldsymbol{Q} \tag{21}$$

where $\hat{\boldsymbol{H}}_0$ is obtained from (11) through the spectral factorization of the array data sample covariance matrix $\boldsymbol{R}$.

To completely specify the initial condition $\boldsymbol{H}_0$, $\boldsymbol{Q}$ remains to be estimated. We accomplish this by working with the transformed data set given by (13). We recall the discussion following (14). In the transformed data space and noiseless case, the centers $\boldsymbol{c}_i$ of the clusters are the vertices of an hypercube, where adjacent vertices are at a distance of 2.

Our strategy to estimate $\boldsymbol{Q}$ is to find in the space of the transformed data $\boldsymbol{x}(k)$ one set of $(P+1)$ adjacent clusters. Of course, there are as many of these sets as there are vertices in the hypercube, i.e., there are $C_{\mathcal{A}} = 2^P$ such sets. Finding one is enough to determine $\boldsymbol{Q}$, as we will see. We use a nearest neighborhood clustering algorithm to compute estimates of $P+1$ adjacent vertices $\boldsymbol{c}_0, \boldsymbol{c}_1, \cdots, \boldsymbol{c}_P$ of the P hypercube. This hypercube is represented for $P=3$ in Fig. 1(b). Notice that the performance of the clustering algorithm benefits from the prior knowledge that we have about the dimensions of the hypercube and the noise statistics. A similar approach is reported in [1], where the authors propose a clustering algorithm to determine all the $C_{\mathcal{A}}$ vertices of the parallelipiped in data space. The algorithm in [1] is much too demanding in terms of computational complexity and does not take advantage of the well-defined geometry in the signal space. From the estimates of any $P+1$ *adjacent* vertices, say, $\boldsymbol{c}_0, \boldsymbol{c}_1, \cdots, \boldsymbol{c}_P$, we compute the P vectors

$$\boldsymbol{u}_i = \boldsymbol{c}_i - \boldsymbol{c}_0, \qquad i = 1, \cdots, P. \tag{22}$$

In the noiseless case, replacing (14) in (22), we get

$$\boldsymbol{u}_i = \boldsymbol{Q}(\boldsymbol{s}_i - \boldsymbol{s}_0), \qquad i = 1, \cdots, P \tag{23}$$

where $\boldsymbol{s}_i$ and $\boldsymbol{s}_0$, because they are *adjacent* vertices of the sources's alphabet hypercube, only differ from each other in one element. This means that

$$\boldsymbol{u}_i = \pm 2\boldsymbol{q}_i, \qquad i = 1, \cdots, P \tag{24}$$

where $\boldsymbol{q}_i$ is one column of $\boldsymbol{Q}$ (not necessarily the ith column).

In general, and due to the presence of noise, the vectors $\{\boldsymbol{u}_i\}$ are not orthogonal. We apply the Gram–Schmidt algorithm to derive from the vectors $\boldsymbol{u}_i, i = 1, \cdots, P$ a set of orthonormal vectors $\boldsymbol{q}_1, \cdots, \boldsymbol{q}_P$ specifying the principal directions of the hypercube [see Fig. 1(b)]. These vectors can be taken as estimates of the columns of the unitary matrix $\boldsymbol{Q}$. For the noiseless case, as we have seen, this procedure obtains the matrix $\boldsymbol{Q}$ up to a permutation matrix (entries ± 1). The Gram–Schmidt algorithm acts like a normalization procedure and takes care of the factor 2. Thus, as claimed before, the vectors $\boldsymbol{q}_i$ give the matrix $\boldsymbol{Q}$ up to a permutation of its columns. In terms of the application that we are considering, this is not an obstacle. The ambiguity resulting from the permutation of the columns of $\boldsymbol{Q}$ is resolved by headers identifying each source, and the uncertainty about the sign is solved using differential encoding schemes.

The performance of this initialization scheme is evaluated in Section IV. Except for low noise situations, the estimate $\boldsymbol{H}_0$ given by (21) is not good enough to provide acceptable bit error rates. However, when used as an initial condition in the global initialization mode, it enables the EM algorithm to converge to the global maximum of the likelihood function with a small number of iterations. This behavior is maintained even for moderate SNR's. This will be shown by computer simulations in Section IV.

C. Reconstruction of Multiple Symbol Sequences

Assuming that the multisource array channel transfer function has been successfully estimated, the symbol sequence is reconstructed by maximizing the joint log-likelihood function. Under the symbol independence and white Gauss noise assumption, each symbol vector is detected by maximizing the pdf in (16). In fact, and after convergence of the EM algorithm, for each transmitted binary vector $\boldsymbol{s}_k, k = 1, 2, \cdots, K$, this maximization is equivalent to selecting, at each time k, the symbol $\boldsymbol{s}_i \in C_{\mathcal{A}}$ corresponding to the maximum mode $\text{Gauss}_i(\hat{\boldsymbol{H}}, \boldsymbol{z}(k))$ in (19).

We provide a useful interpretation of how AChDMA reconstructs the multiple source symbol sequence.

1) Beamforming Interpretation: When $\boldsymbol{H}$ is known, the solution to the multiple hypotheses detection problem given by (5) can be reduced to the minimization

$$\boldsymbol{s}(k) = \arg \min_{\boldsymbol{s}_i \in \mathcal{A}} \|\boldsymbol{z}(k) - \boldsymbol{H}\boldsymbol{s}_i\|^2. \tag{25}$$

This means that if $\boldsymbol{H}$ is known, we need to search in $\mathcal{A}$ only for the symbol $\boldsymbol{s}_i$ that in the space spanned by the columns of $\boldsymbol{H}$ minimizes the distance to the array data vector $\boldsymbol{z}(k)$.

We give a beamforming interpretation to the minimization defined in (25). When $\boldsymbol{H}$ is known, this minimization can be accomplished by the two step approach:

i) Compute an unconstrained solution for the minimization of $\|\boldsymbol{z}(k) - \boldsymbol{H}\boldsymbol{s}(k)\|^2$ in (25), here denoted by $\hat{\boldsymbol{s}}(k)$;

ii) Find the $\boldsymbol{s}_i \in \mathcal{A}$ closest to $\hat{\boldsymbol{s}}(k)$ in the metric of $\boldsymbol{H}^H\boldsymbol{H}$, i.e., for which

$$(\boldsymbol{s}_i - \hat{\boldsymbol{s}}(k))^H(\boldsymbol{H}^H\boldsymbol{H})(\boldsymbol{s}_i - \hat{\boldsymbol{s}}(k))$$

is minimum.

It is straightforward to verify that the unconstrained minimization step **i)** leads to

$$\hat{\boldsymbol{s}}(k) = (\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H\boldsymbol{z}(k). \tag{26}$$

This is identified as the output of the generalized multibeam minimum noise power (mnp) distortionless beamformer [9]. Let $\boldsymbol{G}$ be the $(N \times P)$ matrix specifying the mnp beamformer, i.e.,

$$\hat{\boldsymbol{s}}(k) = \boldsymbol{G}^H\boldsymbol{z}(k). \tag{27}$$

From (26), it follows that

$$\boldsymbol{G}^H = (\boldsymbol{H}^H\boldsymbol{H})^{-1}\boldsymbol{H}^H$$

minimizes the output noise power

$$\text{tr}(\sigma^2\boldsymbol{G}^H\boldsymbol{G})$$

subject to $\boldsymbol{G}^H\boldsymbol{H} = \boldsymbol{I}_P$.

As mentioned at the beginning of this section, the data symbols are estimated directly by maximizing the Gauss

modes so that the computations in (27) are not actually performed. However, the beamforming operation (27) implicit in (25) in the reconstruction of the data sequences provides an interesting interpretation of AChDMA, as we explain next.

The blind estimation of the array/channel transfer matrix $\boldsymbol{H}$ tracks the propagation channel used by each mobile source. The propagation channel used by each of these sources is represented in our model by a column of $\boldsymbol{H}$; see (3). The implicit beamforming operation described above separates these channels: Each generalized beam tracks all the multipath arrivals, not just the direct path, corresponding to a given mobile while canceling (attenuating) all the remaining signals.

D. AChDMA: System Considerations

There are several considerations that should be made to clarify the operation of AChDMA. The first set of comments regards the AChDMA suitability for real-time implementation. The second set of comments refers to the choice of values for some internal parameters and how these relate to real world operating environments.

1) Real-Time Implementation: Three properties of the EM algorithm are worth mentioning. The first is the fast convergence that it exhibits when combined with the initialization procedure of Section III-B. This fast convergence will be illustrated in Section IV, where we show that, usually, the EM algorithm converges in a couple of iterations. The second property is a direct consequence of the finite mixture model of the array data. It follows that each mode can be computed simultaneously in time. As a result, the right-hand side of (20) can be obtained using a parallel computational architecture. The third advantage is that symbol detection is achieved using internal variables of the EM algorithm—the Gauss modes in (19). This means that, in reconstructing the symbol sequence, there is no additional computational effort. In particular, as noted in Section III-C, the computations in (26) are not actually performed.

These three properties of the EM algorithm (fast convergence, parallel structure, and automatic sequence reconstruction) make the EM-based blind AChDMA suitable for real-time implementation.

2) Operating Conditions: We now consider how the rate of change of the propagation channel affects the choice of parameter values in AChDMA. The rate of change is determined by the maximum Doppler shift D present. In Section IV, we calculate this Doppler for realistic mobile radio communications scenarios. A typical value is $D = 50$ Hz. To represent well the channel variations, we need to estimate the channel response at twice this Doppler (Nyquist rate). This means that, at every sampling time $T_s = 1/(2D)$ s, the estimate of the channel transfer function $\boldsymbol{H}$ should be updated. In practice, we update this estimate more frequently, i.e., every $\Delta = \alpha T_s$, with α anywhere from one tenth to one half. During these Δ s, AChDMA assumes that the channel transfer function $\boldsymbol{H}$ does not change significantly. Let $K_{\max}$ be the number of array data samples in Δ s. If R_b is the system data rate, $K_{\max} = R_b\Delta$. Values for $K_{\max}$ depend strongly on R_b. For $R_b = 1$ Mb/s, a typical $K_{\max}$ is on the order of several thousands, whereas for $R_b = 25$ Kb/s, $K_{\max}$ will be on the order of 100. We now discuss the implications of D on the several stages of AChDMA.

3) Global Initialization: The apparent drawback of the global initialization procedure proposed in Section III-B is its relative computational complexity. The global initialization procedure requires the array data sample covariance matrix, which is then spectrally factorized. Say this matrix is computed from $K = K_{\text{cov}}$ samples. There are two limitations on K_{cov}. Clearly, $K_{\text{cov}} \leq K_{\max}$. The smaller K_{cov} is with respect to $K_{\max}$, the larger is the time available to perform the global initialization. The second limitation is less obvious and has to do with the number of sources P and the cardinality of the sources alphabet $C_{\mathcal{A}} = 2^P$. The number K_{cov} should be large compared with $C_{\mathcal{A}}$ so that with high probability, we can find $P+1$ adjacent clusters in the transformed data space. We will see in Section IV that these constraints are easily taken care of in practical scenarios. In the scenarios considered there, a reasonable value for $K_{\text{cov}} \leq 100$.

4) Tracking Mode: From a practical point of view, it is not necessary to reinitialize, at every $K_{\max}$ samples, the EM algorithm with the global initialization procedure of Section III-B. The changes in geometry between $K_{\max}$ adjacent blocks are small. This means that the EM algorithm can be restarted using the estimate of $\boldsymbol{H}$ obtained with the previous block of data, i.e., it operates in a *tracking* mode. Hence, the EM algorithm is seldom reinitialized with the global initialization procedure of Section III-B. Typically, this global initialization procedure is used only at startup and when the number of sources sharing the radio channel changes as mobiles enter or leave the cell.

5) EM Algorithm: The EM algorithm uses $K = K_{\text{em}}$ snapshots of the array data. This number K_{em} is again bounded above by $K_{\max}$. On the other hand, it should be large enough to provide an accurate estimate of $\boldsymbol{H}$ and large compared with $C_{\mathcal{A}}$. Again, these constraints are easily satisfied with realistic scenarios. In our simulations in Section IV, we found that $K_{\text{em}} \approx K_{\text{cov}}$ is quite reasonable.

6) Reconstruction of the Symbol Sequences: For each block of $K_{\max}$ samples, the symbols of the multiple P sources are detected using the estimate $\boldsymbol{H}$ obtained with the first K_{em} symbols.

IV. Performance Evaluation

In this section, we evaluate the performance of the AChDMA scheme proposed in the paper. First, we study the convergence properties of the EM algorithm, comparing it with the ILSP scheme proposed in [26]. Then, we present a study on the stability of the mobile communications channel, which is used to predict the channel estimates update rate. Finally, we focus on the capability of the EM algorithm in tracking the generalized steering vectors associated with the mobile sources and on the bit error rates (BER's) that are achieved by AChDMA.

1) System Parameters and Cell Geometry: The simulations are performed in the context of a cell in a cellular radio mobile communications system. The system operates at a

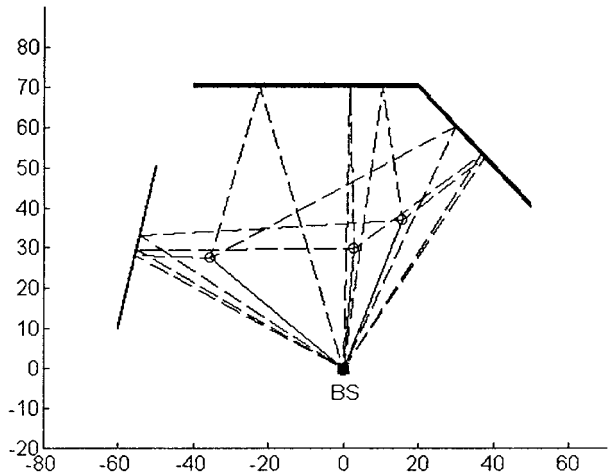


Fig. 2. Typical channel geometry in the simulated cell.

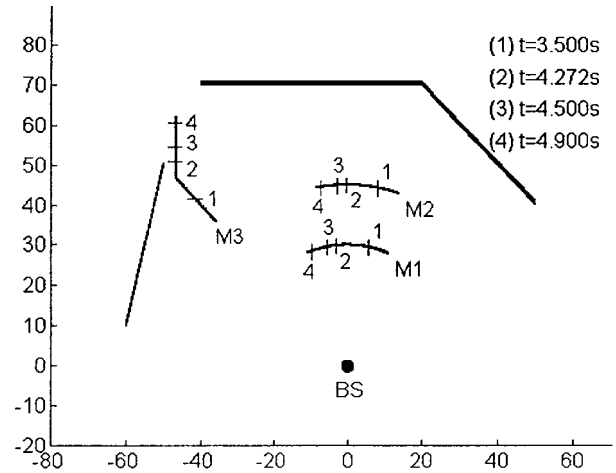


Fig. 3. Trajectories of the mobiles in the simulated scenario.

carrier frequency $f_0 = 1$ GHz and accommodates transmission rates of $R_b = 1$ Mb/s with differential encoding BPSK modulation format. These are in the range of typical values for the Universal Mobile Telecommunications System (UMTS) under study by several European research teams. We will also consider a data rate of $R_b = 25$ Kb/s. The mobiles present in the cell generate signals with unit power that are transmitted simultaneously in time and in the same frequency band.

The geometry of the cell is in Fig. 2. There are three reflectors, each one absorbing half of the impinging power, and the array is placed at point (0,0) in the diagram of Fig. 2. For each mobile, we assume one reflection at each reflector. The array is linear and uniform, aligned with the horizontal axis. We will consider arrays with $N = 4, 8,$ and 11 identical omnidirectional captors separated by half wavelength $\lambda/2$.

2) Scenario with Three Mobiles: Two mobiles M_1 and M_2 describe circular counterclock wise trajectories around the receiver with the same velocity 40 Km/h. These speeds are common in city traffic with slow automobiles circling a rotary. The radii of these trajectories are 30 m for mobile M_1 and 40 m for mobile M_2. A third mobile M_3 is getting out of the rotary along a radial trajectory with a velocity 55 Km/h (see Fig. 3). The signal for mobile M_3 experiences significant attenuation. The simulations last for 2 s. Notice that during the simulation, mobile M_1 overcomes M_2. This leads to geometries where the direct paths have practically the same direction, arriving at the array at essentially the same angle. These are hard geometries to handle by DOA-based SDMA since the two mobiles fall in the same sector of the array.

We estimate the Doppler shift for the velocities and carrier frequency indicated above. The Doppler is particularly sensitive to radial velocities and to large number of reflected rays. For the scenario of Fig. 3, the Doppler is not negligible, in particular, M_3 has a radial trajectory and, due to the number of *reflected* rays associated with each mobile, this is also true for mobiles M_1 and M_2. For example, with the 55 Km/h radial speed of M_3, the Doppler shift is approximately 50 Hz. The channel sampling time is then about $T_s = 10$ ms. In the studies below, we variously update the channel estimates at Δ intervals ranging from 1–10 ms.

Our results will show that AChDMA copes well with these situations, maintaining track of the array channels and keeping the mobile sources separate. This is a consequence of AChDMA processing the information provided in all paths of the multipath structure, whereas, essentially, DOA-based SDMA methods utilize only the direct path.

We now illustrate with this scenario and for two different SNR's the performance of the EM algorithm; in particular, we study the following points:

- convergence rate;
- implementation issues;
- channel stability;
- tracking performance;
- symbol error rates achieved.

3) Convergence of the EM Algorithm: We first evaluate the performance and the convergence rate of the EM algorithm, comparing it with ILSP [26]. As a performance measure, we use the squared error

$$\varepsilon = \sum_{p=1}^{P} ||\boldsymbol{h}_p - \hat{\boldsymbol{h}}_p||^2$$

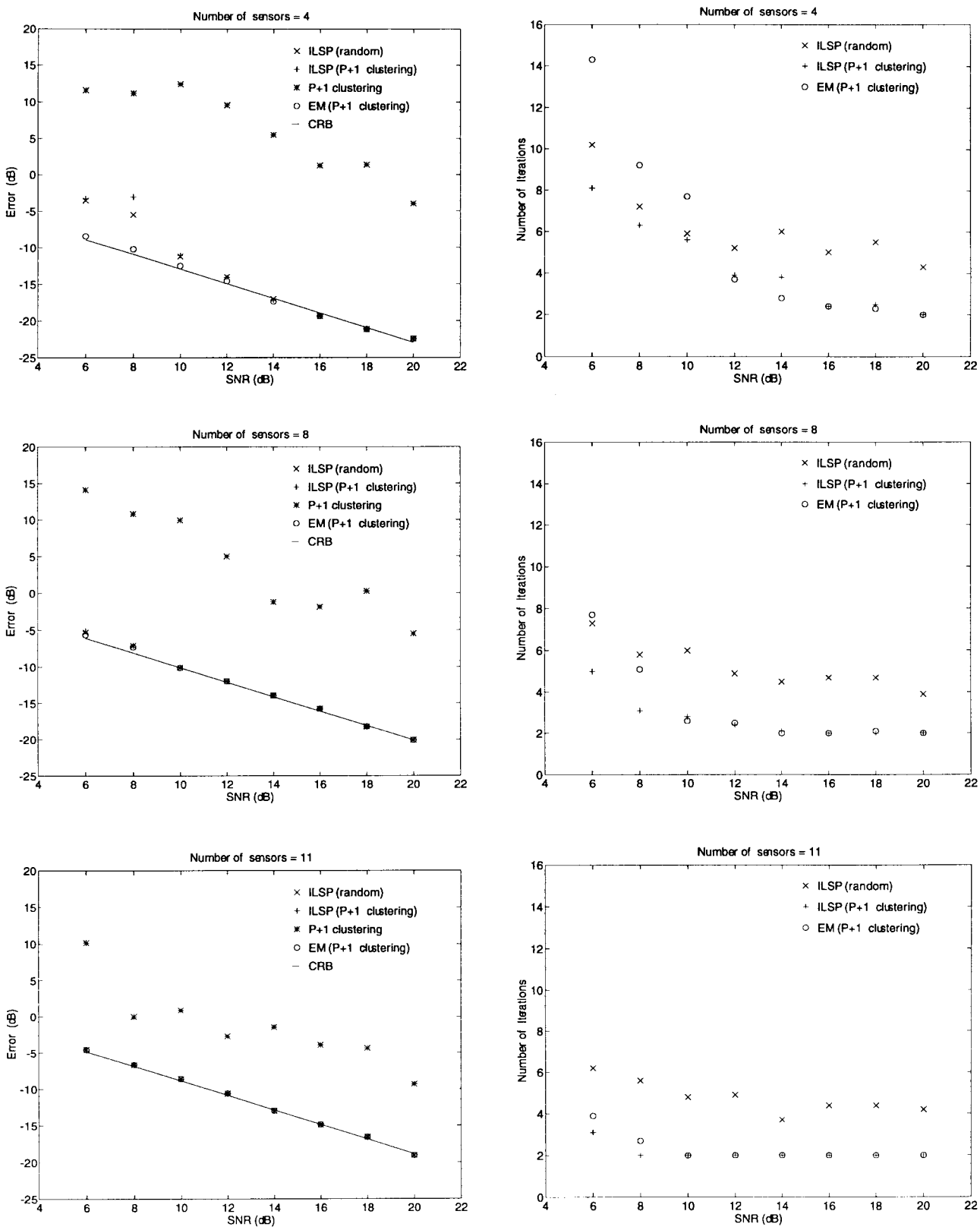
as a function of the global SNR

$$\text{SNR} = \frac{\sum_{p=1}^{P} ||\boldsymbol{h}_p||^2}{N\sigma^2}.$$

For this problem, we derive the Cramér-Rao Bound (CRB), assuming that at each time instant, the transmitted symbol is known. In this case, it is easy to compute that

$$\text{CRB} = P\frac{N\sigma^2}{K}. \tag{28}$$

The plots in Fig. 4(a) and (b) study the convergence of the EM and the ILSP algorithms, respectively, with synthetic data. Different values for the number of sensors are considered:



(a) (b)

Fig. 4. (a) Variance of the channel estimation error. (b) Convergence rate.

$N = 4, 8, 11$. We generated randomly the initial conditions for ten different scenarios. These are defined in Table I in polar coordinates. The frame of reference is that of Fig. 2, where we present the typical channel geometry. For each of the ten scenarios, we computed the matrix $\boldsymbol{H}$ and the values of σ^2 that span the range of values used for the SNR. We then

TABLE I
DOA'S OF THE DIRECT PATHS RANDOMLY GENERATED FOR SCENARIOS 1–10

	1	2	3	4	5	6	7	8	9	10
M_1 $(d = 30m)$	92.5^o	84.5^o	41.0^o	112.2^o	5.1^o	59.3^o	120.7^o	125.5^o	18.9^o	41.4^o
M_2 $(d = 40m)$	154.9^o	142.1^o	174.9^o	109.8^o	136.5^o	76.0^o	39.9^o	145.9^o	134.9^o	41.3^o
M_3 $(d = 45m)$	79.8^o	67.1^o	31.1^o	5.1^o	128.6^o	121.0^o	147.2^o	21.1^o	112.0^o	18.8^o

calculated the corresponding CRB's from (28), as well as the channel estimates provided by each one of the algorithms.

Fig. 4 presents the results averaged over these ten scenarios. Fig. 4(a) plots, as a function of the SNR, the CRB and the error (in decibels) for four different algorithms:

1) the clustering global initialization procedure described in Section III, which is identified as "$P+1$ clustering";
2) the ILSP algorithm when initialized randomly, which is identified as "ILSP(random)";
3) ILSP when initialized with the clustering algorithm, which is identified as "ILSP($P+1$ clustering)";
4) the EM algorithm initialized by the clustering procedure, which is identified as "EM($P+1$ clustering),"

Except for the "$P+1$ clustering" procedure, the remaining algorithms were stopped when the absolute difference of the estimates provided by two consecutive iterations was smaller than a given threshold 10^{-7}. This gives a way of counting the number of iterations necessary for each algorithm to lock in a minimum of the estimation error.

From the plots in Fig. 4(a), we can conclude that the "$P+1$ clustering" initialization procedure alone (top curve in the plot) yields poor results when compared with the CRB. However, it initializes efficiently both the ILSP and the EM algorithm since the corresponding error curves match well with the CRB. The "EM($P+1$ clustering)" algorithm achieves a smaller CRB than ILSP, especially when the number of sensors is small ($N = 4$) and at lower values of SNR. "ILSP(random)," as proposed by [26], also matches well with the CRB. The clustering initialization procedure provides faster convergence than random initialization, as studied in Fig. 4(b). When initialized with the clustering algorithm of Section III, the ILSP and the EM algorithms converge to the CRB in fewer number of iterations [bottom two curves in Fig. 4(b)] than ILSP initialized randomly [top curve in Fig. 4(b)].

4) Implementation: When properly initialized with the clustering algorithm of Section III, both ILSP and EM have nice convergence properties. However, EM provides significant computational advantages over ILSP. ILSP is a sequential algorithm, whereas the EM iteration can be implemented using a parallel architecture. Moreover, the ILSP approach does not provide accurate symbol detection because it is necessary to run at least one iteration of the ILSE algorithm to achieve acceptable symbol error rates (SER) [26]. On the contrary, the EM scheme reconstructs the transmitted sequences without any additional computational effort, as discussed in Section III. We compare EM with ILSP/ILSE in terms of computation time by counting the number of complex multiplications. We count as one those multiplications that can be performed simultaneously when a parallel architecture is used for EM and ILSE. This study yields the ratio

$$\frac{T_{\text{ILSP/ILSE}}}{T_{\text{EM}}} = \frac{\mathcal{O}((3N_{it}+1)KP^2)}{\mathcal{O}(2N_{it}KP^2)}$$

where we used $N = P + 1 \simeq P$, and N_{it} stands for the number of iterations until convergence is achieved. Typically [see Fig. 4(b)], $N_{it} = 2$. This means that EM exhibits computational savings over ILSP/ILSE on the order of 50% in KP^2. Since, as argued before, $K \gg 2^P$, this gain may be significant when the number of sources is large.

5) Channel Stability: We study the stability of the channel as a function of the discretization interval Δ. This helps predicting the update rate for the channel estimates. There are two issues. The first is a consequence of the Doppler shifts caused by the motions. The second relates to the nonlinear effects induced by the time varying structure of the propagation channel as reflection paths are created or disappear.

As discussed at the beginning of this section, the maximum Doppler for the mobile scenario described above (see Fig. 3) is approximately $D = 50$ Hz. By the Nyquist criterion, the discretization interval Δ of the channel response is upper limited by the sampling interval $T_s = 1/(2D) = 10$ ms. We test this by considering four values for the discretization interval: $\Delta = 1, 2, 5, 10$ ms. For each instance, we monitor the second-order statistics for the errors associated with 1) the norms of the generalized steering vectors and 2) the variation of the orientations of each steering vector in data space during two successive time instants t and $t+1$, as measured by the cosine of the angles between $\boldsymbol{h}_p(t)$ and $\boldsymbol{h}_p(t+1)$. These statistics are averaged over 2 s. The results are in Table II.

We first consider the mean and variance of the estimates of the norms of the generalized eigenvectors. Table II(a) shows that these values remain practically constant over the range of discretization intervals tested. In other words, from the point of view of the norms of the generalized eigenvectors, there is no need to oversample the channel response beyond what is predicted by the Nyquist criterion.

Table II(b) shows the mean and variance for the directional variability parameters: the cosines of the angles of the generalized steering vectors at two successive time instants. For $\Delta \leq 5$ ms, these angles remain small over the entire range of discretization intervals tested. In fact, even at $\Delta = 10$ ms, only M_3 loses approximate co-linearity between channel updates. In this case, the problem is that during the 2-s averaging interval, the relative geometry associated with the trajectory of M_3 and the environment causes several discontinuities in the channel used by M_3 due to reflection paths that are lost. To keep track of the time-varying channel, we need to oversample at a higher rate than that predicted by the maximum Doppler shift. From Table II(b), oversampling by a factor of 2 ($\Delta = 5$ ms) seems to be sufficient. This will be confirmed in the

TABLE II
CHANNEL STABILITY STUDY. (a) AMPLITUDE VARIABILITY OF THE GENERALIZED STEERING VECTORS. (b) DIRECTIONAL VARIABILITY OF THE GENERALIZED STEERING VECTORS

	$\lvert h_1\rvert_{dB}$		$\lvert h_2\rvert_{dB}$		$\lvert h_3\rvert_{dB}$	
	Mean	Variance	Mean	Variance	Mean	Variance
$\Delta = 1\,msec$	0.0114	0.5429	−0.9293	1.7075	−3.1238	3.8293
$\Delta = 2\,msec$	0.0111	0.5425	−0.9289	1.7060	−3.1266	3.8365
$\Delta = 5\,msec$	0.0107	0.5418	−0.9288	1.7039	−3.1299	3.8488
$\Delta = 10\,msec$	0.0100	0.5407	−0.9286	1.7005	−3.1379	3.9057

(a)

	$cos\,(angle_1\,[t,t+1])$		$cos\,(angle_2\,[t,t+1])$		$cos\,(angle_3\,[t,t+1])$	
	Mean	Variance	Mean	Variance	Mean	Variance
$\Delta = 1\,msec$	0.9980	0.0000002	0.9977	0.0000006	0.9917	0.0000528
$\Delta = 2\,msec$	0.9919	0.0000033	0.9909	0.0000105	0.9674	0.0005641
$\Delta = 5\,msec$	0.9508	0.0001190	0.9448	0.0003839	0.8099	0.0158000
$\Delta = 10\,msec$	0.8241	0.0014000	0.8008	0.0048000	0.4463	0.0741000

(b)

sequel. In other words, and in conclusion, even for frequency nonselective channels (as assumed here), we may have to be moderately conservative when determining from the Doppler shift D the channel *estimate* update rates Δ, and therefore, the length $K_{\max}$ of the data blocks over which we can use the same estimated $\hat{\boldsymbol{H}}$.

6) Tracking Performance of the EM Algorithm: Here, we study the ability of the EM algorithm to track the time variations of the generalized steering vectors. We use the same simulation scenario as in the channel stability study described in the last paragraph. We conducted experiments with two values for SNR: SNR $= 10$ dB and SNR $= 4$ dB. These values were fixed at the initial time instant. For each experiment, the SNR varies as the mobile sources move along their respective trajectories. We use the same performance measures as before: the norm of the estimates of the generalized steering vectors and the cosine of the generalized angle formed by each column of $\boldsymbol{H}$ and its respective estimate. At the initial time instant, the EM algorithm was initialized using the clustering initialization procedure of Section III. This was done using 100 data points. At each subsequent update, the EM algorithm is initialized with the last available channel estimate (tracking mode). When running the EM algorithm, 100 data points were also used. The averages (mean and variance) were taken over the 2-s interval of the simulation. The results are in Table III, where, in the left column, Δ represents the update period.

Table III(a) and (b) presents the results when the initial SNR is 10 dB. It considers four values of $\Delta = 1, 2, 5, 10$ ms. In all situations, almost perfect channel tracking is achieved as demonstrated by the near zero mean and variance of the error of the estimated norm of the generalized steering vectors and of the estimated cosine of the generalized angles. The estimates of the generalized steering vectors are closely aligned with their respective true vectors. Similar conclusions hold when SNR $= 4$ dB and $\Delta \leq 5$ ms; see Table III(c) and (d).

During the experiments, it was never necessary to reinitialize the EM algorithm with the global procedure AChDMA running in tracking mode. On average, it took in between two and three iterations for the EM algorithm to converge. This means that our algorithm tracks channel fluctuations in real time. The results of these simulations confirm the results of the last paragraph, namely, that we can specify the channel update rates by studying the channel stability as a function of Δ.

7) Symbol Error Rate: Finally, we consider AChDMA in terms of symbol error rate (SER). We say that a P symbol is in error if at least one bit is changed. The simulation scenario is again that of Fig. 3. Fig. 5 shows the theoretical union bound for SER as a function of the minimum distances $(d_{\min})$ between any two vertices of the array data space parallelipiped (see Fig. 1) observed during the simulation. This is done for several values of $N = 4$, 8, 11. As before, we took the initial SNR's to be 10 and 4 dB. Recall that the SNR is defined as the sum of the SNR's associated with each source. Therefore, these values of SNR correspond in practice to moderate to small values of the individual SNR's. When SNR $= 10$ dB, the union bound predicts very good theoretical results, in particular, for the case where $N = 11$. With generality, we can say that for a given scenario, increasing N improves the performance. The fact is that the parallelipiped in array space suffers a volumetric expansion when N increases.[1] This is confirmed by the curves in Fig. 5, where the intervals spanned by $d_{\min}$ are shifted to the right as N becomes larger.

The curves in Fig. 5 also help predict the theoretical value of the outage probability. The outage probability is defined as the fraction of time for which the SER union bound is greater than a given threshold. For example, with $N = 11$ and SNR $= 10$ dB, the predicted outage probability is 0.137 for a threshold $= 10^{-6}$ and is negligible (i.e., 0) for a threshold $= 10^{-5}$. These are acceptable values for the outage probability, which can be significantly improved, even when SNR $= 4$ dB by using adequate error control schemes.

We now address the sensitivity of the receiver with respect to the channel complexity, i.e., the number of reflected paths per mobile. In Fig. 6(a), we consider 100 random locations

[1] The proof of this fact, being simple, is omitted here.

TABLE III
TRACKING ABILITY OF THE EM ALGORITHM. (a), (c) AMPLITUDE ERROR OF THE ESTIMATED GENERALIZED STEERING VECTORS; (b), (d) DIRECTIONAL ERROR OF THE ESTIMATED GENERALIZED STEERING VECTORS

	$\lvert h_1\rvert_{dB} - \lvert\hat{h}_1\rvert_{dB}$		$\lvert h_2\rvert_{dB} - \lvert\hat{h}_2\rvert_{dB}$		$\lvert h_3\rvert_{dB} - \lvert\hat{h}_3\rvert_{dB}$	
SNR = 10dB	Mean	Variance	Mean	Variance	Mean	Variance
$\Delta = 1\,msec$	−0.0050	0.0035	−0.0056	0.0044	−0.0131	0.0076
$\Delta = 2\,msec$	−0.0053	0.0033	−0.0067	0.0049	−0.0139	0.0082
$\Delta = 5\,msec$	0.0034	0.0039	−0.0068	0.0047	−0.0089	0.0080
$\Delta = 10\,msec$	−0.0071	0.0040	0.0017	0.0041	−0.0098	0.0076

(a)

	$cos\left(angle\left[h_1, \hat{h}_1\right]\right)$		$cos\left(angle\left[h_2, \hat{h}_2\right]\right)$		$cos\left(angle\left[h_3, \hat{h}_3\right]\right)$	
SNR = 10dB	Mean	Variance	Mean	Variance	Mean	Variance
$\Delta = 1\,msec$	0.9995	$0.0003e-4$	0.9994	$0.0007e-4$	0.9990	$0.0032e-4$
$\Delta = 2\,msec$	0.9995	$0.0003e-4$	0.9994	$0.0007e-4$	0.9990	$0.0035e-4$
$\Delta = 5\,msec$	0.9995	$0.0003e-4$	0.9994	$0.0007e-4$	0.9990	$0.0035e-4$
$\Delta = 10\,msec$	0.9995	$0.0002e-4$	0.9994	$0.0007e-4$	0.9990	$0.0028e-4$

(b)

	$\lvert h_1\rvert_{dB} - \lvert\hat{h}_1\rvert_{dB}$		$\lvert h_2\rvert_{dB} - \lvert\hat{h}_2\rvert_{dB}$		$\lvert h_3\rvert_{dB} - \lvert\hat{h}_3\rvert_{dB}$	
SNR = 4dB	Mean	Variance	Mean	Variance	Mean	Variance
$\Delta = 5\,msec$	−0.0015	0.0156	−0.0237	0.0186	−0.0356	0.0316

(c)

	$cos\left(angle\left[h_1, \hat{h}_1\right]\right)$		$cos\left(angle\left[h_2, \hat{h}_2\right]\right)$		$cos\left(angle\left[h_3, \hat{h}_3\right]\right)$	
SNR = 4dB	Mean	Variance	Mean	Variance	Mean	Variance
$\Delta = 5\,msec$	0.9981	$0.0046e-4$	0.9977	$0.0128e-4$	0.9959	$0.0549-4$

(d)

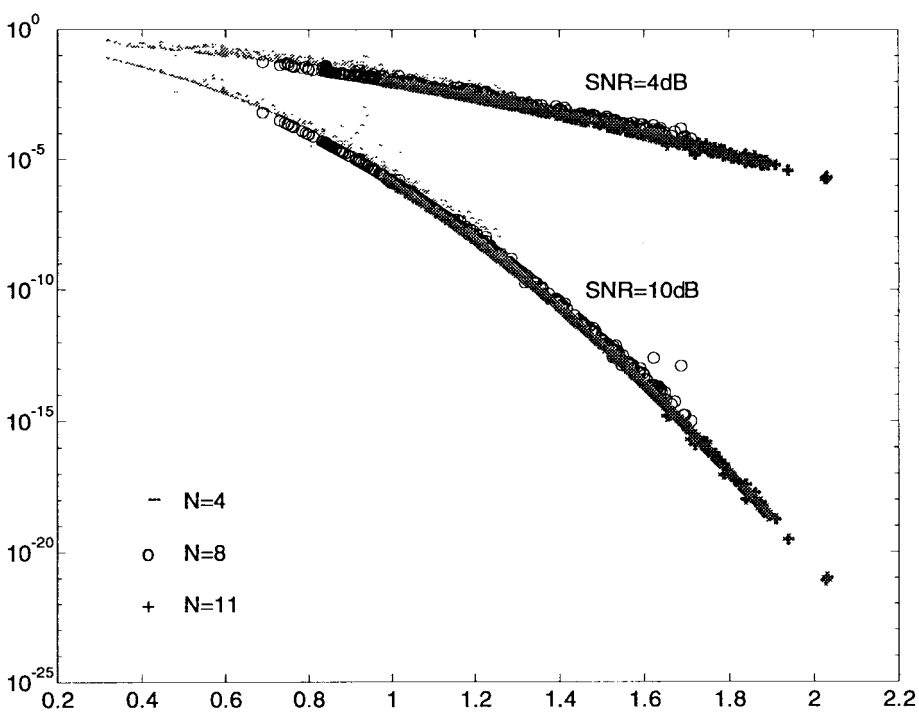
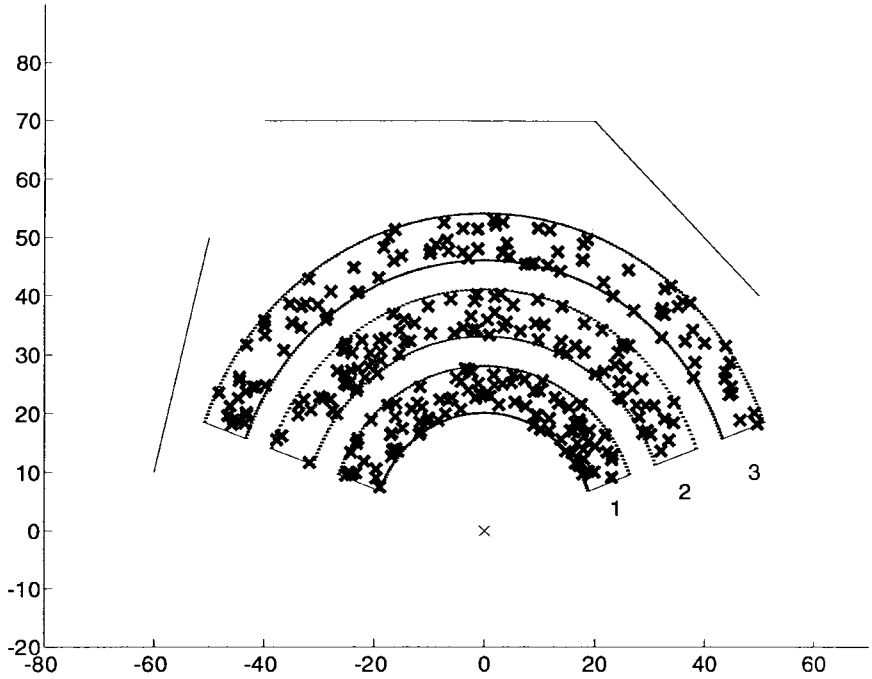
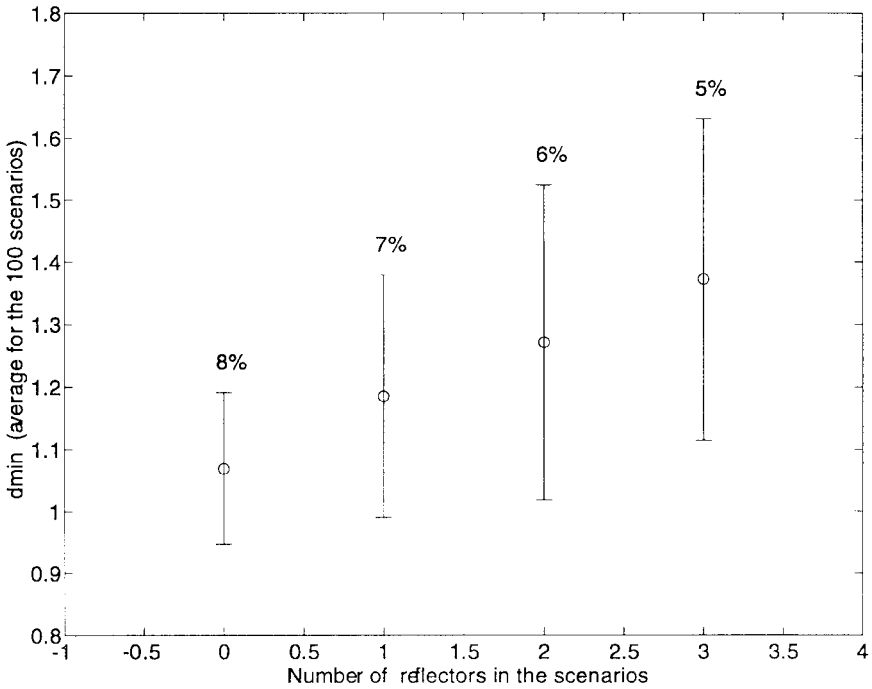


Fig. 5. Union bound of the symbol error probability.

of the mobiles. These are uniformly distributed along the circular intervals shown in the figure. For this set of random locations, we study four cases: zero, one, two, and three reflectors. The SNR is kept fixed at SNR = 10 dB. Fig. 6(b)



(a)



(b)

Fig. 6. Sensitivity to channel complexity. (a) Random location of mobiles. (b) Statistics of the minimum symbol separation in array space.

shows the second-order statistics of $d_{\min}$ obtained for the random scenarios described above as a function of the number of reflectors (the circles indicate the average of $d_{\min}$, and the vertical segments represent the corresponding standard deviation). As expected, and from the point of view of AChDMA, the reception conditions improve as the channel becomes more complex. This is confirmed by the percentage of situations for which the probability union bound is greater than 10^{-7}, which varies from 8% (line of sight) to 5% (three reflectors).

To conclude the performance evaluation of AChDMA, we study the impact on the SER of the rate at which we update the channel estimates. This rate is important because in between updates, we reconstruct the transmitted bit sequences assuming the channel remains invariant. Clearly, the longer the time between updates, the more significant are the changes in the channel. We selected four scenarios. These scenarios are started from the positions indicated by the numbers $1, \cdots, 4$ on the trajectories of the mobiles at the time instants indicated at the top right corner of Fig. 3, for example, scenario 1 starts at

TABLE IV
IMPACT OF CHANNEL ESTIMATES UPDATE RATES IN BER FOR FOUR SCENARIOS IN FIG. 3: SNR = 10 dB

Scenario 1	$BER_1(\%)$	$BER_2(\%)$	$BER_3(\%)$	$\hat{\sigma}^2$	$Upper\ bound\ ML\ (\hat{\sigma}^2)$
$[3.500, 3.501]$	0.0000	0.0000	0.0000	0.0200	$1.32e-10$
$[3.500, 3.502]$	0.0000	0.0000	0.0000	0.0231	$2.11e-09$
$[3.500, 3.503]$	0.0000	0.0000	0.0000	0.0281	$4.67e-08$
$[3.500, 3.504]$	0.0000	0.0000	0.0050	0.0349	$8.16e-07$
$[3.500, 3.505]$	0.0000	0.0000	0.2220	0.0431	$7.92e-06$

$Upper\ bound\ ML\ (\sigma^2) = 1.97e-11$

Scenario 2	$BER_1(\%)$	$BER_2(\%)$	$BER_3(\%)$	$\hat{\sigma}^2$	$Upper\ bound\ ML\ (\hat{\sigma}^2)$
$[4.272, 4.273]$	0.0000	0.0000	0.0000	0.0200	$5.93e-09$
$[4.272, 4.274]$	0.0000	0.0000	0.0000	0.0229	$4.98e-08$
$[4.272, 4.275]$	0.0000	0.0000	0.0000	0.0282	$7.84e-07$
$[4.272, 4.276]$	0.0000	0.0000	0.0425	0.0334	$4.98e-06$
$[4.272, 4.277]$	0.0000	0.0000	0.2100	0.0392	$2.26e-05$

$Upper\ bound\ ML\ (\sigma^2) = 2.07e-09$

Scenario 3	$BER_1(\%)$	$BER_2(\%)$	$BER_3(\%)$	$\hat{\sigma}^2$	$Upper\ bound\ ML\ (\hat{\sigma}^2)$
$[4.500, 4.501]$	0.0000	0.0000	0.0000	0.0239	$1.24e-08$
$[4.500, 4.502]$	0.0000	0.0000	0.0000	0.0260	$4.49e-08$
$[4.500, 4.503]$	0.0000	0.0000	0.0000	0.0295	$2.51e-07$
$[4.500, 4.504]$	0.0000	0.0000	0.0025	0.0343	$1.57e-06$
$[4.500, 4.505]$	0.0000	0.0000	0.0200	0.0402	$8.34e-06$

$Upper\ bound\ ML\ (\sigma^2) = 3.65e-09$

Scenario 4	$BER_1(\%)$	$BER_2(\%)$	$BER_3(\%)$	$\hat{\sigma}^2$	$Upper\ bound\ ML\ (\hat{\sigma}^2)$
$[4.900, 4.901]$	0.0000	0.0000	0.0000	0.0159	$1.74e-08$
$[4.900, 4.902]$	0.0000	0.0000	0.0000	0.0181	$1.03e-07$
$[4.900, 4.903]$	0.0000	0.0000	0.0000	0.0215	$9.69e-07$
$[4.900, 4.904]$	0.0000	0.0000	0.0275	0.0264	$8.52e-06$
$[4.900, 4.905]$	0.0000	0.0000	0.5240	0.0324	$5.03e-05$

$Upper\ bound\ ML\ (\sigma^2) = 4.68e-09$

$t = 3.5$ s. These scenarios have the following characteristics. In scenario 1, mobile M_3 is responsible for a severe Doppler shift; in scenario 2, one reflection associated with M_3 is lost, which corresponds to a channel discontinuity; in scenarios 3 and 4, the channel is more stable than in the previous cases, but the SNR associated with M_3 is very small.

The true channel matrix $\boldsymbol{H}$ is updated at the data rate, i.e., 1000 times per ms, whereas the estimates of $\boldsymbol{H}$ are computed using the first $K_{\text{em}} = 100$ data points. These estimates are then used to reconstruct the data blocks received during the remaining of the updating intervals. We considered the five cases $\Delta = 1, 2, 3, 4,$ or 5 ms.

Tables IV and V summarize the results obtained for SNR = 10 dB and SNR = 4 dB, respectively. The tables indicate the theoretical SER union bound computed at the initial time instant for each scenario. This bound cannot be achieved in practice since we keep the estimates of $\boldsymbol{H}$ constant, even though the channel fluctuates in between updates. As a result of these fluctuations, the clusters centered at the vertices of the parallelipiped specified by $\hat{\boldsymbol{H}}$ experience a larger dispersion. We model this dispersion by increasing the noise variance. Doing this, we can obtain estimates $\hat{\sigma}^2$ for the noise/channel mismatch process and use these estimates to evaluate the actual SER union bound. As might be expected and as seen from Tables IV and V, $\hat{\sigma}^2$ increases with Δ. In both tables, the right-hand side column shows the effect of this in the SER union bound. When SNR = 10 dB (see Table IV), acceptable values of the SER are achieved even for $\Delta = 5$ ms. The losses observed in this case are basically due to the channel time variation. When SNR = 4 dB (see Table V), the SER is practically insensitive to channel fluctuations, where the main factor responsible for poor performance is the low value of the SNR = 4 dB. We conclude that for moderate values of SNR $\simeq$ 10 dB, the AChDMA scheme proposed in this paper yields good performance in terms of the SER, even when the rate at which we update the channel estimates is close to the maximum predicted by the Doppler shift analysis. This is confirmed by Table IV. In terms of BER's, which are estimated by counting the errors in the reconstructed sequences, the corresponding values are extremely small, except for mobile M_3 when $\Delta = 5$ ms. For $\Delta \leq 4$ ms, we get for mobile M_3 that the $\text{BER}_3 \simeq 10^{-4}$ or smaller for all scenarios.

A distinct experiment was conducted under the same conditions but with a data rate of 25 Kbps. We let SNR = 10

TABLE V
IMPACT OF CHANNEL ESTIMATES UPDATE RATES IN BER FOR FOUR SCENARIOS IN FIG. 3: SNR = 4 dB

Scenario 1	$BER_1(\%)$	$BER_2(\%)$	$BER_3(\%)$	$\hat{\sigma}^2$	$Upper\ bound\ ML\ (\hat{\sigma}^2)$
$[3.500, 3.501]$	0.0000	0.0056	0.1083	0.0769	$5.38e-04$
$[3.500, 3.502]$	0.0000	0.0028	0.1750	0.0801	$6.43e-04$
$[3.500, 3.503]$	0.0000	0.0019	0.4704	0.0851	$6.79e-04$
$[3.500, 3.504]$	0.0000	0.0014	1.3111	0.0918	$1.39e-03$
$[3.500, 3.505]$	0.0011	0.0011	3.2872	0.0997	$2.07e-03$

$Upper\ bound\ ML\ (\sigma^2) = 4.68e-04$

Scenario 2	$BER_1(\%)$	$BER_2(\%)$	$BER_3(\%)$	$\hat{\sigma}^2$	$Upper\ bound\ ML\ (\hat{\sigma}^2)$
$[4.272, 4.273]$	0.0000	0.0000	0.1971	0.0779	$1.80e-03$
$[4.272, 4.274]$	0.0000	0.0000	0.4647	0.0811	$2.15e-03$
$[4.272, 4.275]$	0.0000	0.0000	1.3980	0.0863	$2.81e-03$
$[4.272, 4.276]$	0.0000	0.0015	2.2868	0.0912	$3.54e-03$
$[4.272, 4.277]$	0.0000	0.0024	3.4212	0.0968	$4.47e-03$

$Upper\ bound\ ML\ (\sigma^2) = 1.61e-03$

Scenario 3	$BER_1(\%)$	$BER_2(\%)$	$BER_3(\%)$	$\hat{\sigma}^2$	$Upper\ bound\ ML\ (\hat{\sigma}^2)$
$[4.500, 4.501]$	0.0000	0.0000	0.2364	0.0927	$1.86e-03$
$[4.500, 4.502]$	0.0000	0.0000	0.3303	0.0947	$2.05e-03$
$[4.500, 4.503]$	0.0000	0.0000	0.4727	0.0983	$2.43e-03$
$[4.500, 4.504]$	0.0000	0.0000	0.8439	0.1031	$2.97e-03$
$[4.500, 4.505]$	0.0000	0.0000	1.6315	0.1089	$3.71e-03$

$Upper\ bound\ ML\ (\sigma^2) = 1.88e-03$

Scenario 4	$BER_1(\%)$	$BER_2(\%)$	$BER_3(\%)$	$\hat{\sigma}^2$	$Upper\ bound\ ML\ (\hat{\sigma}^2)$
$[4.900, 4.901]$	0.0000	0.0053	0.2368	0.0614	$2.15e-03$
$[4.900, 4.902]$	0.0000	0.0053	0.3750	0.0635	$2.51e-03$
$[4.900, 4.903]$	0.0000	0.0044	0.6930	0.0671	$3.17e-03$
$[4.900, 4.904]$	0.0000	0.0046	1.6178	0.0719	$4.20e-03$
$[4.900, 4.905]$	0.0000	0.0063	3.9711	0.0775	$5.29e-03$

$Upper\ bound\ ML\ (\sigma^2) = 2.02e-03$

dB. During the 4 ms of the channel update interval, 100 data points are collected. Based on this data, as before, the channel estimates were computed, and the corresponding transmitted symbols reconstructed. We tested four independent runs. Each run lasts for 2 s, corresponding to 500 blocks of 4 ms of data. In each of the four independent runs, the EM algorithm was initialized with the global initialization procedure of Section III-B only once, when processing the first data block. AChDMA operates in tracking mode in the remaining of the experiment. The symbol error rate (SER) observed was 0% over the 200 000 transmitted bits. This shows that the AChDMA scheme is also efficient at moderately small transmission data rates, the important issue being the balance between the data rate, required channel update period, and number of users.

The experimental study confirms that AChDMA is a useful strategy to expand frequency or time reuse since, like other SDMA techniques, it enables several sources in a particular cell to share a common time/frequency channel. It has the ability to track in real time the channels present in the cell. Our results also show that when the update rates for the channel estimates are chosen in accordance with the measurements of the channel stability, good performance in terms of SER can be achieved even at moderate values of SNR.

V. CONCLUSIONS

This paper introduced AChDMA, a multiple access blind algorithm for SDMA. AChDMA exploits the multipath in the problem to separate the mobile sources. It is specially suited for nonstationary channels, which is typical in mobile communications systems.

We explained how AChDMA resolves its three major challenges:

1) estimation of the generalized steering vectors associated with the mobiles;
2) tracking dynamically these steering vectors as the sources move;
3) ability for real-time operation.

The sources are separated by an EM-based blind algorithm with a clustering initialization procedure described in Section III. Even at moderate SNR values, the EM algorithm converges globally in a few number of iterations and tracks satisfactorily the changes in the multipath structure induced by the varying

Doppler and the emergence/annihilation of reflected paths. The EM implementation of AChDMA is parallelizable and computationally efficient when compared with other blind SDMA schemes. Finally, AChDMA reconstructs the data from internal variables of the EM algorithm directly available after convergence has been achieved. These three properties of AChDMA (fast convergence, parallel structure, and automatic sequence reconstruction) make it suitable for real-time operation.

AChDMA as described requires $N \geq P+1$ array elements, where P is the number of mobiles being tracked. In practice, the number of array elements must be moderately higher so that good performance of AChDMA is achieved. Even for a significant number of mobile users, arrays of this size are not a problem since current integration technology of patch antenna elements [19] enables the manufacturing of small wavelength array antennas with a large number of elements. As a final comment, just like other SDMA schemes, AChDMA can be used in hybrid TDMA/FDMA cellular radio systems to increase in an efficient way the global capacity of the system.

Appendix A
EM Algorithm: Finite Gaussian Mixture Model

Using the EM approach, the ML estimate of $\boldsymbol{H}$ is obtained by maximization of the log-likelihood function of the complete data set $\boldsymbol{Y} = \{\boldsymbol{y}(k)\}_{k=1}^{K}$, where $\boldsymbol{y}(k) = [\boldsymbol{z}(k), \boldsymbol{x}(k)]^T, k = 1, \cdots, K$. Here, $\boldsymbol{Z} = \{\boldsymbol{z}(k)\}_{k=1}^{K}$ is the array data—see (15)—and $\boldsymbol{X} = \{\boldsymbol{x}(k)\}_{k=1}^{K}$ is the unobservable component. If $\boldsymbol{X}$ is properly chosen, it can lead to a maximization problem equivalent to that formulated in (18), whose solution is obtained iteratively. Since the complete data set is not available, the log-likelihood function $\ln p(\boldsymbol{Y}|\boldsymbol{H})$ is replaced by an average log-likelihood function obtained as follows. Let Ω be a set of candidate sequences (e.g., unknown binary data $\boldsymbol{s}_i$) $\boldsymbol{X}$ that could produce the observations $\boldsymbol{Z}$. A current guess $\boldsymbol{H}_l$ is used to define, for each possible sequence $\boldsymbol{X}$ in Ω, the pdf $p(\boldsymbol{X}|\boldsymbol{Z}, \boldsymbol{H}_l)$. With this, we can perform the E-step (expectation) of the EM algorithm yielding the average log-likelihood function

$$U(\boldsymbol{H}|\boldsymbol{H}_l) = \int_{\Omega} \ln(p(\boldsymbol{Y}|\boldsymbol{H}))p(\boldsymbol{X}|Z, \boldsymbol{H}_l)\, d\boldsymbol{X}. \tag{29}$$

The M-step (maximization) updates the current estimate of $\boldsymbol{H}$ according to

$$\boldsymbol{H}_{l+1} = \arg \max_{\boldsymbol{H}} U(\boldsymbol{H}|\boldsymbol{H}_l). \tag{30}$$

Under certain conditions, the EM iteration converges to the ML estimate based on the marginal likelihood function with respect to the observations data [10], [15]

$$\begin{aligned}\lim_{l\to\infty} \boldsymbol{H}_l &= \arg \max_{\boldsymbol{H}} \ln p(\boldsymbol{Z}|\boldsymbol{H}) \\ &= \arg \max_{\boldsymbol{H}} \left\{ \int_{\Omega} \ln p(\boldsymbol{Y}|\boldsymbol{H}) \right\}.\end{aligned}$$

Here, we make $\boldsymbol{X} = \{\boldsymbol{s}(k)\}_{k=1}^{K}$, where $\boldsymbol{s}(k)$ takes equally like values $\boldsymbol{s}_i \in \mathcal{A}$. It is then easy to show that

$$U(\boldsymbol{H}|\boldsymbol{H}_l) \propto -\sum_{k=1}^{H} \frac{\sum_{i=1}^{C_{\mathcal{A}}} \mathrm{Gauss}_i(\boldsymbol{H}_l, \boldsymbol{z}(k)) \|\boldsymbol{z}(k) - \boldsymbol{H}\boldsymbol{s}_i\|^2}{\sum_{i=1}^{C_{\mathcal{A}}} \mathrm{Gauss}_i(\boldsymbol{H}_l, \boldsymbol{z}(k))}. \tag{31}$$

Maximization of (31) with respect to the columns of $\boldsymbol{H}$ yields

$$\boldsymbol{H}_{l+1} = \left(\sum_{k=1}^{K} \boldsymbol{z}(k) \frac{\sum_{i=1}^{C_{\mathcal{A}}} \mathrm{Gauss}_i(\boldsymbol{H}_l, \boldsymbol{z}(k)) s_i^H}{\sum_{i=1}^{C_{\mathcal{A}}} \mathrm{Gauss}_i(\boldsymbol{H}_l, \boldsymbol{z}(k))} \right) \cdot \left(\sum_{k=1}^{K} \frac{\sum_{i=1}^{C_{\mathcal{A}}} \mathrm{Gauss}_i(\boldsymbol{H}_l, \boldsymbol{z}(k)) s_i s_i^H}{\sum_{i=1}^{C_{\mathcal{A}}} \mathrm{Gauss}_i(\boldsymbol{H}_l, \boldsymbol{z}(k))} \right)^{-1} \tag{32}$$

as claimed in (20). In (32), Gauss_i is as defined in (19).

References

[1] K. Anand, G. Mathew, and V. U. Reddy, "Blind separation of multiple co-channel BPSK signals arriving at an antenna array," *IEEE Signal Processing Lett.*, vol. 2, pp. 176–178, Sept. 1995.

[2] A. B. Baggeroer, W. A. Kuperman, and H. Schmidt, "Matched field processing: Source localization in correlated noise as an optimum parameter estimation problem," in *J. Acoust. Soc. Amer.*, no. 83, pp. 571–587, 1988.

[3] V. A. N. Barroso and J. M. F. Moura, "l_2 and l_1 beamformers: Recursive implementation and performance analysis," *IEEE Trans. Signal Processing*, vol. 42, pp. 1323–1334, June 1994.

[4] ——, "Beamforming with correlated arrivals in mobile communications," invited article in chapter on *Array Signal Processing*, M. Kaveh, Ed.,*Digital Signal Processing Handbook*, V. Madisetti and D. Williams, Eds. Boca Raton, FL: CRC, 1997.

[5] V. A. N. Barroso, M. J. Rendas, and J. Gomes, "Impact of array processing techniques on the design of mobile communication systems," in *Proc. MELECON*, Apr. 1994.

[6] V. A. N. Barroso and C. A. C. Belo, "A model based equalization structure for underwater communications," *Acoust. Signal Process. Ocean Exploration*, J. M. F. Moura and I. M. G. Lourtie, Eds., NATO ASI Series. Boston, MA: Kluwer, 1993.

[7] ——, "A channel matched receiver for underwater communications: Sensitivity to uncertainty in the ray arrival structure," in *Proc. Euro. Conf. Underwater Acoust.*, M. Weydert, Ed., 1992, pp. 80–83.

[8] T. Bull, M. Barret, and R. Arnott, "Technology in smart antennas for universal advanced roc. mobile infrastructure (TSUNAMI) overview," in *Proc. RACE Mobile Commun. Summit*, Cascais, Portugal, 1995, pp. 88–97.

[9] J. Capon, "High resolution frequency-wavenumber spectrum analysis," *Proc. IEEE*, vol. 57, pp. 1408–1418, Aug. 1969.

[10] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood estimation from incomplete data via the EM algorithm," *J. Roy. Stat. Soc.*, vol. B39, pp. 1–38, 1977.

[11] J. Fuhl and A. F. Molisch, "Capacity enhancement and BER in a combined SDMA/TDMA system," in *Proc. IEEE VTC*, Atlanta, GA, 1996, pp. 1481–1485.

[12] J. Kennedy and M. C. Sullivan, "Direction finding and smart antennas using software radio architectures," *IEEE Commun. Mag.*, vol. 33, pp. 62–68, May 1995.

[13] B. H. Khalaj, A. Paulraj, and T. Kailath, "Spatio-temporal channel estimation techniques for multiple access spread spectrum systems with antenna arrays," *Proc. IEEE ICC*, Seattle, WA, June 1995, pp. 1520–1524.

[14] W. C. Y. Lee, *Mobile Cellular Telecommunications Systems*. New York: McGraw-Hill, 1989.
[15] R. Little and D. Rubin, *Statistical Analysis with Missing Data*. New York: Wiley, 1987.
[16] G. Xu and S. Li, "Throughput multiplication of wireless LANS for multimedia services: SDMA protocol design," in *Proc. IEEE Globecom*, 1994.
[17] H. Liu and G. Xu, "A deterministic approach to blind symbol estimation," *IEEE Signal Processing Lett.*, vol. 1, pp. 205–207, Dec. 1994.
[18] ——, "Blind estimation of array responses for an asynchronous multiuser system," in *Proc. IEEE Veh. Technol. Conf.*, Chicago, IL, 1995, pp. 862–865.
[19] R. J. Mailloux, *Phased Array Antenna Handbook*. Norwood, MA: Artech House, 1994.
[20] J. M. F. Moura and M. J. D. Rendas, "Optimal filtering in the presence of multipath," in *Proc. ISIT, IEEE Int. Symp. Inform. Theory*, Brighton, U.K., June 1985. *Invited Session on Stochastic Processes for Underwater Signal Processing*, ch. 3, "Stochastic processes in underwater acoustics," C. R. Baker, Ed., *Lecture Notes in Control and Information Sciences*. New York: Springer Verlag, 1986, pp. 64–94.
[21] A. F. Naguib and A. Paulraj, "Recursive adaptive beamforming for wireless CDMA," in *Proc IEEE ICC*, Seattle, WA, June 1995, pp. 1520–1524.
[22] G. Okamoto and G. Xu, "Throughput multiplication of wireless LANS: Spread spectrum with SDMA," in *Proc. IEEE VTC*, Atlanta, GA, 1996, pp. 1371–1375.
[23] M. J. D. Rendas and J. M. F. Moura, "Cramér-Rao bounds for location systems in multipath environments," *IEEE Trans. Signal Processing*, vol. 39, pp. 2593–2610, Dec. 1991.
[24] R. Steele, *Mobile Radio Communications*. New York: IEEE-Pentech, 1992.
[25] S. Swales, M. Beach, D. Edwards, and J. McGreehan, "The performance enhancement of multibeam adaptive base station antennas for cellular mobile radio," *IEEE Trans. Veh. Technol.*, vol. 31, Feb. 1990.
[26] S. Talwar, M. Viberg, and A. Paulraj, "Bind estimation of multiple co-channel digital signals using an antenna array," *IEEE Signal Processing Lett.*, vol. 1, pp. 29–31, Feb. 1994.
[27] D. M. Titterington, A. F. M. Smith, and U. E. Markov, *Statistical Analysis of Finite Mixture Distributions*. New York: Wiley, 1985.
[28] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Part I*. New York: Wiley, 1968.
[29] J. Walker, *Mobile Information Systems*. New York: Artech House, 1990.
[30] A. van der Veen, S. Talwar, and A. Paulraj, "Blind estimation of multiple digital signals transmitted over FIR channels," *IEEE Signal Processing Lett.*, vol. 2, pp. 99–102, May 1995.
[31] P. Zettberg and B. Ottersten, "The spectrum efficiency of a base station antenna array system for spatially selective transmission," *IEEE Trans. Veh. Technol.*, vol. 44, pp. 651–660, Aug. 1995.

Victor A. N. Barroso (M'89) was born in Lisbon, Portugal, on December 22, 1952. He received the E.E. degree in 1976 and the Ph.D. degree in electrical and computer engineering in 1990, both from the Instituto Superior Técnico (IST), Lisbon, Portugal.

In 1976, he joined the faculty of the Department of Electrical and Computer Engineering at IST. He was promoted to Assistant Professor in 1990 and to Associate Professor in 1993. He has taught courses in systems and signal theory, control systems, signal and array processing, computer networks and communications. He is also a Researcher of the Signal and Image Processing Group at the Institute of Systems and Robotics (ISR), Lisbon, Portugal. His research interests are in statistical signal/image and array processing, communications, and underwater acoustics.

Dr. Barroso has been an associate editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING (Underwater Acoustics) since 1996. He is affiliated with several IEEE societies.

José M. F. Moura (S'71–M'75–SM'90–F'94) received the engenheiro electrotécnico degree in 1969 from Instituto Superior Técnico (IST), Lisbon, Portugal, and the M.Sc., E.E., and the D.Sc. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology (MIT), Cambridge, in 1973 and 1975, respectively.

He is presently a Professor of Electrical and Computer Engineering at Carnegie Mellon University (CMU), Pittsburgh, PA, which he joined in 1986. Prior to this, he was on the faculty of IST, where he was an Assistant Professor (1975), Professor Agregado (1978), and Professor Catedrático (1979). He has had visiting appointments at several Institutions, including MIT (Genrad Associate Professor of Electrical Engineering and Computer Science (from 1984 to 1986) and the University of Southern California, Los Angeles (research scholar, Department of Aerospace Engineering, Summers 1978 to 1981). His research interests include statistical signal processing (one- and two-dimensional), digital communications, image and video processing, radar and sonar, and multiresolution techniques. He has organized and codirected two international scientific meetings on signal processing theory and applications. He has over 180 published technical contributions and is co-editor of two books.

Dr. Moura is currently the Editor in Chief for the IEEE TRANSACTIONS ON SIGNAL PROCESSING and a member of the Underwater Acoustics and the Multimedia Signal Processing Technical Committees of the IEEE Signal Processing Society. He was a member of the IEEE Press Board from 1991 to 1995, a technical Associate Editor for the IEEE SIGNAL PROCESSING LETTERS from 1993 to 1995, and an Associate Editor for the IEEE TRANSACTIONS ON SIGNAL PROCESSING from 1988 to 1992. He was a program committee member for the IEEE International Conference on Image Processing (ICIP'95) and for the IEEE International Symposium on Information Theory (ISIT'93). He is a corresponding member of the *Academy of Sciences of Portugal* (Section of Sciences). He is affiliated with several IEEE societies, Sigma Xi, AMS, IMS, and SIAM.

João Xavier (S'97) was born in Bumba, Zaire, on February 2, 1972. He received the E.C.E. degree in 1995 and the M.Sc. degree in electrical and computer engineering in 1997, both from the Instituto Superior Técnico (IST), Lisbon, Portugal. Presently, he is a Ph.D. student at IST.

In 1997, he joined the faculty of the Department of Electrical and Computer Engineering at IST. He is with the Signal and Image Processing Group at the Institute of Systems and Robotics (ISR), Lisbon. His research interests are in statistical signal array processing, performance analysis, digital communications, and blind equalization.

Mr. Xavier is affiliated with the IEEE Signal Processing Society.