Universal Binary Semidefinite Relaxation for ML Signal Detection

Xiaopeng Fan, Junxiao Song, Daniel P. Palomar, *Fellow, IEEE*, and Oscar C. Au, *Fellow, IEEE*

***Abstract*—Semidefinite relaxation (SDR) provides a computationally efficient polynomial-time approximation of the maximum likelihood detector. However, most of the existing works mainly focus on particular signal constellations. In this paper, we propose a universal binary semidefinite relaxation scheme that can handle arbitrary signal constellations in polynomial time. The proposed scheme first binarizes the original signal space to a linearly constrained binary space, and then solves the detection problem through SDR. A specialized dual barrier method is provided to solve the SDR more efficiently. In addition, we propose to apply on-the-fly decision feedback to further reduce the computational complexity and improve the detection performance. The proposed binary SDR, together with on-the-fly decision feedback scheme, can provide comparable or better solutions compared to existing SDR methods specialized to specific constellations such as 16-QAM and 8-PSK in terms of computational complexity and symbol error rate. Furthermore, the proposed scheme is universal and can solve any other constellations such as 12-QAM, 32-QAM, or M-PSK.**

***Index Terms*—Convex optimization, semidefinite relaxation, ML detection, decision feedback.**

I. Introduction

MAXIMUM likelihood (ML) detection in communication systems is optimal in the sense of minimum error probability under the assumption of i.i.d. data symbols. Unfortunately, its computational complexity increases exponentially with the dimension of the signal (and the number of users in multiuser systems). Thus there has been much interest in implementing suboptimal detection algorithms [1]. The most common suboptimal detectors are the linear receivers, i.e., the matched filter (MF), the decorrelator or zero forcing (ZF) detector, and the minimum mean-squared error (MMSE) detector. There are also many other methods such as lattice-based algorithms, alternating variable methods, expectation maximization, and decision feedback equalization. Sphere decoding algorithms [2]–[4] (and references therein) can approximate the ML detection problem in polynomial time [5], although the exact solution still comes at an exponential cost [6].

Recently, semidefinite relaxation (SDR) has been used to approach the optimal ML detector in polynomial time. In [7], the use of SDR was proposed to solve a quadratic (0,1) problem with the cutting plane method to avoid infeasible solutions. This inspired the application of SDR in communication systems. In [8] and [9], SDR was applied for the first time in communications, in particular, for multiuser CDMA detector for BPSK and QPSK to achieve near optimal BER in polynomial time. Besides binary signals, the detection problem of other non-binary signals, such as M-PSK and QAM, can also be solved by SDR after some reformulations and relaxations, as proposed in [10]–[13]. Another kind of SDR detector was proposed in [14] based on the convex dual problem, and achieves equal BER with lower complexity. Aiming at reducing the complexity, a sufficient and necessary condition for optimality of the solution obtained by SDR was given in [15]; this was then applied in [16] to reduce the complexity of ML detector. In [17], a bound-constrained SDR method was proposed for high order constellations. An SDR detector was proposed for arbitrary signal constellations in [18], in which the original signal space is converted into binary space and the relaxed problem is tightened by the so-called Gangster Operator. Recently, an efficient PSK detector was proposed for M-PSK constellations based on a nonconvex low-rank SDR [19]. Surprisingly, in [20] the authors proved the equivalence of seemingly different relaxation methods proposed in [11], [12] and [17] when they are applied on any 4^q-QAM constellation. Semidefinite programs are usually solved via general purpose solvers, e.g., SDPT3 [21], SeDuMi [22] and DSDP [23], based on interior-point methods [24]. Recently, an alternative approach to solve SDPs row by row has been considered in [25], [26].

In this paper, we propose a universal binary SDR detector that can closely approximate the ML detection problem of arbitrary signal constellations in polynomial time (in the signal dimension). The proposed scheme consists of binarization and SDR detector. First the original signal is converted into a linearly constrained binary signal. Then an SDR detector is used with the binary representation based on a simple dual-barrier method combined with on-the-fly decision feedback.

This work has the following contributions: 1) we propose a universal SDR method in the sense that it accommodates any constellation as opposed to existing methods tailored to specific constellations with a performance that is comparable or better than existing methods whenever they are available; 2) we provide a specialized dual barrier method to solve the semidefinite program more efficiently in practice compared with commonly used general purpose solvers; 3) based on the

Manuscript received December 27, 2012; revised June 2 and August 30, 2013. The editor coordinating the review of this paper and approving it for publication was S. Gezici.

This work was supported in part by the National Science Foundation of China (NSFC) under grants 61100095, the Major State Basic Research Development Program of China (973 Program 2009CB320905), the Hong Kong RGC 617810 research grant, and the Program for New Century Excellent Talents in University (NCET) of China (NCET-11-0797).

X. Fan is with the Harbin Institute of Technology (HIT), Harbin, China (e-mail: fxp@hit.edu.cn).

J. Song, D. P. Palomar, and O. C. Au are with the Hong Kong University of Science and Technology (HKUST), Hong Kong (e-mail: {jsong, palomar, eeau}@ust.hk).

Digital Object Identifier 10.1109/TCOMM.2013.092013.120988

dual barrier method, we propose to apply on-the-fly decision feedback scheme which can further reduce the computational complexity and improve the symbol error rate performance; 4) we provide a convenient mapping table to use the binarization method with all typical constellations such as BPSK, QPSK, QAM, and M-PSK.

The rest of this paper is organized as follows: Section II contains the statement of the problem addressed. Section III briefly reviews the existing SDR methods for solving ML signal detection for different constellations. Section IV presents the proposed universal binary SDR method with some specific instances given in Section V. Section VI provides a dual barrier method to solve the SDR efficiently. Section VII provides the experimental results and Section VIII concludes the paper.

Notation: Boldface upper case letters denote matrices, boldface lower case letters denote column vectors, and italics denote scalars. $\mathsf{Re}(\cdot)$ and $\mathsf{Im}(\cdot)$ denote the real and imaginary part respectively. The superscripts $(\cdot)^T$, $(\cdot)^*$ and $(\cdot)^H$ denote transpose, complex conjugate, and Hermitian operations, respectively. $X_{i,j}$ denotes the (*i*-th, *j*-th) element of matrix $\mathbf{X}$ and x_i denotes the *i*-th element of vector $\mathbf{x}$. $\mathbf{X}_{i,:}$ denotes the *i*-th row of matrix $\mathbf{X}$, $\mathbf{X}_{:,j}$ denotes the *j*-th column of matrix $\mathbf{X}$, and $\mathbf{X}_{i:j,k:l}$ denotes the submatrix of $\mathbf{X}$ from $X_{i,k}$ to $X_{j,l}$. $\mathcal{C}$ is complex field. $\mathsf{Tr}(\cdot)$ and $\mathsf{rank}(\cdot)$ denote the trace and rank of a matrix, respectively. $\mathsf{diag}(\mathbf{X})$ is a column vector consisting of all the diagonal elements of $\mathbf{X}$. $\mathsf{vec}(\mathbf{X})$ is a column vector consisting of all the columns of $\mathbf{X}$ stacked. The operator $\mathsf{unvec}(\cdot)$ is the reverse of $\mathsf{vec}(\cdot)$. $\otimes$ and $\odot$ denote Kronecker product and Hadamard product, respectively.

II. Problem formulation

Let $\mathcal{A}$ be the signal alphabet from which n different values $s_1, ..., s_n$ are drawn. The detection problem considered in this paper is to obtain the ML estimation of the signal vector $\mathbf{s} \in \mathcal{A}^n$ (where $\mathbf{s} = [s_1, ..., s_n]^T$) given the linear observation model

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{w} \tag{1}$$

where $\mathbf{y}$ is the m-dimensional received vector signal, $\mathbf{H}$ is the $m \times n$ known channel matrix, $\mathbf{s}$ is the n-dimensional vector of transmitted symbols, and $\mathbf{w}$ is the m-dimensional complex Gaussian noise vector $\mathbf{w} \sim \mathcal{CN}\left(\mathbf{0}, \sigma^2\mathbf{I}\right)$.

The ML detection problem can be formulated as

$$\underset{\mathbf{s}\in\mathcal{A}^n}{\mathsf{minimize}} \ \|\mathbf{y} - \mathbf{H}\mathbf{s}\| \tag{2a}$$

where $\|\cdot\|$ denotes the Euclidean norm or, equivalently,

$$\underset{\mathbf{s}\in\mathcal{A}^n}{\mathsf{minimize}} \quad \mathbf{s}^H\mathbf{H}^H\mathbf{H}\mathbf{s} - 2\mathsf{Re}(\mathbf{y}^H\mathbf{H}\mathbf{s}) + \mathbf{y}^H\mathbf{y}. \tag{2b}$$

This is a combinatorial problem that can be solved by brute force searching over all of the $|\mathcal{A}|^n$ possibilities. This has an exponential complexity in n, which makes it impractical for large values of n. In this paper, we propose a universal solution to this problem (in the sense that it is applicable to any signal alphabet $\mathcal{A}$) via semidefinite relaxation.

III. A review of the SDP relaxation approach

A. Basic SDP relaxation for a quadratic-(0,1) problem

When $\mathcal{A} = \{-1, 1\}$, the problem (2) can be reformulated as (cf. [8], [9])

$$\begin{array}{ll} \underset{\mathbf{x}}{\mathsf{minimize}} & \mathbf{x}^T\mathbf{L}\mathbf{x} \\ \mathsf{subject\ to} & \mathbf{x}\in\{-1,1\}^{n+1} \\ & x_{n+1} = 1. \end{array} \tag{3}$$

where

$$\mathbf{L} = \begin{bmatrix} \mathbf{H}^H\mathbf{H} & -\mathbf{H}^H\mathbf{y} \\ -\mathbf{y}^H\mathbf{H} & \mathbf{y}^H\mathbf{y} \end{bmatrix} \quad \text{and} \quad \mathbf{x} = \begin{bmatrix} \mathbf{s} \\ 1 \end{bmatrix}. \tag{4}$$

Letting $\mathbf{X} = \mathbf{x}\mathbf{x}^T$, problem (3) is equivalent to

$$\begin{array}{ll} \underset{\mathbf{X},\mathbf{x}}{\mathsf{minimize}} & \mathsf{Tr}(\mathbf{L}\mathbf{X}) \\ \mathsf{subject\ to} & \mathbf{X} = \mathbf{x}\mathbf{x}^T \\ & \mathsf{diag}(\mathbf{X}) = \mathbf{1}_{n+1}. \end{array} \tag{5}$$

The constraint $\mathbf{X} = \mathbf{x}\mathbf{x}^T$ is equivalent to $\mathbf{X} \succeq \mathbf{0}$ and $\mathsf{rank}(\mathbf{X}) = 1$. The key idea in semidefinite programming (SDP) relaxation methods is to relax

$$\mathbf{X} = \mathbf{x}\mathbf{x}^T \tag{6}$$

to

$$\mathbf{X} \succeq \mathbf{0}. \tag{7}$$

After this relaxation, the problem becomes an SDP which can be efficiently solved by interior point methods in polynomial time.

B. Candidate solutions from the SDP relaxation

An optimal solution of the original problem is of the form

$$\mathbf{x} = [\mathbf{s}^T 1]^T. \tag{8}$$

However, after solving the relaxed problem only a relaxed solution $\hat{\mathbf{X}}$ is obtained.

If $\hat{\mathbf{X}}$ is rank one, then from (6) one can readily obtain the solution to the original problem. Otherwise, it only provides a lower bound on the objective rather than the solution to the original problem. There are different heuristic techniques to obtain a candidate solution to the original problem from $\hat{\mathbf{X}}$, namely:

1) Simple quantization: This is the simplest heuristic and it is based on the fact that, when $\hat{\mathbf{X}}$ is rank one, the last column of $\hat{\mathbf{X}}$ is $\mathbf{x}$. The method simply normalizes $\hat{\mathbf{X}}_{:,n+1}$, the last column of $\hat{\mathbf{X}}$, by its last element $\hat{X}_{n+1,n+1}$:

$$\hat{\mathbf{x}} = \frac{\hat{\mathbf{X}}_{:,n+1}}{\hat{X}_{n+1,n+1}}. \tag{9}$$

2) Eigenvalue decomposition: This method gives a better performance at the expense of an increased complexity. Suppose λ is the largest eigenvalue of $\hat{\mathbf{X}}$ and $\mathbf{u}$ is the corresponding eigenvector. Like before, the method is based on the fact that, when $\hat{\mathbf{X}}$ is rank one, then $\mathbf{u}$ is a scaled version of $\mathbf{x}$. Thus,

$$\hat{\mathbf{x}} = \frac{\mathbf{u}}{u_{n+1}}. \tag{10}$$

3) Randomization: This method provides the best performance among the three methods with the highest complexity. The idea is to generate N random points $\hat{\mathbf{x}}^{(1)}, ..., \hat{\mathbf{x}}^{(N)}$ with zero mean and covariance matrix given by $\hat{\mathbf{X}}$ and then keep the best one. A convenient way to implement this is to first find the Cholesky factorization $\hat{\mathbf{X}} = \mathbf{V}\mathbf{V}^T$ and then multiply $\mathbf{V}$ by a (usually Gaussian) random vector $\mathbf{r}^{(i)}$ with i.i.d. entries:

$$\hat{\mathbf{x}}^{(i)} = \frac{\mathbf{V}\mathbf{r}^{(i)}}{\mathbf{V}_{n+1,:}\mathbf{r}^{(i)}} \qquad i = 1, ..., N. \tag{11}$$

In any of the three methods (9)-(11), it is clear that $\hat{\mathbf{x}}$ conforms to the structure in (8). Thus one can obtain $\hat{\mathbf{s}}$ as

$$\hat{\mathbf{s}} = \mathsf{quantize}(\hat{\mathbf{x}}_{1:n}), \tag{12}$$

where $\mathsf{quantize}(\cdot)$ denotes an element-wise quantization to the closest element in the constellation $\mathcal{A}$.

C. Particular solution for different constellations

BPSK constellations can be directly solved by SDP relaxation as previously seen. The detection problem of other constellations need some reformulation. Each different constellation requires a different relaxation method such as QPSK [8], [9], M-PSK [10], 16-QAM [11]–[13], [17], [20]. The reader is referred to [27] and [28] for overview and analysis of SDP relaxation methods.

IV. Proposed universal binary SDP relaxation

We now propose a universal binary SDP relaxation method to deal with arbitrary constellations based on a binarization of the original problem.

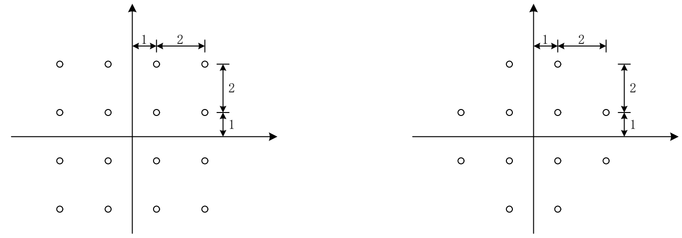
A. Binarization of an arbitrary constellation

The starting point of our universal SDR is the existence of a binary space covering the original non-binary signal space. For any signal space $\mathcal{A} \subseteq \mathcal{C}$ of size $|\mathcal{A}| = M$ there always exists a covering binary space $\mathcal{B}$ defined by a vector $\boldsymbol{\alpha} \in \mathcal{C}^q$ (with q satisfying $\log_2(M) \leq q \leq M + 1$), such that

$$\mathcal{A} \subseteq \mathcal{B} = \{s | s = \boldsymbol{\alpha}^H \mathbf{b}, \quad \mathbf{b} \in \{\pm 1\}^q\}. \tag{13}$$

For example, if $\mathcal{A} = \{s_1, s_2, ..., s_M\}$, we can construct a covering binary space $\mathcal{B}$ by setting $\boldsymbol{\alpha} = [\frac{1}{2}s_1, \frac{1}{2}s_2, \cdots, \frac{1}{2}s_M, \frac{1}{2}\sum_i s_i]^H$, which is not very desirable as it achieves the most expensive binary expansion with $q = M + 1$. (This is equivalent to the binary mapping used in [18].) Interestingly, the binary mapping (i.e., the choice of $\mathcal{B}$ or equivalently $\boldsymbol{\alpha}$) is not unique for a given constellation $\mathcal{A}$, and provides degrees of freedom. In the above example, the dimension of $\boldsymbol{\alpha}$ is $q = M + 1$ (equivalently, $q = M$ in [18]). However, for most constellations we can have tighter binary mappings with smaller q, which translates into less variables and lower complexity. A typical example is the 16-QAM constellation shown in Fig. 1(a). For this constellation, we can construct $\mathcal{B}$ by using $\boldsymbol{\alpha} = [2, 2j, 1, j]^H$ and thus $q = \log_2 M = 4$. This compares favorably with the binary mapping in [18] where the dimension is $q = 16$.

In the case of the 16-QAM constellation shown in Fig. 1(a), it just so happens that $\mathcal{B} = \mathcal{A}$. In other cases, however, like



(a) 16-QAM (b) 12-QAM

Fig. 1. Two examples of QAM constellation.

the constellation 12-QAM shown in Fig. 1(b), $\mathcal{B}$ will contain more points than $\mathcal{A}$, i.e., $\mathcal{B} \supset \mathcal{A}$. In that case, we need to add some constraints to exclude all the undesired points. Note that a naive approach is to use $\mathcal{B}$ in the detection process hoping that the invalid points will never be selected by the detection process. In [18], a quadratic constraint is derived to exclude points for the case that $M = q$. In this paper, we do not restrict to the mapping in [18], but consider arbitrary binary mapping with $\log_2(M) \leq q \leq M + 1$. In the following lemma, we derive linear constraints to remove all the undesired points.

Lemma 1: For any constellation $\mathcal{A} \subseteq \mathcal{C}$ of size $|\mathcal{A}| = M$, there always exists a vector $\boldsymbol{\alpha} \in \mathcal{C}^q$ (with q satisfying $\log_2(M) \leq q \leq M + 1$) and a matrix $\mathbf{D}$ such that [1]

$$\mathcal{A} = \{s | s = \boldsymbol{\alpha}^H \mathbf{b}, \ \mathbf{b}^T \mathbf{D} \leq (q - 2)\mathbf{1}^T, \ \mathbf{b} \in \{\pm 1\}^q\}. \tag{14}$$

Proof: First, note that there always exists $\boldsymbol{\alpha} \in C^q$ and $\mathcal{B}$ satisfying (13), i.e., so that $\mathcal{B}$ is a covering of $\mathcal{A}$.

By defining $\mathcal{D} = \{\mathbf{d} \in \{\pm 1\}^q | s = \boldsymbol{\alpha}^H \mathbf{d}, \quad s \notin \mathcal{A}\}$ as the set of binary points that represent the difference $\mathcal{B} - \mathcal{A}$, we have that, for all $\mathbf{b} \in \{\pm 1\}^q$,

$$\boldsymbol{\alpha}^H \mathbf{b} \in \mathcal{A} \quad \Leftrightarrow \quad \mathbf{b} \notin \mathcal{D} \tag{15}$$

and

$$\mathbf{b} \notin \mathcal{D} \quad \Leftrightarrow \quad \mathbf{b} \neq \mathbf{d}, \, \forall \mathbf{d} \in \mathcal{D} \quad \Leftrightarrow \quad \mathbf{b}^T \mathbf{d} \leq q - 2, \, \forall \mathbf{d} \in \mathcal{D}, \tag{16}$$

where the upper bound $q - 2$ comes from the fact that $\mathbf{b}$ and $\mathbf{d}$ are different in at least one position.

Suppose $\mathcal{D} = \{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_l\}$. By defining the matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_l]$, we have:

$$\mathbf{b}^T \mathbf{d} \leq q - 2, \, \forall \mathbf{d} \in \mathcal{D} \quad \Leftrightarrow \quad \mathbf{b}^T \mathbf{D} \leq (q - 2)\mathbf{1}^T \tag{17}$$

To conclude, (13) together with (15)-(17) leads to (14). ■

The practical implication of Lemma 1 is that an arbitrary constellation $\mathcal{A}$ can always be expressed by all q-dimensional binary points $\mathbf{b}$, $s = \boldsymbol{\alpha}^H \mathbf{b}$, that satisfy some linear inequalities constraints $\mathbf{b}^T \mathbf{D} \leq (q - 2)\mathbf{1}^T$. This has profound practical implications when using the SDP relaxation.

One possible choice for $\mathbf{D}$ in (14) is precisely the $\mathbf{D}$ used in the proof of the lemma, i.e., $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_l]$ where $\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_l$ are the binary points $\mathbf{b}$ that generate the difference $\mathcal{B} - \mathcal{A}$. But there may be more compact representations (with $l < 2^q - M$).

[1] Note that the term $q - 2$ in (14) can be replaced by any $\tilde{q} \in [q - 2, q)$. Nevertheless, the tightest value $\tilde{q} = q - 2$ is expected to provide the best performance and it is indeed the case in our numerical experiments.

It is important to remark that the additional constraints in Lemma 1 required to exclude the undesired points are linear. This holds even if the points are in the interior of the constellation. The reason is that after the binarization, all points become corners in a hypercube which can always be excluded with a hyperplane. In addition, note that the binarization we proposed here is not necessarily the bit-to-symbol mapping that is used in the modulation process. For example, to detect the bits in a vector of Gray-coded QAM symbols, we may use a linear binarization as stated in Lemma 1 to detect the vector of QAM symbols, and then use the inverse Gray mapping on each element of the detected vector to extract the information bits.

B. Relaxation approach

This section considers the resolution of the ML detection problem in (2) by SDP relaxation using the binary representation from Lemma 1.

The following result shows how to rewrite the original problem as a binary one.

Theorem 1: For any constellation $\mathcal{A} \subseteq \mathcal{C}$ of size $M = |\mathcal{A}|$, there always exists a vector $\boldsymbol{\alpha} \in \mathcal{C}^q$ (with q satisfying $\log_2(M) \leq q \leq M+1$) and a matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_l] \in \{\pm 1\}^{q \times l}$ (with $l \leq 2^q - M$) such that the *n*-dimensional ML detection problem in (2) is equivalent to the following linearly constrained *qn*-dimensional binary ML detection problem:

$$\begin{array}{ll} \underset{\tilde{\mathbf{b}}}{\text{minimize}} & \|\mathbf{y} - \tilde{\mathbf{H}}\tilde{\mathbf{b}}\| \\ \text{subject to} & \tilde{\mathbf{b}}^T \tilde{\mathbf{D}} \leq (q-2)\mathbf{1}^T \\ & \tilde{\mathbf{b}} \in \{\pm 1\}^{qn} \end{array} \tag{18}$$

where $\tilde{\mathbf{H}} = \mathbf{H}\tilde{\boldsymbol{\alpha}}^H$, $\tilde{\boldsymbol{\alpha}}^H = \boldsymbol{\alpha}^H \otimes \mathbf{I}_{n\times n}$, $\tilde{\mathbf{D}} = \mathbf{D} \otimes \mathbf{I}_{n \times n}$, and $\tilde{\mathbf{b}} = [\mathbf{b}_1^T, ..., \mathbf{b}_q^T]^T$ (with the n-dimensional vector $\mathbf{b}_q$ containing the q-th bit of the n transmitted symbols).

Proof: According to Lemma 1, with the representation in (14), the signal space $\mathcal{A}^n$ (Cartesian product of $\mathcal{A}$) can be expressed as:

$$\mathcal{A}^n = \{\mathbf{s} | \mathbf{s} = \tilde{\boldsymbol{\alpha}}^H \tilde{\mathbf{b}},\ \tilde{\mathbf{b}}^T \tilde{\mathbf{D}} \leq (q-2)\mathbf{1}^T,\ \tilde{\mathbf{b}} \in \{\pm 1\}^{qn}\} \tag{19}$$

where $\boldsymbol{\alpha} \in \mathcal{C}^q$, $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_l] \in \{\pm 1\}^{q\times l}$, $\log_2 M \leq q \leq M+1$ and $l \leq 2^q - M$. Thus, (1) becomes

$$\mathbf{y} = \mathbf{H}\tilde{\boldsymbol{\alpha}}^H \tilde{\mathbf{b}} + \mathbf{w} \tag{20}$$

and, equivalently,

$$\mathbf{y} = \tilde{\mathbf{H}}\tilde{\mathbf{b}} + \mathbf{w} \tag{21}$$

where

$$\tilde{\mathbf{H}} = \mathbf{H}\tilde{\boldsymbol{\alpha}}^H = \mathbf{H}(\boldsymbol{\alpha}^H \otimes \mathbf{I}_{n\times n}). \tag{22}$$

In (21), $\tilde{\mathbf{b}}$ can be considered as a binary signal and $\tilde{\mathbf{H}}$ is the corresponding virtual channel matrix. The ML detection problem in (2) is then equivalent to (18) which is a linearly constrained ML detection problem with a binary constellation. ∎

Once we get a solution $\tilde{\mathbf{b}}$ to the problem (18), we can reconstruct the original signal by

$$\mathbf{s} = \tilde{\boldsymbol{\alpha}}^H \tilde{\mathbf{b}}. \tag{23}$$

The problem in (18) can be further reformulated as a rank-constrained SDP as follows.

Corollary 1: For any constellation $\mathcal{A} \subseteq \mathcal{C}$ of size $M = |\mathcal{A}|$, there always exists a vector $\boldsymbol{\alpha} \in \mathcal{C}^q$ (with q satisfying $\log_2(M) \leq q \leq M+1$), and a matrix $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_l] \in \{\pm 1\}^{q\times l}$ (with $l \leq 2^q - M$) such that the *n*-order ML detection problem in (2) is equivalent to the following rank-constrained SDP:

$$\begin{array}{ll} \underset{\mathbf{x},\mathbf{X}}{\text{minimize}} & \text{Tr}(\mathbf{XL}) \\ \text{subject to} & \mathbf{X} = \mathbf{x}\mathbf{x}^T \\ & \mathbf{X}_{qn+1,1:qn}\tilde{\mathbf{D}} \leq (q-2)\mathbf{1}^T \\ & \text{diag}(\mathbf{X}) = \mathbf{1} \end{array} \tag{24}$$

where $\mathbf{X}$ is a real $(qn+1) \times (qn+1)$ matrix variable, $\mathbf{x} = [\tilde{\mathbf{b}}^T 1]^T$, $\{\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_l\} = \{\mathbf{d} \in \{\pm 1\}^q | \boldsymbol{\alpha}^H \mathbf{d} \notin \mathcal{A}\}$,

$$\mathbf{L} = \begin{bmatrix} \tilde{\mathbf{H}}^H \tilde{\mathbf{H}} & -\tilde{\mathbf{H}}^H \mathbf{y} \\ -\mathbf{y}^H \tilde{\mathbf{H}} & \mathbf{y}^H \mathbf{y} \end{bmatrix}, \tag{25}$$

$\tilde{\mathbf{H}} = \mathbf{H}\tilde{\boldsymbol{\alpha}}^H$, $\tilde{\boldsymbol{\alpha}}^H = \boldsymbol{\alpha}^H \otimes \mathbf{I}_{n\times n}$, $\tilde{\mathbf{D}} = \mathbf{D} \otimes \mathbf{I}_{n\times n}$, and $\tilde{\mathbf{b}} = [\mathbf{b}_1^T, ..., \mathbf{b}_q^T]^T$ (with the n-dimensional vector b_q containing the q-th bit of the n transmissions).

Proof: First, according to Theorem 1, the original *n*-dimensional ML detection problem in (2) is equivalent to the *qn*-dimensional binary ML detection problem in (18). Then, we introduce two new variables $\mathbf{x} = [\tilde{\mathbf{b}}^T\ 1]^T$ and $\mathbf{X} = \mathbf{x}\mathbf{x}^T$. This means that

$$\mathbf{X} = \begin{bmatrix} \tilde{\mathbf{b}}\tilde{\mathbf{b}}^T & \tilde{\mathbf{b}} \\ \tilde{\mathbf{b}}^T & 1 \end{bmatrix} \tag{26}$$

and $\tilde{\mathbf{b}} = \mathbf{X}_{1:qn,qn+1}$.

With the above conversions, $\tilde{\mathbf{b}} \in \{\pm 1\}^{qn}$ is equivalent to $\text{diag}(\mathbf{X}) = \mathbf{1}$, and the problem in (18) is equivalent to (24). ∎

The problem (24) is not convex due to the rank-one constraint $\mathbf{X} = \mathbf{x}\mathbf{x}^T$. However, through the rank relaxation introduced in Section III-A, we get

$$\begin{array}{ll} \underset{\mathbf{X}}{\text{minimize}} & \text{Tr}(\mathbf{XL}) \\ \text{subject to} & \mathbf{X} = \mathbf{X}^T \succeq 0 \\ & \mathbf{X}_{qn+1,1:qn}\tilde{\mathbf{D}} \leq (q-2)\mathbf{1}^T \\ & \text{diag}(\mathbf{X}) = \mathbf{1}, \end{array} \tag{27}$$

where the optimization variable $\mathbf{X}$ is a real-valued symmetric matrix. This is an SDP that can be efficiently solved in polynomial time [24].

In practice, $\mathbf{L}$ can be complex-valued, but $\mathbf{X}$ and $\mathbf{D}$ are real-valued by definition. Since $\text{Tr}(\mathbf{XL})$ will always be real, we can equivalently write $\text{Tr}(\mathbf{XL}) = \text{Tr}(\mathbf{X}\text{Re}(\mathbf{L}))$. In other words, we can just take the real part of $\mathbf{L}$ and use that instead of $\mathbf{L}$.

After solving (27), if $\mathbf{X}$ is rank-one, then we can reconstruct the original signal by

$$\mathbf{s} = \text{quantize}(\tilde{\boldsymbol{\alpha}}^H \mathbf{X}_{1:qn,qn+1}). \tag{28}$$

If $\mathbf{X}$ is not rank-one, we can, for example, employ a symbol-based randomization to get a candidate solution. The details of the symbol-based randomization are given next.

C. Generation of candidate solutions via symbol-based randomization

We now propose a symbol based randomization as follows. First, we do a conversion from the binary signal space to the symbol space:

$$\hat{\mathbf{S}} = \hat{\boldsymbol{\alpha}}^H \mathbf{X} \hat{\boldsymbol{\alpha}} \tag{29}$$

where

$$\hat{\boldsymbol{\alpha}} = \begin{bmatrix} \tilde{\boldsymbol{\alpha}} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}. \tag{30}$$

Then we generate N candidate solutions based on $\hat{\mathbf{S}}$. Let $\hat{\mathbf{S}} = \mathbf{V}\mathbf{V}^T$ be the Cholesky factorization of $\hat{\mathbf{S}}$. We generate N Gaussian random vectors $\mathbf{r}^{(1)}, ..., \mathbf{r}^{(N)}$ of dimension $n+1$ with zero mean and identity covariance matrix, and calculate candidate solutions as

$$\mathbf{s}^{(i)} = \mathsf{quantize}\left(\frac{\mathbf{V}_{1:n,:}\mathbf{r}^{(i)}}{\mathbf{V}_{n+1,:}\mathbf{r}^{(i)}}\right) \qquad i = 1, ..., N. \tag{31}$$

Finally, we select the one with lowest objective value:

$$\mathbf{s}^{\star} = \underset{\mathbf{s}^{(i)},\, i=1,\ldots,N}{\mathsf{argmin}} \|\mathbf{y} - \mathbf{H}\mathbf{s}^{(i)}\|. \tag{32}$$

V. On solving different constellations

In this section, we will give some examples on solving different constellations. The basic steps to solve the problem are as follows:

1) Given a constellation, find $\boldsymbol{\alpha}$ (as described in the next subsections and summarized in Table I). Then construct $\mathbf{D}$, e.g., according to Lemma 1. Let $\tilde{\boldsymbol{\alpha}}^H = \boldsymbol{\alpha}^H \otimes \mathbf{I}_{n\times n}$ and $\tilde{\mathbf{D}} = \mathbf{D} \otimes \mathbf{I}_{n\times n}$.

2) Construct $\mathbf{L}$ as in (25) with the virtual channel $\tilde{\mathbf{H}}$ in (22).

3) Solve the relaxed problem (27) and reconstruct the signal by (28) or via randomization as in (29)-(32).

A. BPSK and QPSK constellations

For BPSK with signal constellation $\mathcal{A} = \{-1, 1\}$, the signal space is already a binary space, so the binarization is $\boldsymbol{\alpha} = [1]$. With this binarization, all $\mathbf{s} \in \mathcal{A}^n$ can be expressed in the form $\mathbf{s} = \tilde{\boldsymbol{\alpha}}^H \tilde{\mathbf{b}}$ where $\tilde{\mathbf{b}} \in \{\pm 1\}^n$. Since the signal space is already a binary space, the problem can be directly solved by SDR.

For QPSK with signal constellation $\mathcal{A} = \{\pm 1 \pm j\}$, an obvious binarization is $\boldsymbol{\alpha} = [1, j]^H$. In Corollary 1 we have

$$\tilde{\mathbf{H}} = [\mathbf{H}, j\mathbf{H}], \quad \mathbf{D} = \varnothing. \tag{33}$$

Then it follows that we can solve the problem by dropping the rank one constraint in Corollary 1. We can see that this is exactly the same solution as in [8].

B. M-PAM constellations

For M-PAM constellation with $\mathcal{A} = \{-2^q + 1, \cdots, -3, -1, 1, 3, \cdots, 2^q - 1\}$ and $M = 2^q$, one possible binarization is $\boldsymbol{\alpha}^H = \left[2^{q-1}, \ldots, 2^1, 2^0\right]$. With this binarization, all $\mathbf{s} \in \mathcal{A}^n$ can be expressed as $\mathbf{s} = \tilde{\boldsymbol{\alpha}}^H \tilde{\mathbf{b}}$ where $\tilde{\mathbf{b}} \in \{\pm 1\}^{qn}$.

For example, for 8-PAM, one may choose $\boldsymbol{\alpha} = [4, 2, 1]^H$ and then any n-dimensional 8-PAM signal $\mathbf{s}$ can be expressed as $\mathbf{s} = \tilde{\boldsymbol{\alpha}}^H \tilde{\mathbf{b}}$, where $\tilde{\boldsymbol{\alpha}}^H = \boldsymbol{\alpha}^H \otimes \mathbf{I}_{n\times n}$ and $\tilde{\mathbf{b}} \in \{\pm 1\}^{3n}$. And in Corollary 1 we have

$$\tilde{\mathbf{H}} = [4\mathbf{H}, 2\mathbf{H}, \mathbf{H}], \quad \mathbf{D} = \varnothing. \tag{34}$$

Then we can solve the problem by solving the relaxed problem (27).

C. M-QAM constellations

For M-QAM constellations, all the signal points are on a lattice. Thus it is very natural to map the signal space to a higher order binary space. For a rectangular M-QAM constellation with $\mathcal{A} = \{s | s = r_1 + r_2 j,\ r_1 \in \nabla_1, r_2 \in \nabla_2\}$ where $\nabla_i = \{-2^{k_i} + 1, \cdots, -3, -1, 1, 3, \cdots, 2^{k_i} - 1\}$ and $M = 2^{k_1 + k_2}$, one possible binarization is $\boldsymbol{\alpha}^H = [2^{k_1 - 1}, \cdots, 2, 1, 2^{k_2 - 1}j, \ldots, 2j, j]$. With this binarization, all $\mathbf{s} \in \mathcal{A}^n$ can be expressed in the form $\mathbf{s} = \tilde{\boldsymbol{\alpha}}^H \tilde{\mathbf{b}}$ where $\tilde{\mathbf{b}} \in \{\pm 1\}^{qn}$, $q = k_1 + k_2$.

For instance, for the 16-QAM constellation in Fig. 1(a), one binarization is $\boldsymbol{\alpha} = [2, 2j, 1, j]^H$. With this binarization, any n-dimensional signal $\mathbf{s}$ (in the Cartesian product of 16-QAM constellation) can be expressed in the form $\mathbf{s} = \tilde{\boldsymbol{\alpha}}^H \tilde{\mathbf{b}}$ where $\tilde{\boldsymbol{\alpha}}^H = \boldsymbol{\alpha}^H \otimes \mathbf{I}_{n\times n}$ and $\tilde{\mathbf{b}} \in \{\pm 1\}^{4n}$. In addition, for 16-QAM, in Corollary 1 we have

$$\tilde{\mathbf{H}} = [2\mathbf{H}, 2j\mathbf{H}, \mathbf{H}, j\mathbf{H}], \quad \mathbf{D} = \varnothing. \tag{35}$$

Then we can solve (27) and reconstruct the signal by (28) or via randomization. This solution is the same as the one in [12].

Different from these rectangular QAM constellations, for other non-rectangular M-QAM constellations (e.g., 12-QAM and 32-QAM, known as cross QAM constellations) the matrix $\mathbf{D}$ (and $\tilde{\mathbf{D}}$) is not empty and the constraint $\mathbf{X}_{qn+1,1:qn}\tilde{\mathbf{D}} \leq (q-2)\mathbf{1}^T$ becomes enabled. One possible approach to deal with these constellations is first to find the smallest rectangular QAM constellation that can cover the non-rectangular constellation and then exclude the points not in the constellation by including the corresponding binary vectors in matrix $\mathbf{D}$.

For example, for 12-QAM in Fig. 1(b), the smallest rectangular QAM constellation that can cover it is 16-QAM. Thus, we may choose $\boldsymbol{\alpha} = [2, 2j, 1, j]^H$. Then to exclude the points in 16-QAM but not in 12-QAM, we include the binary vectors that generate the difference $\mathcal{A}_{\text{16-QAM}} - \mathcal{A}_{\text{12-QAM}}$ in the columns of matrix $\mathbf{D}$, i.e.,

$$\mathbf{D} = \begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix}. \tag{36}$$

Regarding the matrix $\mathbf{D}$, instead of directly including all the points we want to exclude, there may be more compact representations for some constellations, e.g., the 32-QAM constellation in Table I where there are 32 excluded points but only 8 columns in $\mathbf{D}$.

With $\boldsymbol{\alpha}$ and matrix $\mathbf{D}$, we can then solve problem (27) and reconstruct the signal by (28) or via randomization as in (29)-(32).

TABLE I
BINARIZATION FOR DIFFERENT CONSTELLATIONS

Constellation	$\mathcal{A}$	$\boldsymbol{\alpha}$	$\mathbf{D}$
BPSK	$\mathcal{A}=\{-1,1\}$	1	$\varnothing$
4-PAM	$\mathcal{A}=\{-3,-1,1,3\}$	$[2,1]^T$	$\varnothing$
8-PAM	$\mathcal{A}=\{-7,-5,-3,-1,1,3,5,7\}$	$[4,2,1]^T$	$\varnothing$
QPSK	$\mathcal{A}=\{\pm 1 \pm j\}$	$[1,j]^H$	$\varnothing$
8-QAM	$\mathcal{A}=\{s \vert s=s_1+js_2, s_1\in\mathcal{S}_1,\ s_2\in\mathcal{S}_2\}$ $\mathcal{S}_1=\{-3,-1,1,3\},\ \mathcal{S}_2=\{-1,1\}$	$[2,1,j]^H$	$\varnothing$
16-QAM	$\mathcal{A}=\{s \vert s=s_1+js_2, s_1\in\mathcal{S},\ s_2\in\mathcal{S}\}$ $\mathcal{S}=\{-3,-1,1,3\}$	$[2,2j,1,j]^H$	$\varnothing$
64-QAM	$\mathcal{A}=\{s \vert s=s_1+js_2, s_1\in\mathcal{S},\ s_2\in\mathcal{S}\}$ $\mathcal{S}=\{-7,-5,-3,-1,1,3,5,7\}$	$[4,4j,2,2j,1,j]^H$	$\varnothing$
256-QAM	$\mathcal{A}=\{s \vert s=s_1+js_2, s_1\in\mathcal{S},\ s_2\in\mathcal{S}\}$ $\mathcal{S}=\{\pm(2i+1),\ i=0,1,...,7\}$	$[8,8j,4,4j,2,2j,1,j]^H$	$\varnothing$
12-QAM	$\mathcal{A}=\{s \vert s\in\mathcal{A}_{16-\mathrm{QAM}}, \vert s\vert^2<18\}$	$[2,2j,1,j]^H$	$\begin{bmatrix} 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \end{bmatrix}$
32-QAM	$\mathcal{A}=\{s \vert s\in\mathcal{A}_{64-\mathrm{QAM}}, \vert s\vert^2<50\}$	$[4,4j,2,2j,1,j]^H$	$\begin{bmatrix} 1 & 1 & -1 & -1 & 0 & 0 & 4 & -4 \\ 1 & -1 & 1 & -1 & 4 & -4 & 0 & 0 \\ 1 & 1 & -1 & -1 & 0 & 0 & 4 & -4 \\ 1 & -1 & 1 & -1 & 4 & -4 & 0 & 0 \\ -1 & -1 & 1 & 1 & 0 & 0 & 4 & -4 \\ -1 & 1 & -1 & 1 & 4 & -4 & 0 & 0 \end{bmatrix}$
8-PSK	$\mathcal{A}=\{s \vert s=e^{j\frac{(2i+1)\pi}{8}}, i=0,1,...,7\}$	$[a,-b,aj,bj]^H$ $(a=\frac{\sqrt{2}}{2}cos(\frac{\pi}{8})$ $b=\frac{\sqrt{2}}{2}sin(\frac{\pi}{8}))$	$\begin{bmatrix} 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \end{bmatrix}$

D. M-PSK constellations

For M-PSK constellations, all the points are on the unit circle. Suppose the constellation is $\mathcal{A}=\{s_1,s_2,\cdots,s_M\}$ where $s_k=e^{\frac{\pi}{M}(2k-1)j}$. Let

$$\begin{aligned} \bar{\mathbf{A}} &= [s_1,s_2,\cdots,s_h], \\ \bar{\mathbf{B}} &= \begin{bmatrix} +1 & +1 & +1 & \cdots & +1 \\ -1 & +1 & +1 & \cdots & +1 \\ -1 & -1 & +1 & \cdots & +1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & -1 & -1 & +1 \end{bmatrix}_{h\times h} \\ \text{and}\quad \bar{\boldsymbol{\alpha}} &= \frac{1}{2}\begin{bmatrix} s_1^*+s_h^* \\ s_2^*-s_1^* \\ s_3^*-s_2^* \\ \vdots \\ s_h^*-s_{h-1}^* \end{bmatrix} \end{aligned} \tag{37}$$

where $h=M/4$ (we assume that M is a multiple of 4). Then

$$\bar{\mathbf{A}}=\bar{\boldsymbol{\alpha}}^H\bar{\mathbf{B}}. \tag{38}$$

By defining

$$\begin{aligned} \mathbf{A} &= [\bar{\mathbf{A}},-\bar{\mathbf{A}}^*,-\bar{\mathbf{A}},\bar{\mathbf{A}}^*], \\ \mathbf{B} &= \begin{bmatrix} \bar{\mathbf{B}} & -\bar{\mathbf{B}} & -\bar{\mathbf{B}} & \bar{\mathbf{B}} \\ \bar{\mathbf{B}} & \bar{\mathbf{B}} & -\bar{\mathbf{B}} & -\bar{\mathbf{B}} \end{bmatrix} \\ \text{and}\quad \boldsymbol{\alpha} &= \begin{bmatrix} \mathrm{Re}\{\bar{\boldsymbol{\alpha}}\} \\ \mathrm{Im}\{\bar{\boldsymbol{\alpha}}\}j \end{bmatrix}, \end{aligned} \tag{39}$$

we have

$$\mathbf{A}=\boldsymbol{\alpha}^H\mathbf{B}. \tag{40}$$

Since $\mathbf{A}$ contains all the elements in $\mathcal{A}$, any $s\in\mathcal{A}$ can be expressed as $s=\boldsymbol{\alpha}^H\mathbf{b}$, where $\mathbf{b}\in\{\pm 1\}^{M/2}$. Therefore, $\boldsymbol{\alpha}$ is a binarization of the M-PSK, and the dimension of the binarization is $q=M/2$.

For example, for 8-PSK, according to (37) and (39), we have

$$\bar{\boldsymbol{\alpha}}=\frac{1}{2}\begin{bmatrix} s_1^*+s_2^* \\ s_2^*-s_1^* \end{bmatrix}=\begin{bmatrix} e^{-\frac{1}{4}\pi j}cos(\frac{\pi}{8}) \\ -e^{-\frac{1}{4}\pi j}sin(\frac{\pi}{8})j \end{bmatrix},$$

and

$$\boldsymbol{\alpha}=\begin{bmatrix} \mathrm{Re}\{\bar{\boldsymbol{\alpha}}\} \\ \mathrm{Im}\{\bar{\boldsymbol{\alpha}}\}j \end{bmatrix}=\frac{\sqrt{2}}{2}\begin{bmatrix} \cos(\frac{\pi}{8}) \\ -\sin(\frac{\pi}{8}) \\ -j\cos(\frac{\pi}{8}) \\ -j\sin(\frac{\pi}{8}) \end{bmatrix}.$$

Let $a=\frac{\sqrt{2}}{2}\cos(\frac{\pi}{8})$ and $b=\frac{\sqrt{2}}{2}\sin(\frac{\pi}{8})$. Then $\boldsymbol{\alpha}=[a,-b,aj,bj]^H$, and we can solve (27), with

$$\tilde{\mathbf{H}}=[a\mathbf{H},-b\mathbf{H},aj\mathbf{H},bj\mathbf{H}] \tag{41a}$$

$$\mathbf{D}=\begin{bmatrix} 1 & 1 & -1 & -1 & 1 & -1 & 1 & -1 \\ -1 & 1 & -1 & 1 & -1 & 1 & 1 & -1 \\ 1 & 1 & -1 & -1 & -1 & 1 & -1 & 1 \\ 1 & -1 & 1 & -1 & -1 & 1 & 1 & -1 \end{bmatrix}. \tag{41b}$$

Then we can solve problem (27) and reconstruct the signal by (28) or via randomization as in (29)-(32).

VI. EFFICIENT IMPLEMENTATION OF THE SDP RELAXATION FORMULATION VIA A DUAL BARRIER METHOD

The SDP relaxation problem in (27) can be solved by general-purpose solvers, e.g., SDPT3 [21], SeDuMi [22] and DSDP [23], based on interior-point methods. However, it is possible in this particular case to derive a simple and efficient tailored implementation that can be coded in just a few lines of Matlab and beats general-purpose solvers in speed.

A. Solving SDP by a dual barrier method

The following proposed implementation is based on solving the dual problem with a barrier method [29].

Let us start by recalling the relaxed problem (27):

$$\begin{array}{ll} \underset{\mathbf{X}}{\text{minimize}} & \mathsf{Tr}\left(\mathbf{LX}\right) \\ \text{subject to} & \mathsf{diag}\left(\mathbf{X}\right)=\mathbf{1} \\ & \mathbf{X}\succeq\mathbf{0} \\ & \tilde{\mathbf{D}}^T\mathbf{X}_{1:qn,qn+1}\leq\mathbf{h}, \end{array} \tag{42}$$

where we have defined $\mathbf{h}=(q-2)\mathbf{1}_{nl\times 1}$. Observe that one implicit constraint in an SDP is the symmetry of the optimization variable $\mathbf{X}$. Now, for convenience, let us rewrite the last linear inequality constraint in a more symmetric way:

$$\begin{array}{ll} \underset{\mathbf{X}}{\text{minimize}} & \mathsf{Tr}\left(\mathbf{LX}\right) \\ \text{subject to} & \mathsf{diag}\left(\mathbf{X}\right)=\mathbf{1} \\ & \mathbf{X}=\mathbf{X}^T\succeq\mathbf{0} \\ & \tilde{\mathbf{D}}^T\mathbf{X}_{1:qn,qn+1}+\left(\mathbf{X}_{qn+1,1:qn}\tilde{\mathbf{D}}\right)^T\leq 2\mathbf{h}. \end{array} \tag{43}$$

The dual problem is

$$\begin{array}{ll} \underset{\mathbf{v},\boldsymbol{\lambda}}{\text{maximize}} & -\mathbf{v}^T\mathbf{1}-2\boldsymbol{\lambda}^T\mathbf{h} \\ \text{subject to} & \boldsymbol{\Phi}=\mathbf{L}+\mathsf{diag}\left(\mathbf{v}\right)+\mathbf{M}\left(\tilde{\mathbf{D}}\boldsymbol{\lambda}\right)\succeq\mathbf{0} \\ & \boldsymbol{\lambda}\geq\mathbf{0} \end{array} \tag{44}$$

where we have defined the linear M-operator as

$$\mathbf{M}\left(\mathbf{a}\right)=\begin{bmatrix}\mathbf{0} & \mathbf{a}\\ \mathbf{a}^T & 0\end{bmatrix}. \tag{45}$$

Note that the symmetry constraint is automatically satisfied and does not need to be imposed in the dual formulation.

The KKT conditions corresponding to (43)-(44) are

$$\begin{array}{rcl} \boldsymbol{\Phi} &=& \mathbf{L}+\mathsf{diag}\left(\mathbf{v}\right)+\mathbf{M}\left(\tilde{\mathbf{D}}\boldsymbol{\lambda}\right)\succeq\mathbf{0} \\ \boldsymbol{\lambda} &\geq& \mathbf{0} \\ \mathbf{X} &\succeq& \mathbf{0},\quad \mathsf{diag}\left(\mathbf{X}\right)=\mathbf{1} \\ 2\mathbf{h} &\geq& \tilde{\mathbf{D}}^T\mathbf{X}_{1:qn,qn+1}+\left(\mathbf{X}_{qn+1,1:qn}\tilde{\mathbf{D}}\right)^T \\ \mathbf{0} &=& \boldsymbol{\lambda}^T\left(\tilde{\mathbf{D}}^T\mathbf{X}_{1:qn,qn+1}+\left(\mathbf{X}_{qn+1,1:qn}\tilde{\mathbf{D}}\right)^T-2\mathbf{h}\right) \\ \boldsymbol{\Phi}\mathbf{X} &=& \mathbf{0}. \end{array} \tag{46}$$

Now, the proposed algorithm is based on solving the dual problem (44) with a barrier method [29]. In particular, the algorithm will solve the following problem based on the Newton method:

$$\underset{\mathbf{v},\boldsymbol{\lambda}}{\text{minimize}} \quad f\left(\mathbf{v},\boldsymbol{\lambda}\right) \tag{47}$$

where

$$\begin{array}{rcl} f\left(\mathbf{v},\boldsymbol{\lambda}\right) &=& t\left(\mathbf{v}^T\mathbf{1}+2\boldsymbol{\lambda}^T\mathbf{h}\right)-\sum_i\log\left(\lambda_i\right) \\ && -\mathsf{logdet}\left(\mathbf{L}+\mathsf{diag}\left(\mathbf{v}\right)+\mathbf{M}\left(\tilde{\mathbf{D}}\boldsymbol{\lambda}\right)\right) \end{array}$$

where t is a parameter that controls the goodness of the solution of the barrier method and $(\mathbf{v},\boldsymbol{\lambda})$ are the dual variables. At the optimal point of problem (47), the following stationary conditions will be satisfied:[2]

$$\begin{array}{rcl} \nabla_{\mathbf{v}}f &=& t\mathbf{1}-\mathsf{diag}\left(\left(\mathbf{L}+\mathsf{diag}\left(\mathbf{v}\right)+\mathbf{M}\left(\tilde{\mathbf{D}}\boldsymbol{\lambda}\right)\right)^{-1}\right)=\mathbf{0} \\ \nabla_{\lambda_i}f &=& 2th_i-\dfrac{1}{\lambda_i} \\ && -\mathsf{Tr}\left(\left(\mathbf{L}+\mathsf{diag}\left(\mathbf{v}\right)+\mathbf{M}\left(\tilde{\mathbf{D}}\boldsymbol{\lambda}\right)\right)^{-1}\mathbf{M}\left(\tilde{\mathbf{D}}_{:,i}\right)\right). \end{array}$$

Denoting $\mathbf{v}^{\star}(t)$, $\boldsymbol{\lambda}^{\star}(t)$ the optimal solution to problem (47), $\boldsymbol{\Phi}^{\star}(t)=\mathbf{L}+\mathsf{diag}\left(\mathbf{v}^{\star}(t)\right)+\mathbf{M}\left(\tilde{\mathbf{D}}\boldsymbol{\lambda}^{\star}(t)\right)$ and defining

$$\mathbf{X}^{\star}(t)=\frac{1}{t}\left(\mathbf{L}+\mathsf{diag}\left(\mathbf{v}^{\star}(t)\right)+\mathbf{M}\left(\tilde{\mathbf{D}}\boldsymbol{\lambda}^{\star}(t)\right)\right)^{-1},$$

it is not difficult to see that the original KKT conditions (46) are satisfied (note that $\tilde{\mathbf{D}}^T\mathbf{X}^{\star}_{1:qn,qn+1}(t)+\left(\mathbf{X}^{\star}_{qn+1,1:qn}(t)\tilde{\mathbf{D}}\right)^T=2\mathbf{h}-\left(t\boldsymbol{\lambda}^{\star}(t)\right)^{-1}<2\mathbf{h}$) with the exception of the two complementarity conditions that are replaced by $\boldsymbol{\Phi}^{\star}(t)\mathbf{X}^{\star}(t)=\frac{1}{t}\mathbf{I}_{qn+1\times qn+1}$ and $\boldsymbol{\lambda}^{\star T}(t)\left(\tilde{\mathbf{D}}^T\mathbf{X}^{\star}_{1:qn,qn+1}(t)+\left(\mathbf{X}^{\star}_{qn+1,1:qn}(t)\tilde{\mathbf{D}}\right)^T-2\mathbf{h}\right)=-\frac{ln}{t}$. It then follows that the optimal value of (47) for a given t is not further away from the desired optimal value $p^{\star}$ of (43) by more than $(qn+1+ln)/t$ [29]:

$$f_0\left(\mathbf{X}^{\star}(t)\right)-p^{\star}\leq\frac{qn+1+ln}{t}.$$

At this point, to derive the algorithm we just need to find closed-form and compact expressions for the gradient and Hessian of the objective $f(\mathbf{v},\boldsymbol{\lambda})$ in (47).[3] Defining $\mathbf{Y}=\left(\mathbf{L}+\mathsf{diag}\left(\mathbf{v}\right)+\mathbf{M}\left(\tilde{\mathbf{D}}\boldsymbol{\lambda}\right)\right)^{-1}$ for compactness of notation, the gradients and Hessians are

$$\begin{array}{rcl} \nabla_{\mathbf{v}}f &=& t\mathbf{1}-\mathsf{diag}\left(\mathbf{Y}\right) \\ \nabla_{\boldsymbol{\lambda}}f &=& 2t\mathbf{h}-2\tilde{\mathbf{D}}^T\mathbf{Y}_{1:qn,qn+1}-\boldsymbol{\lambda}^{-1} \\ \mathsf{H}_{\mathbf{vv}}f &=& \mathbf{Y}\odot\mathbf{Y} \\ \mathsf{H}_{\mathbf{v}\boldsymbol{\lambda}}f &=& 2\mathsf{diag}\left(\mathbf{Y}_{:,qn+1}\right)\mathbf{Y}_{:,1:qn}\tilde{\mathbf{D}} \\ \mathsf{H}_{\boldsymbol{\lambda}\boldsymbol{\lambda}}f &=& 2Y_{qn+1,qn+1}\tilde{\mathbf{D}}^T\mathbf{Y}_{1:qn,1:qn}\tilde{\mathbf{D}}+\mathsf{diag}\left(\boldsymbol{\lambda}^{-2}\right) \\ && +2\left(\tilde{\mathbf{D}}^T\mathbf{Y}_{1:qn,qn+1}\right)\left(\mathbf{Y}_{qn+1,1:qn}\tilde{\mathbf{D}}\right). \end{array}$$

Algorithm 1 summarizes the implementation of a simple but efficient barrier method for the resolution of the dual problem (44) making use of the gradients and Hessians previously derived. In the algorithm, *backtracking* stands for backtracking line search, which is used to choose the step size and readers may refer to Algorithm 9.2 in [29] for details. For each iteration, the computation is dominated by some matrix operations, namely, the matrix multiplications in the construction of the Hessian, the logdet in the objective of (47) which is needed in backtracking and the two matrix inverse operations. Thus, the computational complexity of

[2]The gradients follows easily from the differential of the logdet function $d\mathsf{logdet}\left(\mathbf{X}\right)=\mathsf{Tr}(\mathbf{X}^{-1}d\mathbf{X})$ [30].

[3]The Hessians follows easily from the differential of the matrix inverse $d\mathbf{X}^{-1}=-\mathbf{X}^{-1}\left(d\mathbf{X}\right)\mathbf{X}^{-1}$ [30].

each iteration is of order $O(k^3 + k^2 ln)$, where $k = qn + 1$. In a practical implementation, since $\mathbf{L} + \mathrm{diag}(\mathbf{v}) + \mathbf{M}(\tilde{\mathbf{D}}\boldsymbol{\lambda})$ and the Hessian are positive definite, one can compute the logdet and matrix inverse via Cholesky decomposition, which is more efficient and numerically stable.

Note that the algorithm is quite simple and easily implementable. However, it is not making use of the specific structure of the data, which could potentially lead to further improvement in terms of efficiency of implementation and speed of execution. One example is the Kronecker structure of the channel matrix $\tilde{\mathbf{H}}$ and the matrix $\tilde{\mathbf{D}}$.

B. On-the-fly decision feedback

By solving the SDR in (27), our purpose is to find $\tilde{\mathbf{b}} \in \{\pm 1\}^{qn}$ minimizing (18). With the solution $\mathbf{X}$ to the problem (27), one simple way to reconstruct $\tilde{\mathbf{b}}$ is the simple quantization method in section III-B, i.e., $\tilde{\mathbf{b}} = \mathrm{sgn}(\mathbf{X}_{1:qn,qn+1})$, which means it is enough to find the sign of each element in $\mathbf{X}_{1:qn,qn+1}$. Besides, when solving (27), high SNR bits (in $\mathbf{X}_{1:qn,qn+1}$) may converge to correct value (-1 or +1) earlier than low SNR bits in high probability.

In light of this, we propose an on-the-fly decision feedback scheme that solves (27) once and performs decision feedback during the execution of the dual barrier method. Every time we go into the outer loop in Algorithm 1, i.e., Newton method converges for the current barrier parameter t, we check $\mathbf{X}_{1:qn,qn+1}$. Once any element in $\mathbf{X}_{1:qn,qn+1}$ is larger than a predefined threshold τ $(0 < \tau \leq 1)$ or smaller than $-\tau$, it will be set to be +1 or -1 accordingly. A similar technique was proposed in [19] for M-PSK constellations based on the principal eigenvector.

After fixing some elements of $\mathbf{X}_{1:qn,qn+1}$, the problem and the constraints need to be updated. Given the index set of the fixed bits $\mathcal{S}_{\text{fix}}$, we update the received vector signal $\mathbf{y} := \mathbf{y} - \sum_{i \in \mathcal{S}_{\text{fix}}} X_{i,qn+1}\tilde{\mathbf{H}}_{:,i}$, remove the ith column of $\tilde{\mathbf{H}}$ for $i \in \mathcal{S}_{\text{fix}}$, update matrix $\mathbf{L}$ by removing the ith row and column for $i \in \mathcal{S}_{\text{fix}}$ and recomputing the last row and column based on newly updated $\tilde{\mathbf{H}}$ and $\mathbf{y}$. Regarding matrix $\tilde{\mathbf{D}}$ and vector $\mathbf{h}$ in the inequality constraint, we first update $\mathbf{h}$ by $\mathbf{h} = \mathbf{h} - \sum_{i \in \mathcal{S}_{\text{fix}}} X_{i,qn+1}\tilde{\mathbf{D}}_{i,:}^T$ and remove the ith row of $\tilde{\mathbf{D}}$ for $i \in \mathcal{S}_{\text{fix}}$, then we further remove the jth column of $\tilde{\mathbf{D}}$ and corresponding elements of $\mathbf{h}$ for $j \in \left\{ j | \sum_i \left| \tilde{D}_{i,j} \right| \leq h_j \right\}$, which correspond to redundant constraints.

With this scheme, the dimension of the problem is reduced during the execution of the dual barrier method. Since the computational complexity per iteration is cubic with respect to the problem dimension, once some bits are fixed, the computational cost of following iterations will be reduced significantly. When all the bits have been determined, the algorithm terminates. It may also happen that the dual barrier method terminates but there are still some undetermined bits, especially when SNR is low. In this case, we determine the remaining bits via randomization.

VII. Experimental Results

To verify the detection and complexity performance of the proposed universal binary SDR (BSDR) detector, we conducted various simulations on MIMO systems. First, we compare the computational complexity of the proposed dual barrier method with some general purpose SDP solvers. Second, for the on-the-fly decision feedback scheme, the influence of the parameter τ on SER performance and computational cost is investigated. Finally, we compare the SER performance of the proposed BSDR detector with some benchmarks for different constellations.

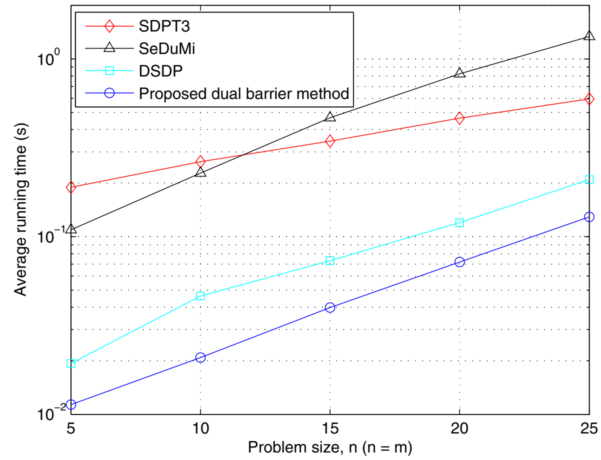


Fig. 2. Average running time, SNR=15dB, 16-QAM.

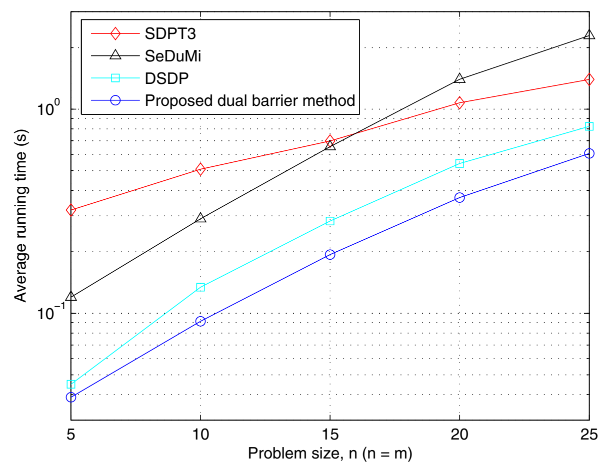


Fig. 3. Average running time, SNR=15dB, 12-QAM.

The simulation settings are as follows. The entries of the channel matrix $\mathbf{H}$ are i.i.d. and drawn from a circularly symmetric zero-mean complex normal distribution of unit variance, i.e., $\mathcal{CN}(0,1)$. The SNR is defined as SNR$\triangleq 10\log_{10}\left(nE_s/\sigma^2\right)$, where n is the length of the transmitted symbol vector $\mathbf{s}$, E_s is the mean symbol energy of the constellation and the entries of the noise vector are i.i.d. $\mathcal{CN}\left(0,\sigma^2\right)$. For each SNR, we perform up to 100,000 Monte Carlo simulations to get the average SER. When randomization approach is applied, we randomly generate 100 candidate solutions. The simulations were done on a desktop with 3.4GHz Pentium 4 CPU and 2.5GB RAM.

A. Computational Complexity

We compare the computational complexity of the proposed dual barrier method with three general purpose SDP solvers, namely SDPT3 [21], SeDuMi [22], and DSDP [23]. It is worth

Algorithm 1 Dual barrier method for the SDP relaxation in (42)

Input: $\mathbf{L} \in \mathbf{C}^{qn+1 \times qn+1}$, $\tilde{\mathbf{D}} \in \mathbf{R}^{(qn) \times ln}$, $\mathbf{h} \in \mathbf{R}^{ln}$, accuracy
Output: $\mathbf{X} \in \mathbf{R}^{qn+1 \times qn+1}$
$\mathbf{L} = \mathsf{real}(\mathbf{L})$
% Initial values
$t = 1$
$\boldsymbol{\lambda} = \mathbf{1}_{ln}$
$\mathbf{v} = (1 - \lambda_{min}\left(\mathbf{L} + \mathbf{M}(\tilde{\mathbf{D}}\boldsymbol{\lambda})\right))\mathbf{1}_{qn+1}$
$\mathbf{X} = (1/t)\left(\mathbf{L} + \mathsf{diag}(\mathbf{v}) + \mathbf{M}(\tilde{\mathbf{D}}\boldsymbol{\lambda})\right)^{-1}$
begin loop
% Gradient and Hessian (inner loop)
$\mathbf{Z} = \mathbf{X}_{:,1:qn}\tilde{\mathbf{D}}$
$$\nabla f = t\begin{bmatrix} \mathbf{1} - \mathsf{diag}(\mathbf{X}) \\ 2\mathbf{h} - 2\mathbf{Z}_{qn+1,:}^T - (t\boldsymbol{\lambda})^{-1} \end{bmatrix}$$
$$\mathsf{H}f = t^2\begin{bmatrix} \mathbf{X} \odot \mathbf{X} & 2\mathsf{diag}(\mathbf{X}_{:,qn+1})\mathbf{Z} \\ 2\mathsf{diag}(\mathbf{X}_{:,qn+1})\mathbf{Z}^T & 2X_{qn+1,qn+1}\tilde{\mathbf{D}}^T\mathbf{Z}_{1:qn,:} + 2\mathbf{Z}_{qn+1,:}^T\mathbf{Z}_{qn+1,:} + \mathsf{diag}\left((t\boldsymbol{\lambda})^{-2}\right) \end{bmatrix}$$
% Direction, step size, and update of variables
$\boldsymbol{\delta} = -(\mathsf{H}f)^{-1}\nabla f$
$s = \mathsf{backtracking}(t, \mathbf{v}, \boldsymbol{\lambda}, \mathbf{L}, \tilde{\mathbf{D}}, \mathbf{h}, \boldsymbol{\delta}, \nabla f)$
$$\begin{bmatrix} \mathbf{v} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{v} \\ \boldsymbol{\lambda} \end{bmatrix} + s\boldsymbol{\delta}$$
Newton_accuracy $= (-1/2)\nabla f^T \boldsymbol{\delta}$
if Newton_accuracy < accuracy
% Check the stopping criterion and update t (outer loop)
current_accuracy $= ((qn + 1 + ln)/t +$ Newton_accuracy$)$ / $\max\left(1, \left|\mathbf{v}^T\mathbf{1} + 2\boldsymbol{\lambda}^T\mathbf{h}\right|\right)$
if current_accuracy < accuracy, **then terminate**, **else** $t = 10t$.
end if
$\mathbf{X} = (1/t)\left(\mathbf{L} + \mathsf{diag}(\mathbf{v}) + \mathbf{M}(\tilde{\mathbf{D}}\boldsymbol{\lambda})\right)^{-1}$
end loop

noting that our dual barrier method is simply implemented in Matlab, while SDPT3 and SeDuMi have some key subroutines in Fortran and C incorporated via Matlab Mex files and DSDP is entirely written in C. For all the solvers, we terminate when the solution accuracy is smaller than 10^{-3}. We fix SNR = 15dB and choose square channel matrix, i.e., $n = m$. Figs. 2 and 3 show the average running time of different solvers to solve the SDP (43) as a function of the problem size for 16-QAM and 12-QAM, respectively. In the 16-QAM case, since the matrix $\mathbf{D}$ is $\varnothing$, the inequality constraints in (43) can be removed and the resulting SDP will be easier to solve than the SDP in 12-QAM case. From Figs. 2 and 3, we can see that it is indeed faster to solve the SDP associated with 16-QAM and for both cases (with and without inequality constraints) our dual barrier method is the fastest, since we have exploited the problem structure explicitly while deriving the algorithm.

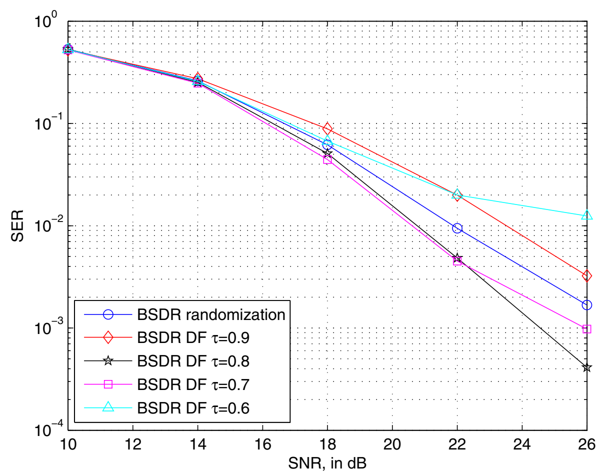


Fig. 4. SER performance, $n = m = 6$, 16-QAM.

B. On-the-fly Decision Feedback

SER performance and average running time of the on-the-fly decision feedback scheme with different threshold τ are shown in Figs. 4-7 for a MIMO system with $n = m = 6$ and two different constellations, i.e., 16-QAM and 8-PSK. From Figs. 5 and 7, we can see that the average running time decreases when either the threshold τ becomes smaller or the SNR gets higher. It is easy to understand, since with smaller τ and higher SNR, we can fix more bits at an early stage, which will reduce the size of the problem more. For BSDR with randomization, the average running time increases with respect to the SNR. This is because at higher SNR, the matrix $\mathbf{L}$ is more ill-conditioned and the optimal objective value becomes smaller, thus more iterations are needed to solve the SDP to the same accuracy. Regarding the SER performance, we can see from

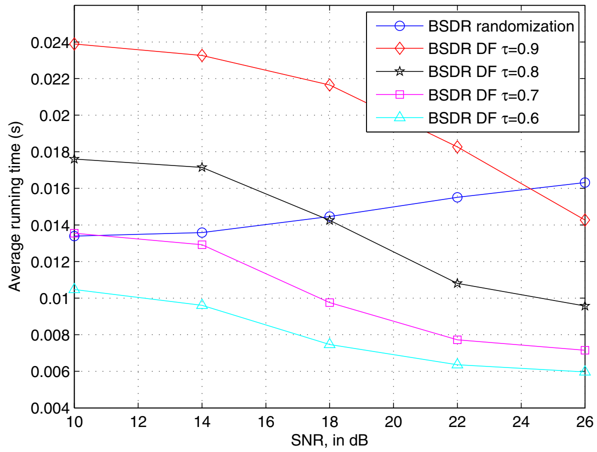


Fig. 5. Average running time, $n = m = 6$, 16-QAM.

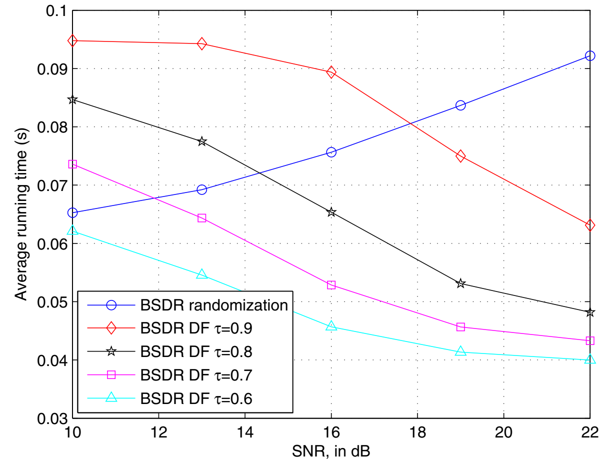


Fig. 7. Average running time, $n = m = 6$, 8-PSK.

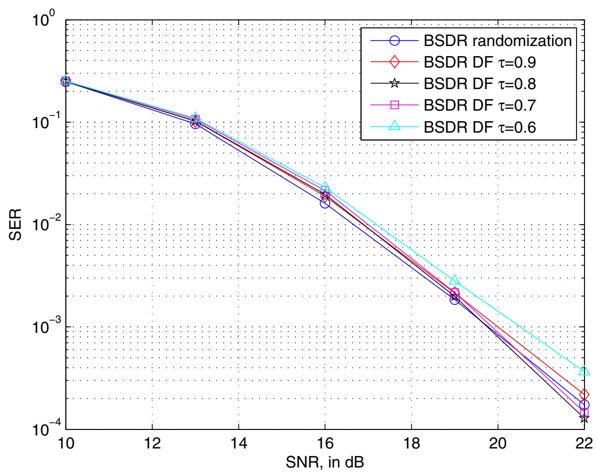


Fig. 6. SER performance, $n = m = 6$, 8-PSK.

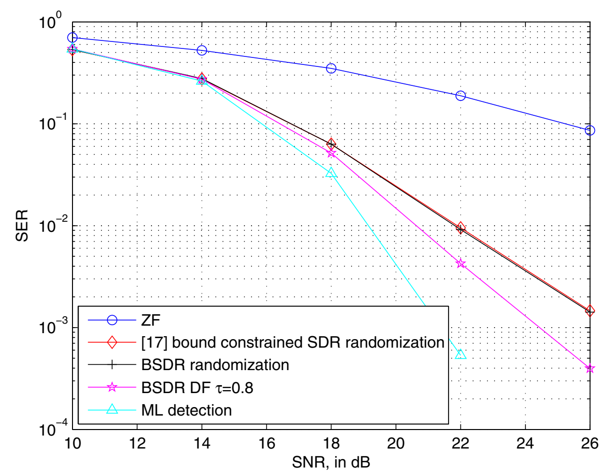


Fig. 8. SER performance, $n = m = 6$, 16-QAM.

Figs. 4 and 6 that in the 16-QAM case, the SER performance of the on-the-fly decision feedback scheme varies significantly for different values of threshold τ and the best performance is achieved when $\tau = 0.8$, which is better than the randomization approach. While in the 8-PSK case, different values of τ lead to very similar SER performance, which is very close to the SER performance of the randomization approach. Take both criteria into consideration, $\tau = 0.7$ or 0.8 are good choice for both 16-QAM and 8-PSK, which gives better or close SER performance and lower or comparable computational complexity compared with the original binary SDR with randomization, especially at high SNR.

C. Symbol Error Performance

In Fig. 8, we compare our proposed BSDR method with the zero forcing (ZF), the bound constrained SDR method in [17], and the maximum likelihood (ML) detection for a MIMO system with $n = m = 6$ and 16-QAM constellation. As expected, our proposed BSDR method gives exactly the same SER performance as the bound constrained SDR when randomization is used, since in 16-QAM case our BSDR method is the same as the one in [12] and [12] is proved to be equivalent to [17] for any 4^q-QAM constellation in [20]. With on-the-fly decision feedback scheme, we can further improve the performance by about 2dB at SER = 10^{-3}.

Fig. 9 considers the 8-PSK constellation. The size of the MIMO system is still $n = m = 6$. In this test, we compare our BSDR method with the norm constrained SDR method in [10]. We can see that our BSDR method (either with randomization or on-the-fly decision feedback) outperforms [10] by about 0.6dB and within 1dB from the ML detection at SER = 10^{-3}.

Fig. 10 shows the SER performance on a MIMO system with $n = m = 6$ and 12-QAM constellation. Since no SDR based detection method has been proposed before for 12-QAM, we just compare our BSDR method with zero forcing and exact ML detection. We can see that our BSDR method with on-the-fly decision feedback is within 2dB from the ML detection at SER = 10^{-3}.

VIII. Conclusions

In this paper, we have proposed a universal binary SDR scheme to deal with arbitrary signal constellations. The proposed scheme first binarizes the original signal space to a linearly constrained binary space, and then solves the detection problem through semidefinite relaxation. The proposed binary SDR reduces to the existing SDR methods for BPSK, QPSK and 16QAM, and can deal with any other constellations such

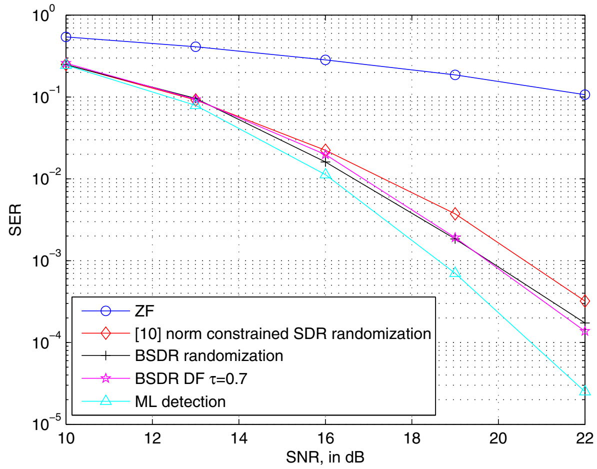


Fig. 9. SER performance, $n = m = 6$, 8-PSK.

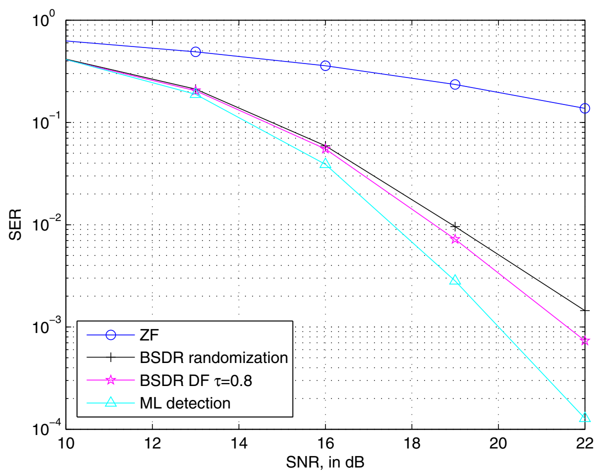


Fig. 10. SER performance, $n = m = 6$, 12-QAM.

as 12-QAM, 32-QAM, or M-PSK. A tailored dual barrier method has been provided to solve the SDR efficiently in practice and we propose to apply on-the-fly decision feedback during the execution of the dual barrier method to further reduce the computational cost and improve the detection performance. Simulation results show that our binary SDR, together with on-the-fly decision feedback scheme, can provide a comparable or better solution for 16-QAM and 8-PSK compared with existing methods in terms of computational complexity and symbol error rate.

REFERENCES

[1] S. Verdú, *Multiuser Detection*. Cambridge University Press, 1998.

[2] W. H. Mow, "Maximum likelihood sequence estimation from the lattice viewpoint," *IEEE Trans. Inf. Theory*, vol. 40, no. 5, pp. 1591–1600, Sept. 1994.

[3] M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inf. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.

[4] A. D. Murugan, H. E. Gamal, M. O. Damen, and G. Caire, "A unified framework for tree search decoding: rediscovering the sequential decoder," *IEEE Trans. Inf. Theory*, vol. 52, no. 3, pp. 933–953, Mar. 2006.

[5] J. Jalden, L. G. Barbero, B. Ottersten, and J. S. Thompson, "Full diversity detection in MIMO systems with a fixed-complexity sphere decoder," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing*.

[6] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Process.*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[7] C. Helmberg and F. Rendl, "Solving quadratic (0, 1)-problems by semidefinite programs and cutting planes," *Mathematical Programming*, vol. 82, no. 3, pp. 291–315, Mar. 1998.

[8] P. H. Tan and L. K. Rasmussen, "The application of semidefinite programming for detection in CDMA," *IEEE J. Sel. Areas Commun.*, vol. 19, no. 8, pp. 1442–1449, Aug. 2001.

[9] W. K. Ma, T. N. Davidson, K. M. Wong, Z.-Q. Luo, and P. C. Ching, "Quasi-maximum-likelihood multiuser detection using semi-definiterelaxation with application to synchronous CDMA," *IEEE Trans. Signal Process.*, vol. 50, no. 4, pp. 912–922, Apr. 2002.

[10] W. K. Ma, P. C. Ching, and Z. Ding, "Semidefinite relaxation based multiuser detection for M-ary PSK multiuser systems," *IEEE Trans. Signal Process.*, vol. 52, no. 10, pp. 2862–2872, Oct. 2004.

[11] A. Wiesel, Y. C. Eldar, and S. Shamai, "Semidefinite relaxation for detection of 16-QAM signaling in MIMO channels," *IEEE Signal Process. Lett.*, vol. 12, no. 9, pp. 653–656, May 2005.

[12] Z. Mao, X. Wang, and X. Wang, "Semidefinite programming relaxation approach for multiuser detection of QAM signals," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4275–4279, Dec. 2007.

[13] Y. Yang, C. Zhao, P. Zhou, and W. Xu, "MIMO detection of 16-QAM signaling based on semidefinite relaxation," *IEEE Signal Process. Lett.*, vol. 14, no. 11, pp. 797–800, Jun. 2007.

[14] X. M. Wang, W. S. Lu, and A. Antoniou, "A near-optimal multiuser detector for DS-CDMA systems using semidefinite programming relaxation," *IEEE Trans. Signal Process.*, vol. 51, no. 9, pp. 2446–2450, Sept. 2003.

[15] J. Jalden, C. Martin, and B. Ottersten, "Semidefinite programming for detection in linear systems-optimality conditions and space-time decoding," in *2003 IEEE International Conference on Acoustics, Speech and Signal Processing*.

[16] J. Jalden, B. Ottersten, and W. K. Ma, "Reducing the average complexity of ML detection using semidefinite relaxation," in *2005 IEEE International Conference on Acoustics, Speech and Signal Processing*.

[17] N. D. Sidiropoulos and Z. Q. Luo, "A semidefinite relaxation approach to MIMO detection for high-order QAM constellations," *IEEE Signal Process. Lett.*, vol. 13, no. 9, pp. 525–528, Sept. 2006.

[18] A. Mobasher, M. Taherzadeh, R. Sotirov, and A. K. Khandani, "A near-maximum-likelihood decoding algorithm for MIMO systems based on semi-definite programming," *IEEE Trans. Inf. Theory*, vol. 53, no. 11, pp. 3869–3886, Nov. 2007.

[19] M. Kisialiou, X. Luo, and Z.-Q. Luo, "Efficient implementation of quasi-maximum-likelihood detection based on semidefinite relaxation," *IEEE Trans. Signal Process.*, vol. 57, no. 12, pp. 4811–4822, Dec. 2009.

[20] W. K. Ma, C. C. Su, J. Jaldén, T. H. Chang, and C. Y. Chi, "The equivalence of semidefinite relaxation MIMO detectors for higher-order QAM," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 6, pp. 1038–1052, Jun. 2009.

[21] R. H. Tütüncü, K. C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using SDPT3," *Mathematical Programming*, vol. 95, no. 2, pp. 189–217, 2003.

[22] J. F. Sturm, "Using SeDuMi 1. 02, A Matlab toolbox for optimization over symmetric cones," *Optimization Methods and Software*, vol. 11, no. 1, pp. 625–653, 1999.

[23] S. Benson, Y. Ye, and X. Zhang, "Solving large-scale sparse semidefinite programs for combinatorial optimization," *SIAM J. Optimization*, vol. 10, no. 2, pp. 443–461, 2000.

[24] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz, "An interior-point method for semidefinite programming," *SIAM J. Optimization*, vol. 6, no. 2, pp. 342–361, 1996. Available: http://epubs.siam.org/doi/abs/10.1137/0806020

[25] Z. Wen, D. Goldfarb, S. Ma, and K. Scheinberg, "Row by row methods for semidefinite programming," *Industrial Engineering*, pp. 1–21, Jan 2009.

[26] H. Wai, W. Ma, and A. So, "Cheap semidefinite relaxation mimo detection using row-by-row block coordinate descent," in *Proc. 2011 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 3256–3259.

[27] Z. Luo, W. Ma, A. So, Y. Ye, and S. Zhang, "Semidefinite relaxation of quadratic optimization problems," *IEEE Signal Process. Mag.*, vol. 27, no. 3, pp. 20–34, Mar. 2010.

[28] D. Palomar and Y. Eldar, *Convex Optimization in Signal Processing and Communications*. Cambridge University Press, 2010.

[29] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.

[30] J. R. Magnus and H. Neudecker, *Matrix Differential Calculus with Applications in Statistics and Econometrics*. Wiley, 1988.

Xiaopeng Fan (S'07-M'09) received the B.S. and M.S. degrees from the Harbin Institute of Technology (HIT), China, in 2001 and 2003 respectively, and the Ph.D degree from Hong Kong University of Science and Technology (HKUST) in 2009.

In 2009, he joined the Department of Computer Science, Harbin Institute of Technology (HIT), where he is currently an Associate Professor. From 2003 to 2005, he worked with Intel China Software Laboratory(ICSL) as Software Engineer.

His current research interests are in image/video processing and wireless communication. He has authored or co-authored over 60 technical journal and conference papers. His transform and error resilience algorithms are accepted into the China AVS standard. He served as associate editor of IEEE AVS standard from 2012. He is a recipient of the New Century Excellent Talents in University (NCET) Program of China and the MSRA 'StarTrack' Visiting Young Faculty Program in 2011.

Junxiao Song (S'13) received the B.S. degree in Control Science and Engineering from Zhejiang University (ZJU), Hangzhou, China, in 2011. Currently, he is pursuing the Ph.D. degree with the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology (HKUST). His research interests are inconvex optimization, signal processing and distributed algorithms, with applications in communication and financial engineering.

Daniel P. Palomar (S'99-M'03-SM'08-F'12) received the Electrical Engineering and Ph.D. degrees from the Technical University of Catalonia (UPC), Barcelona, Spain, in 1998 and 2003, respectively. He is an Associate Professor in the Department of Electronic and Computer Engineering at the Hong Kong University of Science and Technology (HKUST), Hong Kong, which he joined in 2006. Since 2013 he is a Fellow of the Institute for Advance Study (IAS) at HKUST. He had previously held several research appointments, namely, at King's College London (KCL), London, UK; Technical University of Catalonia (UPC), Barcelona; Stanford University, Stanford, CA; Telecommunications Technological Center of Catalonia (CTTC), Barcelona; Royal Institute of Technology (KTH), Stockholm, Sweden; University of Rome "La Sapienza", Rome, Italy; and Princeton University, Princeton, NJ. His current research interests include applications of convex optimization theory, game theory, and variational inequality theory to financial systems and communication systems. Dr. Palomar is an IEEE Fellow, a recipient of a 2004/06 Fulbright Research Fellowship, the 2004 Young Author Best Paper Award by the IEEE Signal Processing Society, the 2002/03 best Ph.D. prize in Information Technologies and Communications by the Technical University of Catalonia (UPC), the 2002/03 Rosina Ribalta first prize for the Best Doctoral Thesis in Information Technologies and Communications by the Epson Foundation, and the 2004 prize for the best Doctoral Thesis in Advanced Mobile Communications by the Vodafone Foundation and COIT. He serves as an Associate Editor of IEEE TRANSACTIONS ON INFORMATION THEORY, and has been an Associate Editor of IEEE TRANSACTIONS ON SIGNAL PROCESSING, a Guest Editor of the *IEEE Signal Processing Magazine* 2010 Special Issue on ""Convex Optimization for Signal Processing," the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2008 Special Issue on "Game Theory in Communication Systems," and the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS 2007 Special Issue on "Optimization of MIMO Transceivers for Realistic Communication Networks."

Oscar C. Au (F'11) received his B.A.Sc. from Univ. of Toronto in 1986, his M.A. and Ph.D. from Princeton Univ. in 1988 and 1991 respectively. After being a postdoctoral researcher at Princeton Univ. for one year, he joined the Hong Kong University of Science and Technology (HKUST) as an Assistant Professor in 1992. He is/was a Professor of the Dept. of Electronic and Computer Engineering, Director of Multimedia Technology Research Center (MTrec), and Director of the Computer Engineering (CPEG) Program in HKUST. His main research contributions are on video and image coding and processing, watermarking and light weight encryption, speech and audio processing. He has published 60+ technical journal papers, 350+ conference papers, and 70+ contributions to international standards. His fast motion estimation algorithms were accepted into the ISO/IEC 14496-7 MPEG-4 international video coding standard and the China AVS-M standard. His light-weight encryption and error resilience algorithms are accepted into the China AVS standard. He was Chair of Screen Content Coding AdHoc Group in the JCTVC for the ITU-T H.265 HEVC video coding standard. Dr. Au is a Fellow of the Institute of Electrical and Electronic Engineering (IEEE), a Fellow of the Hong Kong Institute of Engineers (HKIE), and a Board 0f Governor member of the Asia Pacific Signal and Information Processing Association (APSIPA). He is/was Associate Editors of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY (TCSVT), IEEE TRANSACTIONS ON IMAGE PROCESSING (TIP), and IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS, PART 1 (TCAS1). He is/was Chair of IEEE CAS Technical Committee on Multimedia Systems and Applications (MSATC), Chair of IEEE SPS TC on Multimedia Signal Processing (MMSP), and Chair of APSIPA TC on Image, Video and Multimedia (IVM). He is a member of CAS TC on Video Signal Processing and Communications (VSPC), CAS TC on Digital Signal Processing (DSP), SPS TC on Image, Video and Multidimensional Signal Processing (IVMSP), SPS TC on Information Forensics and Security (IFS), and ComSoc TC on Multimedia Communications (MMTC).