Efficient Soft-Output Demodulation of MIMO QPSK via Semidefinite Relaxation

Mehran Nekuii, *Member, IEEE*, Mikalai Kisialiou, Timothy N. Davidson, *Member, IEEE*, and Zhi-Quan Luo, *Fellow, IEEE*

***Abstract*—Two efficient list-based "soft"-output demodulators are developed for iterative receivers in multiple-input multiple-output (MIMO) communication systems with QPSK signaling. The proposed demodulators are based on the semidefinite relaxation (SDR) technique, and hence their computational costs are bounded by a low-order polynomial of the number of bits transmitted per channel use. The first demodulator applies the SDR technique once per demodulation-decoding iteration, and generates list members via the randomization procedure that is inherent in the SDR technique. The second demodulator is based on an approximation of that randomization procedure by a set of independent Bernoulli trials, and this approximation reduces the number of semidefinite programs that need to be solved to just one per channel use. List-free implementations that reduce the memory requirements of list demodulators with moderate to long lists are also developed. Analysis suggests that the proposed "Single-SDR" demodulator should offer good performance at moderate computational cost, especially for larger systems. This is quantified using simulations of a richly scattered environment, in which the performance of the Single-SDR demodulator is similar to that of the list sphere decoder with moderate sized lists and better than that of the minimum mean square error soft interference canceler. The average computational cost of a straightforward implementation of the Single-SDR demodulator is competitive with that of the list sphere decoder with moderate sized lists, and the distribution of its computational cost is quite concentrated around the average.**

***Index Terms*—Iterative demodulation and decoding (IDD), multiple-input multiple-output (MIMO) communication, multiuser detection, semidefinite relaxation, soft-output demodulation.**

Manuscript received January 24, 2011; revised May 28, 2011 and August 14, 2011; accepted September 14, 2011. Date of publication October 19, 2011; date of current version November 18, 2011. This work was supported in part by the Natural Sciences and Engineering Research Council (NSERC) and the National Science Foundation (NSF) under Grant DMS-1015346. The work of T. N. Davidson was supported by the Canada Research Chairs program. Preliminary versions of portions of this work appear in Proceedings of the IEEE Workshop Signal Processing Advances in Wireless Communications 2007, and Proceedings of the IEEE International Conference of Acoustics, Speech, and Signal Processing 2008. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Riccardo Raheli.

M. Nekuii was with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4L8, Canada. He is now with Cavium, Inc., Montreal, QC H9P 2W8, Canada (e-mail: nekuii@ieee.org).

M. Kisialiou was with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55414 USA. He is now with Intel Corporation, Portland, OR 97208 USA (e-mail: kisialiou@gmail.com).

T. N. Davidson is with the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON L8S 4L8, Canada (e-mail: davidson@mcmaster.ca).

Z.-Q. Luo is with the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, MN 55414 USA (e-mail: luozq@umn.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSTSP.2011.2172772

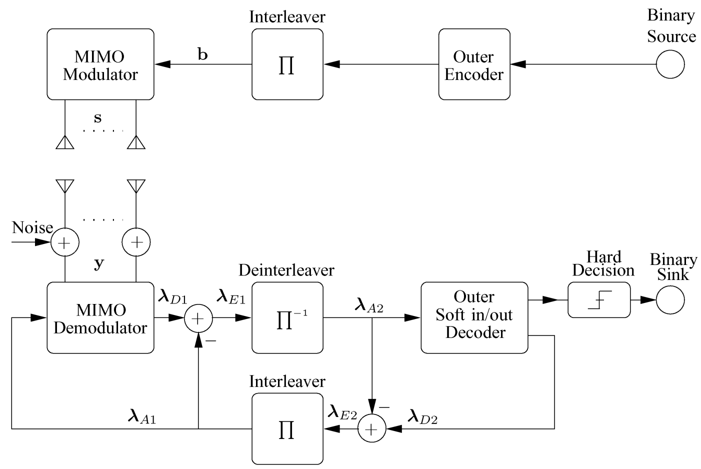


Fig. 1. MIMO BICM-IDD transceiver.

I. Introduction

The provision of multiple antennas at both the transmitter and receiver of a wireless communication system offers the potential for reliable communication at data rates that are substantially higher than those of single antenna systems [1]. One of the core challenges in the design of such multiple-input multiple-output (MIMO) systems is to obtain good performance at high data rates without incurring unreasonable computational cost. A popular transceiver architecture for moving toward that goal is that of MIMO bit-interleaved coded modulation (BICM) with iterative "soft" demodulation and decoding (IDD), e.g., [2]; see also Fig. 1. Although the IDD receiver architecture has many desirable features, the computational cost of the (exact) soft demodulator increases exponentially with the number of encoded bits transmitted per channel use, and hence there has been considerable interest in the development of approximate soft demodulation schemes with lower complexity; e.g., [2]–[20].

One approach to lower-complexity soft-output demodulation is to apply the so-called "max-log" approximation [21], under which the posterior log likelihood ratio (LLR) of each bit is approximated by the difference between the optimal values of a pair of "hard"-output demodulation problems; e.g., [8]–[10], [17]. However, each of these hard-output demodulation problems is also hard in the NP sense. Tree search methods (e.g., [22]), such as sphere decoding (e.g., [23], [24]), can be used to find optimal solutions to these problems (e.g., [8], [9], and [17]), but both the average and worst-case computational costs of searching for the optimal solution remain exponential in the problem size [25], and the "tail" of the distribution of the computational cost can be quite significant at low SNRs or for large problem sizes; e.g., [26]. In order to reduce the computational cost of tree search methods, a variety of reduced tree search strategies have been proposed for generating good suboptimal solutions to the hard-output demodulation problems; e.g., [12], [17], [27], [28]. An alternative approach [10] to addressing the computational cost of the hard-output demodulation problems is to employ the semidefinite relaxation (SDR) technique (e.g., [29]), which has also been considered in a variety of other hard-output demodulation contexts; e.g., [30]–[33]. The growth of the computational cost of the SDR technique is (upper) bounded by a low-order polynomial in the problem size, and this technique has the advantage that (upper) bounds on the approximation error are available [34]–[36]. However, when the demodulator in [10] is used in an IDD receiver, the number of semidefinite programs that must be solved in each demodulation-decoding iteration grows linearly in the number of encoded bits transmitted per channel use.

A different approach to approximate soft(-output) demodulation is to apply the principles of list decoding, in which one seeks to efficiently identify a list of bit-vectors that dominate the LLRs; e.g., [2]. The LLRs can then be approximated by marginalizing over the list, or by applying the "max-log" approximation over the list. Most of the existing techniques are based on the use of modified tree search algorithms to identify members of the list; e.g., [2], [4]–[7], [12]–[14], [16]. In some list demodulation schemes for iterative receivers (e.g., [2], [12], and [13]) the list is generated once per channel use, in the first demodulation-decoding iteration, and is stored for use in the subsequent iterations. In other schemes (e.g., [4]–[7], and [16]), the list is regenerated in each demodulation-decoding iteration, which enables the information available from the decoder output at the previous iteration to be used in the construction of the list.

In this paper, we develop an alternative approach to list-based soft demodulation of MIMO QPSK that is based on semidefinite relaxation. We propose two new demodulators, both of which regenerate the list in each demodulation-decoding iteration, using the information available from the previous iteration of the decoder. The first demodulator applies the semidefinite relaxation technique once per demodulation-decoding iteration, and generates list members via the randomization procedure [34], [35] that is inherent in SDR techniques. The second demodulator is based on an approximation of this randomization procedure by a set of independent Bernoulli trials. This approximation allows us to reduce the number of semidefinite programs to be solved to just one per channel use. We also develop a list-free implementation of a broad class of list demodulators that includes the proposed demodulators, and we show that this implementation reduces the memory requirements of demodulators with moderate to long lists. Insight from the analysis of the approximation accuracy of SDR-based hard demodulation of MIMO QPSK [35], [36] and from the bound on its computational cost (e.g., [29], [37]) suggests that the proposed demodulators should offer a tradeoff between performance and computational cost with some desirable characteristics, especially for larger systems. Our simulations show that this is indeed the case. In particular, in a richly scattered environment the proposed demodulators provide performance similar to that of the list sphere decoder with moderate sized lists, and better than that of the minimum mean square error soft interference canceler. The average computational cost of a straightforward implementation of the second of the proposed demodulators is competitive with that of the list sphere decoder with moderate sized lists, and the distribution of that cost is quite concentrated.

The proposed approach to list generation is substantially different from existing approaches, and in order to effectively establish the principles of the proposed demodulators, in this paper we will focus on the case of QPSK signalling. Extensions to systems that employ higher order QAM constellations will be discussed in the Conclusion.

The paper is organized as follows. In Section II, we provide an overview of the MIMO-BICM-IDD system, and in Section III we review the SDR approach to hard demodulation of MIMO QPSK. In Sections IV and V, we develop the proposed demodulators, which we will call the List-SDR and Single-SDR demodulators, respectively. In Section VI, we describe a list-free implementation of list demodulation that can be applied to the proposed methods, and in Section VII the computational cost of the proposed demodulators is analyzed. The results of simulation experiments that compare the performance and computational cost of the proposed demodulators against those of several existing demodulators will be presented in Section VIII.

II. System Model and Iterative Receiver

We consider a narrowband system with N_t transmit antennas and N_r receive antennas. The signal vector transmitted at the nth channel use is denoted by $\mathbf{s}_n$, and the corresponding received vector is

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

where the channel matrix $\mathbf{H}_n$ is assumed to be known at the receiver, and $\mathbf{v}_n$ is a vector of additive white circular complex Gaussian noise samples with variance σ^2 per real dimension. We will consider a MIMO-BICM-IDD transceiver for this system, e.g., [2]; see Fig. 1. For simplicity, we will focus on the spatial multiplexing transmission scheme [38], but the extension to systems that employ a space-time code from the linear dispersion class [39] is direct. We will let $\mathbf{b}_n$ denote the sub-block of the interleaved outer codeword that is to be transmitted in the nth channel use, and we will let $\mathcal{M}(\mathbf{b})$ denote the mapping used by the MIMO modulator; i.e., $\mathbf{s}_n = \mathcal{M}(\mathbf{b}_n)$. In this paper, we will focus on systems in which this mapping is to QPSK symbols.

The role of the (coherent) soft demodulator in Fig. 1 is to estimate the posterior log likelihood ratio of each interleaved encoded bit; e.g., [2] and [40]. Since the channel in (1) is memoryless, this can be done on a per-channel-use basis, and for notational simplicity we will consider a generic channel use and will drop the subscript n. The LLR of b_i, the ith bit in $\mathbf{b}$, can be written as

$$\log \frac{P(b_i = +1|\mathbf{y})}{P(b_i = -1|\mathbf{y})} = \log \frac{\sum_{\mathbf{b} \in \mathcal{L}_{i,+1}} p(\mathbf{y}|\mathbf{b}) p(\mathbf{b})}{\sum_{\mathbf{b} \in \mathcal{L}_{i,-1}} p(\mathbf{y}|\mathbf{b}) p(\mathbf{b})} \tag{2}$$

where $\mathcal{L}_{i,\pm 1} = \{\mathbf{b} \in \mathcal{L} | b_i = \pm 1\}$, and $\mathcal{L} = \{\mathbf{b} \in \{-1,+1\}^{2N_t}\}$ denotes the (complete) list of possible transmitted bit-vectors. (The conditioning on $\mathbf{H}_n$ has been left implicit in (2).) From the model in (1) we have that $p(\mathbf{y}|\mathbf{b}) \propto \exp(-\|\mathbf{y}-\mathbf{H}\mathcal{M}(\mathbf{b})\|^2/(2\sigma^2))$, and the conventional estimate of $p(\mathbf{b})$ (e.g., [40]) is proportional to $\exp(\boldsymbol{\lambda}_{A1}^T\mathbf{b}/2)$, where $\boldsymbol{\lambda}_{A1}$ is the vector of extrinsic LLRs provided by the previous iteration of the decoder. Hence, the demodulator computes or approximates (e.g., [2])

$$\lambda_{D1,i} = \log \frac{\sum_{\mathcal{L}_{i,+1}} \exp\left(-D(\mathbf{b})/(2\sigma^2)\right)}{\sum_{\mathcal{L}_{i,-1}} \exp\left(-D(\mathbf{b})/(2\sigma^2)\right)} \tag{3}$$

where

$$D(\mathbf{b}) \triangleq \|\mathbf{y} - \mathbf{H}\mathcal{M}(\mathbf{b})\|_2^2 - \sigma^2\boldsymbol{\lambda}_{A1}^T\mathbf{b}. \tag{4}$$

The cardinalities of $\mathcal{L}_{i,\pm1}$ in (3) grow exponentially in the number of encoded bits transmitted per channel use, and hence so does the computational cost of the exact soft demodulator. As a result there has been considerable interest in approximations of (3) that can be computed (or themselves approximated) with less effort. The resulting soft demodulators can be classified according to the nature of the approximations they employ. To help place the proposed demodulators in context, we will draw attention to the classes based on the following two approximations:

$$\lambda_{D1,i} \approx \log \frac{\sum_{\hat{\mathcal{L}}_{i,+1}} \exp\left(-D(\mathbf{b})/(2\sigma^2)\right)}{\sum_{\hat{\mathcal{L}}_{i,-1}} \exp\left(-D(\mathbf{b})/(2\sigma^2)\right)} \tag{5}$$

$$\approx \frac{1}{2\sigma^2}\left(\min_{\mathbf{b}\in\hat{\mathcal{L}}_{i,-1}} D(\mathbf{b}) - \min_{\mathbf{b}\in\hat{\mathcal{L}}_{i,+1}} D(\mathbf{b})\right) \tag{6}$$

where $\hat{\mathcal{L}} \subseteq \mathcal{L}$. (The approximation of (5) by (6) is often referred to as the "max-log" approximation.) Although many existing soft demodulators employ one of these approximations, there are several that do not, including the minimum mean square error soft (parallel) interference cancellation (MMSE-SIC) demodulator [3], [41]–[43], and its approximations [44], [45], and the partial marginalization demodulators [15], [20].

One class of soft demodulators is based on selecting $\hat{\mathcal{L}} = \mathcal{L}$ and solving the two binary quadratic optimization problems in (6) for each encoded bit. As mentioned in the Introduction, optimal solutions to these "hard" demodulation problems can be obtained using tree-search algorithms, as they are in [8], [9], [17], and suboptimal solutions can be found at lower computational cost using reduced tree search algorithms (e.g., [12], [17], [27], [28]), or by employing semidefinite relaxation [10].

A second class of soft demodulators is based on efficiently selecting a list $\hat{\mathcal{L}}$ of bit-vectors that generate small values for $D(\mathbf{b})$ and then approximating the LLR either by marginalizing over $\hat{\mathcal{L}}_{i,\pm1}$, as in (5), e.g., [5], or by performing an exhaustive search over $\hat{\mathcal{L}}_{i,\pm1}$ to solve the minimization problems in (6); e.g., [2]. The key challenge in this class of methods is the efficient selection of the members of $\hat{\mathcal{L}}$, and most existing approaches are based on tree-search ideas; e.g., [2], [4]–[7], [12]–[14], and [16].

The two demodulators proposed in this paper fall into the second class of approximate soft demodulators, but they are based on semidefinite relaxation rather than a tree search. (The SDR-based demodulator in [10] falls into the first class.) A key step in the development of these soft demodulators is to exploit the properties of the randomization step that is inherent in the semidefinite relaxation technique [34], [35], and before we introduce them we will provide a brief overview of the application of the semidefinite relaxation to hard demodulation [30], [31].

III. Hard Demodulation Using SDR

Consider the real-valued equivalent representation for (1) with QPSK signaling:

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\mathbf{b} + \tilde{\mathbf{v}} \tag{7}$$

where $\tilde{\mathbf{y}}$, $\mathbf{b}$ and $\tilde{\mathbf{v}}$ are the concatenations of the real and imaginary parts of $\mathbf{y}$, $\mathbf{s}$ and $\mathbf{v}$, respectively, and we have considered an arbitrary channel use. Given prior information on the bit probabilities in the form of $\boldsymbol{\lambda}_{A1}$ in (4), the bit-vector $\mathbf{b}$ that maximizes the *a posteriori* probability is the solution to the following binary optimization problem:

$$\min_{\mathbf{b}\in\{+1,-1\}^{2N_t}} D(\mathbf{b}) = \min_{\mathbf{b}\in\{+1,-1\}^{2N_t}} \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\mathbf{b}\|_2^2 - \sigma^2\boldsymbol{\lambda}_{A1}^T\mathbf{b}. \tag{8}$$

Using the definitions [10], [30]

$$\tilde{\mathbf{b}} \triangleq \begin{bmatrix} \breve{\mathbf{b}} \\ c \end{bmatrix}, \quad \mathbf{Q} \triangleq \begin{bmatrix} \tilde{\mathbf{H}}^T\tilde{\mathbf{H}} & \mathbf{a} \\ \mathbf{a}^T & 0 \end{bmatrix} \tag{9a}$$

$$\breve{\mathbf{b}} \triangleq c\mathbf{b}, \quad \mathbf{a} \triangleq -\tilde{\mathbf{H}}^T\tilde{\mathbf{y}} - 0.5\sigma^2\boldsymbol{\lambda}_{A1} \tag{9b}$$

in which $c \in \{+1, -1\}$, the problem in (8) can be stated as the following (NP-hard) binary quadratic programming (BQP) problem:

$$\min_{\tilde{\mathbf{b}}\in\{+1,-1\}^{2N_t+1}} \tilde{\mathbf{b}}^T\mathbf{Q}\tilde{\mathbf{b}}. \tag{10}$$

Using the substitution $\mathbf{X} = \tilde{\mathbf{b}}\tilde{\mathbf{b}}^T$, the problem in (10) can be reformulated as

$$\min_{\mathbf{X}\in\mathbb{S}_{2N_t+1}} \quad \mathrm{Trace}(\mathbf{XQ}) \tag{11a}$$

$$\text{s.t.} \quad \mathbf{X} \succcurlyeq \mathbf{0}, \quad \mathrm{rank}(\mathbf{X}) = 1 \tag{11b}$$

$$[\mathbf{X}]_{ii} = 1, \quad i = 1, \ldots, 2N_t + 1 \tag{11c}$$

where $\mathbb{S}_m \subset \mathbb{R}^{m\times m}$ denotes the set of symmetric matrices of size $m \times m$, and $\mathbf{X} \succcurlyeq \mathbf{0}$ denotes that $\mathbf{X}$ is positive semidefinite. In (11), the computational difficulties of (10) manifest themselves in the rank-1 constraint. The semidefinite relaxation approach to approximating the solution to (10) is to relax the rank-1 constraint and solve the following semidefinite program (SDP):

$$\min_{\mathbf{X}\in\mathbb{S}_{2N_t+1}} \quad \mathrm{Trace}(\mathbf{XQ}) \tag{12a}$$

$$\text{s.t.} \quad \mathbf{X} \succcurlyeq \mathbf{0} \tag{12b}$$

$$[\mathbf{X}]_{ii} = 1, \quad i = 1, \ldots, 2N_t + 1. \tag{12c}$$

This problem is convex and can be efficiently solved (in $O(N_t^{3.5})$ operations) using the interior point method in [37]; see Section VII. (See also [33] for some recent developments.) When the optimal solution to (12), denoted $\mathbf{X}_{\mathrm{opt}}$, is rank 1, its factorization generates an optimal solution to (10). In the more common event that the solution to (12) is not rank 1, a randomization procedure [34], [35] can be used to extract an approximation of the solution to (10) from $\mathbf{X}_{\text{opt}}$. That procedure involves the construction of a Cholesky factorization $\mathbf{X}_{\text{opt}} = \mathbf{V}^T\mathbf{V}$, and the generation of a sequence of random vectors $\mathbf{u}$ from the uniform distribution on the unit hypersphere. For each vector $\mathbf{u}$ we compute $\tilde{\mathbf{x}} = \text{sign}(\mathbf{V}^T\mathbf{u})$, construct the vector $\mathbf{x} = \tilde{x}_{2N_t+1} \times [\tilde{x}_1, \ldots, \tilde{x}_{2N_t}]^T$, and compute $D(\mathbf{x})$ using (4). If this value of $D(\mathbf{x})$ is smaller than the smallest encountered in the previous steps, then $\mathbf{x}$ is retained as $\mathbf{b}_{\text{sdr}}$, the current approximation of the optimal solution to (10). A key feature of the SDR approach to hard demodulation of MIMO QPSK is that there are several theoretical bounds on the approximation accuracy [29], [35], [36] that reinforce the good performance that is seen in practice; e.g., [10], [19], [26], [30], [31], [33].

In [10], Steingrimsson *et al.* developed a soft MIMO demodulator from the first class in Section II that was based on the semidefinite relaxation technique described above. For each channel use, that demodulator solves $2N_t + 1$ SDPs per demodulation-decoding iteration, and hence we will call that scheme the "multi-SDR" method. In the next section, we will propose a soft demodulation scheme from the second class (i.e., a list-based demodulator) that requires the solution to just one SDP in each demodulation-decoding iteration for each channel use. In Section V, we will propose a list-based scheme that requires the solution of only one SDP per channel use.

IV. List-SDR Method for Soft Demodulation

One of the properties of the SDR approach to hard demodulation is that, on average, the bit-vectors generated by the randomization procedure yield small values for the objective in (8). This suggests that, on average, those bit-vectors are good candidates for membership of the list in a list-based approach to soft MIMO demodulation, and this is the insight behind the proposed approach: the List-SDR demodulator uses the randomization procedure to construct a list $\hat{\mathcal{L}}$, and then approximates the LLRs using either (5) or (6).

The procedure for constructing the list used by the List-SDR demodulator is summarized in Table I. The first phase of that procedure is to construct a preliminary list, $\hat{\mathcal{L}}'$, by storing each (unique) bit-vector generated by the randomization procedure (Lines 1–8). The randomization procedure is repeated until the number of members of the preliminary list, $|\hat{\mathcal{L}}'|$, reaches K or the number of randomizations reaches M, where K and M are pre-specified constants, with $K \leq M$. The second phase of the list construction procedure is to enrich the list by identifying the elements in $\hat{\mathcal{L}}'$ with the J smallest metrics, and adding to the list all those bit vectors that are at a Hamming distance of one from (at least) one of these J "best" bit vectors (and are not already members of the list; see Lines 9 and 10). This enrichment is based on ideas in [11] and can be implemented by "flipping" individual bits of each element of the J best bit vectors. Since $|\hat{\mathcal{L}}'| \leq K$, the size of the enriched list satisfies $|\hat{\mathcal{L}}| \leq |\hat{\mathcal{L}}'| + 2JN_t \leq K + 2JN_t$.

TABLE I
List Generation Component of the List-SDR Algorithm

Data: $\mathbf{X}_{\text{opt}}$, the solution to (12), or an approximation thereof.
Parameters: M, the number of randomization iterations; K, the maximum size of the preliminary list; J, the number of list members to be used in the enrichment phase.
Output: $\hat{\mathcal{L}}$, the enriched list.

1: Initialize $\hat{\mathcal{L}}'$ and $\hat{\mathcal{L}}$ empty, $m = 0$, and $k = 0$.
2: Compute a (Cholesky) factor $\mathbf{V}$ of $\mathbf{X}_{\text{opt}}$ such that $\mathbf{X}_{\text{opt}} = \mathbf{V}^T\mathbf{V}$.
3: **while** $k < K$ and $m < M$ **do**
4: Choose a random vector $\mathbf{u}$ from the uniform distribution on the unit sphere.
5: Compute $\tilde{\mathbf{x}} = \text{sign}(\mathbf{V}^T\mathbf{u})$ and increment m.
6: Construct $\mathbf{x} = \tilde{x}_{2N_t+1} \times [\tilde{x}_1, \ldots, \tilde{x}_{2N_t}]^T$.
7: If $\mathbf{x} \notin \hat{\mathcal{L}}'$, then update $\hat{\mathcal{L}}'$, $\hat{\mathcal{L}}' \leftarrow \{\hat{\mathcal{L}}', \mathbf{x}\}$, and increment k.
8: **end while**
9: Compute (and store) $D(\mathbf{b})$ in (4) for each element of $\hat{\mathcal{L}}'$ and select those vectors with the J smallest metrics.
10: Construct $\hat{\mathcal{L}}$ as the union of $\hat{\mathcal{L}}'$ and all the single bit-flippings of those J "best" bit-vectors.

Once the enriched list has been constructed, the LLRs can be approximated using either (5) or (6), where advantage can be taken of the prior computation of the metrics, $D(\mathbf{b})$, for the elements of $\hat{\mathcal{L}}'$ on Line 9 of the list construction procedure. These approximate LLRs can then be "clipped" in the conventional manner (e.g., [6]), if desired. If we are to use the approximation in (5) we must ensure that the list members are unique. This is done on Line 7 in Table I, and is done implicitly on Line 10, as well. To facilitate that operation, we store the list members in a binary heap (e.g., [46]), and we choose the heap metric to be $\mathbf{w}^T(\mathbf{x} + \mathbf{1})$, where $[\mathbf{w}]_k = 2^{k-2}$ and $\mathbf{1}$ is a conformally sized vector of ones, so that only binary comparisons are needed. (One need not explicitly compute $\mathbf{w}^T(\mathbf{x} + \mathbf{1})$.) To determine the J best bit vectors, we use a second binary heap in which the heap metric is $D(\mathbf{b})$.[1] In the case of the "max-log" approximation of the LLRs in (6), uniqueness of the list members is not required, and hence Lines 7 and 10 can be simplified, but ensuring uniqueness in the manner described above does avoid redundant evaluations of the metric $D(\mathbf{b})$ when the maxima in (6) are calculated by enumeration over the list.

V. Single-SDR Method for Soft Demodulation

An interesting property of the SDR approach to approximating the solution to a binary quadratic problem is that an analytic expression can be obtained for the mean value of each element of the candidate bit-vectors $\mathbf{x}$ that are generated by the randomization procedure described in Section III. The mean value of the ith element can be computed by using the fact that if the inner products of the random vector $\mathbf{u}$ with columns $\mathbf{v}_i$ and $\mathbf{v}_{2N_t+1}$ of the Cholesky factor $\mathbf{V}$ have the same sign then $x_i = +1$, otherwise $x_i = -1$; cf. [34], [35]. Since the random vector $\mathbf{u}$ is uniformly distributed on the unit sphere, the mean value for x_i over the randomization iterations depends on the angle, $\theta_{i,2N_t+1}$, between $\mathbf{v}_i$ and $\mathbf{v}_{2N_t+1}$ and can be written as

$$\mu_i = \frac{\pi - 2\theta_{i,2N_t+1}}{\pi}. \tag{13}$$

[1] Instead of using two heaps, one could manage the list using a single heap with the heap metric $D(\mathbf{b})$, but doing so would require real-valued comparisons throughout, and may involve redundant computation of $D(\mathbf{b})$.

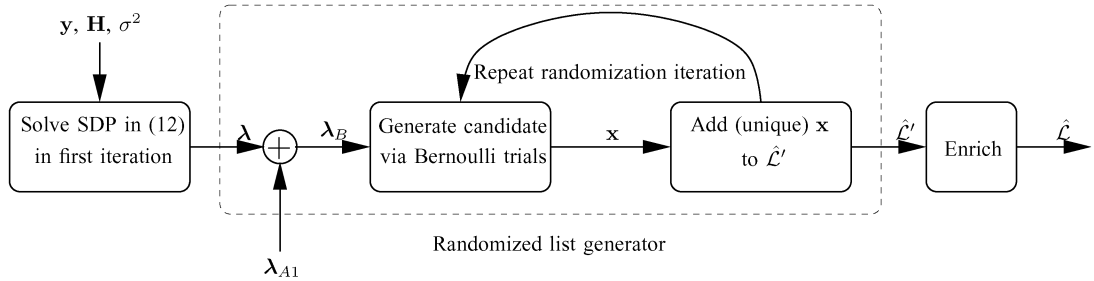


Fig. 2. List generation scheme using the Single-SDR algorithm.

Using the fact that $\mathbf{v}_i^T\mathbf{v}_{2N_t+1} = \|\mathbf{v}_i\|\|\mathbf{v}_{2N_t+1}\|\cos(\theta_{i,2N_t+1})$, and the fact that the constraint $[\mathbf{X}]_{ii} = 1$ in (12) ensures that all $\|\mathbf{v}_i\| = 1$, the mean value can be expressed directly in terms of the columns of $\mathbf{V}$:

$$\mu_i = \frac{2}{\pi}\arcsin\left(\mathbf{v}_i^T\mathbf{v}_{2N_t+1}\right). \tag{14}$$

The first observation in the development of the proposed Single-SDR demodulator is that the expression in (14) suggests that for the purposes of soft demodulation, one could consider generating a sequence of bit vectors with properties similar to those generated by the formal randomization process by making the approximation that the elements of $\mathbf{x}$ are independent, and generating each element of $\mathbf{x}$ via a scalar (antipodal) Bernoulli trial. Such an approach would avoid the cost of computing $\mathbf{V}^T\mathbf{u}$ in each instance of the formal randomization procedure.

The second observation is that this Bernoulli trial approach provides an opportunity to separate the processing of the information provided by the channel output from the processing of the extrinsic information fed back from the previous iteration of the decoder. At each iteration, the decoder updates the extrinsic information that it provides to the demodulator (which we have denoted by $\boldsymbol{\lambda}_{A1}$). The expression for $D(\mathbf{b})$ in (4) suggests that the demodulation procedure needs to be repeated at each iteration (as it is in [4], [5], [7] and in the List-SDR algorithm proposed in Section IV). However, as we will show below, the Bernoulli trial approach to randomization allows us to extract the SDP from the iterative demodulation and decoding loop so that we need only solve one SDP per channel use.

The architecture of the proposed list generation technique is illustrated in Fig. 2. It consists of an SDR demodulator (which is invoked only in the first iteration), and a randomized list generator. The randomized list generator takes two inputs: 1) the vector $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_{2N_t}]^T$ containing the mean values in (14) in LLR form, i.e.,

$$\lambda_i = \log\left(\frac{1+\mu_i}{1-\mu_i}\right) \tag{15}$$

and 2) the vector $\boldsymbol{\lambda}_{A1}$ containing the extrinsic information (in LLR form) from the previous iteration of the decoder. The randomized demodulator then computes Bernoulli distributions that reflect these inputs [see (17)], and generates a sequence of random binary vectors according to those distributions.

By construction, the extrinsic information provided by the decoder is independent of the soft information from the channel [40]. Therefore, if the randomized demodulator is to generate candidate bit-vectors via Bernoulli trials that reflect both the information from the channel and the extrinsic information from the decoder, the LLR representation of the mean of that Bernoulli distribution should be

$$\boldsymbol{\lambda}_B = \boldsymbol{\lambda} + \boldsymbol{\lambda}_{A1}. \tag{16}$$

The ith entry of the corresponding mean vector $\boldsymbol{\mu}_B$ is

$$\mu_{B,i} = 1 - 2/\left(1 + \exp(\lambda_{B,i})\right). \tag{17}$$

Having computed $\boldsymbol{\mu}_B$, the demodulator randomly generates the bit-vectors that will form the preliminary list, $\hat{\mathcal{L}}'$.[2] The ith bit of each of these vectors is generated by running an independent (antipodal) Bernoulli trial with mean $\mu_{B,i}$. An enriched list $\hat{\mathcal{L}}$ is then constructed by adding to the list all the single bit-flippings of the J "best" bit vectors in $\hat{\mathcal{L}}'$. A statement of list generation using this algorithm is provided in Table II. After construction of the list $\hat{\mathcal{L}}$, the soft information from demodulator can be approximated using (5) or (6). As mentioned in Section IV, if the max-log approximation in (6) is used, the list is allowed to contain repeated entries, and Lines 6 and 9 in Table II can be simplified, if desired.

TABLE II
LIST GENERATION COMPONENT OF THE SINGLE-SDR ALGORITHM

Data: $\boldsymbol{\lambda}$ in (15); $\boldsymbol{\lambda}_{A1}$, the vector of extrinsic LLRs from the previous iteration of the decoder.
Parameters: M, the number of randomization iterations; K, the maximum size of the preliminary list; J, the number of list members to be used in the enrichment phase.
Output: $\hat{\mathcal{L}}$, the enriched list.

1: Initialize $\hat{\mathcal{L}}'$ and $\hat{\mathcal{L}}$ empty, $m = 0$, and $k = 0$.
2: Compute $\boldsymbol{\lambda}_B$ in (16) and subsequently $\boldsymbol{\mu}_B$ using (17).
3: **while** $k < K$ and $m < M$ **do**
4: Generate each element of $\mathbf{x}$, x_i, independently according to the (antipodal) Bernoulli distribution with mean $\mu_{B,i}$.
5: Increment m.
6: If $\mathbf{x} \notin \hat{\mathcal{L}}'$, then update $\hat{\mathcal{L}}'$, $\hat{\mathcal{L}}' \leftarrow \{\hat{\mathcal{L}}', \mathbf{x}\}$, and increment k.
7: **end while**
8: Compute (and store) $D(\mathbf{b})$ in (4) for each element of $\hat{\mathcal{L}}'$ and select those vectors with the J smallest metrics.
9: Construct $\hat{\mathcal{L}}$ as the union of $\hat{\mathcal{L}}'$ and all the single bit-flippings of those J "best" bit-vectors.

VI. LIST-FREE IMPLEMENTATION

One of the bottlenecks in the implementation of list-based soft demodulators is the memory required to store the list. In

[2]Since the randomization procedures of the List-SDR and Single-SDR demodulators are different, they may produce different preliminary lists. However, for simplicity we will retain the notation from Sections II and IV and we will use the context to distinguish between the lists.

this section, we will develop a "list-free" implementation of a large class of list demodulators that employ the "max-log" approximation; cf. (6). The list-free approach reduces the memory required to implement demodulators with moderate to long lists, and is immediately applicable to the List-SDR and Single-SDR demodulators.

We consider the broad class of list demodulators in which the candidate list members are generated one at a time. When such demodulators approximate the LLRs by employing the "max-log" approximation over the list (cf. (6)), one need not wait for the whole list to be constructed before solving the $4N_t$ optimization problems in (6). Instead, the optimal values of these problems can be updated as each list member is generated. The list member can then be discarded, because all we need to store are the $4N_t$ real values that are the current optimal values for the problems in (6). If each of these real values is approximated using a B-bit (possibly floating-point) representation, the memory required for this "list-free" implementation is $4N_tB$ bits. In contrast, the regular implementation requires $2N_t|\hat{\mathcal{L}}|$ bits, where $|\hat{\mathcal{L}}|$ is the size of the list. Therefore, the memory required by the list-free implementation is less than that required for the regular implementation whenever $|\hat{\mathcal{L}}| > 2B$. Another advantage of the list-free implementation is that the size of the list can be adapted dynamically, without the need for additional memory management. This offers the potential for the demodulator to dynamically adjust its operating point on its performance-complexity tradeoff curve in response to changes in the characteristics of the channel or in the requirements of the application.

An algorithm for a list-free Single-SDR demodulator is provided in Table III. That algorithm includes an enrichment step (Lines 7–15) that ensures that all the bit vectors that would have been in the list generated by the regular Single-SDR demodulator in Table II are generated by the list-free implementation. In the form in Table III, the list-free demodulator does not attempt to avoid repeated computation of the metric $D(\mathbf{b})$ for the same bit vector $\mathbf{b}$, and hence some additional computational cost may be incurred. This is quantified in the following section.

TABLE III
LIST-FREE IMPLEMENTATION OF THE SINGLE-SDR ALGORITHM

Data: $\boldsymbol{\lambda}$ in (15); $\boldsymbol{\lambda}_{A1}$, the vector of extrinsic LLRs from the previous iteration of the decoder.
Parameters: M, the number of randomization iterations; J, the number of list members to be used in the enrichment scheme.
Output: $\boldsymbol{\lambda}_{D1}$, the vector of log likelihood-ratios

function UPDATE_FS($\mathbf{x}$, $D(\mathbf{x})$, $\mathbf{f}_{+1}$, $\mathbf{f}_{-1}$)
  for $i = 1, 2, \ldots, 2N_t$ **do**
    if $x_i = +1$ **then** $[\mathbf{f}_{+1}]_i \leftarrow \min\{[\mathbf{f}_{+1}]_i, D(\mathbf{x})\}$
    else $[\mathbf{f}_{-1}]_i \leftarrow \min\{[\mathbf{f}_{-1}]_i, D(\mathbf{x})\}$.
    end if
  end for
  return $[\mathbf{f}_{+1}, \mathbf{f}_{-1}]$
end function

1: Compute $\boldsymbol{\lambda}_B$ in (16) and subsequently $\boldsymbol{\mu}_B$ using (17).
2: Initialize $\mathbf{f}_{+1} = \{+\infty\}^{2N_t}$, $\mathbf{f}_{-1} = \{+\infty\}^{2N_t}$, $m = 0$, and a binary (max) heap $\mathcal{J}$ with J elements equal to $+\infty$.
3: **while** $m < M$ **do**
4:   Generate each element of $\mathbf{x}$, x_i, independently according to the (antipodal) Bernoulli distribution with mean $\mu_{B,i}$.
5:   Compute $D(\mathbf{x})$ in (4) and increment m.
6:   $[\mathbf{f}_{+1}, \mathbf{f}_{-1}] =$ UPDATE_FS($\mathbf{x}$, $D(\mathbf{x})$, $\mathbf{f}_{+1}$, $\mathbf{f}_{-1}$).
7:   **if** $D(\mathbf{x})$ is less than the maximal element of $\mathcal{J}$ **then**
8:     Remove the maximum element from the heap.
9:     Insert $D(\mathbf{x})$ into the heap.
10:     **for** $i = 1, 2, \ldots, 2N_t$ **do**,
11:       Set $\check{\mathbf{x}}^{(i)} = \mathbf{x}$ and then set $\check{x}_i^{(i)} = -x_i$.
12:       Compute $D(\check{\mathbf{x}}^{(i)})$ in (4).
13:       $[\mathbf{f}_{+1}, \mathbf{f}_{-1}] =$ UPDATE_FS($\check{\mathbf{x}}^{(i)}$, $D(\check{\mathbf{x}}^{(i)})$, $\mathbf{f}_{+1}$, $\mathbf{f}_{-1}$).
14:     **end for**
15:   **end if**
16: **end while**
17: Compute $\boldsymbol{\lambda}_{D1} = \big(\mathbf{f}_{+1} - \mathbf{f}_{-1}\big)/(2\sigma^2)$.

VII. COMPUTATIONAL COST

In this section, we analyze the computational costs of the proposed List-SDR and Single-SDR demodulators, and compare these costs with those of the Multi-SDR demodulator in [10] and the MMSE-SIC demodulator in [3]. These comparisons provide some insight into the relative costs of implementing these demodulators on a general purpose sequential machine, as various parameters of the system grow. For compatibility with the list-free implementation, we will consider List-SDR and Single-SDR demodulators that employ the "max-log" approximation in (6) and invest in list management in order to avoid redundant computation of the metric $D(\mathbf{b})$; see Sections IV and V.

We begin our analysis by stating the computational cost of each of the components of the algorithms. If we let ϵ denote the accuracy to which the SDP is solved, the worst case computational cost of solving the SDP in (12) using the interior point method in [37] is $O((2N_t+1)^{3.5}\log\epsilon^{-1})$ floating-point operations. The cost of the subsequent Cholesky decomposition is $O((2N_t+1)^3)$. The cost of generating each bit-vector in the conventional randomization procedure used in the Multi-SDR [10] and List-SDR methods is $O((2N_t+1)^2)$. In contrast, the cost of the simplified randomization step in the Single-SDR method is $O(2N_t)$. In order to evaluate the cost of the management aspects of the algorithms, we observe that sequentially constructing a binary heap of size n requires at most $O(n\log n)$ operations, whereas if the data is already available, the heap can be constructed in $O(n)$ operations [46]. We also observe that after deleting the root of a heap, the heap can be reconstructed in $O(\log n)$ operations. Finally, we evaluate the cost of computing the "max-log" approximation of the LLRs; cf. (6). Since $\mathbf{b}$ is binary, computing the metric $D(\mathbf{b})$ requires $2N_t(2N_t+1)$ (signed) real additions. In any given iteration of the List-SDR or Single-SDR demodulators, the number of metrics that needs to be evaluated is the number of elements of the enriched list $\hat{\mathcal{L}}$. Although $|\hat{\mathcal{L}}|$ is a random number, for both of those demodulators it is bounded by $P = K + 2JN_t$. In any given iteration of the list-free implementation of the Single-SDR demodulator, if we let $\breve{Q}$ denote the number of times the condition on Line 7 of Table III is satisfied, then the number of metrics to be computed is $Q = M + 2\breve{Q}N_t$. Since $\breve{Q} \leq M$, we have that $Q \leq M(2N_t+1)$.

By evaluating how many times each of the above operations must be performed in each of T demodulation-decoding iterations, we obtain the expressions in Table IV for the computational cost per channel use of each component of the SDR-based demodulators. To verify some of the entries of

TABLE IV
COMPUTATIONAL COST PER CHANNEL USE OF THE COMPONENTS OF VARIOUS SDR-BASED MIMO SOFT DEMODULATORS FOR A SYSTEM WITH N_t TRANSMIT ANTENNAS AND T DEMODULATION-DECODING ITERATIONS

Demodulator	**Computational Cost***				
	SDP†	Cholesky†	Randomization†	List Management‡	Metric Computation§
Multi-SDR [10]	$O(TN_t^{4.5})$	$O(TN_t^4)$	$O(TMN_t^3)$	—	$O(TMN_t^3)$
List-SDR (Table I)	$O(TN_t^{3.5})$	$O(TN_t^3)$	$O(TMN_t^2)$	$O(TP\log P)$; $O(TK)+O(TJ\log K)$¶	$O(TPN_t^2)$
Single-SDR (Table II)	$O(N_t^{3.5})$	$O(N_t^3)$	$O(TMN_t)$	$O(TP\log P)$; $O(TK)+O(TJ\log K)$¶	$O(TPN_t^2)$
List-free Single-SDR (Table III)	$O(N_t^{3.5})$	$O(N_t^3)$	$O(TMN_t)$	0; $O(TM\log J)$¶	$O(TMN_t^3)$

* Notation. M: pre-specified limit on number of randomization steps; K: pre-specified bound on size of preliminary list ($K \leq M$); J: size of set selected for enrichment ($J \leq K$); P: maximum size of enriched list ($P = K + 2JN_t$). For reference, the cost of the corresponding MMSE-SIC demodulator [3] is $O(TN_t^4)$.
† Floating-point operations
‡ Binary vector comparisons; floating-point scalar comparisons
§ Signed floating-point additions.
¶ If $J = K$ these operations are not required.

Table IV we recall that in each demodulation-decoding iteration, the Multi-SDR method in [10] requires the solution of one SDP of size $2N_t + 1$ and $2N_t$ SDPs of size $2N_t$. After solving each SDP it performs M randomization iterations and computes $D(\mathbf{b})$ for all the generated bit-vectors. The List-SDR approach proposed in Section IV requires the solution of only one SDP of size $2N_t + 1$ per demodulation-decoding iteration, and the total number of evaluations of $D(\mathbf{b})$ required in T iterations is $\sum_{t=1}^{T} |\hat{\mathcal{L}}^{(t)}| \leq TP$, where $|\hat{\mathcal{L}}^{(t)}|$ is the size of the enriched list at iteration t, and $P = K + 2JN_t$. The Single-SDR demodulator requires only one SDP to be solved per channel use and uses a simplified randomization scheme, and hence the computational cost per channel use of the SDP, Cholesky and randomization steps is significantly reduced. The list-free implementation of the Single-SDR demodulator does not require storage of the list, but it may require some additional evaluations of the metric function.

In most practical implementations, the cost of solving the SDPs will be the dominant component of the computational cost of the SDR-based demodulators. As shown in the second column of Table IV, for the multi-SDR, List-SDR and Single-SDR demodulators, the computational costs per channel use of solving the SDPs are $O(TN_t^{4.5})$, $O(TN_t^{3.5})$, and $O(N_t^{3.5})$, respectively. To help place those costs in context, the computational cost per channel use of the direct form of the (conditional [45]) MMSE-SIC demodulator in [3] is $O(TN_t^4)$. The development of an efficient implementation of that demodulator that has a computational cost per channel use of $O(TN_t^3)$ was recently reported in [43].

VIII. SIMULATIONS

Insight from the approximation accuracy [35], [36] of the SDR approach to hard demodulation and the analysis of the computational cost in the previous section suggest that the proposed list-based SDR soft demodulators should offer good performance at a moderate computational cost, especially for larger systems. In this section, we seek to quantify that insight by comparing the performance and computational cost of the proposed demodulators with those of several existing demodulators. Since the proposed list-based demodulators employ SDR, a natural comparison is with the Multi-SDR demodulator [10], which invokes the "max log" approximation of the LLRs in (6) and uses SDR to approximate the solutions to the hard decision problems therein. We also make comparisons with two demodulators that have often been used as benchmarks, the list sphere decoder in [2], and the (conditional [45]) MMSE-SIC demodulator in [3]. For the MMSE-SIC demodulator, we employ the efficient implementation in [43].

We consider MIMO BICM systems that employ spatial multiplexing transmission of Gray-labeled QPSK symbols over a narrowband spatially uncorrelated Rayleigh fading channel; e.g., [2]. The outer codes are conventional punctured parallel concatenated turbo codes of rates 1/2 and 2/3. The rate-1/2 turbo code is that used in [2], whose constituent codes are the rate-1/2 recursive systematic convolutional code of constraint length 2 that has feedforward and feedback generator polynomials with octal representations 5 and 7, respectively. The rate-2/3 code is derived from the rate-1/2 code by further puncturing of the non-systematic bits. The input block length for the turbo codes is 8192 information bits, and the (different) interleavers in the turbo code and in the BICM transmitter are generated randomly in each Monte-Carlo iteration. The conventional BCJR algorithm [47] is used to decode the constituent codes, and eight turbo decoding iterations are performed before the extrinsic information is passed back to the demodulator. For the list-based demodulators (i.e., those proposed herein and the list sphere decoder), we adopt the common "clipping" approach to mitigating LLR estimation errors incurred by insufficiently rich lists, e.g., [6]. Based on the analysis in [6] and our own experiments, we clip the estimated LLRs to the interval $[-5, +5]$.

A. 8×8 MIMO System

In this section, we consider a MIMO system with $N_t = 8$ antennas at the transmitter, $N_r = 8$ antennas at the receiver, and a spatially uncorrelated Rayleigh fading channel model. We consider both an "ergodic" model, in which the channel realization changes independently at each channel use, and a quasi-static

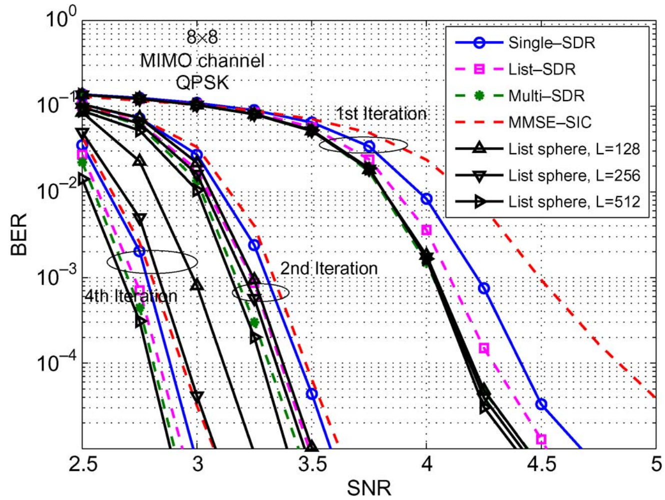


Fig. 3. Comparison of the BER performance of various demodulators for the 8×8 ergodic Rayleigh fading channel model and the rate-1/2 outer code.

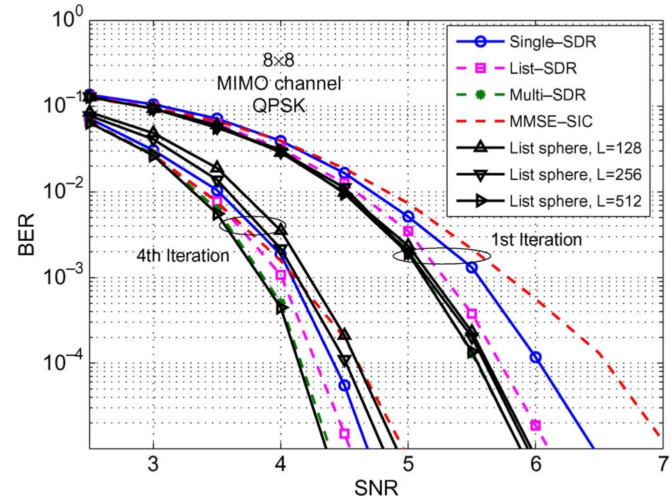


Fig. 4. Comparison of the BER performance of various demodulators for the 8×8 quasi-static Rayleigh fading channel model and the rate-1/2 outer code.

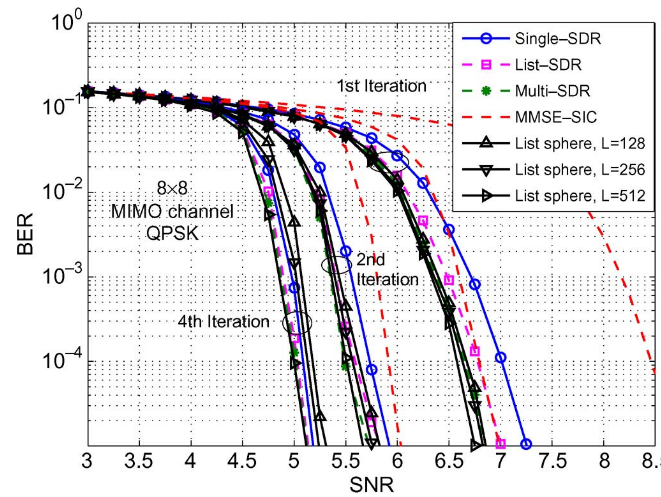


Fig. 5. Comparison of the BER performance of various demodulators for the 8×8 ergodic Rayleigh fading channel model and the rate-2/3 outer code.

model in which each outer codeword "sees" only one channel realization. (Although these models are somewhat idealized, they constitute useful limiting cases in the evaluation of demodulator performance.) We compare the BER performance of: 1) the Single-SDR demodulator; 2) the List-SDR demodulator; 3) the Multi-SDR demodulator in [10]; 4) three instances of the list sphere decoder in [2], one with the list size $L = 512$ that was considered in [2] and the others with $L = 256$ and $L = 128$, respectively; and 5) the MMSE-SIC demodulator in [3] with intrinsic information exchange [48]. Based on experiments reported in [49], the SDPs in the SDR-based demodulators were solved to an accuracy of $\epsilon = 10^{-2}$ and $M = 50$ randomizations were performed. For the List-SDR and Single-SDR demodulators we set $K = M$ and in the list enrichment step "bit-flipping" was performed on the best $J = 10$ members of the preliminary list. For all the list-based demodulators (the two proposed demodulators and the list sphere decoders) the LLRs were estimated using the "max-log" approximation in (6). If the approximation in (5) is used instead, the performance of these list demodulators improves a little, but their relative performance is similar.

First, we consider the case of the rate-1/2 outer code. The BERs of the considered demodulators under the ergodic channel model are shown in Fig. 3, from which it is apparent that after four iterations the BER of the Single-SDR demodulator is better than that of the list sphere decoder with $L = 128$, is slightly better than that of the list sphere decoder with $L = 256$ and the MMSE-SIC demodulator, and is close to that of the other demodulators. As shown in Fig. 4, the relative performance of the proposed demodulators under the quasi-static channel model is qualitatively similar.

For the case of the rate-2/3 outer code, the performance comparisons between the proposed demodulators and the list sphere decoder show similar trends, as illustrated by the results in Fig. 5 for the case of the ergodic channel model. However, with the higher rate outer code the relative performance of the MMSE-SIC is somewhat degraded. In particular, the other demodulators provide better performance after two iterations than the MMSE-SIC demodulator provides after four. In fact, ten iterations of the MMSE-SIC demodulator are required to obtain the performance of two iterations of the other demodulators.

An interesting feature of Figs. 3–5 is the degradation in the relative performance of the list sphere decoders with short lists as the number of demodulate-decode iterations increases. This is due in large part to the fact that the list sphere decoder in [2] generates its list once per channel use. In contrast, at each iteration the List-SDR and Single-SDR demodulators generate new lists that are adapted to the output of the decoder in the previous iteration. A key feature of the Single-SDR demodulator is that the new list is generated without the need to solve another SDP. Although there are a number of list-based tree-search demodulators that update their lists in each iteration (e.g., [4], [5], [7], and [16]), they must perform a tree search at each iteration, rather than at each channel use, and hence they may incur a larger computational cost than the benchmark list sphere decoder in [2]; see, e.g., [16].

Since the considered demodulators operate in substantially different manners, direct comparison of their computational costs can be rather awkward, but in order to provide one such comparison, we explicitly counted the number of floating point operations (FLOPs)[3] required by each demodulator to perform each component of its algorithm at each demodulation iteration in each channel use. For the SDR-based demodulators we used the straightforward primal-dual interior point algorithm in [37] to solve the SDPs, and we have included the FLOPs required to solve the SDPs and the FLOPs required to perform the Cholesky decompositions and the randomization steps, to manage the list, to compute the metrics $D(\mathbf{b})$, and to compute the max-log approximation on the list; cf. (6). For the list sphere decoder in [2] we have included the FLOPs required to construct the list (which is only performed once per channel use), including those required to perform the QR decomposition of the channel, and the FLOPs required to compute the metrics and the max-log approximation in each demodulation iteration. The list sphere decoder that we have considered is "genie-aided" in the sense that it is provided with an appropriate radius for the sphere (cf. [2]) at no computational cost. As such, the depicted results for the list sphere decoder may be somewhat optimistic. For the MMSE-SIC demodulator [3], we have used the efficient implementation in [43], and we count the FLOPs required to compute and subtract the mean of the interfering symbols, and those required to compute and implement the unbiased linear MMSE estimator of the resulting zero-mean signal.

In Fig. 6, we have plotted the average computational cost per channel use for $T = 4$ demodulation-decoding iterations of each of the considered demodulators against the SNR under the ergodic channel model for the case of the rate-2/3 outer code. As would be expected from the structure of the demodulators, the corresponding costs for the system with the rate-1/2 outer code over the SNRs of interest in that case are similar; cf. [49]. Fig. 6 shows that the average computational cost of our straightforward implementation of the Single SDR demodulator lies between that of the list sphere decoders of list sizes $L = 512$ and $L = 128$ and lies just below that of the list sphere decoders of list sizes $L = 256$, but is greater than that of the efficient implementation of the MMSE-SIC demodulator. (Recall that we have provided the list sphere decoders with an appropriate radius at no computational cost.) Furthermore, unlike the list sphere decoder, the distribution of the computational cost of the SDR demodulation methods is concentrated around the mean. To illustrate that fact, we have plotted in Fig. 7 the empirical complementary cumulative distribution of the computational cost per channel use of the considered demodulators at an SNR of 5 dB, which is in the "waterfall" region of the BER curves in Fig. 5.

The structure of the MMSE-SIC demodulator means that the computational cost of each iteration is the same. In contrast, for a list sphere decoder with typical list sizes the second and subsequent iterations each incur only a marginal computational cost because they use the list that was generated in the first iteration. Our explicit counting of the number of required operations verifies the earlier insight that the dominant component of the computational cost of the SDR-based demodulators is the cost of solving the SDP or SDPs. As a result, the cost of each subsequent iteration of the List-SDR and Multi-SDR demodulators is similar to that of the first, whereas the second and subsequent iterations of the Single-SDR demodulator each incur only a marginal computational cost.

[3]That is, each real-valued arithmetic operation. (For each demodulator, the number of additions and multiplications is quite similar.) Although the number of FLOPs is a reasonable metric for demodulators that are implemented on a general purpose sequential machine, this metric overlooks any structure in the algorithm that might facilitate efficient implementation in application-specific hardware.

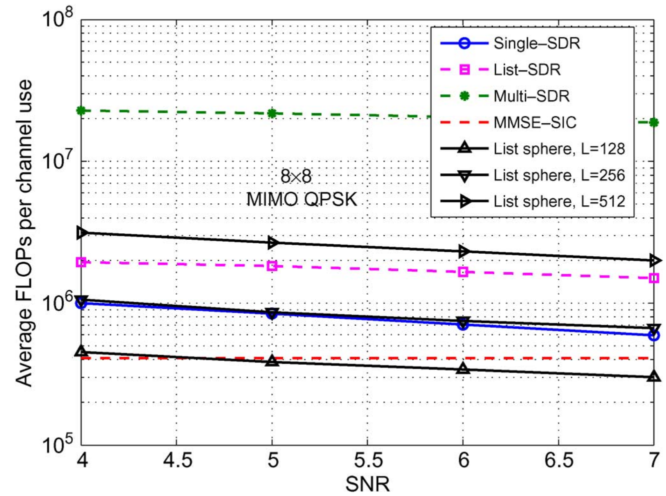


Fig. 6. Comparison of the average computational cost per channel use of the proposed demodulators and that of the Multi-SDR, list sphere decoding, and MMSE-SIC demodulators for the 8×8 ergodic Rayleigh fading channel model, the rate-2/3 outer code, and $T = 4$ demodulation-decoding iterations.

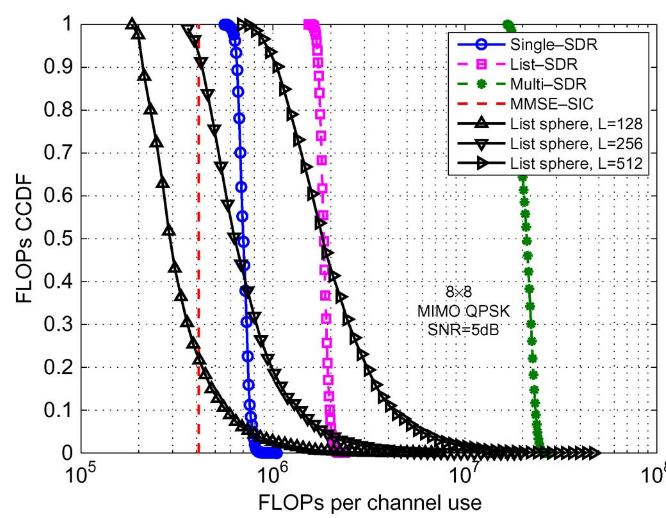


Fig. 7. Empirical complementary cumulative distribution of the number of FLOPs per channel use in the 8×8 ergodic Rayleigh fading channel model, the rate-2/3 outer code, and $T = 4$ demodulation and decoding iterations at an SNR of 5 dB.

B. 4×4 MIMO System

In this section, we illustrate that the tradeoffs between performance and computational cost achieved by the proposed demodulators remain competitive in smaller systems. We consider a MIMO-BICM system with $N_t = N_r = 4$ and the rate-1/2 outer code operating over the ergodic Rayleigh fading channel model. In this case, the full demodulation list has only 256 elements, and hence full-list demodulation can also be used as a benchmark. In the SDR demodulators we solved the SDPs to an accuracy of $\epsilon = 10^{-2}$, employed $M = 25$ randomizations

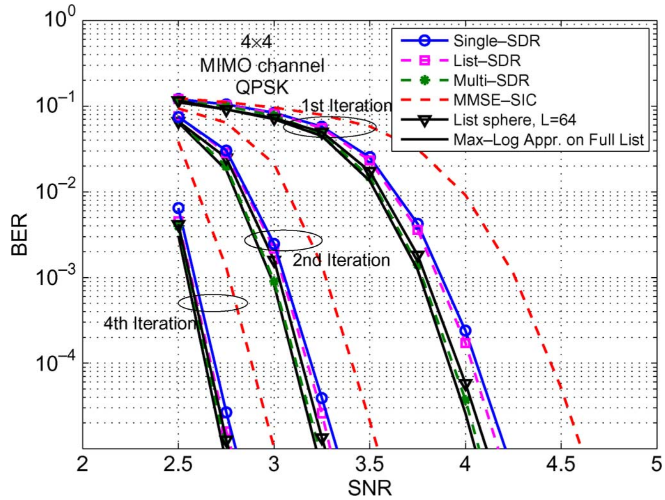


Fig. 8. Comparison of the BER performance of various demodulators for the 4×4 ergodic Rayleigh fading channel model and the rate-1/2 outer code.

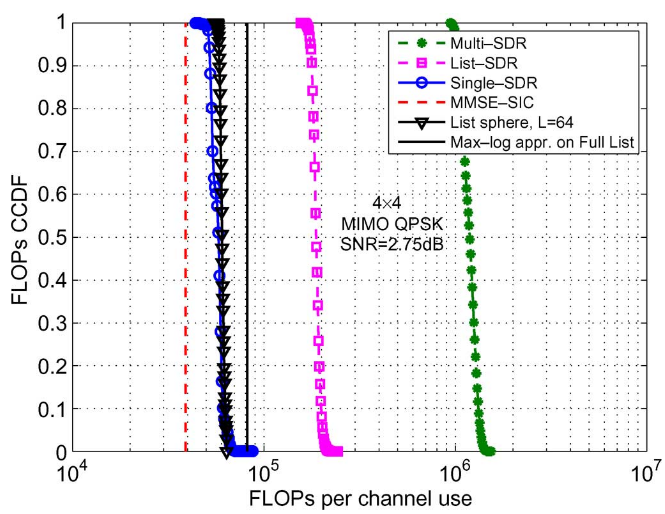


Fig. 9. Empirical cumulative distribution of the number of FLOPs per channel use in the 4×4 ergodic Rayleigh fading channel model, the rate-1/2 outer code, and $T = 4$ demodulation and decoding iterations at an SNR of 2.75 dB.

(with $K = M$), and chose $J = 5$ in the enrichment procedure. The average BERs of the various demodulators are plotted in Fig. 8. As in the 8×8 case, the performance of the Single-SDR demodulator is close to that of the best of the considered demodulators. However, we observe that in this 4×4 case, the relative performance of the MMSE-SIC demodulator is weaker than in the corresponding 8×8 case (cf. Fig. 3), due to the fact that there are fewer interfering symbols, and hence the inherent approximation that the residual interference is Gaussian is less accurate in this case.

A comparison of the average computational cost of the considered demodulators over a range of SNRs reveals similar trends to those depicted in Fig. 6 for the 8×8 case. This is apparent from Fig. 9, where we have plotted the empirical complementary cumulative distribution of the computational costs at an SNR of 2.75 dB. (Once again, we have provided the list sphere decoder with an appropriate radius at no computational cost.)

IX. Conclusion

In this paper, we have proposed two computationally efficient soft MIMO demodulators based on an adaptation of the SDR method for hard demodulation to list-based soft demodulation. These demodulators are designed for iterative receivers and at each iteration they regenerate the list, incorporating information from the previous iteration of the decoder. We have also presented a list-free implementation of the proposed demodulators that reduces the memory requirements of demodulators with moderate-to-long lists.

In contrast to the list sphere decoder, the (worst case) computational cost of the proposed demodulators is bounded by a (low-order) polynomial of the number of bits to be demodulated, and in contrast to the SDR-based demodulator in [10], one of the proposed demodulators requires the solution of one semidefinite program (SDP) per demodulation-decoding iteration for each channel use and the other requires the solution of only one SDP per channel use. Along with insight from the approximation accuracy of SDR [35], [36], these properties suggest that the proposed demodulators should offer good performance at moderate computational cost, especially for large systems. This was quantified by simulations of straightforward implementations of the demodulators in a richly scattered environment. In particular, the performance of the Single-SDR demodulator is similar to that of the list sphere decoder in [2] with moderate sized lists and better than that of the minimum mean square error soft interference canceler [3], [43]. Its average computational cost is competitive with that of the list sphere decoder with moderate sized lists, and the distribution of its computational cost is quite concentrated around the average. These results demonstrate the potential of the proposed demodulators and suggest that they are worthy of further algorithmic development and performance analysis.

The proposed SDR approach to list-based soft MIMO demodulation of QPSK symbols exploits the randomization step that is employed in the SDR approach to hard MIMO demodulation and multiuser detection; e.g., [30]. There are several ways in which the SDR approach to hard MIMO demodulation of QPSK symbols can be extended to hard demodulation of higher order QAM symbols [50]–[52]. By applying the principles discussed herein to those different extensions, a rich family of SDR approaches to list-based soft demodulation for MIMO-BICM systems with higher-order QAM signalling can be obtained; e.g., [49] and [53]. In the case of hard demodulation, several prominent SDR extensions have been shown to be equivalent [49], [52], and hence they are also equivalent in the context of a Single-SDR soft demodulator. However, in the context of a List-SDR soft demodulator these extensions enable different approximations of the prior information, and hence they provide different performance [53]. The evaluation of the performance-complexity tradeoffs within the family of SDR-based demodulators for higher order QAM is on going, and will be reported in due course.

Acknowledgment

The authors would like to thank J. Veloce of McMaster University for his assistance with some of the simulation experiments.

References

[1] I. E. Telatar, "Capacity of multiple antenna Gaussian channels," *Eur. Trans. Telecomm.*, vol. 10, pp. 585–595, Nov. 1999.

[2] B. M. Hochwald and S. ten Brink, "Achieving near capacity on a multiple-antenna channel," *IEEE Trans. Commun.*, vol. 51, no. 3, pp. 389–399, Mar. 2003.

[3] X. Wang and H. V. Poor, "Iterative (turbo) soft interference cancellation and decoding for coded CDMA," *IEEE Trans. Commun.*, vol. 47, no. 7, pp. 1046–1061, Jul. 1999.

[4] S. Bäro, J. Hagenauer, and M. Witzke, "Iterative detection of MIMO transmission using a list-sequential (LISS) detector," in *Proc. IEEE Int. Conf. Commun.*, Anchorage, AK, May 2003, vol. 4, pp. 2653–2657.

[5] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 2299–2311, Nov. 2004.

[6] Y. L. C. de Jong and T. J. Willink, "Iterative tree search detection for MIMO wireless systems," *IEEE Trans. Commun.*, vol. 53, no. 6, pp. 930–935, Jun. 2005.

[7] J. Hagenauer and C. Kuhn, "The list-sequential (LISS) algorithm and its application," *IEEE Trans. Commun.*, vol. 55, no. 5, pp. 918–928, May 2007.

[8] J. Jaldén and B. Ottersten, "Parallel implementation of a soft output sphere decoder," in *Proc. Asilomar Conf. Signal Syst. Comput.*, Monterey, CA, Oct. 2005, pp. 581–585.

[9] R. Wang and G. B. Giannakis, "Approaching MIMO channel capacity with soft detection based on hard sphere decoding," *IEEE Trans. Commun.*, vol. 54, no. 4, pp. 587–590, Apr. 2006.

[10] B. Steingrimsson, Z.-Q. Luo, and K. M. Wong, "Soft quasi-maximum-likelihood detection for multiple-antenna channels," *IEEE Trans. Signal Process.*, vol. 51, no. 11, pp. 2710–2719, Nov. 2003.

[11] D. J. Love, S. Hosur, A. Batra, and R. W. Heath, Jr., "Space-time Chase decoding," *IEEE Trans. Wireless Commun.*, vol. 4, no. 5, pp. 2035–2039, Sep. 2005.

[12] Z. Guo and P. Nilsson, "Algorithm and implementation of the K-best sphere decoding for MIMO detection," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 3, pp. 491–503, Mar. 2006.

[13] L. G. Barbero and J. S. Thompson, "Extending a fixed-complexity sphere decoder to obtain likelihood information for turbo-MIMO systems," *IEEE Trans. Veh. Technol.*, vol. 57, no. 5, pp. 2804–2814, Sep. 2008.

[14] D. L. Milliner, E. Zimmermann, J. R. Barry, and G. Fettweis, "A fixed-complexity smart candidate adding algorithm for soft-output MIMO detection," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 6, pp. 1016–1025, Dec. 2009.

[15] E. G. Larsson and J. Jaldén, "Fixed-complexity soft MIMO detection via partial marginalization," *IEEE Trans. Signal Process.*, vol. 56, no. 8, pp. 3397–3407, Aug. 2008.

[16] M. Nekuii and T. N. Davidson, "A multistack algorithm for soft MIMO demodulation," *IEEE Trans. Veh. Technol.*, vol. 58, no. 5, pp. 2592–2597, Jun. 2009.

[17] C. Studer and H. Bölcskei, "Soft-input soft-output single tree-search sphere decoding," *IEEE Trans. Inf. Theory*, vol. 56, no. 10, pp. 4827–4842, Oct. 2010.

[18] W. Zhang and X. Ma, "Low-complexity soft-output decoding with lattice-reduction-aided detectors," *IEEE Trans. Commun.*, vol. 58, no. 9, pp. 2621–2629, Sep. 2010.

[19] P. Fertl, J. Jaldén, and G. Matz, "Capacity-based performance comparison of MIMO-BICM demodulators," in *Proc. IEEE Workshop Signal Process. Adv. Wireless Commun.*, Recife, Brazil, Jul. 2008, pp. 166–170, see also http://arxiv.org/abs/0903.2711.

[20] D. Persson and E. G. Larsson, "Partial marginalization soft MIMO detection with higher order constellations," *IEEE Trans. Signal Process.*, vol. 59, no. 1, pp. 453–458, Jan. 2011.

[21] P. Robertson, E. Villebrun, and P. Hoeher, "A comparison of optimal and suboptimal map decoding algorithms operating in the log domain," in *Proc. IEEE Int. Conf. Commun.*, Seattle, WA, Jun. 1995, vol. 2, pp. 1009–1013.

[22] A. D. Murugan, H. El 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.

[23] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[24] 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.

[25] J. Jaldén 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.

[26] W. K. Ma, B. N. Vo, T. N. Davidson, and P. C. Ching, "Blind ML detection of orthogonal space-time block codes: Efficient high-performance implementations," *IEEE Trans. Signal Process.*, vol. 54, no. 2, pp. 738–751, Feb. 2006.

[27] R. Gowaikar and B. Hassibi, "Statistical pruning for near-maximum likelihood decoding," *IEEE Trans. Signal Process.*, vol. 55, no. 6, pp. 2661–2675, Jun. 2007.

[28] L. G. Barbero and J. S. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Trans. Wireless Commun.*, vol. 7, no. 6, pp. 2131–2142, Jun. 2008.

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

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

[31] 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.

[32] J. Jaldén and B. Ottersten, "The diversity order of the semidefinite relaxation detector," *IEEE Trans. Inf. Theory*, vol. 54, no. 4, pp. 1406–1422, Apr. 2008.

[33] 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.

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

[35] Y. E. Nesterov, "Semidefinite relaxation and nonconvex quadratic optimization," *Optim. Methods Softw.*, vol. 9, pp. 141–160, 1998.

[36] M. Kisialiou and Z.-Q. Luo, "Probabilistic analysis of semidefinite relaxation for binary quadratic minimization," *SIAM J. Optim.*, vol. 20, no. 4, pp. 1906–1922, 2010.

[37] C. Helmberg, F. Rendl, R. Vanderbei, and H. Wolkowicz, "An interior point method for semidefinite programming," *SIAM J. Optim.*, vol. 6, no. 2, pp. 342–361, 1996.

[38] G. J. Foschini, G. Golden, R. Valenzuela, and P. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Sel. Areas Commun.*, vol. 17, no. 11, pp. 1841–1852, Nov. 1999.

[39] B. Hassibi and B. Hochwald, "High-rate codes that are linear in space and time," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1804–1824, Jul. 2002.

[40] J. Hagenauer, "The turbo principle: Tutorial introduction and state of the art," in *Proc. Int. Symp. Turbo Codes Rel. Topics*, Brest, France, Sep. 1997, pp. 1–11.

[41] H. El-Gamal and E. Geraniotis, "Iterative multiuser detection for coded CDMA signals in AWGN and fading channels," *IEEE J. Sel. Areas Commun.*, vol. 18, no. 1, pp. 30–41, Jan. 2000.

[42] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inf. Theory*, vol. 48, no. 7, pp. 1772–1793, Jul. 2002.

[43] C. Studer, S. Fateh, and D. Seethaler, "ASIC implementation of soft-input soft-output MIMO detection using MMSE parallel interference cancellation," *IEEE J. Solid-State Circuits*, vol. 46, no. 7, pp. 1754–1765, Jul. 2011.

[44] M. Tüchler, A. C. Singer, and R. Koetter, "Minimum mean squared error equalization using *a priori* information," *IEEE Trans. Signal Process.*, vol. 50, no. 3, pp. 673–683, Mar. 2002.

[45] G. Caire, R. R. Müller, and T. Tanaka, "Iterative multiuser joint decoding: Optimal power allocation and low-complexity implementation," *IEEE Trans. Inf. Theory*, vol. 50, no. 9, pp. 1950–1973, Sep. 2004.

[46] T. H. Cormen, C. F. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.

[47] L. Bahl, J. Coke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. IT-20, no. 3, pp. 284–287, Mar. 1974.

[48] M. Witzke, S. Bäro, F. Schreckenbach, and J. Hagenauer, "Iterative detection of MIMO signals with linear detectors," in *Proc. Conf. Rec. 36th Asilomar Conf. Signals, Syst., Comput.*, Pacific Grove, CA, Nov. 2002, pp. 289–293.

[49] M. Nekuii, "Soft demodulation schemes for MIMO communication systems," Ph.D. dissertation, McMaster Univ., Montreal, QC, Canada, Aug. 2008.

[50] 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, Sep. 2005.

[51] 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, Sep. 2006.

[52] 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, Dec. 2009.

[53] M. Nekuii and T. N. Davidson, "A semidefinite relaxation approach to efficient soft demodulation of MIMO 16-QAM," in *Proc. IEEE Int. Conf. Commun.*, Dresden, Jun. 2009, pp. 1–6.

Mehran Nekuii (S'04–M'09) received the B.Sc. and M.Sc. degrees in electrical engineering from Sharif University of Technology, Tehran, Iran, in 2000 and 2003, respectively, and the Ph.D. degree in electrical and computer engineering from McMaster University, Hamilton, ON, Canada, in 2008.

He is now working as a Scientist in the Wireless Broadband Group, Cavium, Inc., Montreal. His research interests are signal processing in communications systems, optimization techniques in MIMO systems, and physical layer in OFDM-based communication systems.

Mikalai Kisialiou received the B.Sc. and M.Sc. degrees (with distinction) in applied mathematics and physics from the Moscow Institute of Physics and Technology, Moscow, Russia, in 2001 and the Ph.D. degree in electrical engineering from the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis MN, in 2007.

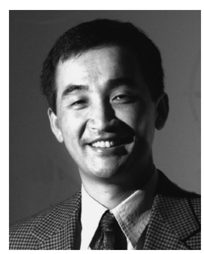
In 2001, he started his Ph.D. program in the Department of Electrical and Computer Engineering, McMaster University, Hamilton ON, Canada. After graduation, he joined the Circuit Simulation Group in the Department of Design and Technology Solutions at Intel Corporation, Hillsboro OR, United States. His research interests include optimization theory, signal processing, wireless communications, advanced numerical methods, algorithm analysis and efficient software implementation.

Dr. Kisialiou received the ICASSP'05 Best Student Paper Award for his work on theoretical analysis of semidefinite relaxation applied to maximum-likelihood detection.

Timothy N. Davidson (M'96) received the B.Eng. (Hons. I) degree in electronic engineering from the University of Western Australia (UWA), Perth, in 1991 and the D.Phil. degree in engineering science from the University of Oxford, Oxford, U.K., in 1995.

He is currently a Professor in the Department of Electrical and Computer Engineering, McMaster University, Hamilton, ON, Canada, where he holds the (Tier II) Canada Research Chair in Communication Systems and is currently serving as Associate Director of the School of Computational Engineering and Science. His research interests lie in the general areas of communications, signal processing, and control.

Dr. Davidson received the 1991 J. A. Wood Memorial Prize from UWA, and the 1991 Rhodes Scholarship for Western Australia. He has served as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS II. He has also served as a Guest Co-Editor of issues of the IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS and the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING. Dr. Davidson is currently serving as Vice-Chair of the IEEE Signal Processing Society's Technical Committee on Signal Processing for Communications and Networking. He is a Registered Professional Engineer in the Province of Ontario.



Zhi-Quan Luo (F'07) received the B.Sc. degree in applied mathematics from Peking University, Beijing, China, in 1984 and the Ph.D. degree in operations research from the Massachusetts Institute of Technology, Cambridge, in 1989.

He is a Professor in the Department of Electrical and Computer Engineering, University of Minnesota, Minneapolis, where he holds an endowed ADC Chair in digital technology. From 1989 to 2003, he was with the Department of Electrical and Computer Engineering, McMaster University, Montreal, QC, Canada, where he later served as the Department Head and held a senior Canada Research Chair in Information Processing. His research interests lie in the union of optimization algorithms, data communication, and signal processing.

Dr. Luo is a fellow of SIAM. He is a recipient of the IEEE Signal Processing Society's Best Paper Award in 2004 and 2009, the EURASIP Best Paper Award and the ICC's Best Paper Award in 2011. He was awarded the 2010 Farkas Prize from the INFORMS Optimization Society. He currently chairs the IEEE Signal Processing Society's Technical Committee on Signal Processing for Communications and Networking (SPCOM). He has held editorial positions for several international journals including the *Journal of Optimization Theory and Applications*, *Mathematics of Computation*, IEEE TRANSACTIONS ON SIGNAL PROCESSING, *SIAM Journal on Optimization*, *Management Sciences* and *Mathematics of Operations Research*.