

The Application of Semidefinite Programming for Detection in CDMA

Peng Hui Tan, *Student Member, IEEE*, and Lars K. Rasmussen, *Senior Member, IEEE*

Abstract—In this paper, a detection strategy based on a semidefinite relaxation of the CDMA maximum-likelihood (ML) problem is investigated. Cutting planes are introduced to strengthen the approximation. The semidefinite program arising from the relaxation can be solved efficiently using interior point methods. These interior point methods have polynomial computational complexity in the number of users. The simulated bit error rate performance demonstrates that this approach provides a good approximation to the ML performance.

Index Terms—Code-division multiple access, multiuser detection, semidefinite programming.

I. INTRODUCTION

IN A DIRECT-SEQUENCE code-division multiple access (DS/CDMA) system, users are assigned unique signature waveforms that are used to modulate their transmitted symbols. It is, however, not possible to ensure orthogonality among received signature waveforms in a mobile environment, and thus, multiple access interference (MAI) arises. MAI limits the cellular capacity of a system using conventional Rake detection [1], and leads to strict power control requirements to alleviate the near-far problem.

A better detection strategy is to detect jointly multiple users since the sufficient statistics for the detection of one user contain information about all the others as well. In [2], Verdú proposed and analyzed the optimal multi-user detector, which is a maximum likelihood (ML) sequence detector. This receiver minimizes, with respect to all possible transmitted bit sequences, the squared Euclidean distance¹ between the actual and the hypothesized received signal vectors. As such, the presence of all the K users simultaneously sharing the channel is accounted for in arriving at the ML receiver.

For an asynchronous transmission of L information bits per user using binary phase-shift keying (BPSK) modulation and spreading, the optimum detection problem is equivalent to minimizing a quadratic program with the information-bit vector \mathbf{d} constrained to the set $\{-1, 1\}^{LK}$. This is a binary constrained minimization that is also known as a binary quadratic program (BQP) in the area of optimization. It has

been shown that this problem is in general nondeterministic polynomial (NP)-hard [3], it is much too complex for practical asynchronous DS-CDMA systems with even a moderate number of users. For certain special cases of the correlation matrix, it has been shown that ML detection can be obtained by polynomial-time algorithms [4], [5]. An iterative structure that is guaranteed to deliver ML detection on some bits was suggested in [6]. However, decisions on all bits are not guaranteed. Approximations to the ML problem have also been suggested in [7] based on the EM algorithm, and in [8] based on iterative transformations of the quadratic minimization problem, such that the unconstrained solution to the transformed problem monotonically approaches the desired solution.

Another possibility to approximately solve the discrete ML problem is to relax the constraint on the valid set of feasible solutions to obtain a tractable nonlinear continuous problem. In [9]–[11], the solution is allowed to lie within a closed convex set. One way is to confine the solution vector to lie within a sphere of radius $\|\mathbf{d}\|$, passing through all possible data points. The algorithm can also constrain the data estimate vector to lie within a hypercube described by the data points. Recently, Helmborg *et al.* [12] have used the combination of semidefinite relaxation and cutting planes to obtain a tight bound for the solution of a BQP. Here, we will apply these results to CDMA detection.

The paper is organized as follows. In the following section, the CDMA channel model is described. Section III develops detection strategies based on a semidefinite relaxation of the optimum detection problem. The implementation of the strategies are discussed in more detail in Section IV, while numerical examples are presented in Section V. Concluding remarks are found in Section VI.

II. SYSTEM MODEL

Let us consider a CDMA channel that is shared by K simultaneous users. Each user is assigned a signature waveform $p_k(t)$ of duration T , where T is the symbol interval. A signature waveform may be expressed as

$$p_k(t) = \sum_{n=0}^{N-1} a_k(n)p(t - nT_c), \quad 0 \leq t \leq T$$

where $\{a_k(n), 0 \leq n \leq N-1\}$ is a code sequence consisting of N chips that take values $\{\pm 1\}$, $p(t)$ is a pulse of duration T_c , where T_c is the chip interval. Thus, we have N chips per symbol and $T = NT_c$. Without loss of generality, we assume that all K signature waveforms have unit energy.

The information sequence of the k th user is denoted by $\{d_k(m)\}$, where the value of each information symbol may

Manuscript received December 14, 2000; revised May 10, 2001. This work was supported in parts by the Swedish Research Council for Engineering Sciences under Grant 271-1999-390 and Grant 217-1997-538. This paper was presented in part at the IEEE International Symposium on Information Theory, June 2001, Washington, DC.

The authors are with the Telecommunication Theory Group, Department of Computer Engineering, Chalmers University of Technology, Gothenburg, Sweden (e-mail: phtan@ce.chalmers.se; larsr@ce.chalmers.se).

Publisher Item Identifier S 0733-8716(01)07234-1.

¹Assuming an additive white Gaussian noise (AWGN) channel.

be chosen from the set \mathcal{D} . It is convenient to consider the transmission of a block of symbols of some arbitrary length, say L . The corresponding equivalent lowpass, transmitted waveform may be expressed as

$$s_k(t) = c_k \sum_{i=1}^L d_k(i) p_k(t - iT)$$

where c_k is the complex channel coefficient. The composite transmitted signal for the K users may be expressed as

$$\begin{aligned} s(t) &= \sum_{k=1}^K s_k(t - \tau_k) \\ &= \sum_{k=1}^K c_k \sum_{i=1}^L d_k(i) p_k(t - iT - \tau_k) \end{aligned}$$

where $\{\tau_k\}$ are the transmission delays, which satisfy the condition $0 \leq \tau_k < T$ for $1 \leq k \leq K$. Without loss of generality, we assume that $0 \leq \tau_1 \leq \tau_2 \leq \dots \leq \tau_K < T$. This is the model for a multiuser transmitted signal in asynchronous mode. In the special case of synchronous transmission, $\tau_k = 0$ for $1 \leq k \leq K$.

The transmitted signal is assumed to be corrupted by AWGN. Hence, the received signal may be expressed as

$$r(t) = s(t) + n(t)$$

where $n(t)$ is the noise, with power spectral density σ^2 .

The sampled output of the matched filter (MF) of the k th user during the i th signal interval is

$$y_k(i) = \int_{iT+\tau_k}^{(i+1)T+\tau_k} r(t) s_k(t - iT - \tau_k) dt, \quad 1 \leq k \leq K, \quad 1 \leq i \leq L.$$

Using vector notation, the LK MF outputs can be expressed as

$$\mathbf{y} = \mathbf{R}\mathbf{C}\mathbf{d} + \mathbf{z}$$

where \mathbf{C} is a matrix that contains the channel coefficients of the users, and \mathbf{R} is a $KL \times KL$ correlation matrix. The Gaussian noise vector \mathbf{z} has zero mean and autocorrelation matrix

$$E[\mathbf{z}\mathbf{z}^H] = \sigma^2 \mathbf{R}.$$

This model can easily be extended to model transmission over multipath channels. The extension is discussed in more detail in [13].

The optimum ML detector selects the ML hypothesis $\hat{\mathbf{d}}$ given the matched filter output, and perfect knowledge of \mathbf{C} and \mathbf{R}

$$\hat{\mathbf{d}} = \arg \max_{\mathbf{d} \in \mathcal{D}} p(\mathbf{y} | \mathbf{d}).$$

Since we are considering an AWGN channel, the negative log-likelihood function based on $p(\mathbf{y} | \mathbf{d})$ is described as $F(\mathbf{d}) = \mathbf{d}^H \mathbf{C}^H \mathbf{R} \mathbf{C} \mathbf{d} - 2 \operatorname{Re}\{\mathbf{y}^H \mathbf{C} \mathbf{d}\}$. The constrained ML problem is then described as

$$\hat{\mathbf{d}} = \arg \min_{\mathbf{d} \in \mathcal{D}} \mathbf{d}^H \mathbf{C}^H \mathbf{R} \mathbf{C} \mathbf{d} - 2 \operatorname{Re}\{\mathbf{y}^H \mathbf{C} \mathbf{d}\}. \quad (1)$$

The solution to the above problem (2) requires a search over all the \mathcal{D}^{KL} possible combinations of the components of the

vector \mathbf{d} . It is, thus, clear that the computational complexity increases exponentially with the number of users. The complexity, however, does not necessarily grow exponentially with the block length L as the causal memory in the inherent finite state machine is limited by K only [2].

A. Real Channels and Binary Symbols

To highlight the mathematical details of the application of semidefinite programming, we will only consider synchronous CDMA system unless explicitly stated otherwise. In synchronous transmission, $\tau_k = 0$, and each interfering user produces exactly one symbol which interferes with the desired symbol. In single-path AWGN channels, it is sufficient to consider the signal received in one signal interval only and determined the optimum receiver. When the channel coefficients are real valued and the symbols \mathbf{d} are binary, i.e., $\mathcal{D} \in \{-1, 1\}^K$, the optimization problem of (1) can be rewritten as

$$\hat{\mathbf{d}} = \arg \min_{\mathbf{d} \in \{\pm 1\}^K} \mathbf{d}^T \mathbf{C}^T \mathbf{R} \mathbf{C} \mathbf{d} - 2 \mathbf{y}^T \mathbf{C} \mathbf{d}. \quad (2)$$

B. Complex Channels and QPSK Symbols

In this case, the symbols of each user is modulated using e.g., quaternary phase-shift keying (QPSK), $\mathcal{D} = \{\pm 1 \pm j\}^K$, and e.g., QPSK spreading. The ML criterion (1) can be expressed in the same format

$$\begin{aligned} \mathbf{u} &= \arg \min_{\mathbf{d} \in \{\pm 1 \pm j\}^K} \mathbf{d}^H \mathbf{C}^H \mathbf{R} \mathbf{C} \mathbf{d} - 2 \operatorname{Re}\{\mathbf{y}^H \mathbf{C} \mathbf{d}\} \\ &= \arg \min_{\tilde{\mathbf{d}} \in \{\pm 1\}^{2K}} \tilde{\mathbf{d}}^T \tilde{\mathbf{R}} \tilde{\mathbf{d}} - 2 \tilde{\mathbf{d}}^T \tilde{\mathbf{y}}. \end{aligned}$$

In (1), we have converted the complex components of dimensions K and $K \times K$ into vectors and matrices of dimensions $2K$ and $2K \times 2K$ as shown:

$$\begin{aligned} \tilde{\mathbf{d}} &= [\operatorname{Re}\{\mathbf{d}^T\} \quad \operatorname{Im}\{\mathbf{d}^T\}]^T \\ \tilde{\mathbf{R}} &= \begin{bmatrix} \operatorname{Re}\{\mathbf{C}^H \mathbf{R} \mathbf{C}\} & -\operatorname{Im}\{\mathbf{C}^H \mathbf{R} \mathbf{C}\} \\ \operatorname{Im}\{\mathbf{C}^H \mathbf{R} \mathbf{C}\} & \operatorname{Re}\{\mathbf{C}^H \mathbf{R} \mathbf{C}\} \end{bmatrix} \\ \tilde{\mathbf{y}} &= [\operatorname{Re}\{\mathbf{C}^H \mathbf{y}\}^T \quad \operatorname{Im}\{\mathbf{C}^H \mathbf{y}\}^T]^T. \end{aligned}$$

The derivations in Section III generalizes to include such cases. To simplify notation and emphasize the mathematical details of the semidefinite programming technique, we will, however, focus on a synchronous system with real channel coefficients and BPSK data modulation. It is notationally challenging but comprehensively straightforward to extend the results to include complex modulation.

III. SEMIDEFINITE PROGRAMMING FOR MULTIUSER DETECTION

This section considers multiuser detection from a combinatorial optimization point of view. We propose the use of semidefinite programming to seek the optimum solution. This technique has the potential to reduce computational complexity without sacrificing performance. Current references for semidefinite programming are [14] and [15]. A survey concentrating on the connection of semidefinite programming to combinatorial optimization can be found in [16].

A. Basic Semidefinite Relaxation

Let us start with the problem in (2) where $\mathbf{Q} = \mathbf{C}^T \mathbf{R} \mathbf{C}$, $\mathbf{c} = \mathbf{C} \mathbf{y}$ and $\mathbf{u} = \hat{\mathbf{d}}$.

We then have

$$\mathbf{u}^* = \arg \min_{\mathbf{u}} \mathbf{u}^T \mathbf{Q} \mathbf{u} - 2\mathbf{c}^T \mathbf{u} \quad \text{s.t. } \mathbf{u} \in \{-1, 1\}^{n-1} \quad (3)$$

where $K = n - 1$. The following results apply for any given $\mathbf{Q} = \mathbf{Q}^T$ [12]. Including a redundant dummy variable u_n , we can express problem (3) as

$$\begin{aligned} [\mathbf{u}^*, u_n^*] &= \arg \min_{\mathbf{u}, u_n} [\mathbf{u}^T \quad u_n] \begin{bmatrix} \mathbf{Q} & -\mathbf{c} \\ -\mathbf{c}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ u_n \end{bmatrix} \\ \text{s.t. } & \begin{bmatrix} \mathbf{u} \\ u_n \end{bmatrix} \in \{-1, 1\}^n, \quad u_n = 1. \end{aligned} \quad (4)$$

Since the cost function is symmetric, $u_n = 1$ need not to be maintained explicitly. Based on (4), we reformulate (3) as

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} \mathbf{x}^T \mathbf{L} \mathbf{x} \quad \text{s.t. } \mathbf{x} \in \{-1, 1\}^n \quad (5)$$

where

$$\mathbf{L} = \begin{bmatrix} \mathbf{Q} & -\mathbf{c} \\ -\mathbf{c}^T & 0 \end{bmatrix} \quad \text{and} \quad \mathbf{x} = [\mathbf{u}^T \quad u_n]^T.$$

A semidefinite relaxation of (5) observes that

$$\mathbf{x}^T \mathbf{L} \mathbf{x} = \text{tr}\{\mathbf{L} \mathbf{x} \mathbf{x}^T\}.$$

For $\mathbf{x} \in \{-1, 1\}^n$, the matrix $\mathbf{x} \mathbf{x}^T$ is positive semidefinite, its diagonal entries are equal to 1, and it is a rank one matrix [16]. Now let $\mathbf{X} = \mathbf{x} \mathbf{x}^T$ be a matrix that satisfies these three characteristic properties. Then, (5) becomes

$$\begin{aligned} \mathbf{X}_1^* &= \arg \min_{\mathbf{X}} \text{tr}\{\mathbf{L} \mathbf{X}\} \\ \text{s.t. } & \text{diag}(\mathbf{X}) = \mathbf{e}_n, \quad \text{rank}(\mathbf{X}) = 1, \quad \mathbf{X} \succeq 0 \end{aligned} \quad (6)$$

where

$\text{diag}(\mathbf{X})$ vector consisting of the diagonal elements of \mathbf{X} ;

\mathbf{e}_n all ones vector of length n ;

$\mathbf{X} \succeq 0$ \mathbf{X} is positive semidefinite.

The subscript of \mathbf{X}_1^* refers to the rank one constraint. The equivalence of (5) and (6) is shown in general in Lemma 3.1 in [17].

Dropping the rank one constraint yields the basic semidefinite relaxation [12] of (6)

$$\mathbf{X}^* = \arg \min_{\mathbf{X}} \text{tr}\{\mathbf{L} \mathbf{X}\} \quad \text{s.t. } \text{diag}(\mathbf{X}) = \mathbf{e}_n, \quad \mathbf{X} \succeq 0. \quad (7)$$

This is known as a semidefinite program in the matrix variable \mathbf{X} because it is a linear problem in \mathbf{X} with the additional semidefiniteness constraint $\mathbf{X} \succeq 0$. For $n = 3$ ($K = 2$), we define

$$\mathbf{X} = \begin{bmatrix} 1 & x & y \\ x & 1 & z \\ y & z & 1 \end{bmatrix}.$$

The so-called primal feasible set of x, y and z based on (7) is illustrated in Fig. 1. The surface of the set is determined by

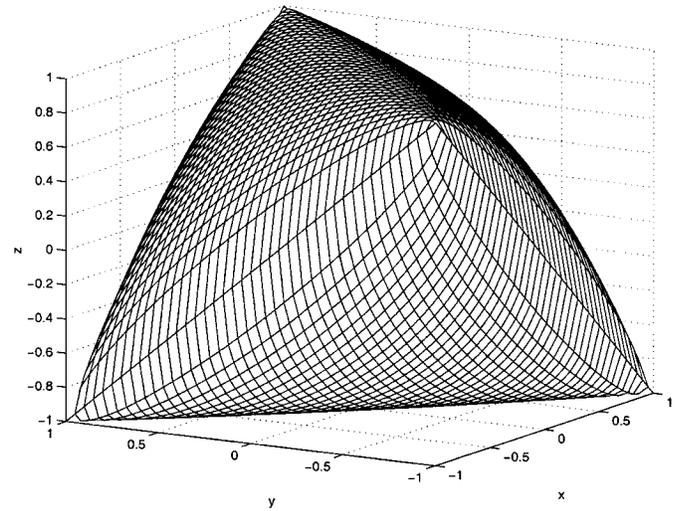


Fig. 1. Primal feasible set of the semidefinite program (7) for $n = 3$. The surface is the solution of $\det(\mathbf{X}) = 0$.

$\det \mathbf{X} = 0$ Note that now $\mathbf{X}^* \neq \mathbf{x}^* \mathbf{x}^{*T}$. The task is then to find \mathbf{X}^* in this feasible set.

B. Cutting Planes

Tightening of the relaxation introduces cutting planes that have become a standard technique for solving combinatorial optimization problems through semidefinite programs [12]. An optimal solution to the relaxed problem in (7) is computed iteratively. If the solution is not feasible for the original problem in (6), the feasible region for (7) is reduced so that the current solution is no longer feasible. This is done by finding inequalities that are valid for the original problem in (6), but excluding the current point from the feasible region. The goal is to approximate the solution to (6) by using a tightest possible continuous relaxation of the feasible set of integral points. Clearly, the algorithm needs to keep the number of valid linear inequalities for the feasible set, also called the cutting planes, as small as possible to limit the computational complexity.

The class of linear inequalities used as cutting planes in this paper is described as follows. For $\mathbf{h} \in \mathcal{Z}^n$ with exactly three nonzero components,² which can be either $+1$ or -1 , the inequalities $\mathbf{h}^T \mathbf{X} \mathbf{h} = \text{tr}\{\mathbf{X} \mathbf{h} \mathbf{h}^T\} \geq 1$ are called triangle inequalities [18]. For a set of $h_i, h_j, h_k \in \{-1, 1\}$ and given that \mathbf{X} is symmetrical, the set of triangle inequalities can be described as

$$\begin{aligned} x_{ij} + x_{ik} + x_{jk} &\geq -1 \\ x_{ij} - x_{ik} - x_{jk} &\geq -1 \\ -x_{ij} + x_{ik} - x_{jk} &\geq -1 \\ -x_{ij} - x_{ik} + x_{jk} &\geq -1. \end{aligned} \quad (8)$$

The set of all points satisfying all triangle inequalities define the so-called metric polytope [18]. An illustration of the metric polytope for $n = 3$ is shown in Fig. 2. It can be easily seen that the metric polytope is a subset of the primal feasible set shown in Fig. 1. Since there are $\binom{n}{3}$ possible values of \mathbf{h} which yield $4\binom{n}{3}$ triangle inequalities, the optimization over the metric polytope can be done in polynomial time.

² \mathcal{Z}^n is the integer space of dimension n .

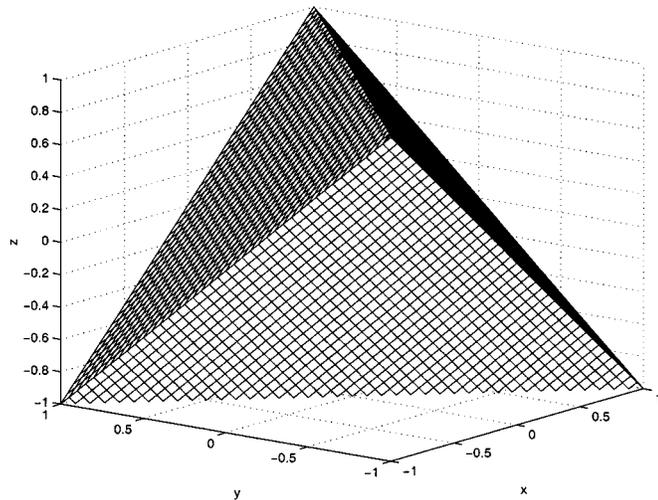


Fig. 2. Metric polytope formed by the triangle inequalities (8) for $n = 3$.

IV. INTERIOR-POINT METHOD

The development of interior-point methods has made it possible to solve semidefinite programs efficiently and with polynomial complexity. A semidefinite program (7) can be solved by employing the primal–dual path-following algorithm of [19] as a basic optimization tool. We start with a description of this interior-point method, and then explain the procedures for including the triangle inequalities (8).

A. Initial Semidefinite Relaxation

The interior-point method in [19] is designed to handle primal–dual pairs, where (7) is called the primal problem and its dual problem is described as

$$(\mathbf{v}^*, \mathbf{Z}^*) = \arg \max_{\mathbf{v}, \mathbf{Z}} \mathbf{e}_n^T \mathbf{v} \quad \text{s.t. } \mathbf{Z} = \mathbf{L} - \text{Diag}(\mathbf{v}), \quad \mathbf{Z} \succeq 0. \quad (9)$$

Here, $\text{Diag}(\mathbf{v})$ denotes the diagonal matrix with diagonal elements obtained from \mathbf{v} .

Interior-point methods start off at an interior point of the cone of positive semidefinite matrices [15]. The set of positive semidefinite matrices can be interpreted as a convex cone in $\mathcal{R}^{\binom{n+1}{2}}$ [15]. The identity matrix \mathbf{I} is contained in the cone and is a typical starting point. The iterates are the approximate minimizers of a sequence of auxiliary problems that contain an additional term ($-\mu \log \det(\mathbf{X})$) in the cost function. This term is called the barrier term and it consists of a barrier parameter $\mu > 0$ and a barrier function $-\log \det(\mathbf{X})$. The auxiliary problem is then stated as

$$\mathbf{X}_\mu^* = \arg \min_{\mathbf{X}} \mathbf{L}\mathbf{X} - \mu \log \det(\mathbf{X}) \quad \text{s.t. } \text{diag}(\mathbf{X}) = \mathbf{e}_n, \quad \mathbf{X} \succ 0. \quad (10)$$

The barrier function grows to infinity as \mathbf{X} approaches the boundary of the semidefinite cone. The barrier parameter μ controls the distance of the optimal solution of the auxiliary problem to the boundary. For a sequence of auxiliary barrier

problems with $\mu \rightarrow 0$, the original cost function eventually dominates in the interior of the feasible set. It follows that the sequence of minimizers of the barrier problems converges to an optimizer of the original problem (7).

This paper uses the algorithm proposed in [19] to solve (7). We transform the barrier problem into an unconstrained problem for $\mathbf{X} \succ 0$ by introducing a Lagrange multiplier vector \mathbf{v} for the equality constraints,

$$L_\mu(\mathbf{X}, \mathbf{v}) = \text{tr}\{\mathbf{L}\mathbf{X}\} - \mu \log \det(\mathbf{X}) + \mathbf{v}^T [\mathbf{e}_n - \text{diag}(\mathbf{X})].$$

For given $\mathbf{v} \in \mathcal{R}^n$, the function $L_\mu(\mathbf{X}, \mathbf{v})$ is a smooth convex function [15]. The first-order necessary conditions for optimality are called the Karush–Kuhn–Tucker (KKT) conditions. By matrix calculus the gradient of $\log \det(\mathbf{X})$ is $\nabla_{\mathbf{X}} \log \det(\mathbf{X}) = \mathbf{X}^{-1}$ [19], and we arrive at the system

$$\begin{aligned} \nabla_{\mathbf{X}} L_\mu &= \mathbf{L} - \mu \mathbf{X}^{-1} - \text{Diag}(\mathbf{v}) = \mathbf{Z} - \mu \mathbf{X}^{-1} = \mathbf{0} \\ \nabla_{\mathbf{v}} L_\mu &= \mathbf{e}_n - \text{diag}(\mathbf{X}) = \mathbf{0}. \end{aligned}$$

In a primal–dual formulation, we set $\mathbf{Z} = \mu \mathbf{X}^{-1}$ [19] and rewrite the KKT-conditions of the primal and dual barrier problems in the following form³

$$\begin{aligned} \text{diag}(\mathbf{X}) &= \mathbf{e}_n, \quad \mathbf{X} \succ 0 \\ \mathbf{Z} + \text{Diag}(\mathbf{v}) &= \mathbf{L}, \quad \mathbf{Z} \succ 0 \\ \mathbf{X}\mathbf{Z} &= \mu \mathbf{I}. \end{aligned} \quad (11)$$

The first equation requires primal feasibility, the second requires dual feasibility, and the third equation is called the perturbed complementarity condition [19].

We denote the solution of the equation system (11) for some fixed μ by $(\mathbf{X}_\mu^*, \mathbf{v}_\mu^*, \mathbf{Z}_\mu^*)$. \mathbf{X}_μ^* is the unique optimal solution to the primal barrier problem, and $(\mathbf{v}_\mu^*, \mathbf{Z}_\mu^*)$ is the optimal solution to the analog dual barrier problem. \mathbf{X}_μ^* and \mathbf{Z}_μ^* are feasible primal and dual points of the original problem with a gap of $\text{tr}(\mathbf{X}\mathbf{Z}) = n\mu$ between the objective values. The set of solutions $(\mathbf{X}_\mu^*, \mathbf{v}_\mu^*, \mathbf{Z}_\mu^*)$ for $\mu > 0$ forms the so-called central path, which is a smooth curve [20]. It was shown in [15] that for $\mu \rightarrow 0$ the central path converges to a point $(\mathbf{X}^*, \mathbf{v}^*, \mathbf{Z}^*)$, where \mathbf{X}^* is an optimal solution of the original primal problem (7) and $(\mathbf{v}^*, \mathbf{Z}^*)$ is an optimal solution of the original dual problem (9).

To find the optimal solution to the system in (11), Newton's method [19] is used. After having found an approximate solution for some specified value of μ , this parameter is reduced and a new solution is found, using the previous solution as a starting point. This iterative procedure is continued until μ is sufficiently small.

The complementarity condition $\mathbf{X}\mathbf{Z} = \mu \mathbf{I}$ can be written in many equivalent ways [16]. In this paper, we will use the Helmberg–Kojima–Monteiro (HKM) method proposed independently by Helmberg *et al.* [19], Kojima *et al.* [20], and later reintroduced by Monteiro [21].⁴ Therefore, we briefly go through the main steps to perform one Newton iteration,

³Starting from the dual barrier problem, one would arrive at the same system.

⁴The method is denoted by the first letter of each author.

using the HKM method. We want to compute an approximate solution of (11), i.e., a solution to

$$F_\mu(\mathbf{X}, \mathbf{v}, \mathbf{Z}) = \begin{bmatrix} \text{diag}(\mathbf{X}) - \mathbf{e}_n \\ \mathbf{Z} + \text{Diag}(\mathbf{v}) - \mathbf{L} \\ \mathbf{X}\mathbf{Z} - \mu\mathbf{I} \end{bmatrix} = \mathbf{0}.$$

Newton's method computes a step direction $(\Delta\mathbf{X}, \Delta\mathbf{v}, \Delta\mathbf{Z})$ by solving $F_\mu + \nabla F_\mu[\Delta\mathbf{X}, \Delta\mathbf{v}, \Delta\mathbf{Z}]^T = 0$. Here, the step direction can be determined by the linearized system

$$\begin{aligned} \text{diag}(\Delta\mathbf{X}) &= \mathbf{0} \\ \Delta\mathbf{Z} &= \text{Diag}(\Delta\mathbf{v}) \\ \Delta\mathbf{X}\mathbf{Z} + \mathbf{X}\Delta\mathbf{Z} &= \mu\mathbf{I} - \mathbf{X}\mathbf{Z}. \end{aligned} \quad (12)$$

The second equation is used to eliminate $\Delta\mathbf{Z}$. The third equation gives

$$\Delta\mathbf{X} = \mu\mathbf{Z}^{-1} - \mathbf{X} - \mathbf{X}\text{Diag}(\Delta\mathbf{v})\mathbf{Z}^{-1}. \quad (13)$$

This is substituted into the first equation, leading to a linear equation for $\Delta\mathbf{v}$

$$(\mathbf{X} \circ \mathbf{Z}^{-1})\Delta\mathbf{v} = \mu \text{diag}(\mathbf{Z}^{-1}) - \mathbf{e}_n$$

where \circ denotes the Hadamard-product of matrices [22]. It can be shown that this system is positive definite [19], but gets increasingly ill-conditioned: as μ tends to 0, \mathbf{Z} will become nearly singular, which is a problem since \mathbf{Z}^{-1} is used explicitly. For the numerical examples presented in Section V, this was however not a problem. Given $\Delta\mathbf{v}$, we back-substitute for $\Delta\mathbf{Z}$ and $\Delta\mathbf{X}$. Equation (13) gives in general a nonsymmetric $\Delta\mathbf{X}$. According to the HKM method, we replace $\Delta\mathbf{X}$ by its symmetrical part

$$\Delta\mathbf{X} = (\Delta\hat{\mathbf{X}} + \Delta\hat{\mathbf{X}}^T)/2$$

where $\Delta\hat{\mathbf{X}}$ is obtained from (13). The search direction is then $(\Delta\mathbf{X}, \Delta\mathbf{v}, \Delta\mathbf{Z})$. Even though this direction is not the Newton direction any more (because we changed $\Delta\mathbf{X}$), it can still be shown that this search direction enjoys global convergence [19].⁵

Having determined the desired search direction triple $(\Delta\mathbf{X}, \Delta\mathbf{v}, \Delta\mathbf{Z})$, we will step to the new triple $(\mathbf{X} + \Delta\mathbf{X}, \mathbf{v} + \Delta\mathbf{v}, \mathbf{Z} + \Delta\mathbf{Z})$ except that it might violate the positive definiteness property required of \mathbf{X} and \mathbf{Z} . Hence, we perform a line search to find constants α_p and α_d such that $\mathbf{X} + \alpha_p\Delta\mathbf{X}$ and $\mathbf{Z} + \alpha_d\Delta\mathbf{Z}$ are positive definite. Given α_p and α_d , we step to the new point

$$\begin{aligned} \mathbf{X}^+ &= \mathbf{X} + \alpha_p\Delta\mathbf{X} \\ \mathbf{v}^+ &= \mathbf{v} + \alpha_d\Delta\mathbf{v} \\ \mathbf{Z}^+ &= \mathbf{Z} + \alpha_d\Delta\mathbf{Z}. \end{aligned}$$

We then update $\mu = \text{tr}\{\mathbf{X}\mathbf{Z}\}/(2n)$ and repeat. The algorithm continues until the current triple $(\mathbf{X}^+, \mathbf{v}^+, \mathbf{Z}^+)$ satisfies primal feasibility, dual feasibility and the duality gap is sufficiently

⁵For local super-linear convergence, see [20].

small. This completes the description of the HKM-direction interior-point algorithm. The complete algorithm is listed below.

Algorithm IV.1:

Input: \mathbf{L} and some starting point $(\mathbf{X}_0, \mathbf{v}_0, \mathbf{Z}_0)$ with $\mathbf{X}_0 \succ \mathbf{0}$ and $\mathbf{Z}_0 \succ \mathbf{0}$

1. Set $k = 0$
2. Choose $\mu_k = \text{tr}\{\mathbf{X}_k\mathbf{Z}_k\}/(2n)$.
3. Compute $(\Delta\mathbf{X}, \Delta\mathbf{v}, \Delta\mathbf{Z})$ by solving linearized system (12).
4. Choose some $\alpha_p, \alpha_d \in (0, 1]$ so that $\mathbf{X}_k + \alpha_p\Delta\mathbf{X}$ and $\mathbf{Z}_k + \alpha_d\Delta\mathbf{Z}$ remain positive definite.
5. Set $(\mathbf{X}_{k+1}, \mathbf{v}_{k+1}, \mathbf{Z}_{k+1}) := (\mathbf{X}_k + \alpha_p\Delta\mathbf{X}, \mathbf{v}_k + \alpha_p\Delta\mathbf{v}, \mathbf{Z}_k + \alpha_d\Delta\mathbf{Z})$.
6. If $\text{tr}\{\mathbf{X}_{k+1}\mathbf{Z}_{k+1}\} < \varepsilon$ then **stop**.
7. Set $k := k + 1$ and **goto** step 2.

The computationally crucial steps are as follows. Here, we assume that matrix multiplication is of complexity order of n^3 [16]. In each iteration, we need to compute the matrix \mathbf{Z}^{-1} and solve a linear system of size n . This will take $\mathcal{O}(n^3)$ operations. For the line search, we must check whether a cholesky decomposition of \mathbf{X} and \mathbf{Z} exist. If the number of backtracking steps is independent of n , then the line search takes $\mathcal{O}(n^3)$ operations.⁶ The algorithm therefore has a computational complexity which is polynomial in the number of users, K . The complexity is further quantified based on simulation results in Section V.

For an asynchronous system, the computational complexity is polynomial in the product of K and L , where L is the total block size of the data symbols. In order to reduce this complexity, we can consider the received signal in processing windows of length $P \leq L$ symbol durations, where P is called the detector memory length [23]. The window length P can be set to be sufficiently large so that we can neglect the edge effects caused by the finite memory of the detector. In practice, P is significantly smaller than L .

To solve (7), it remains to specify a starting point. We make the following choices. For a starting point on the primal side, we have $\mathbf{X} = \mathbf{I}$. Then, we find a feasible assignment for the dual variables \mathbf{Z} and \mathbf{v} . Since \mathbf{v} is unconstrained, we can choose v_i such that \mathbf{Z} is diagonally dominant and, therefore, positive definite. We now compute Newton steps until the duality gap is sufficiently small, i.e.,

$$\text{tr}\{\mathbf{L}\mathbf{X}\} - \mathbf{e}_n^T \mathbf{v} = \text{tr}\{\mathbf{X}\mathbf{Z}\} \times \varepsilon \quad (14)$$

where ε is an application specified accuracy.⁷

Since $\mathbf{X}^* \neq \mathbf{x}^*\mathbf{x}^{*T}$, we have to approximate \mathbf{x}^* from \mathbf{X}^* . One way is to assume that \mathbf{X}^* has rank one and let \mathbf{x}^* to be the sign of the last column of the \mathbf{X}^* . This is based on the knowledge of the dummy variable $u_n = 1$. Another way is to select \mathbf{x}^* to be the sign of the eigenvector corresponding to the largest eigenvalue of \mathbf{X}^* . In our simulations, we find that the second method gives a better estimate of \mathbf{x}^* .

⁶Practical experience indicates that this is usually the case [16].

⁷In this paper, $\varepsilon = 10^{-6}$.

B. Adding Cutting Planes

We now discuss how to combine semidefinite programming and cutting planes to strengthen the relaxation of (7). We follow the approach proposed by Helmburg and Rendl [12].

In general, we find a lot more violated inequalities than we are willing to include in our relaxation. To keep the approach reasonably efficient, it is important to select just a small number of promising inequalities from the vast set of violated inequalities.

In the following, we assume that m cutting planes of triangular type (8) have been added to the initial relaxation (7). These inequalities are described by the linear constraint $\mathbf{s} = \mathcal{H}(\mathbf{X}) - \mathbf{e}_m$, where $\mathbf{s} \geq \mathbf{0}$ is a vector of slack variables [12].⁸ The linear constraint $\mathcal{H}(\mathbf{X})$ is defined as

$$\mathcal{H}(\mathbf{X}) = \begin{bmatrix} \text{tr}\{\mathbf{X}\mathbf{h}_1\mathbf{h}_1^T\} \\ \vdots \\ \text{tr}\{\mathbf{X}\mathbf{h}_m\mathbf{h}_m^T\} \end{bmatrix} = \begin{bmatrix} \mathcal{H}_1(\mathbf{X}) \\ \vdots \\ \mathcal{H}_m(\mathbf{X}) \end{bmatrix}.$$

The corresponding adjoint operator $\mathcal{H}^T(\mathbf{w})$ [19] has the form

$$\mathcal{H}^T(\mathbf{w}) = \sum_{i=1}^m w_i \mathbf{h}_i \mathbf{h}_i^T$$

where $\mathbf{w} \in \mathcal{R}^m$. The primal semidefinite relaxation then reads

$$(\mathbf{X}^*, \mathbf{s}^*) = \arg \min_{\mathbf{X}, \mathbf{s}} \text{tr}\{\mathbf{L}\mathbf{X}\}$$

$$\text{s.t. } \text{diag}(\mathbf{X}) = \mathbf{e}_n, \mathbf{s} = \mathcal{H}(\mathbf{X}) - \mathbf{e}_m, \quad \mathbf{X} \succeq \mathbf{0}, \quad \mathbf{s} \geq \mathbf{0} \quad (15)$$

and its dual is

$$(\mathbf{v}^*, \mathbf{w}^*, \mathbf{Z}^*) = \arg \max_{\mathbf{v}, \mathbf{w}} \mathbf{e}_n^T \mathbf{v} + \mathbf{e}_m^T \mathbf{w}$$

$$\text{s.t. } \mathbf{Z} = \mathbf{L} - \text{Diag}(\mathbf{v}) - \mathcal{H}^T(\mathbf{w}), \quad \mathbf{t} - \mathbf{w} = \mathbf{0}, \quad \mathbf{Z} \succeq \mathbf{0}, \quad \mathbf{t} \geq \mathbf{0}.$$

The variables $\mathbf{v} \in \mathcal{R}^n$ and $\mathbf{w} \in \mathcal{R}_+^m$ are the Lagrange multipliers corresponding to the diagonal and the cutting plane constraints, respectively. The slack variables are $\mathbf{s} \in \mathcal{R}_+^m$ and $\mathbf{t} \in \mathcal{R}_+^m$.

In cutting plane algorithms, it is of fundamental importance that the basic operation, adding a few cutting planes and optimizing, can be carried out in reasonable time. Since the new inequalities separate the current iterate \mathbf{X}_c from the feasible set, this point is then infeasible after the addition of the cutting planes. Therefore, we have to construct a new feasible primal point \mathbf{X}_c^+ for restarting the iteration.

For the dual variables, we simply set the new v_i and w_i corresponding to the new inequalities to zero, and the corresponding t_i to one. This results in an infeasible dual point because feasibility requires $t_i = w_i$. However, these infeasibilities are removed after roughly three iterations of the Newton method.

To keep the number of inequalities small, we select only the most violated inequalities. Note that the identity matrix \mathbf{I} is the centroid of the primal feasible set [12]. We now address how to measure the degree of violation of a violated inequality. Instead of calculating the normal distance between \mathbf{X}_c and the inequality under consideration,⁹ the violation is measured with

respect to the centroid \mathbf{I} . This proves to be a better criterion in practice [12]. To calculate the degree of violation, we perform the following steps. Suppose that inequality $\mathcal{H}_i(\cdot) - 1$ is checked for violation. If feasibility is not violated,¹⁰ i.e., $\mathcal{H}_i(\mathbf{X}_c) - 1 \geq -\delta$, we look at the next inequality $i + 1$. If the inequality i is violated, i.e., $\mathcal{H}_i(\mathbf{X}_c) - 1 < -\delta$

$$\nu_i = \frac{\mathcal{H}_i(\mathbf{X}_c) - 1}{\mathcal{H}_i(\mathbf{X}_c) - \mathcal{H}_i(\mathbf{I})}$$

is computed which is now always positive. This is because $\mathcal{H}_i(\mathbf{X}_c) < 1$ and $\mathcal{H}_i(\mathbf{I}) = 3$. Then, the violated inequalities are sorted in decreasing order with respect to the ν_i s and a specified number of inequalities with the largest ν_i s are selected. This is called a large add of inequalities.

After the large add, we restart the interior-point iteration of (15). The update of \mathbf{X} is no longer feasible so we start from \mathbf{I} again. After q iterations, we check whether new triangle inequalities are significantly violated. If so, we change the model (15) again by including the additional violated inequalities. This is called a small add. After the small add, we start the iterations again by setting $\mathbf{X} = \mathbf{I}$.

In our algorithm, each large add adds n violated inequalities and is followed by some number of small adds which add $n/3$ inequalities each after consecutive steps of three iteration. Then, we calculate the optimal solution with these inequalities included. At termination, we select \mathbf{x}^* to be the sign of the eigenvector corresponding to the largest eigenvalue of the primal matrix \mathbf{X}^* as previously discussed.

V. SIMULATIONS

In this section, we report numerical results on the bit error rate (BER) performance of the detector based on semidefinite relaxation. A synchronous CDMA system with randomly selected long spreading codes are used. Two different scenarios are considered, a lightly loaded case with $K = 10$ and $N = 32$, as well as a more loaded case where $K = 24$ and $N = 32$. These systems are selected to make it possible to compare the performance to the optimal detector based on exhaustive search.

The BER performance of the semidefinite relaxation based and ML detectors are depicted in Fig. 3. For the lightly loaded case with $K = 10$, the basic semidefinite relaxation SDP1 is approximating the ML detector performance closely. No cutting plane is used except for $\text{SNR} = 9$ dB. First, we solve \mathbf{X} without using any cutting plane, if the rank of the sign of \mathbf{X}^* is greater than one, one large add is used in (15) to solve for \mathbf{X} . On average, it takes around 22 iterations to converge to the desired duality gap (14). For the more loaded case ($K = 24$), SDP1 does not provide a good approximation. The number of iterations required has also increased to around 25. Here, we have to use a combination of semidefinite relaxation and cutting planes SDP2, which needs around 70 iterations for convergence to obtain a good approximation. The number of small adds used is determined as follows. Initially, a large add is used in (15). Similarly, if the rank of the sign of \mathbf{X}^* is greater than one, ten small adds are added to solve for \mathbf{X} again. For this case, we have also reached ML performance. For both $K = 10$ and $K = 24$,

⁸The notation $\mathbf{s} \geq \mathbf{0}$ denotes that all elements of \mathbf{s} are greater than or equal to zero.

⁹This distance is $\mathcal{H}_i(\mathbf{X}_c) - 1$.

¹⁰ δ is an application specific parameter and we set it to 10^{-6} .

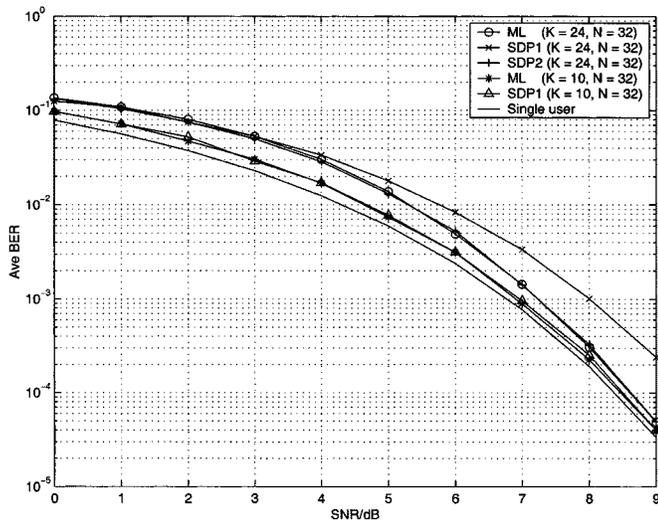


Fig. 3. Average BER of the SDP based detectors.

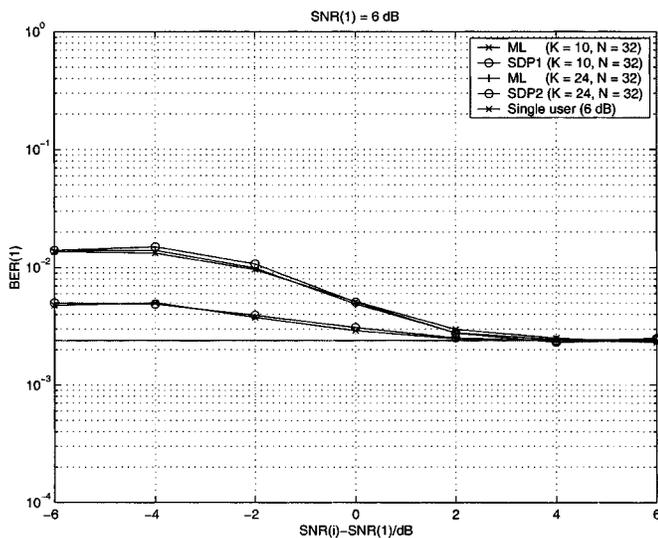
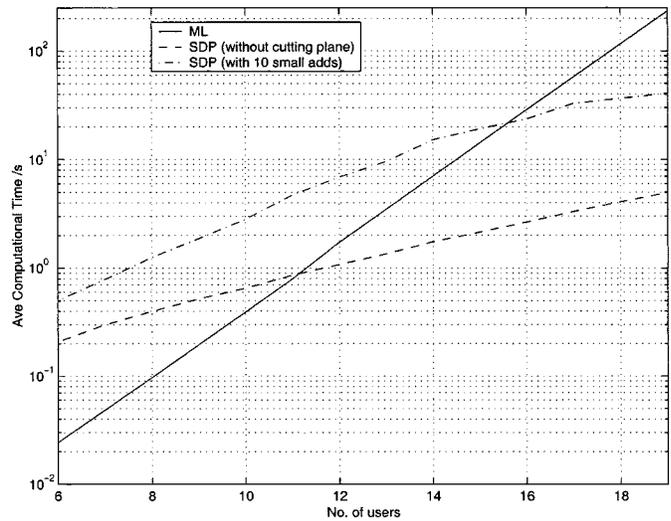


Fig. 4. Average BER of the SDP based detectors.

ML performance is reached with polynomial complexity in the number of users.

To illustrate near-far resistance, we consider the case where all but user 1 have the same signal-to-noise ratio (SNR). User 1 stays at a fixed SNR. In Fig. 4, the BER of the first user with $\text{SNR}(1) = 6$ dB is shown against the ratio of the strength of the interfering users' signals to the first user's signal strength ($\text{SNR}(i) - \text{SNR}(1)$ in dB). The simulation results show no appreciable performance difference between semidefinite relaxation and ML detection for $K = 10$ and $K = 24$.

Simulations are used to evaluate the average computational time of the ML detector using exhaustive search and the detectors based on semidefinite relaxation. The result is shown in Fig. 5. The simulations are run in the MATLAB environment on a 600-MHz Pentium III personal computer with 512 Mb of RAM. The computational time of ML using exhaustive search increases exponentially with the number of users. When K is large, the computational times of SDP based detectors are significantly lower than that of the exhaustive search. Based on a

Fig. 5. Average computation time against K .

series of simulations as above, the complexity can be approximated through curve fitting techniques. It follows that the complexity (in time/s) is approximated as

$$\begin{aligned} C_{\text{op}} &\approx 4 \times 10^{-4} 2^K \quad (\text{Exhaustive search}) \\ C_{\text{sdp}} &\approx 1.4 \times 10^{-3} K^3 \quad (\text{SDP}) \\ C_{\text{sdp,cp}} &\approx 7.2 \times 10^{-3} K^3 \quad (\text{SDP with 10 small cuts}) \end{aligned}$$

VI. CONCLUSION

In this paper, we have introduced a semidefinite relaxation of the optimal ML detection problem as a complexity-limiting alternative. Cutting planes were added to strengthen this relaxation. An interior-point method with polynomial complexity was used to solve the semidefinite programs arising from the relaxation. The numerical examples show that there is no appreciable performance difference between the semidefinite relaxation and optimal ML detection. For a lightly loaded case, the basic semidefinite program is sufficient to approximate the ML solution. When the load increases, cutting planes are required. In both cases, it was demonstrated that practical ML performance can be achieved with polynomial complexity in the number of users.

REFERENCES

- [1] S. Moshavi, "Multi-user detection for DS-SSMA communications," *IEEE Commun. Mag.*, vol. 34, pp. 132–136, Oct. 1996.
- [2] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 85–96, Jan. 1986.
- [3] —, "Computational complexity of optimum multiuser detection," *Algorithmica*, vol. 4, pp. 303–312, 1989.
- [4] C. Sankaran and T. Ephremides, "Solving a class of optimum multiuser detection problems with polynomial complexity," *IEEE Trans. Inform. Theory*, vol. 44, pp. 1958–1961, Sept. 1998.
- [5] S. Ulukus and R. D. Yates, "Optimum multiuser detection is tractable for synchronous CDMA using m -sequences," *IEEE Commun. Lett.*, vol. 2, pp. 89–91, Apr. 1998.
- [6] R. Nilsson, F. Sjögren, O. Edfors, P. Ödling, H. Erikon, S. K. Wilson, and P. O. Börjesson, "A low complexity threshold detector making MLSD decision in a multiuser environment," *Proc. IEEE Vehicular Technology Conf.*, pp. 333–337, May 1998.

- [7] L. B. Nelson and H. V. Poor, "Iterative multiuser receivers for CDMA channels: An EM-based approach," *IEEE Trans. Commun.*, vol. 44, pp. 1700–1710, Dec. 1996.
- [8] Z. L. Shi, W. Du, and P. F. Driessen, "A new multistage detector for synchronous CDMA communications," *IEEE Trans. Commun.*, vol. 44, pp. 538–541, May 1996.
- [9] A. Yener, R. D. Yates, and S. Ulukus, "A nonlinear programming approach to CDMA multiuser detection," in *Proc. Asilomar Conf. Signal, Systems and Computers*, Pacific Grove, CA, Oct. 1999, pp. 1579–1583.
- [10] P. H. Tan, L. K. Rasmussen, and T. J. Lim, "Iterative interference cancellation as maximum-likelihood detection in CDMA," presented at the Int. Conf. Information, Communication, Signal Processing, Singapore, Dec. 1999.
- [11] —, "Constrained maximum-likelihood detection in CDMA," *IEEE Trans. Commun.*, vol. 49, pp. 142–153, Jan. 2001.
- [12] C. Helmberg and F. Rendl, "Solving quadratic (0, 1)-problems by semidefinite programs and cutting planes," *Math. Program.*, vol. 82, no. 3, pp. 219–315, 1998.
- [13] L. K. Rasmussen, P. D. Alexander, and T. J. Lim, *CDMA Techniques for 3rd Generation Mobile System*, F. Swarts, P. van Rooyen, I. Oppermann, and M. Lotter, Eds. Norwell, MA: Kluwer, 1998. A Linear model for CDMA signals received with multiple antennas over multipath fading channels.
- [14] Y. E. Nesterov and A. S. Nemirovskii, *Interior Point Polynomial Algorithms in Convex Programming*, *SIAM Studies in Applied Mathematics*. Philadelphia, PA: SIAM, 1994.
- [15] C. Helmberg, Semidefinite programming, in ZIB Preprint SC99-49, 1999.
- [16] F. Rendl, "Semidefinite programming and combinatorial optimization," *Appl. Num. Math.*, no. 29, pp. 255–281, 1999.
- [17] M. Laurent and S. Poljak, "On a positive semidefinite relaxation of the cut polytope," *Linear Algebra Appl.*, vol. 223, pp. 439–461, 1995.
- [18] M. Deza, M. Laurent, and S. Poljak, "The cut cone iii: On the triangle facets," *Graph. Combin.*, vol. 7, no. 1, pp. 125–142, 1996.
- [19] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz, "An interior point method for semidefinite programming," *SIAM J. Optim.*, vol. 7, no. 3, pp. 342–361, 1996.
- [20] M. Kojima, S. Shindoh, and S. Hara, "Interior-point methods for the monotone linear complementarity problem in symmetric matrices," *SIAM J. Optim.*, vol. 8, no. 7, pp. 86–125, 1997.
- [21] R. Monteiro, "Primal-dual path following algorithms for semidefinite programming," *SIAM J. Optim.*, vol. 8, no. 7, pp. 663–678, 1997.
- [22] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1985.
- [23] M. J. Juntti, B. Aazhang, and J. O. Lilleberg, "Iterative implementation of linear multiuser detection for dynamic asynchronous CDMA systems," *IEEE Trans. Commun.*, vol. 46, pp. 503–508, Apr. 1998.

Peng Hui Tan (S'99) received the B.Eng. and M.Eng. degrees in electrical and electronic engineering from National University of Singapore, Singapore in 1998 and 1999, respectively. He is currently working toward the Ph.D. degree with the Telecommunication Theory Group, Department of Computer Engineering, Chalmers University of Technology, Gothenburg, Sweden.

Lars K. Rasmussen (S'92–M'93–SM'01) was born on March 8, 1965, in Copenhagen, Denmark. He received the M.Eng. degree in 1989 from the Technical University of Denmark, Lyngby, and the Ph.D. degree from Georgia Institute of Technology, Atlanta, GA, in 93.

From 1993 to 1995, he was a Research Fellow with the University of South Australia, Adelaide, Australia. From 1995 to 1998, he was a Senior Member of Technical Staff with the Center for Wireless Communications, National University of Singapore, Singapore. He is now an Associate Professor with Chalmers University of Technology, Gothenburg, Sweden.