Correpondence

A Near-Optimal Multiuser Detector for DS-CDMA Systems Using Semidefinite Programming Relaxation

X. M. Wang, W.-S. Lu, and A. Antoniou

***Abstract*—A multiuser detector for direct-sequence code-division multiple-access systems based on semidefinite programming (SDP) is proposed. It is shown that maximum likelihood (ML) detection can be carried out by "relaxing" the associated integer programming problem to a dual SDP problem, which leads to a detector of polynomial complexity. Computer simulations that demonstrate that the proposed detector offers near-optimal performance with considerably reduced computational complexity compared with that of existing primal-SDP-relaxation based detectors are presented.**

***Index Terms*—Duality theory, maximum likelihood detection, multiuser detection, semidefinite programming.**

I. Introduction

DEMODULATION in the presence of multiple access interference (MAI) is of central importance in multiuser wireless communications. The optimal maximum likelihood (ML) multiuser detector for direct-sequence code-division multiple-access (DS-CDMA) channels has been proposed by Verdù [1], [2]. Except for some special circumstances where the crosscorrelation matrix of the user signatures is well structured [3], ML detection is carried out by solving a combinatorial optimization problem involving a quadratic objective function of a binary variable vector. Recently, several authors have proposed a new solution method where the integer programming problem involved in the ML detection is "relaxed" into a semidefinite programming (SDP) problem [4]–[6].

Although the SDP relaxation based detector is of polynomial complexity, the associated SDP problem involves a matrix variable of dimension $(K+1)\times(K+1)$, where K denotes the number of users in the system. Hence, it requires a large amount of computation even for a moderate number of users. In this correspondence, it is shown that the detection can be carried out by solving a dual SDP problem that involves only a vector variable of size $K+1$, and therefore, the computational complexity can be significantly reduced compared with that required by the existing SDP relaxation method. Computer simulations are presented to demonstrate the performance and efficiency of the proposed detector.

II. Signal Model and Problem Formulation

We consider a synchronous DS-CDMA system in which K users transmit antipodal signals through a single-path, frequency-nonselective, slowly Rayleigh fading channel. The bit interval of each user is T_b s, and each signal is assigned a normalized signature waveform $s(t)$ given by

$$s(t)=\sum_{i=1}^{N}(-1)^{c_i}p_{T_c}[t-(i-1)T_c] \quad \text{for } t\in[0,T_b] \tag{1}$$

where $p_{T_c}(t)$ is the chip waveform that takes a nonzero value between $0\le t\le T_c$ and is zero elsewhere, $\{c_1,c_2,\ldots,c_N\}$ is a binary sequence, and $N=T_b/T_c$ is the spreading gain. The received baseband signal is given by

$$y(t)=\sum_{k=1}^{K}A_k b_k s_k(t)+n(t) \tag{2}$$

where b_k is an information bit, A_k is the signal amplitude of the kth user's signal, and $n(t)$ is an additive white Gaussian noise (AWGN) process with zero-mean and variance σ^2.

The demodulation begins by filtering the received signal $y(t)$ with a bank of K matched filters. The output of the kth matched filter is given by

$$\begin{aligned} y_k &= \int_0^{T_b} y(t)s_k(t)\,dt \\ &= A_k b_k+\sum_{j\neq k}A_j b_j \rho_{jk}+n_k \end{aligned} \tag{3}$$

where $\rho_{jk}=\int_0^{T_b} s_j(t)s_k(t)\,dt$, and $n_k=\int_0^{T_b} n(t)s_k(t)\,dt$. In matrix form, the model in (3) can be written as

$$\mathbf{y}=\mathbf{RAb}+\mathbf{n} \tag{4}$$

where $\mathbf{R}=\{\rho_{jk}\}$ is the synchronous crosscorrelation matrix, $\mathbf{b}=[b_1\ b_2\ \cdots\ b_K]^T$ is a vector whose elements are the information bits of the K users, and $\mathbf{n}=[n_1\ n_2\ \cdots\ n_K]^T$ is a vector of zero-mean Gaussian random variables whose crosscorrelation matrix is $E[\mathbf{nn}^T]=\sigma^2\mathbf{R}$.

The objective of multiuser detection is to identify the transmitted information vector $\mathbf{b}$ from $y(t)$ in (2), which is the superposition of the received user signals and ambient channel noise. Among various multiuser detectors, the ML detector has been shown to have optimal demodulation performance and is often used as a baseline for comparison of other detectors in DS-CDMA systems. In the ML detector, the detection is carried out by solving a combinatorial optimization problem given by

$$\text{minimize } \mathbf{x}^T\mathbf{Hx}+\mathbf{x}^T\mathbf{p} \tag{5a}$$

$$\text{subject to: } x_i\in\{1,-1\} \quad \text{for } i=1,2,\ldots,K \tag{5b}$$

where x_i denotes the ith entry of $\mathbf{x}, \mathbf{H}=\mathbf{ARA}$, and $\mathbf{p}=-2\mathbf{Ay}$. Because of the binary constraints in (5b), the computational complexity involved in the ML detection increases exponentially with respect to the number of users. Therefore, the implementation of the ML detector becomes prohibitive even for a moderate number of users.

Manuscript received February 27, 2002; revised February 7, 2003. This work was supported by PMC-Sierra, Micronet, NCE Program, and the Natural Sciences and Engineering Research Council of Canada. The associate editor coordinating the review of this paper and approving it for publication was Dr. Sergios Theodoridis.

The authors are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, B.C. V8W 3P6 Canada.

Digital Object Identifier 10.1109/TSP.2003.815382

III. MULTIUSER DETECTOR USING SEMIDEFINITE PROGRAMMING RELAXATION

A. Semidefinite Programming

Semidefinite programming (SDP) is a class of mathematical methods for the solution of optimization problems where the objective functions are linear, and the constraints are linear matrix inequalities. A typical SDP problem formulation is given by

$$\text{minimize } \mathbf{c}^T\mathbf{x} \tag{6a}$$

$$\text{subject to: } \mathbf{F}(\mathbf{x}) = \mathbf{F}_0 + \sum_{i=1}^{n} x_i\mathbf{F}_i \succeq \mathbf{0} \tag{6b}$$

where $\mathbf{x} = [x_1\ x_2\ \cdots\ x_n]^T$ denotes the variable vector, $\mathbf{c} \in R^{n\times 1}$, and $\mathbf{F}_i \in R^{m\times m}$ for $0 \le i \le n$ are symmetric matrices. The notation $\mathbf{F}(\mathbf{x}) \succeq \mathbf{0}$ denotes that matrix $\mathbf{F}(\mathbf{x})$ is positive semidefinite. Efficient interior-point optimization algorithms and software for SDP have been developed in the past several years [7]–[10].

B. SDP Relaxation of MAX-CUT Problem

The MAX-CUT problem is a well-known integer programming problem in graph theory. It can be formulated as [11]

$$\text{minimize tr}(\mathbf{WX}) \tag{7a}$$

$$\text{subject to: } \mathbf{X} \succeq \mathbf{0} \tag{7b}$$

$$x_{ii} = 1 \quad \text{for } 1 \le i \le n \tag{7c}$$

$$\text{rank}(\mathbf{X}) = 1 \tag{7d}$$

where the elements of $\mathbf{W}$ are the weights associated with a graph with $w_{ii} = 0$ and x_{ii} denotes the ith diagonal element of $\mathbf{X}$.

In [11], Geomans and Williamson proposed a relaxation of the above problem by removing the rank constraint in (7d), which leads to

$$\text{minimize tr}(\mathbf{WX}) \tag{8a}$$

$$\text{subject to: } \mathbf{X} \succeq \mathbf{0} \tag{8b}$$

$$x_{ii} = 1 \quad \text{for } 1 \le i \le n. \tag{8c}$$

Note that the objective function in (8a) is a linear function of $\mathbf{X}$, and the constraints in (8b) and (8c) can be combined into a linear matrix inequality (LMI) as

$$\sum_{i>j}\sum x_{ij}\mathbf{F}_{ij} + \mathbf{I} \succeq \mathbf{0} \tag{9}$$

where, for each (i,j) with $i > j$, $\mathbf{F}_{ij}$ is a symmetric matrix with its (i,j)th and (j,i)th entries being ones and all the remaining entries being zeros. Therefore, the problem in (8) is of the type given in (6), and it, therefore, is an SDP problem. For this reason, the problem in (8) is known as an *SDP relaxation* of the problem in (7).

C. SDP-Relaxation-Based Multiuser Detector

Let

$$\hat{\mathbf{X}} = \begin{bmatrix} \mathbf{x}\mathbf{x}^T & \mathbf{x} \\ \mathbf{x}^T & 1 \end{bmatrix} \quad \text{and} \quad \mathbf{C} = \begin{bmatrix} \mathbf{H} & \mathbf{p}/2 \\ \mathbf{p}^T/2 & 1 \end{bmatrix} \tag{10}$$

By noting that tr$(\mathbf{AB})$ = tr$(\mathbf{BA})$, the objective function in (5a) can be expressed as

$$\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{p} = \text{tr}(\mathbf{C}\hat{\mathbf{X}}). \tag{11}$$

Using an argument similar to that in Section III-B, the constraints in (5b) can be converted to

$$\hat{\mathbf{X}} \succeq 0, \quad \hat{x}_{ii} = 1 \quad \text{for } 1 \le i \le K \tag{12a}$$

$$\text{rank}(\hat{\mathbf{X}}) = 1 \tag{12b}$$

where $\hat{x}_{ii}$ denotes the ith diagonal element of $\hat{\mathbf{X}}$. By removing the rank constraint in (12b), we obtain an SDP relaxation of (5) as

$$\text{minimize tr}(\mathbf{C}\hat{\mathbf{X}}) \tag{13a}$$

$$\text{subject to: } \hat{\mathbf{X}} \succeq \mathbf{0} \tag{13b}$$

$$\hat{x}_{ii} = 1 \quad \text{for } i = 1, 2, \ldots, K+1. \tag{13c}$$

If we collect the variables involved in $\hat{\mathbf{X}}$ into a vector $\mathbf{x}$, then the objective function in (13a) can be expressed as $\mathbf{c}^T\mathbf{x}$ for some constant vector $\mathbf{c}$. In addition, as argued at the end of Section III-B, the constraints in (13b) and (13c) can be expressed as the LMI in (9). Thus, the problem in (13) can be expressed in the form of (6), and therefore, it is an SDP problem.

D. Binary Solution

The variables in the original problem in (5) assume values of only 1 or -1, whereas the variable $\mathbf{X}$ in the SDP minimization problem (13) has nonbinary values. In what follows, we describe two approaches that can be used to generate a binary solution for (5) based on the solution $\hat{\mathbf{X}}$ of the SDP problem in (13).

Let the solution of (13) be denoted as $\hat{\mathbf{X}}^*$. It follows from (10) that $\hat{\mathbf{X}}^*$ is a $(K+1)\times(K+1)$ symmetric matrix of the form

$$\hat{\mathbf{X}}^* = \begin{bmatrix} \mathbf{X}^* & \mathbf{x}^* \\ \mathbf{x}^{*T} & 1 \end{bmatrix} \tag{14}$$

with $(\hat{x}^*)_{ii} = 1$ for $i = 1, 2, \ldots, K$. In view of the structure of (14), the first approach is simply to apply operator sgn$(\cdot)$ to $\mathbf{x}^*$ in (14), namely,

$$\hat{\mathbf{b}} = \text{sgn}[\hat{\mathbf{X}}^*(1:K, K+1)] \tag{15}$$

where $\hat{\mathbf{X}}^*(1:K, K+1)$ denotes the vector formed by using the first K entries in the last column of $\hat{\mathbf{X}}^*$.

A better binary solution, at the cost of increased computation, can be obtained by using the eigen-decomposition of matrix $\hat{\mathbf{X}}^*$, i.e., $\hat{\mathbf{X}}^* = \mathbf{USU}^T$, where $\mathbf{U}$ is an orthogonal matrix, and $\mathbf{S}$ is a diagonal matrix with the eigenvalues of $\hat{\mathbf{X}}^*$ on its diagonal in decreasing order [13]. It is well known that an optimal rank-one approximation of $\hat{\mathbf{X}}^*$ in the 2-norm sense is given by $\lambda_1\mathbf{u}_1\mathbf{u}_1^T$, where λ_1 is the largest eigenvalue of $\hat{\mathbf{X}}^*$, and $\mathbf{u}_1$ is the eigenvector associated with λ_1 [13]. If we denote the vector formed by using the first K entries of $\mathbf{u}_1$ as $\tilde{\mathbf{u}}$ and the last entry of $\mathbf{u}_1$ as u_{K+1}, i.e., $\mathbf{u}_1 = [\tilde{\mathbf{u}}^T\ \ u_{K+1}]^T$, then the optimal rank-one approximation of $\hat{\mathbf{X}}^*$ can be expressed as

$$\begin{aligned} \hat{\mathbf{X}}^* \approx \lambda_1\mathbf{u}_1\mathbf{u}_1^T &= \lambda_1 \begin{bmatrix} \tilde{\mathbf{u}}\tilde{\mathbf{u}}^T & u_{K+1}\tilde{\mathbf{u}} \\ u_{K+1}\tilde{\mathbf{u}}^T & u_{K+1}^2 \end{bmatrix} \\ &= \lambda_1 u_{k+1}^2 \begin{bmatrix} \tilde{\mathbf{x}}_1\tilde{\mathbf{x}}_1^T & \tilde{\mathbf{x}}_1 \\ \tilde{\mathbf{x}}_1^T & 1 \end{bmatrix} \end{aligned} \tag{16}$$

where $\tilde{\mathbf{x}}_1 = \tilde{\mathbf{u}}/u_{k+1}$. Since $\lambda_1 > 0$, upon comparing (16) with (14), we see that vector $\tilde{\mathbf{x}}_1$ is a reasonable approximation of $\mathbf{x}^*$. Therefore, a binary solution of (5) can be generated as

$$\hat{\mathbf{b}} = \begin{cases} \text{sgn}[\mathbf{u}_1(1:K)], & \text{if } \mathbf{u}_1(K+1) > 0 \\ -\text{sgn}[\mathbf{u}_1(1:K)], & \text{if } \mathbf{u}_1(K+1) < 0 \end{cases} \tag{17}$$

IV. EFFICIENT SDP SOLUTION VIA DUALITY

Although interior-point algorithms [8]–[10] can be applied to the SDP problem at hand and their computational complexity increases in a nonexponential manner, numerical difficulties may arise in solving (13) because of the large number of variables involved, even for the detection of a moderate number of users. In this section, we present a new algorithm that can be used to obtain a solution of the SDP problem at hand more efficiently. The proposed method entails two steps, as follows: First, the so-called dual of the SDP problem is solved, and second, the solution obtained for the dual problem is converted to the solution of the primal SDP problem.

The SDP problem in (13) can be expressed as

$$\text{minimize tr}(\mathbf{C}\hat{\mathbf{X}}) \tag{18a}$$

$$\text{subject to: } \hat{\mathbf{X}} \succeq \mathbf{0} \tag{18b}$$

$$\text{tr}(\mathbf{A}_i\mathbf{X}) = 1 \quad \text{for } i = 1, \ldots, K+1 \tag{18c}$$

where $\mathbf{A}_i$ is a diagonal matrix whose diagonal entries are all zeros except the ith entry, which is one. It follows from [10] and [14] that the dual of the problem in (18) is given by

$$\text{minimize } -\mathbf{b}^T\mathbf{y} \tag{19a}$$

$$\text{subject to: } \mathbf{S} = \mathbf{C} - \sum_{i=1}^{K+1} y_i\mathbf{A}_i \tag{19b}$$

$$\mathbf{S} \succeq \mathbf{0} \tag{19c}$$

where $\mathbf{y} = [y_1 \ \cdots \ y_{K+1}]^T$, and $\mathbf{b} = [1 \ \cdots \ 1]^T \in \mathcal{C}^{(K+1)\times 1}$. Note that the dual problem in (19) involves only $K+1$ variables, and it can be solved efficiently by using interior-point algorithms such as the projective method proposed by Nemirovskii and Gahinet [15].

In order to obtain the solution of the primal SDP problem in (18), the Karush-Kuhn-Tucker (KKT) conditions for the solutions of the problems in (18) and (19) need to be examined. The KKT conditions state that $\{\hat{\mathbf{X}}^*, \mathbf{y}^*\}$ solves the problems in (18) and (19) if and only if they satisfy the conditions

$$\sum_{i=1}^{K+1} y_i^*\mathbf{A}_i + \mathbf{S}^* = \mathbf{C} \tag{20a}$$

$$\text{tr}(\mathbf{A}_i\hat{\mathbf{X}}^*) = 1 \quad \text{for } i = 1, \ldots, K+1 \tag{20b}$$

$$\mathbf{S}^*\hat{\mathbf{X}}^* = \mathbf{0} \tag{20c}$$

$$\hat{\mathbf{X}}^* \succeq \mathbf{0} \quad \text{and} \quad \mathbf{S}^* \succeq \mathbf{0}. \tag{20d}$$

From (20a), we have

$$\mathbf{S}^* = \mathbf{C} - \sum_{i=1}^{K+1} y_i^*\mathbf{A}_i. \tag{21}$$

Since the solution $\mathbf{y}^*$ is typically obtained from an *iterative* algorithm, $\mathbf{y}^*$ can at best be a good approximate solution of (20), which means that $\mathbf{y}^*$ is in the *interior* of the feasible region. Consequently, matrix $\mathbf{S}^*$ remains *positive definite*. Therefore, the set $\{\mathbf{y}^*, \mathbf{S}^*, \hat{\mathbf{X}}^*\}$ can be regarded as a point in the feasible region that is sufficiently close to the limiting point of the central path for the problems in (18) and (19). Recall that the central path is defined as a parameterized set $\{\mathbf{y}(\tau), \mathbf{S}(\tau), \hat{\mathbf{X}}(\tau) \text{ for } \tau > 0\}$ that satisfies the modified KKT conditions [10]

$$\sum_{i=1}^{K+1} y_i(\tau)\mathbf{A}_i + \mathbf{S}(\tau) = \mathbf{C} \tag{22a}$$

$$\text{tr}(\mathbf{A}_i\hat{\mathbf{X}}(\tau)) = 1 \quad \text{for } i = 1, \ldots K+1 \tag{22b}$$

$$\mathbf{S}(\tau)\hat{\mathbf{X}}(\tau) = \tau\mathbf{I} \tag{22c}$$

$$\hat{\mathbf{X}}(\tau) \succeq \mathbf{0} \quad \text{and} \quad \mathbf{S}(\tau) \succeq \mathbf{0}. \tag{22d}$$

The relation between the equations in (20) and those in (22) becomes apparent as one realizes that the entire central path defined by (22) lies in the interior of the feasible region, and as $\tau \to 0$, the path converges to the solution set $\{\mathbf{y}^*, \mathbf{S}^*, \hat{\mathbf{X}}^*\}$ that satisfies (20).

From (22c), it follows that

$$\hat{\mathbf{X}}(\tau) = \tau\mathbf{S}^{-1}(\tau) \tag{23}$$

which suggests an approximate solution of (18) as

$$\hat{\mathbf{X}} = \tau(\mathbf{S}^*)^{-1}. \tag{24}$$

In (24), τ is a sufficiently small constant such that $\tau > 0$ and $\mathbf{S}^*$ is given by (21). In order for matrix $\hat{\mathbf{X}}$ in (24) to satisfy the equality constraints in (18c), $\hat{\mathbf{X}}$ needs to be slightly modified by using a scaling matrix $\mathbf{\Pi}$, i.e.,

$$\hat{\mathbf{X}}^* = \mathbf{\Pi}(\mathbf{S}^*)^{-1}\mathbf{\Pi} \tag{25}$$

where $\mathbf{\Pi} = \text{diag}\{\xi_1^{1/2} \ \cdots \ \xi_{K+1}^{1/2}\}$, and ξ_i is the ith diagonal entry of $(\mathbf{S}^*)^{-1}$. In (25), we have pre- and post-multiplied $(\mathbf{S}^*)^{-1}$ by $\mathbf{\Pi}$ to ensure that matrix $\hat{\mathbf{X}}^*$ remains *symmetric* and *positive definite*. It is worth noting that by imposing the equality constraints (18c) on $\hat{\mathbf{X}}$, the parameter τ in (24) is incorporated in the scaling matrix $\mathbf{\Pi}$. The steps to compute an approximate solution $\hat{\mathbf{X}}$ of the SDP problem (18) can be summarized as follows.

i) Form matrix $\mathbf{C}$ using (10).
ii) Solve the dual SDP problem in (19), and denote its solution as $\mathbf{y}^*$.
iii) Compute $\mathbf{S}^*$ using (21).
iv) Compute $\hat{\mathbf{X}}^*$ using (25).
v) Compute $\hat{\mathbf{b}}$ using (15) or (17).

Two remarks can be made pertaining to the computational complexity of the proposed algorithm and the accuracy of the solution obtained. To a large extent, the computational complexity of our algorithm is determined by steps ii) and iv), where a $(K+1)$-variable SDP problem is solved, and a $(K+1) \times (K+1)$ positive definite matrix is inverted, respectively. Compared with the computations required to directly solve the $K(K+1)/2$-variable SDP problem in (18), the new method reduces the computational complexity by a considerable amount. Concerning the accuracy of the solution, we note that it is the binary solution that determines the performance of the multiuser detector. Since the binary solution is the result of the sign operation [see (15) and (17)], the approximation introduced in (25) is expected to have a negligible effect on the accuracy of the solution.

The use of semidefinite programming was also explored recently in [5] and [6]. The method in [5] also uses relaxation, but a randomization method is used to convert the solution of the SDP problem into the binary detection output as opposed to our eigen-decomposition-based method. In the method reported in [6], which is also based on the eigen-decomposition, a cutting plane method is used the improve the detection performance of the SDPR detector for a system with a large number of users, and satisfactory results have been reported. However, methods for reducing the computational complexity involved in solving the SDP problem, which is of key concern as will be demonstrated in the next section, have not been explored in these papers.

V. Simulation Results

Computer simulations were carried out to evaluate the performance of the SDP-relaxation (SDPR)-based multiuser detector in terms of bit-error rate (BER) and computational complexity and to compare the proposed detectors with the ML detector, the conventional matched-filter detector, the linear decorrelating detector [16], and the linear MMSE detector [17].

Two types of SDPR detectors were implemented using the MATLAB LMI control toolbox [9]: one based on the primal and the other based on the dual SDP formulation. In what follows, these detectors will be referred to as the SDPR-P and SDPR-D detectors. Since the ultimate goal of the detector is to estimate the *binary-valued* information vector $\mathbf{b}$, a fairly large convergence tolerance $\varepsilon = 10^{-2}$ was used in order to keep the number of iterations, and, hence, the computational complexity, low. In addition, an upper bound $K_u = 5$ was imposed on the number of iterations. In our simulations, 10^5 runs were performed to evaluate the average performance for each of the ML, linear MMSE (LMMSE),

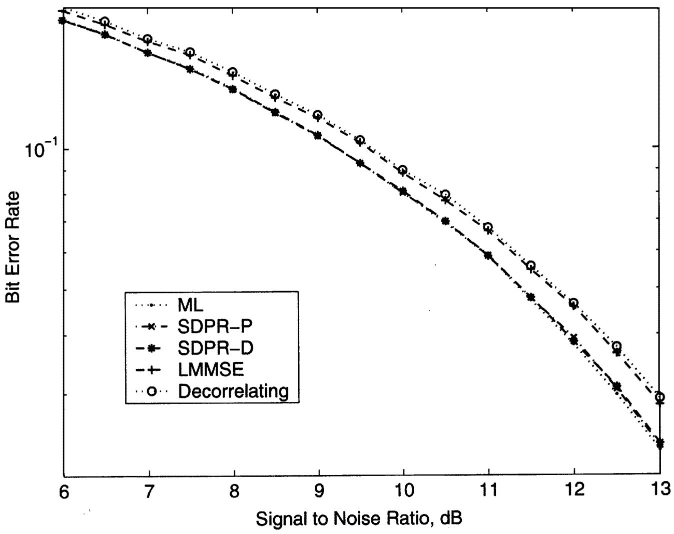

Fig. 1. BER versus SNR for six-user synchronous DS-CDMA system in AWGN channel.

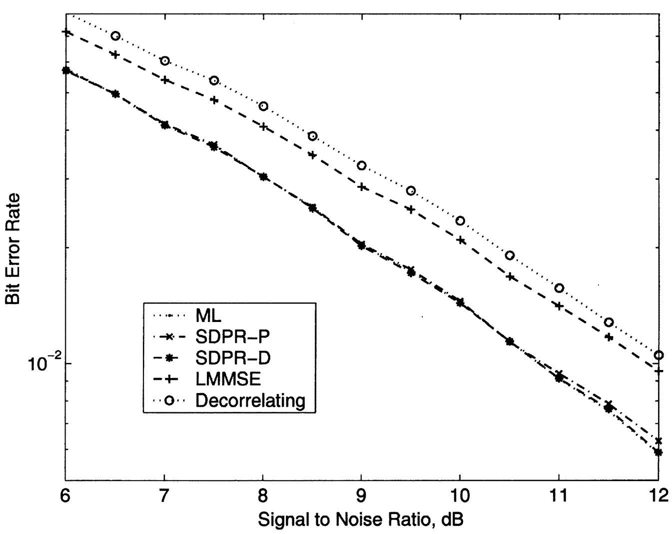

Fig. 2. BER versus SNR for eight-user synchronous DS-CDMA system in flat Rayleigh fading channel.

matched-filter (MF), decorrelating (DD), SDPR-P, and SDPR-D detectors.

In the first set of simulations, a six-user synchronous system using 15-chip Gold sequences for signatures was considered. The received signal powers of the six users were set to 5, 3, 1.8, 0.6, 0.3, and 0.2. The last user with power 0.2 was designated as the desired user. The average BER versus SNR for the various detectors is plotted in Fig. 1.

The second set of simulations concerned an eight-user synchronous system in a frequency-flat Rayleigh fading channel. The user signatures used were the same as in the first simulation and the received signal power of the eight users were set to 5, 3, 1.8, 1, 0.6, 0.3, 0.2, and 0.1. The fourth user (the one with unity power) was designated as the desired user, and the coherent time of the fading channel was set to ten times the bit duration. The signal-to-noise ratio (SNR) was assumed to be known to the MMSE detector. The average BER versus SNR for the various detectors is plotted in Fig. 2.

The third set of simulations were carried out to examine the near–far resistance of the SDPR-P and SDPR-D detectors for a six-user synchronous system using the same user signatures as before. The SNR of the desired user signal was fixed to 8 dB higher than channel noise.

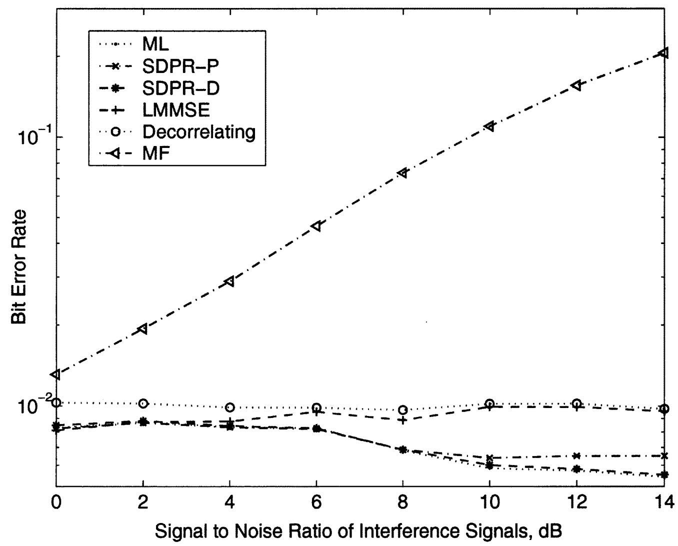

Fig. 3. Near–far performance of six-user synchronous DS-CDMA system in AWGN channel.

The SNR of the five interference user signals were identical, and the SNR varied from 0 to 14 dB during the transmission. The average BER versus SNR for the ML, LMMSE, DD, MF (conventional matched filter), SDPR-P, and SDPR-D detectors obtained is plotted in Fig. 3.

From these simulations, it is observed that the demodulation performance of the SDPR-P and SDPR-D detectors is consistently very close to that of the ML detector and is superior relative to that of the decorrelating and MMSE detectors. It is also observed that the bit error rate of the SDPR-D detector is slightly smaller than that of the SDPR-P detector, in particular, when the SNR is high. We believe that this performance improvement can be attributed to the fact that the size and, therefore, the complexity of the SDP problem involved in the SDPR-D formulation is considerably smaller than its primal counterpart, and therefore, the SDPR-D detector is more robust with respect to numerical errors. This is particularly so when the SNR is high because the data used in the simulation are less random, and the numerical errors become a more dominant source of performance degradation.

The computational complexity of the various detectors was evaluated in terms of the CPU time and the floating-point operations (flops) reported by the MATLAB LMI Control toolbox. The logarithm of the averaged CPU time and computation flops for the ML, SDPR-P, and SDPR-D detectors are plotted versus the number of active users in Fig. 4. The number of users for the ML detector was restricted to the range 10 to 17 to avoid the extremely high computational effort involved, but for the sake of comparison, the curve was extrapolated as shown in Fig. 4, assuming that the trend established for the range 10 to 17 users continues for a larger numbers of users.

As can be observed in Fig. 4, the SDPR detectors are considerably more efficient than the ML detector when the number of users is large, and the SDPR-D detector is much more efficient than the SDPR-P detector. The SDPR-P detector requires 8.6% and 0.05% the computational effort required by the ML detector for 15 and 25 users, respectively. On the other hand, the SDPR-D detector requires 10.43% and 7.86% of the computational effort required by the SDPR-P detector, again for 15 and 25 users, respectively.

Based on the simulation results obtained, the computational complexity of the three detectors can be quantified in terms of the approximate CPU time, which is given by

$$\mathcal{C}_{\mathrm{ML}} \approx 4.5 \times 10^{-4} \cdot 2^{K} \tag{26a}$$

$$\mathcal{C}_{\mathrm{SDPR-P}} \approx 4.2 \times 10^{-5} K^{3.9} \tag{26b}$$

$$\mathcal{C}_{\mathrm{SDPR-D}} \approx 2.6 \times 10^{-5} K^{3.3} \tag{26c}$$

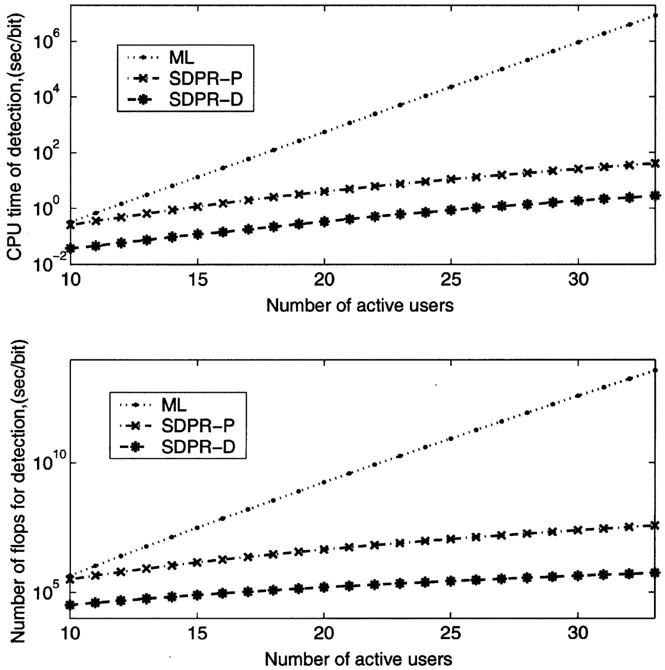

Fig. 4. Computational complexity of the demodulation for the ML, SDPR-P, and SDPR-D detectors.

where K is the number of users. In effect, the computational complexity for the ML detector increases exponentially with the number of users, whereas that of the SDPR detectors is of polynomial order.

It should be mentioned here that the computational effort pertaining to the SDPR-D detector can be further reduced by taking the special structure of matrix $\mathbf{A}_i$ into consideration, but this possibility has yet to be explored.

VI. Conclusion

A multiuser detector for DS-CDMA systems based on SDP has been proposed. It has been shown that the ML detection can be carried out by "relaxing" the associated integer programming problem to a dual SDP problem, which leads to a detector of polynomial complexity. Computer simulations that demonstrate that the proposed detector offers near-optimal performance with considerably reduced computational complexity, compared with that of existing primal SDP relaxation-based detectors, have been presented.

References

[1] S. Verdú, "Minimum probability of error for asynchronous Gaussian multiple-access channels," *IEEE Trans. Inform. Theory*, vol. IT-32, pp. 85–96, Jan. 1986.

[2] ——, *Multiuser Detection*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[3] T. Kailath and H. V. Poor, "Detection of stochastic processes," *IEEE Trans. Inform. Theory*, vol. 44, pp. 2230–2259, Oct. 1998.

[4] X. M. Wang, W.-S. Lu, and A. Antoniou, "A near-optimal multiuser detector for CDMA channels using semidefinite programming relaxation," in *Proc. ISCAS*, vol. 4, May 2001, pp. 298–301.

[5] W.-K. Ma, T. N. Davidson, K. M. Wong, Z. Q. Luo, and P. C. Ching, "Efficient quasimaximum-likelihood multiuser detection by semi-definite relaxation," in *Proc. ICC*, vol. 1, June 2001, pp. 6–10.

[6] H. T. Peng and L. K. Rasmussen, "The application of semidefinite programming for detection in CDMA," *IEEE J. Select. Areas Commun.*, vol. 19, pp. 1442–1449, Aug. 2001.

[7] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, pp. 49–95, 1996.

[8] K. C. Toh, M. J. Todd, and R. H. Tütüncü, SDPT3 Version 2.1—A MATLAB Software for Semidefinite Programming, Sept. 1999.

[9] P. Gahinet, A. Nemirovski, A. J. Laub, and M. Chilali, *Manual of LMI Control Toolbox*. Natick, MA: MathWorks Inc., May 1990.

[10] H. Wolkowicz, R. Saigal, and L. Vandenberghe, Eds., *Handbook on Semidefinite Programming*. Hingham, MA: Kluwer, 2000.

[11] M. X. Geomans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problem using semidefinite programming," *J. ACM*, vol. 42, pp. 1115–1145, 1995.

[12] ——, ".878-approximation algorithms for MAX-CUT and MAX-2SAT," in *Proc. 26th ACM Symp. Theory Comput.*, 1994, pp. 422–431.

[13] R. A. Horn and C. R. Johnson, *Matrix Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1990.

[14] C. Helmberg, *Semidefinite Programming for Combinatorial Optimization*, Berlin, Germany: Konrad-Zuse-Zentrum, Oct. 2000.

[15] A. Nemirovskii and P. Gahinet, "The projective method for solving linear matrix inequalities," *Math. Programming Series B*, vol. 77, pp. 163–190, 1997.

[16] R. Lupas and S. Verdú, "Linear multiuser detectors for synchronous code-division multiple-access channels," *IEEE Trans. Inform. Theory*, vol. 35, pp. 123–136, Jan. 1989.

[17] H. V. Poor and S. Verdú, "Probability of error in MMSE multiuser detection," *IEEE Trans. Inform. Theory*, vol. 43, pp. 858–871, May 1997.

OFDM Transmitters: Analog Representation and DFT-Based Implementation

Yuan-Pei Lin and See-May Phoong

***Abstract*—The implementation of OFDM transmitters typically consists of a discrete DFT matrix and a digital-to-analog (DAC) converter. Many existing results on the analysis of OFDM systems, e.g., spectral roll-off, are based on a convenient analog representation. In this paper, we show that the analog representation and the DFT-based OFDM transmitters are equivalent only in special cases. Using the analog system to analyze the DFT-based OFDM system may not be valid if there is no equivalent analog representation.**

***Index Terms*—Analog representation, DFT-based implementation, OFDM, pulse shaping, window.**

I. Introduction

The orthogonal frequency division multiplexing (OFDM) systems [1]–[3] are well known for applications in wireless local area networks (LANs) and broadcast of digital audio and digital video. Fig. 1 shows the schematic of an analog OFDM transmitter with M subcarriers. Let-

Manuscript received November 10, 2002; revised February 20, 2003. The work was supported in parts by the NSC under Grants 90-2213-E-009-108 and 90-2213-E-002-097, by the Ministry of Education, under Contract 89-E-FA06-2-4, Taiwan, R.O.C., and the Lee and MTI Center for Networking Research. The associate editor coordinating the review of this paper and approving it for publication was Dr. Naofal M. W. Al-Dhahir.

Y.-P. Lin is with the Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C. (e-mail: ypl@cc.nctu.edu.tw).

S.-M. Phoong is with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, R.O.C.

Digital Object Identifier 10.1109/TSP.2003.815392