EFFICIENT SOFT DEMODULATION OF MIMO QPSK VIA SEMIDEFINITE RELAXATION[1]

Mehran Nekuii[*] *Mikalai Kisialiou*[†] *Timothy N. Davidson*[*] *Zhi-Quan Luo*[†]

[*]Dept. Electrical and Computer Eng.
McMaster University
Hamilton, ON, Canada

[†]Dept. Electrical and Computer Eng.
University of Minnesota
Minneapolis, MN, USA

ABSTRACT

We develop a computationally efficient and memory efficient approach to (near) maximum *a posteriori* probability demodulation for MIMO systems with QPSK signalling, based on semidefinite relaxation. Existing approaches to this problem require either storage of a large list of candidate bit-vectors, or the solution of multiple binary quadratic problems. In contrast, the proposed demodulator does not require the storage of a candidate list, and involves the solution of a single (efficiently solvable) semidefinite program per channel use. Our simulation results show that the resulting computational and memory efficiencies are obtained without incurring a significant degradation in performance.

Index Terms— MIMO soft demodulation; bit-interleaved coded modulation; iterative demodulation and decoding; semidefinite relaxation

1. INTRODUCTION

Wireless communication systems with multiple transmit and multiple receive antennas have the potential to provide data rates that are substantially higher than those of the single antenna systems. The core challenge in designing these systems is to achieve these rates with reasonable computational complexity. A standard transceiver architecture for moving towards that goal is to use an outer binary code, an interleaver, and a multiple-input multiple-output (MIMO) modulator at the transmitter, and iterative "soft" demodulation and decoding (IDD) at the receiver. That is, MIMO bit-interleaved coded modulation (BICM), e.g., [1]; see also Fig. 1. However, the computational cost of the maximum *a posteriori* probability (MAP) soft demodulator increases exponentially with the number of bits transmitted per channel use, and hence there has been considerable interest in the design of approximate (quasi-MAP) soft demodulation schemes with lower complexity; e.g., [1–8].

One approach to quasi-MAP soft demodulation is to apply the so-called "max-log" approximation to the log-likelihood ratio. Doing so converts the soft demodulation problem into a set of "hard" demodulation problems (e.g., [4–6]), but each of these hard demodulation problems is, in itself, a binary quadratic problem, and hence is difficult to solve exactly. Tree search methods (e.g., [9]), such as sphere decoding, can be used to find solutions to these problems (e.g., [4,5]), but both the average and worst-case computational complexities remain exponential in the problem size and can be a serious impediment to implementation at low SNRs and for large problems [10]. Semidefinite relaxation methods can also be used [6], and they have the advantage that the growth of the computational cost is bounded by a low-order polynomial in the problem size. (The implementation in [11] has a number of desirable features.)

An alternative approach to quasi-MAP soft demodulation is to apply the principles of list decoding, in which one seeks to efficiently identify a list of bit-vectors that dominate the log-likelihood ratios that are computed by the MAP demodulator. Most of the existing list-based approaches have used tree search algorithms to identify the list members (e.g., [1–3]), but as mentioned above, these algorithms can be rather costly at low SNRs and for large problems. Furthermore, the size of the list required for accurate approximation of the MAP demodulator means that these methods have significant memory requirements. An alternative to the use of tree search algorithms for generating the list is to use the randomization procedure implicit in the semidefinite relaxation approach to hard demodulation [7]. While the complexity of that approach is polynomial in the problem size, the memory required to store the list can still be significant.

In this paper, we will develop an alternative quasi-MAP demodulator based on semidefinite relaxation (SDR). The proposed demodulator has the advantages that: i) the complexity is bounded by a low-order polynomial in the problem size; ii) it requires only one semidefinite program (SDP) to be solved for each channel use; and iii) it does not require the storage of a list of candidate bit-vectors. The proposed demodulator operates as follows: The first step is to solve the SDP that one would solve if one were to perform "hard" demodulation from the channel output via SDR; e.g., [11]. Rather than using that solution to (approximately) solve the hard demodulation problem, we use it to generate a vector of probabilities that describe the randomization procedure implicit in hard demodulation via SDR [12, 13]. This vector of probabilities is combined with the prior probabilities from the preceding iteration of the decoder, and the combined probabilities are then used to generate a sequence of candidate bit vectors via independent scalar Bernoulli trials. By updating its approximation of the soft output in sequence, the proposed demodulator can operate without having to store these candidate bit vectors in a list. Our simulation results will show that the resulting computational and memory efficiencies are obtained without incurring a significant degradation in performance.

2. SYSTEM MODEL

We consider a narrow-band multiple antenna system with N_t transmit antennas, N_r receive antennas, and transmitted symbol vector $\mathbf{s}$. The received vector $\mathbf{y}$ can be written as:

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

where $\mathbf{H}$ is the $N_r \times N_t$ matrix of channel coefficients and is assumed to be known at the receiver, and $\mathbf{v}$ is a vector of additive white circular complex Gaussian noise samples with variance σ^2 per real dimension. We will consider a MIMO-BICM transmission scheme with iterative demodulation and decoding (e.g., [1]),

[1]This work was supported in part by NSERC and the NSF (DMS-0610037). The work of the third author is also supported by the Canada Research Chairs program.

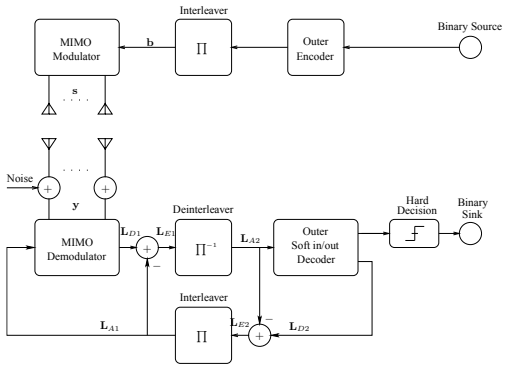

Fig. 1. MIMO BICM-IDD transceiver.

in which bit vectors $\mathbf{b}$ selected from the encoded and interleaved bit stream are mapped to transmitted symbol vectors $\mathbf{s}$ using the mapping $\mathbf{s} = \mathcal{M}(\mathbf{b})$; see Fig. 1. For simplicity we will focus on V-BLAST transmission of QPSK symbols, but extensions to more general linear dispersion codes can follow from [14], and extensions to higher-order constellations can be developed from the semidefinite relaxation approaches to hard demodulation in [15] and [16].

We will consider the standard "soft" iterative demodulation and decoding architecture for the receiver; e.g., [1]. Since binary outer codes are used, the role of the "soft demodulator" is to compute the posterior log-likelihood ratio (LLR) of each interleaved encoded bit. By capturing in a vector $\boldsymbol{\lambda}$ the extrinsic soft information on the elements of $\mathbf{b}$ from the decoder, these LLRs can be written as (e.g., [1])

$$L_i = \log \frac{\Sigma_{\mathcal{L}_{i,+1}} \exp(-D(\mathbf{b})/(2\sigma^2))}{\Sigma_{\mathcal{L}_{i,-1}} \exp(-D(\mathbf{b})/(2\sigma^2))}, \tag{2}$$

where $\mathcal{L} = \{\mathbf{b} \in \{-1,+1\}^{2N_t}\}$ denotes the (complete) list of possible transmitted bit-vectors, $\mathcal{L}_{i,\pm 1} = \{\mathbf{b} \in \mathcal{L} | b_i = \pm 1\}$ and

$$D(\mathbf{b}) \triangleq \|\mathbf{y} - \mathbf{H}\mathcal{M}(\mathbf{b})\|_2^2 - \sigma^2 \boldsymbol{\lambda}^T \mathbf{b}. \tag{3}$$

It can be seen from (2) that as the number of bits per channel use increases, the computational cost of (2) increases exponentially. The standard approaches to reducing the complexity of the soft demodulator involve the use one of the following approximations:

$$L_i \simeq \log \frac{\Sigma_{\hat{\mathcal{L}}_{i,+1}} \exp(-D(\mathbf{b})/(2\sigma^2))}{\Sigma_{\hat{\mathcal{L}}_{i,-1}} \exp(-D(\mathbf{b})/(2\sigma^2))} \tag{4}$$

$$\simeq \frac{1}{2\sigma^2}\Big(\min_{\mathbf{b}\in\hat{\mathcal{L}}_{i,-1}} D(\mathbf{b}) - \min_{\mathbf{b}\in\hat{\mathcal{L}}_{i,+1}} D(\mathbf{b})\Big), \tag{5}$$

where $\hat{\mathcal{L}} \subseteq \mathcal{L}$. These equations reveal two general classes of approximate soft demodulators. The first is based on selecting $\hat{\mathcal{L}} = \mathcal{L}$ and solving the problems in (5) using the direct applications of "hard" demodulation techniques. This requires the solution of two binary quadratic problems for each bit. Each of these solutions can be obtained using a tree search algorithm (e.g., [4, 5]), or can be approximated by semidefinite relaxation (e.g., [6]). The second class of soft demodulators is based on efficiently selecting a list $\hat{\mathcal{L}}$ of bit-vectors with small values of $D(\mathbf{b})$ and then approximating the LLR using either marginalization over $\hat{\mathcal{L}}_{i,\pm 1}$ in (4) (e.g., [3]), or by performing an exhaustive search over $\hat{\mathcal{L}}_{i,\pm 1}$ to solve the minimization problems in (5), (e.g., [1]). Often, methods in the second class are more computationally efficient than those in the first, but they may require significant memory resources to store the generated list members.

The proposed demodulator falls into the second class, in the sense that it generates a sequence of candidate bit-vectors, but it does not store a list. The value of $D(\mathbf{b})$ for each bit-vector is computed as it is generated, and only two real vectors are stored. (These vectors store the smallest obtained values for the minimization problems in (5).) The candidate bit-vectors are generated by Bernoulli trials, the probabilities of which are determined from the solution of a single SDP per channel use, and from the extrinsic information from the previous iteration of the decoder. The single SDP arises from the semidefinite relaxation of a binary quadratic problem, and is described in the next section.

3. SOLVING BINARY QUADRATIC PROBLEMS VIA SDR

For simplicity, we consider the real-valued equivalent model for (1) where QPSK signalling is used: $\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\mathbf{b} + \tilde{\mathbf{v}}$, where $\tilde{\mathbf{y}}$ and $\tilde{\mathbf{v}}$ are concatenations of the real and imaginary parts of $\mathbf{y}$ and $\mathbf{v}$ in (1), respectively. The minimization of $D(\mathbf{b})$ can be formulated as

$$\min_{\mathbf{b}\in\{+1,-1\}^{2N_t}} D(\mathbf{b}) = \min_{\tilde{\mathbf{b}}\in\{+1,-1\}^{2N_t+1}} \tilde{\mathbf{b}}^T \mathbf{Q} \tilde{\mathbf{b}}, \tag{6}$$

where

$$\tilde{\mathbf{b}} \triangleq \begin{bmatrix} \check{\mathbf{b}} \\ c \end{bmatrix},\ \check{\mathbf{b}} \triangleq c\mathbf{b},\ \mathbf{Q} \triangleq \begin{bmatrix} \tilde{\mathbf{H}}^T\tilde{\mathbf{H}} & \mathbf{a} \\ \mathbf{a}^T & 0 \end{bmatrix},\ \mathbf{a} \triangleq -\tilde{\mathbf{H}}^T\tilde{\mathbf{y}} - 0.5\sigma^2\boldsymbol{\lambda},$$

and $c \in \{+1, -1\}$. (In the proposed demodulator, this problem will be solved only at the first iteration, and hence $\boldsymbol{\lambda} = \mathbf{0}$.) To construct an approximate solution to (6), we first solve the following SDP, which is a (matrix) relaxation of (6), [12, 13],

$$\min \quad \text{Trace}(\mathbf{XQ}) \tag{7a}$$

$$\text{s.t.} \quad \mathbf{X} \succeq 0, \quad \mathbf{X}_{ii} = 1,\ i = 1, \ldots, 2N_t + 1. \tag{7b}$$

The standard randomization procedure [12, 13] can then be used to extract an approximation of the solution to (6) from the optimal solution to (7), $\mathbf{X}_{\text{opt}}$. That procedure involves the construction of a (Cholesky) factor $\mathbf{V}$ of $\mathbf{X}_{\text{opt}}$, (i.e., $\mathbf{X}_{\text{opt}} = \mathbf{V}^T\mathbf{V}$), and the generation of a sequence of random vectors $\mathbf{u}$ from the uniform distribution on the unit hypersphere. We set $\tilde{\mathbf{x}} \triangleq \text{sign}(\mathbf{V}^T\mathbf{u})$ and define the vector $\hat{\mathbf{x}} \triangleq [\tilde{\mathbf{x}}]_{2N_t+1}\,\tilde{\mathbf{x}}$ to ensure that the last entry is $+1$. We repeat this procedure and at each step we retain the bit-vector $\mathbf{b} = [\hat{\mathbf{x}}]_{1:2N_t}$ that generates the smallest value for $D(\mathbf{b})$. A key feature of the SDR approach is that the expected value of $D(\mathbf{b})$ over the randomizations is guaranteed to be within a (reasonably small) constant factor of the optimal value of (6), [13]. Hence, the generated bit-vectors appear to be good candidates for constructing the list $\hat{\mathcal{L}}$ used to approximate the LLR values in (4) or (5), [7]. An interesting property of the randomization procedure is that an analytic expression is available for the mean value of each element of $\hat{\mathbf{x}}$. In particular, if we let $\mathbf{m}$ denote the vector of means, and if we denote the $2N_t + 1$ columns of $\mathbf{V}$ by $\mathbf{v}_k$, then [12, 13]

$$[\mathbf{m}]_k = \frac{2}{\pi} \arcsin\big(\mathbf{v}_k^T \mathbf{v}_{2N_t+1}\big). \tag{8}$$

4. ITERATIVE QUASI-MAP DEMODULATOR

The first observation in the development of the proposed demodulator is that the expression in (8) suggests that for the purposes of

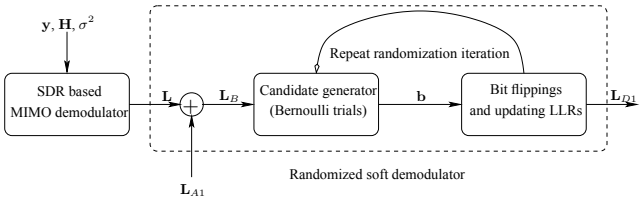

Fig. 2. The proposed quasi-MAP demodulator.

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 each element of $\hat{\mathbf{x}}$ is independent, and generating each element of $\hat{\mathbf{x}}$ via a scalar (antipodal) Bernoulli trial. This approach avoids 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 $\boldsymbol{\lambda}$). The expression for $D(\mathbf{b})$ in (3) suggests that the demodulation procedure needs to be repeated at each iteration (as it is in [2, 3, 7]), but 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 quasi-MAP soft demodulator is illustrated in Fig. 2. It consists of an SDR demodulator (which is invoked only in the first iteration), and a randomized soft demodulator. The randomized soft demodulator takes two inputs: (i) a vector $\mathbf{L}$ containing the mean values in (8) in LLR form, i.e., $[\mathbf{L}]_k = \log\big((1+[\mathbf{m}]_k)/(1-[\mathbf{m}]_k)\big)$; and (ii) the vector $\mathbf{L}_{A1}$ containing the extrinsic information (in LLR form) from the previous iteration of the decoder, see Fig. 2. The randomized demodulator then computes a Bernoulli distribution that reflects these inputs, and generates a sequence of random binary vectors according to that distribution. Each vector, and its neighbors, is then used to (possibly) update the demodulator's output.

By construction, the extrinsic information provided by the decoder is independent of the soft information from the channel [17]. 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

$$\mathbf{L}_B = \mathbf{L} + \mathbf{L}_{A1}. \tag{9}$$

The k-th entry of the corresponding mean vector $\mathbf{m}_B$ is $[\mathbf{m}_B]_k = 1 - 2/(1+\exp([\mathbf{L}_B]_k))$. Having computed $\mathbf{m}_B$, the randomized demodulator generates a sequence of M bit-vectors according to the corresponding Bernoulli distribution. The value of $D(\mathbf{b})$ for each bit-vector is computed in sequence, and the smallest encountered values for the two problems in (5) are stored. Using insight from [8], before evaluating the next randomly generated bit-vector, we evaluate the neighbors of the current bit-vector; i.e., all those vectors within a given Hamming distance. In this paper we will consider those vectors at a Hamming distance of one, as they can be generated by simply flipping one of the bits in the current bit-vector. For convenience, we have summarized the algorithm for the proposed randomized soft demodulator in Table 1.

Table 1. Algorithm for randomized soft demodulator

- Data: $\mathbf{L}$, $\mathbf{L}_{A1}$, M
- Output: $\mathbf{L}_{D1}$

1. Compute $\mathbf{L}_B$ in (9) and subsequently $\mathbf{m}_B$.
2. Initialize $\mathbf{f}_{+1} = \{+\infty\}^{2N_t}$, $\mathbf{f}_{-1} = \{+\infty\}^{2N_t}$, $m = 0$
3. If $m = 0$, set $\mathbf{b} = \text{sign}(\mathbf{m}_B)$. Otherwise, generate each $[\mathbf{b}]_k$ independently according to the (antipodal) Bernoulli distribution with mean $[\mathbf{m}_B]_k$.
4. Compute $D(\mathbf{b})$. For each $k = 1, 2, \ldots, 2N_t$, if $\mathbf{b}_k = 1$ then set $[\mathbf{f}_{+1}]_k = \min\{[\mathbf{f}_{+1}]_k, D(\mathbf{b})\}$, else set $[\mathbf{f}_{-1}]_k = \min\{[\mathbf{f}_{-1}]_k, D(\mathbf{b})\}$.
5. For each $i = 1, 2, \ldots, 2N_t$, set $\check{\mathbf{b}}^{(i)} = \mathbf{b}$ and then $[\check{\mathbf{b}}^{(i)}]_i = -[\mathbf{b}]_i$. Repeat Step 4 for $\check{\mathbf{b}}^{(i)}$.
6. Increment m. If $m < M$ return to 3. Otherwise, return $\mathbf{L}_{D1} = \big(\mathbf{f}_{+1} - \mathbf{f}_{-1}\big)/(2\sigma^2)$.

As mentioned in the introduction, an advantage of SDR methods over tree search methods, such as sphere decoding, is their polynomial (worst-case) complexity. An advantage of the particular SDR-based soft demodulator that we have proposed here in is that its computational (and memory) costs are lower than those of existing SDR-based soft demodulators. In particular, if we let ϵ denote the accuracy to which the SDP is solved, and if we perform K demodulation-decoding iterations, then the computational cost per channel use of the proposed demodulator is

$$O(N_t^{3.5}\log\epsilon^{-1}) + O(KMN_t^3),$$

where the first term represents the complexity of solving the SDP and the second term represents the cost of computing $D(\mathbf{b})$ for each bit-vector generated in the randomization step. In contrast, the computational costs of the List-SDR method [7] and the multi-SDR method [6] are $O(KN_t^{3.5}\log\epsilon^{-1}) + O(KMN_t^3)$ and $O(KN_t^{4.5}\log\epsilon^{-1}) + O(KMN_t^3)$, respectively. These expressions obscure the fact that the generation of the candidate bit-vectors in the randomization procedure of the proposed method is somewhat simpler than that in the List-SDR and multi-SDR methods, but they do reveal the computational advantage of only solving one SDP per channel use. In addition, the proposed method does not require memory to be allocated to store a list of bit-vectors, as is required by the List-SDR method. Since it does not require this memory allocation, the proposed demodulator can dynamically adjust its performance-complexity tradeoff (e.g., by adjusting ϵ and M) in response to changes in the latency requirements or variations in the channel SNR.

In the form of the algorithm in Table 1, the compromise that we have made in order to avoid the need to store the list is that there may be some repetitions of the bit-vectors generated by the Bernoulli trial in Step 3, and hence repeated computation of $D(\mathbf{b})$ in Steps 4 and 5. However, hashing techniques can be used to avoid this repeated computation while requiring only a small amount of additional memory (at most M integers) and a computational cost of only $O(KMN_t)$ per channel use.

5. SIMULATION RESULTS

In this section we compare the performance of the proposed "single SDR" demodulator with several existing demodulators. We consider a MIMO BICM system with V-BLAST transmission of QPSK over a MIMO i.i.d. Rayleigh block fading channel. The transceiver parameters, including those of the outer (Turbo) codes and the iterative demodulation and decoding algorithm are chosen from those used in [1, 3]. We will compare the performance of the proposed

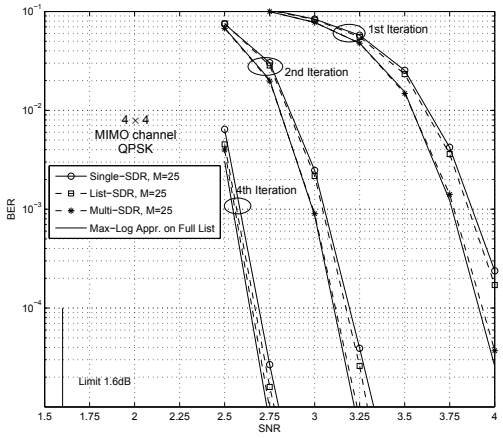

Fig. 3. BER performance for a 4×4 MIMO system

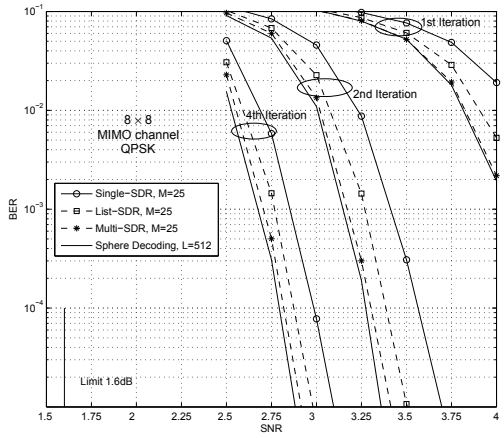

Fig. 4. BER performance for an 8×8 MIMO system

demodulator with that of the following methods: i) the Multi-SDR method [6], which solves several hard demodulation problems in each demodulation iteration to approximate the max-log problems in (5), ii) the List-SDR method [7], which solves one SDR per demodulation iteration and generates candidate bit-vectors using standard randomization iterations, and iii) the sphere decoding method of [3].

For the outer code we use a rate $1/2$ punctured parallel concatenated turbo code with the $(5, 7)$ recursive systematic convolutional code as the component codes and an (input) block length of 8192. We use the conventional BCJR algorithm to decode the constituent convolutional codes of the turbo code, and 8 turbo decoding iterations are performed before we pass extrinsic information back to the demodulator. Figs 3 and 4 compare the bit error rate (BER) performance of these methods after 1, 2 and 4 demodulation-decoding iterations for 4×4 and 8×8 MIMO systems, respectively. In the 4×4 set up the full list size is 256, and instead of using the sphere decoder we perform the max-log approximation on the full list. Each of these curves demonstrates that the proposed soft demodulator provides BER performance close to that of the other methods. To complement our previous analysis of the rate of growth of the computational cost, we explicitly computed the average number of floating point operations (FLOPs) per channel use (Fpcu) for the 8×8 case in Fig. 4 at an SNR of 2.75dB. The Single-SDR method required around 1.2×10^5 Fpcu, the List-SDR method required 4.1×10^5 Fpcu, and the Multi-SDR and Sphere decoding methods required around 5.7×10^6 and 2.5×10^6 Fpcu, respectively.

6. REFERENCES

[1] 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.

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

[3] 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.

[4] J. Jaldén and B. Ottersten, "Parallel implementation of a soft output sphere decoder," in *Proc. Asilomar Conf. Sig. Sys. Comp.*, Monterey, CA, Oct. 2005.

[5] 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.

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

[7] M. Nekuii and T. N. Davidson, "List based soft demodulation of MIMO QPSK via semidefinite relaxation," in *Proc. IEEE Workshop on Signal Processing Advances in Wireless Communications*, Helsinki, Finland, June 2007.

[8] 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, Sept. 2005.

[9] 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. Inform. Theory*, vol. 52, pp. 933–953, Mar. 2006.

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

[11] M. Kisialiou and Z.-Q. Luo, "Efficient implementation of a quasi-maximum-likelihood detector based on semi-definite relaxation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Honolulu, Apr. 2007, vol. IV, pp. 1329–1332.

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

[13] Y. E. Nesterov, "Quality of semidefinite relaxation for nonconvex quadratic optimization," Tech. Rep. CORE, Universite Catholique de Louvain, Brussels, Belgium, Mar. 1997.

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

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

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

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