16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

MULTIUSER DETECTION WITH AN UNKNOWN NUMBER OF ACTIVE USERS: RECEIVER DESIGN

Daniele Angelosante[1], *Ezio Biglieri*[2], *and Marco Lops*[3]

[1] DAEIMI, Università di Cassino, Italy. email: d.angelosante@unicas.it
[2]Departament de Tecnologia, Universitat Pompeu Fabra, Barcelona, Spain. email: e.biglieri@ieee.org
[3]DAEIMI, Università di Cassino, Italy. email: lops@unicas.it

ABSTRACT

In multiuser detection, the set of users active at any time may be unknown to the receiver. In these conditions, optimum reception consists of detecting simultaneously the set of active users and their data, problem that can be solved exactly by applying random-set theory (RST) and Bayesian recursions (BR). However, implementation of optimum receivers may be limited by their complexity, which grows exponentially with the number of potential users. In this paper we examine three strategies leading to reduced-complexity receivers. In particular, we show how a simple approximation of BRs enables the use of Sphere Detection (SD) algorithm, which exhibits satisfactory performance with limited complexity.

1. INTRODUCTION

In multiuser detection (MUD), an important issue is that the set of users that are active at a given time may be unknown to the receiver. A simple solution to the problem of detecting data in a multiuser system with unknown number of active users consists of a two-step procedure, where multiuser detection is preceded by active-user identification [1]. However, the procedure in [1] can be applied for quasi-static network, wherein the number of users do not change often. In [2], the optimum solution to the problem of jointly estimating the set of active user and possibly their data is described: instead of proceeding in two steps, the number of active users and data are detected simultaneously. Indeed, modelling the set of active users as a random finite set (RFS) [3] allows the description of the number of users and their parameters as a single parameter set. Several scenarios can be considered: in the simplest among them, no a priori information about user activity is available, and maximum likelihood detection is used. When a priori information is available in the form of the probability that a user is active, maximum a posteriori (MAP) detection can be performed. If, in addition, a dynamic model is available for users logging in and out of the system, the use of random-set theory allows one to describe the evolution of the a posteriori probability of the set of active users and their data. In order to apply to real systems the concepts described above, it is necessary to design receivers whose performance is close to optimum, while retaining a reasonable complexity. The goal of this paper is to present techniques allowing a low complexity approximation of BRs. Three such techniques are considered. The first two are heuristic approximations of BRs and of the subsequent detection problem, while the third, based on what we call a *zero-order approximation* of BRs, applies the Sphere Detection (SD) algorithm. This paper is organized as follows. Section 2 describes the signal model. Section 3 introduces two heuristic estimation procedures aimed at approximating the optimum detector. Section 4 briefly describes the SD algorithm and how it can be applied to our problem. Numerical results are presented in Section 5, and conclusions are drawn in Section 6.

2. SIGNAL MODEL

In this section we review the MUD approach based on random-set theory (RST), as first advocated in [2].

We assume a random number of users transmitting digital data over a common channel. We denote by K the maximum number of active users, and by $\mathbf{s}(\mathbf{x}_t^{(k)})$ the signal transmitted at discrete time t by the kth user, if active. Each signal has in it a number of known parameters, reflected by a deterministic function $\mathbf{s}(\cdot)$, and a number of random parameters, summarized by $\mathbf{x}_t^{(k)}$. The observed signal at time t, denoted $\mathbf{y}_t$, includes $\mathbf{s}(\mathbf{x}_t^{(k)})$, the signals generated by the users active at t, which are in a random number, and stationary random noise $\mathbf{z}_t$. Thus,

$$\mathbf{y}_t = \sum_k \mathbf{s}(\mathbf{x}_t^{(k)}) + \mathbf{z}_t \tag{1}$$

Let $\mathcal{X}_t$ denote the random-set encapsulating what is unknown about the active users. We write

$$\mathcal{X}_t = \bigcup_{k=1}^{K} \mathcal{X}_t^{(k)} \tag{2}$$

where $\mathcal{X}_t^{(k)}$ is the singleton-or-empty set

$$\mathcal{X}_t^{(k)} = \begin{cases} \{\mathbf{x}_t^{(k)}\} = \{[k, x_t^{(k)}]^T\} & \text{if user } k \text{ is active at time } t \\ \emptyset & \text{otherwise} \end{cases} \tag{3}$$

Here, $\{\mathbf{x}_t^{(k)}\}$ is a singleton whose element is the vector containing the user index k and an unknown (possibly random) parameter $x_t^{(k)}$. The latter takes values in the finite set $\mathcal{M}$, with cardinality $|\mathcal{M}| = M$, representing the digital data transmitted by user k at time t. Moreover, if $|\mathcal{M}| = 1$ the system will operate in trained fashion. In this case the goal of our algorithm will be the estimation of the random set of active users.

Under the assumption of direct-sequence code-division multiple-access (DS-CDMA) with signature sequences of length $N \geq K$, and of zero-mean additive white Gaussian noise with power spectral density $N_0/2$, we can write, for the sufficient statistics of the received signal at time t,

$$\mathbf{y}_t = \mathbf{S}\mathbf{A}\mathbf{x}_t + \mathbf{z}_t \tag{4}$$

where $\mathbf{y}_t$ is the N-dimensional column vector of the observations, $\mathbf{S} \triangleq [\mathbf{s}_1, \ldots, \mathbf{s}_K]$ is a $N \times K$ matrix whose columns contain the signature sequences of all the potential K users, $\mathbf{A}$ is the $K \times K$ diagonal matrix of the user amplitudes, and $\mathbf{x}_t = \mathbf{x}_t(\mathcal{X}_t)$ is a mapping between the RFS $\mathcal{X}_t$ and a K-vector whose kth entry is defined as

$$\{\mathbf{x}_t\}_k = \begin{cases} 0 & \text{if } \mathcal{X}_t^{(k)} = \emptyset \\ x_t^{(k)} & \text{otherwise} \end{cases} \tag{5}$$

Assuming further that the system is power-controlled, i.e., $\mathbf{A} = \mathbf{I}_K$, where $\mathbf{I}_K$ denotes the K-dimensional identity matrix, we obtain

$$f_{\mathbf{Y}_t|\mathcal{X}_t}(\mathbf{y}_t|\mathcal{X}_t) = \frac{1}{\sqrt{\pi N_0}} \exp\{-\|\mathbf{y}_t - \mathbf{S}\mathbf{x}_t\|^2/N_0\} \tag{6}$$

Consider now a dynamic model for the users logging in and out of a multiuser communication system [2]. Denote by $\mathcal{X}_t$ the random set whose elements are the active users with their data at time t, and consider its evolution with time. We assume that from $t-1$ to t some new users log in, while some old users log out. We write

$$\mathcal{X}_t = \mathcal{S}_t \cup \mathcal{N}_t \tag{7}$$

where $\mathcal{S}_t$ is the set of *surviving* users still active from $t-1$, and $\mathcal{N}_t$ is the set of *new* users becoming active at t. The condition $\mathcal{N}_t \cap \mathcal{X}_{t-1} = \emptyset$ is forced, because a user ceasing transmission at time $t-1$ cannot reenter the set of active users at time t.

Suppose that there are k active users at $t-1$, i.e. $\mathcal{X}_{t-1} = \bigcup_{j=1}^{k} \mathcal{X}_{t-1}^{(j)}$. Then we may write, for the set of surviving users,

$$\mathcal{S}_t = \bigcup_{i=1}^{k} \mathcal{S}_t^{(i)} \tag{8}$$

where $\mathcal{S}_t^{(i)}$ denotes either an empty set (if user i has become inactive) or the singleton $\{\mathbf{x}_t^{(i)}\}$ (user i is still active). Let μ denote the "persistence" probability, i.e., the probability that a user survives from $t-1$ to t. We obtain, for the conditional probability of $\mathcal{S}_t$ given that $\mathcal{X}_{t-1} = \mathcal{B}$:

$$f_{\mathcal{S}_t|\mathcal{X}_{t-1}}(\mathcal{C} \mid \mathcal{B}) = \begin{cases} M^{-|\mathcal{C}|}\mu^{|\mathcal{C}|}(1-\mu)^{|\mathcal{B}|-|\mathcal{C}|}, & \mathcal{C} \subseteq \mathcal{B} \\ 0, & \mathcal{C} \nsubseteq \mathcal{B} \end{cases} \tag{9}$$

Denote α the probability that a new user arises. Then, a reasonable model is

$$f_{\mathcal{N}_t|\mathcal{X}_{t-1}}(\mathcal{C} \mid \mathcal{B}) = \begin{cases} M^{-|\mathcal{C}|}\alpha^{|\mathcal{C}|}(1-\alpha)^{K-|\mathcal{B}|-|\mathcal{C}|}, & \mathcal{C} \cap \mathcal{B} = \emptyset \\ 0, & \mathcal{C} \cap \mathcal{B} \neq \emptyset \end{cases} \tag{10}$$

Finally, by assuming that births and deaths of users are conditionally independent given $\mathcal{X}_{t-1} = \mathcal{B}$, the pdf of the union of the independent random sets $\mathcal{S}_t$ and $\mathcal{N}_t$ is obtained from the *generalized convolution* [3]

$$\begin{aligned} f_{\mathcal{X}_t|\mathcal{X}_{t-1}}(\mathcal{C} \mid \mathcal{B}) &= \sum_{\mathcal{W} \subseteq \mathcal{C}} f_{\mathcal{S}_t|\mathcal{X}_{t-1}}(\mathcal{W} \mid \mathcal{B}) f_{\mathcal{N}_t|\mathcal{X}_{t-1}}(\mathcal{C} \setminus \mathcal{W} \mid \mathcal{B}) \\ &= f_{\mathcal{S}_t|\mathcal{X}_{t-1}}(\mathcal{C} \cap \mathcal{B}) f_{\mathcal{N}_t|\mathcal{X}_{t-1}}(\mathcal{C} \setminus (\mathcal{C} \cap \mathcal{B})) \end{aligned} \tag{11}$$

Now, assume that two functions are available to model the system. One models the observation, and has the form of the probability density function $f(\mathbf{y}_t \mid \mathcal{X}_t)$ of the observation $\mathbf{y}_t$ given the realization of the random set $\mathcal{X}_t$ (eq. (6)). The other one is a Markov model for the evolution of $\mathcal{X}_t$ with time, i.e., the conditional density $f(\mathcal{X}_t \mid \mathcal{X}_{t-1})$ (eq. (11)). These two functions can be used as the ingredients of Bayes recursions for countable sets: denoting $\mathbf{y}_{1:t} \triangleq (\mathbf{y}_1, \ldots, \mathbf{y}_t)$ the channel-output observations from time 1 to time t, we have for the conditional a posteriori densities

$$f(\mathcal{X}_{t+1} \mid \mathbf{y}_{1:t}) = \sum_{\mathcal{X}_t} f(\mathcal{X}_{t+1} \mid \mathcal{X}_t) f(\mathcal{X}_t \mid \mathbf{y}_{1:t}) \tag{12}$$

$$f(\mathcal{X}_{t+1} \mid \mathbf{y}_{1:t+1}) \propto f(\mathcal{X}_{t+1} \mid \mathbf{y}_{1:t}) f(\mathbf{y}_{t+1} \mid \mathcal{X}_{t+1}) \tag{13}$$

Thus, the optimum causal detector for $\mathcal{X}_{t+1}$ is

$$\widehat{\mathcal{X}}_{t+1} = \arg\max_{\mathcal{X}_{t+1}} f(\mathcal{X}_{t+1} \mid \mathbf{y}_{1:t+1}) \tag{14}$$

Eq. (12) predicts $\mathcal{X}_{t+1}$ on the basis of its past and of the observations up to time t, while (13) corrects this prediction by accounting for the additional observation made at time $t+1$. Notice, in particular, that the maximization of (12) with respect to $\mathcal{X}_{t+1}$ yields the best prediction of $\mathcal{X}_{t+1}$ based on the observations $\mathbf{y}_{1:t}$.

In [2], the above recursion and the detector in (14) are considered as the optimal solution for the causal estimation of $\mathcal{X}_t$. The problem of (12)-(14) is that they require a complexity for the calculation of (12) and the evaluation of the maximum in (14) which grows exponentially in K. However, one can exploit the structure of the problem to further simplify the detection process. This is the subject of next subsections.

3. ITERATIVE APPROXIMATION STRATEGIES

From now on we denote as $\widehat{\mathcal{X}}_t$ the estimate propagated from the previous step, i.e. $\widehat{\mathcal{X}}_t = \arg\max f(\mathcal{X}_t|\mathbf{y}_{1:t})$ (which is available) and deal with efficient computation of the estimate $\widehat{\mathcal{X}}_{t+1}$. Consider a trained acquisition, i.e. $|\mathcal{M}| = 1$. Inspecting (12) and (13), we see that $\mathcal{O}(2^K)$ complexity is involved both for evaluating the set integral and for the maximum a-posteriori estimate: we are thus faced with the problem of reducing the computational complexity of both operations.

3.1 Algorithm *Iterative 1*

Given an element $\mathcal{G}$ in the collection $\mathcal{P}_K\{1, \ldots, N\}$ of the subsets of $\{1, \ldots, N\}$ with cardinality less than or equal to K, it can be represented through an $N-$dimensional binary vector having 1 in the positions dictated by the elements of $\mathcal{G}$. The correspondence between the set $\mathcal{G}$ and the binary vector $\mathbf{g}$ allows the definition of the collection of sets:

$$\mathcal{C}_d(\mathcal{G}) = \{\mathcal{X} \in \mathcal{P}_K\{1, \ldots, N\} : d_H(\mathbf{x}, \mathbf{g}) = d\} \tag{15}$$

where $d_H(\mathbf{x}, \mathbf{g})$ denotes the Hamming distance between the two vectors representing $\mathcal{X}$ and $\mathcal{G}$. Let us rewrite (12) as follows:

$$\begin{aligned} f(\mathcal{X}_{t+1}|\mathbf{y}_{1:t}) &= f(\widehat{\mathcal{X}}_t|\mathbf{y}_{1:t}) f(\mathcal{X}_{t+1}|\widehat{\mathcal{X}}_t) \\ &\quad + \sum_{i=1}^{K} [\sum_{\mathcal{X}_t \in \mathcal{C}_i(\widehat{\mathcal{X}}_t)} f(\mathcal{X}_{t+1}|\mathcal{X}_t) f(\mathcal{X}_t|\mathbf{y}_{1:t})] \end{aligned} \tag{16}$$

Observe that the number of summands in (12) is exponential. We can reduce the complexity of the calculation of this summation, and yet retain a good approximation for $f(\mathcal{X}_{t+1} \mid \mathbf{y}_{1:t+1})$, if we assume that the discrete-time intervals are so narrow that from time t to time $t+1$ the set of active users changes little, which is tantamount to choosing $\alpha \ll 0.5$ and $\mu \gg 0.5$. Denote by $\widehat{\mathcal{X}}_t$ the MAP estimate of $\mathcal{X}_t$ performed at t. If $\widehat{\mathcal{X}}_t$ is a good estimate of $\mathcal{X}_t$, and the active-user set changes little from t to $t+1$, we may expect that the largest contribution to the sum (12) is offered by the terms corresponding to small values of d, and that the additional summands are (approximately) decreasing as d is increasing. Define

$$f^{(i+1)}(\mathcal{X}_{t+1}|\mathbf{y}_{1:t}) \triangleq \sum_{j=0}^{i+1} \sum_{\mathcal{X}_t \in \mathcal{C}_j(\widehat{\mathcal{X}}_t)} f(\mathcal{X}_{t+1}|\mathcal{X}_t) f(\mathcal{X}_t|\mathbf{y}_{1:t}) \tag{17}$$

A heuristic algorithm, which we refer as *Iterative 1*, can be defined as described herewith:

Algorithm 1 Iterative 1

1: Given $\mathbf{y}_{1:t+1}$ and the MAP estimation of the RFS of the active user at time t, $\widehat{\mathcal{X}}_t$;
2: Set $i=0$ and $\widehat{\mathcal{X}}_{t+1}^{(0)} = \widehat{\mathcal{X}}_t$;
3: Determine the new estimate as:

$$\widehat{\mathcal{X}}_{t+1}^{(i+1)} = \arg \max_{\mathcal{X}_{t+1} \in \bigcup_{j=0}^{i+1} \mathcal{C}_j(\widehat{\mathcal{X}}_t)} f(\mathbf{y}_{t+1}|\mathcal{X}_{t+1}) f^{(i+1)}(\mathcal{X}_{t+1}|\mathbf{y}_{1:t})$$

4: **if** $\widehat{\mathcal{X}}_{t+1}^{(i+1)} = \widehat{\mathcal{X}}_{t+1}^{(i)}$ or if $i = i_{MAX}$
5: stop and $\widehat{\mathcal{X}}_{t+1} = \widehat{\mathcal{X}}_{t+1}^{(i+1)}$
6: **else**
7: Set $i = i+1$ and go to 3.
8: **end if**

In geometric terms, *Iterative 1* consists in restricting the calculation of (12) in a sequence of hyperspheres with increasing radii and common center $\widehat{\mathcal{X}}_t$ and then evaluating the maximum of (13) in the hypersphere only. If this new estimation differs from the estimation at the previous step then the algorithm increments the radius up to a maximum number of iterations, i_{MAX}.

3.2 Algorithm *Iterative 2*

Instead of keeping the center fixed and increasing the radius, a similar algorithm can be defined where the center of the hypersphere is updated at each iteration, while the radius is kept fixed at $d=1$. We will refer to this algorithm, described below, as *Iterative 2*. In it we have defined

$$f'(\mathcal{X}_{t+1}|\mathbf{y}_{1:t}) \triangleq \sum_{\mathcal{X}_t \in \mathcal{C}_1(\widehat{\mathcal{X}}_t) \bigcup \widehat{\mathcal{X}}_t} f(\mathcal{X}_{t+1}|\mathcal{X}_t) f(\mathcal{X}_t|\mathbf{y}_{1:t}) \tag{18}$$

4. SPHERE DETECTION FOR RFS ESTIMATION

The detection problem described by the Bayes recursion in eqs. (12)-(13) cannot be directly solved by Sphere Detection (SD). In this section, we will briefly review the SD algorithm and then introduce an approximation of (12) which will make the problem solvable by SD.

Algorithm 2 Iterative 2

1: Given $\mathbf{y}_{1:t+1}$ and the MAP estimation of the RFS of the active user at time t, $\widehat{\mathcal{X}}_t$;
2: Set $i=0$ and $\widehat{\mathcal{X}}_{t+1}^{(0)} = \widehat{\mathcal{X}}_t$;
3: Determine the new estimate as:

$$\widehat{\mathcal{X}}_{t+1}^{(i+1)} = \arg \max_{\mathcal{X}_{t+1} \in \mathcal{C}_1(\widehat{\mathcal{X}}_{t+1}^{(i)}) \bigcup \widehat{\mathcal{X}}_{t+1}^{(i)}} f(\mathbf{y}_{t+1}|\mathcal{X}_{t+1}) f'(\mathcal{X}_{t+1}|\mathbf{y}_{1:t})$$

4: **if** $\widehat{\mathcal{X}}_{t+1}^{(i+1)} = \widehat{\mathcal{X}}_{t+1}^{(i)}$ or if $i = i_{MAX}$
5: stop and $\widehat{\mathcal{X}}_{t+1} = \widehat{\mathcal{X}}_{t+1}^{(i+1)}$
6: **else**
7: Set $i = i+1$ and go to 3.
8: **end if**

4.1 Sphere detection algorithm

Here we describe a simple version of sphere detection (SD), in a form which will be useful for further developments [1]. Consider the minimization of a function $f(x_1, \ldots, x_K)$ with respect to its K arguments, all taking values in a discrete set with M elements. While brute-force minimization involves the evaluation of all M^K values of f, SD simplifies the problem under the assumption that f can be written in the form of a sum of nonnegative functions with an increasing number of arguments:

$$f(x_1, \ldots, x_K) = \sum_{k=0}^{K-1} f_{K-k, K-k+1, \ldots, K}(x_{K-k}, x_{K-k+1}, \ldots, x_K) \tag{19}$$

The minimization of (19) can be described graphically by using an $(K+1)$-level tree graph whose paths merge into a common uppermost node (*level* 0) to the M^K leaves (*level* K). Each node at level k emanates M branches which join it to a node at level $k+1$, each one being associated with a value of x_{K-k}; hence, each node at level k correspond to a value of the partial sum of the first k terms of (19), and each terminal branch (or *leaf*) to a value of f. Now, brute-force minimization of f can be interpreted as the process of probing all the M^K paths joining the root node to all terminal leaves. SD simplifies the process as follows. Start from the root node and proceed downwards; at level-k node ($k = 0, \ldots, K-1$), only one branch stemming from it is chosen, that associated with the smallest value of $f_{K-k,K-k+1,\ldots,K}$. This leads to a single node at level $k+1$, from which only one branch is chosen according to the same criterion, etc. This is equivalent to the following algorithm, repeated for $k = 0, \ldots, K-1$[2]:

$$\hat{x}_{K-k} = \arg\min_{x_{K-k}} f_{K-k,K-k+1,\ldots,K}(x_{K-k}, \hat{x}_{K-k+1}, \ldots, \hat{x}_K), \tag{20}$$

where $\hat{x}_\ell$ denotes the value chosen for x_ℓ. At the end of this process, we obtain a *preliminary estimate* of the minimum value of f, which we call $\bar{f}$. Next, we proceed to probe the branches that were left out, backtracking from the leaf associated with $\bar{f}$ *and excluding all the branches that will certainly end up into a leaf corresponding to a value of f larger*

[1] SD was first applied to digital detection problems in [4]. For recent developments, see [5, 6, 7] and the references therein. A VLSI implementation is described in [6].

[2] In the Multiuser Detection literature, the algorithm in eq. (20) is refereed as Decision-Feedback (DF) [8]

than $\bar{f}$. To do this, all branches emanating from a node are expunged from the tree ("pruned out") whenever the value of the partial sum at that node is already greater than $\bar{f}$. Whenever a leaf is reached, if this is associated with a value $f < \bar{f}$, then this new value replaces $\bar{f}$, and the procedure is continued.
Several variations of the basic SD algorithm are possible, based on different search schedules. Among these, a breadth-first search, or an *M-best* search, can be implemented in lieu of the depth-first search described *supra*. The M-best search consists of an approximation of the breadth-first search, whereby at each level of the tree only M nodes are kept, viz., those with the smallest partial metrics. This solution may not lead to the minimum (and hence is suboptimum), but has the advantage of requiring a constant number of operations, and hence of reducing the maximum, rather than the average, complexity.

4.2 Zero-order approximation

Inspired by the same principle of Section 3, i.e., under the assumption that the RFS change little from t to $t+1$, we could keep only the largest term of (16), i.e.

$$f(\mathcal{X}_{t+1}|\mathbf{y}_{1:t}) \approx f(\widehat{\mathcal{X}}_t|\mathbf{y}_{1:t})f(\mathcal{X}_{t+1}|\widehat{\mathcal{X}}_t) \propto f(\mathcal{X}_{t+1}|\widehat{\mathcal{X}}_t) \quad (21)$$

We refer eq. (21) as *zero-order approximation*. For future reference, defining

$$\Lambda(\mathcal{X}_{t+1} \mid \mathbf{y}_{1:t+1}) \triangleq -N_0 \ln f(\mathcal{X}_{t+1} \mid \mathbf{y}_{1:t+1}) \quad (22)$$

we have, retaining the zero-order approximation,

$$\Lambda(\mathcal{X}_{t+1} \mid \mathbf{y}_{1:t+1}) \approx \|\mathbf{y}_{t+1} - \mathbf{S}\mathbf{x}_{t+1}\|^2 - N_0 \ln f(\mathcal{X}_{t+1} \mid \widehat{\mathcal{X}}_t) \quad (23)$$

which yields the MAP estimate

$$\widehat{\mathcal{X}}_{t+1} = \arg\min_{\mathcal{X}_{t+1}} \left[\|\mathbf{y}_{t+1} - \mathbf{S}\mathbf{x}_{t+1}\|^2 - N_0 \ln f(\mathcal{X}_{t+1} \mid \widehat{\mathcal{X}}_t)\right] \quad (24)$$

Since the dynamics of each user are independent, we have

$$f(\mathcal{X}_{t+1} \mid \widehat{\mathcal{X}}_t) = \prod_{k=1}^{K} f(\mathcal{X}_{t+1}^{(k)} \mid \widehat{\mathcal{X}}_t^{(k)}) \quad (25)$$

For trained mode we have

$$\begin{aligned} f(\mathcal{X}_{t+1}^{(k)} \mid \widehat{\mathcal{X}}_t^{(k)}) &= \mu^{|\mathcal{X}_{t+1}^{(k)} \cap \widehat{\mathcal{X}}_t^{(k)}|}(1-\mu)^{|\widehat{\mathcal{X}}_t^{(k)}| - |\widehat{\mathcal{X}}_t^{(k)} \cap \mathcal{X}_{t+1}^{(k)}|} \\ &\times \alpha^{|\mathcal{X}_{t+1}^{(k)} \setminus \widehat{\mathcal{X}}_t^{(k)} \cap \mathcal{X}_{t+1}^{(k)}|}(1-\alpha)^{1-|\widehat{\mathcal{X}}_t^{(k)}| - |\mathcal{X}_{t+1}^{(k)} \setminus \widehat{\mathcal{X}}_t^{(k)} \cap \mathcal{X}_{t+1}^{(k)}|} \end{aligned} \quad (26)$$

Perusal of the last equations shows that they are precisely in a form allowing application of the SD algorithm. In fact, by applying a QR decomposition to $\mathbf{S}$ we have $\mathbf{S} = \mathbf{QR}$, with $(\mathbf{R})_{i,j} = 0$, $i > j$ and $\widetilde{\mathbf{y}} = \mathbf{Q}^{\dagger}\mathbf{y}$, and the quantity to be minimized becomes

$$\begin{aligned} &\Lambda(\mathcal{X}_{t+1} \mid \widetilde{\mathbf{y}}_{1:t+1}) \\ &= \sum_{i=1}^{K} \Big|\widetilde{y}_{t+1}(i) - \sum_{j=i}^{K} r_{i,j}x_{t+1}(j)\Big|^2 - N_0 \ln f(\mathcal{X}_{t+1} \mid \widehat{\mathcal{X}}_t) \\ &= \sum_{i=1}^{K} \left[\Big|\widetilde{y}_{t+1}(i) - \sum_{j=i}^{K} r_{i,j}x_{t+1}(j)\Big|^2 - N_0 \ln f(\mathcal{X}_{t+1}^{(i)} \mid \widehat{\mathcal{X}}_t^{(i)})\right] \\ &= \sum_{i=1}^{K} g_i\big(x_{t+1}(K), \ldots, x_{t+1}(i)\big) \end{aligned} \quad (27)$$

where

$$g_i\big(x_{t+1}(K), \ldots, x_{t+1}(i)\big) \triangleq \Big|\widetilde{y}_{t+1}(i) - \sum_{j=i}^{K} r_{i,j}x_{t+1}(j)\Big|^2 - N_0 \ln f(\mathcal{X}_{t+1}^{(i)} \mid \widehat{\mathcal{X}}_t^{(i)}) \quad (28)$$

5. NUMERICAL RESULTS

Consider now the performance of the proposed suboptimal algorithms for MUD in a dynamic environment.
Assume a training phase. Let the spreading sequences be m-sequences with processing gain $N = 7$. The frames have length $T = 10$. Assume the maximum number of allowable user to be $K = 6$. We first simulate a scenario where $\mu = 0.8$ and $\alpha = 0.2$. We examine the set sequence error probability (SSEP) defined as

$$\text{SSEP} = Pr\{\mathcal{X}_{1:T} \neq \widehat{\mathcal{X}}_{1:T}\} \quad (29)$$

and we evaluate the algorithm complexity in term of number of hypotheses to explore. In fig. 1 we show the SSEP versus the signal-to-noise ratio (SNR). We can see how the performance of both *Iterative 1* and *Iterative 2* exhibits a floor for large SNR, while the performance of the SD algorithm is very close to that of the optimal detector in (14). Fig. 2 shows the complexity (in term of explored nodes) of the proposed algorithms with respect to Bayes recursion. While the complexities of *Iterative 1* and *Iterative 2* are independent of the SNR, the complexity of the SD decreases with the SNR, meaning that the decision feedback (DF) initialization is of high quality. Successively, we tested a faster scenario with $\mu = 0.6$ and $\alpha = 0.4$. Fig. 3 shows the SSEP versus SNR of the introduced algorithms and of the optimal detector. We notice that the performances of *Iterative 1* and *Iterative 2* exhibit a degradation with respect to the previous scenario, while the SD is more robust, approaching the performance of the optimal detector. Fig. 4 shows the complexity of the considered algorithms. We notice that the complexity of *Iterative 1* and *Iterative 2* augments with respect to the slow scenario, while the complexity of the SD differs little in the two scenarios.

The results in figs. 1-4 show that, at least for conveniently large SNR, sphere detection applied to the zero-order approximation is definitely superior to both *iterative 1* and *iterative 2*, which are instead preferable in the low SNR regime. This trend is an indirect confirmation of the suitability of the zero-order approximation: otherwise stated, the results seem to demonstrate that the increased accuracy in BR's computation pursued by *iterative 1* and *iterative 2* is *not* paid off by a significant advantage in terms of performance, which is severely limited by the fact that, unlike sphere detection, neither algorithm is ensured to terminate in the absolute minimum.

6. CONCLUSION

We have examined multiuser detectors operating without a priori knowledge of the number of active users, and hence detecting simultaneously the set of active users and their data. Since implementation of optimum detectors can be limited by their complexity, which grows exponentially with the number of potential users, we have described techniques for

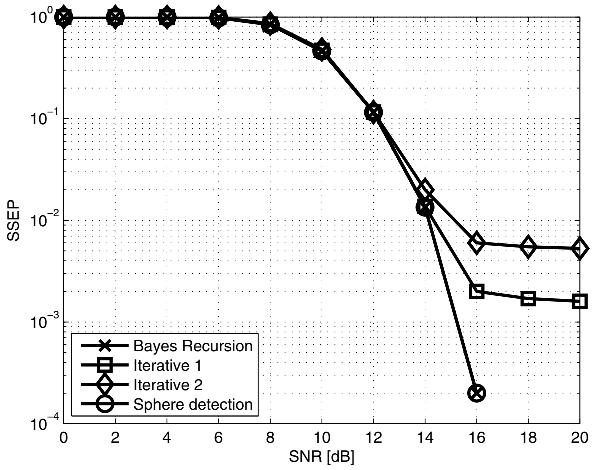

Figure 1: SSEP vs SNR ($\mu = 0.8$ and $\alpha = 0.2$).

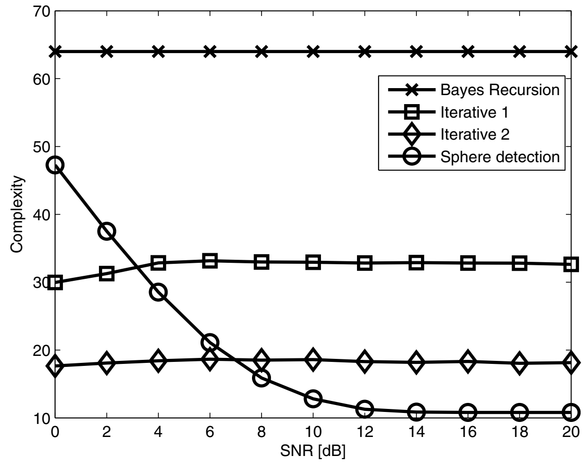

Figure 2: Complexity vs SNR ($\mu = 0.8$ and $\alpha = 0.2$).

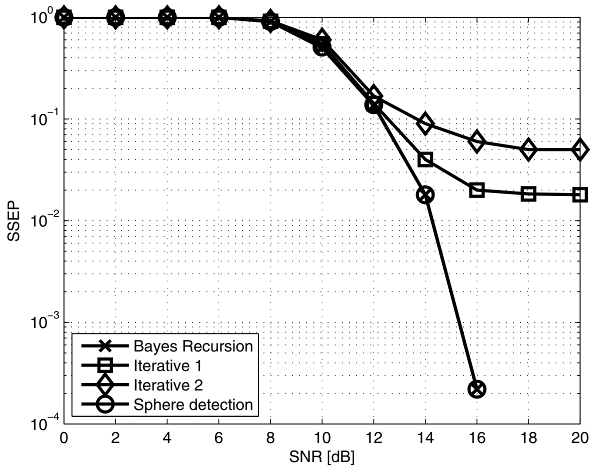

Figure 3: SSEP vs SNR ($\mu = 0.6$ and $\alpha = 0.4$).

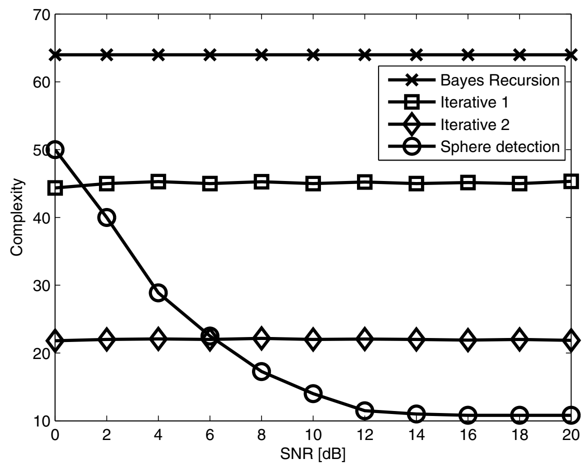

Figure 4: Complexity vs SNR ($\mu = 0.6$ and $\alpha = 0.4$).

the reduction of this complexity. Among the presented suboptimal solutions, the Sphere Detector, based on the zero-order approximation of the Bayes recursion, exhibits marginal loss in performance with respect to the optimum detector, with a complexity that decreases with the signal-to-noise ratio.

Acknowledgment

The work of Ezio Biglieri was supported by the Spanish Ministery of Education and Science under Project TEC2006-01428/TCM, and by the STREP project No. IST-026905 (MASCOT) within the 6th framework program of the European Commission.

REFERENCES

[1] W.-C. Wu and K.-C. Chen, Identification of active users in synchronous CDMA multiuser detection, *IEEE J. Select. Areas Commun.*, Vol. 16, No. 9, pp. 1723-1735, December 1998.

[2] E. Biglieri and M. Lops, Multiuser detection in a dynamic environment. Part I: User identification and data detection, *IEEE Trans. Inform. Theory*, Volume 53, No. 9, pp. 3158-3170, September 2007.

[3] I. R. Goodman, R. P. S. Mahler, and H. T. Nguyen, *Mathematics of Data Fusion*. Dordrecht, The Netherlands: Kluwer, 1997.

[4] E. Viterbo and E. Biglieri, "A universal lattice decoder," in *GRETSI 14-ème Colloque*, Juan-les-Pins, France, September 1993.

[5] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, Vol. 48, No. 8, pp. 2201–2214, August 2002.

[6] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bölcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE Journal Solid-State Circuits*, Vol. 40, No. 7, pp. 1566–1577, July 2005.

[7] M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inform. Theory*, Vol. 49, No. 10, pp. 2389–2402, October 2003.

[8] S. Verdú, *Multiuser Detection*. Cambridge, UK: Cambridge University Press, 1998.