

**SOFT DEMODULATION SCHEMES FOR  
MIMO COMMUNICATION SYSTEMS**

SOFT DEMODULATION SCHEMES FOR MIMO  
COMMUNICATION SYSTEMS

By

MEHRAN NEKUII, B.Sc., M.Sc.

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL & COMPUTER  
ENGINEERING

AND THE SCHOOL OF GRADUATE STUDIES

OF MCMASTER UNIVERSITY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

McMaster University

© Copyright by Mehran Nekuii, August 2008

All Rights Reserved

Doctor of Philosophy (2008)  
(Electrical & Computer Engineering)

McMaster University  
Hamilton, Ontario

TITLE: Soft demodulation schemes for MIMO  
communication systems

AUTHOR: Mehran Nekuii  
B.Sc., M.Sc. (Electrical Engineering)  
Sharif University of Technology, Tehran, Iran

SUPERVISOR: Timothy N. Davidson, Associate Professor

NUMBER OF PAGES: xxii, 214

## **Dedications**

*To my wonderful wife*

*To my mother and father*

*To Peyman and Ladan*

# Abstract

In this thesis, several computationally-efficient approximate soft demodulation schemes are developed for multiple-input multiple-output (MIMO) communication systems. These soft demodulators are designed to be deployed in the conventional iterative receiver ('turbo') architecture, and they are designed to provide good performance at substantially lower computational cost than that of the exact soft demodulator. The proposed demodulators are based on the principle of list demodulation and can be classified into two classes, according to the nature of the list-generation algorithm. One class is based on a tree-search algorithm and the other is based on insight generated from the analysis of semidefinite relaxation techniques for hard demodulation.

The proposed tree-search demodulators are based on a multi-stack algorithm, developed herein, for efficiently traversing the tree structure that is inherent in the MIMO demodulation problem. The proposed scheme was inspired, in part, by the stack algorithm, which stores all the visited nodes in the tree in a single stack and chooses the next node to expand based on a 'best-first' selection scheme. The proposed algorithm partitions this global stack into a stack for each level of the tree. It examines the tree in the natural ordering of the levels and performs a best-first search in each of the stacks. By assigning appropriate priorities to the level at which the search

for the next leaf node re-starts, the proposed demodulators can achieve performance-complexity trade-offs that dominate several existing soft demodulators, including those based on the stack algorithm and those based on ‘sphere decoding’ principles, especially in the low-complexity region.

In the second part of this thesis it is shown that the randomization procedure that is inherent in the semidefinite relaxation (SDR) technique for hard demodulation can be exploited to generate the list members required for list-based soft demodulation. The direct application of this observation yields list-based soft demodulators that only require the solution of one SDP per demodulation-decoding iteration. By approximating the randomization procedure by a set of independent Bernoulli trials, this requirement can be reduced to just one semidefinite program (SDP) per channel use. An advantage of these demodulators over those based on optimal tree-search algorithms is that the computational cost of solving the SDP is a low-order polynomial in the problem size. The analysis and simulation experiments provided in the thesis show that the proposed SDR-based demodulators offer an attractive trade-off between performance and computational cost.

The structure of the SDP in the proposed SDR-based demodulators depends on the signaling scheme, and the initial development focuses on the case of QPSK signaling. In the last chapter of this thesis, the extension to MIMO 16-QAM systems is developed, and some interesting observations regarding some existing SDR-based hard demodulation schemes for MIMO 16-QAM systems are derived. The simulation results reveal that the excellent performance-complexity trade-off of the proposed SDR-based schemes is preserved under the extension to 16-QAM signaling.

# Acknowledgment

I would like to express my deepest appreciation and gratitude to Prof. Tim Davidson for his encouragement, support, constructive supervision and for the many hours spent reviewing ideas, papers and this thesis. The completion of this research would have not been possible without the countless discussions and meetings with him.

I would like to thank my supervisory committee members, Prof. Wong and Prof. Kirubarajan for their guidance and comments. Also many thanks for the constructive discussions and comments from Dr. Kisialiou and Prof. Luo.

The administrative team of the Electrical and Computer Engineering Department deserves all the gratitude for their help in many matters.

Last but not least I would like to thank my very supporting family; I owe a great dept of gratitude to my wife Mahsa for her patience, encouragement and understanding. I appreciate all her help and her support through my research. All my love and thanks to her. I can not thank my parents enough for what they have been doing for me since my birth. I am deeply indebted to them for their love, continuous encouragement and support. I would like to thank my brother Peyman and my sister Ladan for their support and for being my best friends.

# Glossary

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BICM	Bit Interleaved Coded Modulation
BLAST	Bell-Labs Layered Space-Time Architecture
BQP	Binary Quadratic Program
BPSK	Binary Phase-Shift Keying
CDMA	Code Division Multiple Access
CSI	Channel State Information
FLOP	Floating Point Operation
IDD	Iterative Demodulation and Decoding
IP	Interior Point
LD	Linear Dispersion
LISS	List Sequential
LLR	Log Likelihood Ratio
MAP	Maximum a Posteriori Probability
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
MMSE	Minimum Mean Square Error
OFDM	Orthogonal Frequency Division Multiplexing
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase-Shift Keying
SD	Sphere Decoding
SDP	Semidefinite Program
SDR	Semidefinite Relaxation
SIC	Soft Interference Cancellation
SNR	Signal to Noise Ratio
TCM	Trellis Coded Modulation

# Contents

<b>Abstract</b>	<b>v</b>
<b>Acknowledgment</b>	<b>vii</b>
<b>Glossary</b>	<b>ix</b>
<b>Contents</b>	<b>x</b>
<b>List of Tables</b>	<b>xv</b>
<b>List of Figures</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Multiple antenna communication systems . . . . .	2
1.2 Thesis outline . . . . .	6
<b>2 MIMO communication systems</b>	<b>13</b>
2.1 Introduction . . . . .	13
2.2 Single input single output communications systems and coding	15
2.2.1 AWGN channels . . . . .	15
2.2.2 Fading channels . . . . .	16
2.2.3 Modulation and coding for AWGN and fading channels	18

2.2.3.1	Bit interleaved coded modulation (BICM) . . .	20
2.3	Multiple input multiple output communication systems . . . .	25
2.3.1	Narrow-band MIMO system model . . . . .	25
2.3.2	Performance measures for MIMO channels . . . . .	27
2.3.2.1	MIMO channel capacity . . . . .	27
2.3.2.2	Multiplexing gain . . . . .	28
2.3.2.3	Diversity gain . . . . .	29
2.3.2.4	Diversity-multiplexing trade-off . . . . .	30
2.4	Space-time modulation and coding . . . . .	30
2.4.1	Space-time mappings . . . . .	32
2.4.1.1	Space-time constellations . . . . .	33
2.4.1.2	Space-time trellis codes . . . . .	35
2.4.2	Concatenated space-time codes . . . . .	36
2.4.2.1	Layered space-time transmission . . . . .	36
2.4.2.2	Space-time bit-interleaved codes . . . . .	39
2.5	Soft MIMO demodulation . . . . .	41
2.5.1	Optimum soft MIMO demodulation for a MIMO-BICM- IDD scheme . . . . .	43
2.5.2	Hard MIMO demodulation . . . . .	46
2.5.2.1	Tree search algorithms . . . . .	47
2.5.2.2	Greedy algorithm . . . . .	51
2.5.2.3	Semidefinite relaxation . . . . .	52
2.5.3	Approximate soft MIMO demodulation based on hard demodulation . . . . .	54
2.5.4	Approximate soft MIMO demodulation based on list approximations . . . . .	56
2.5.5	Other soft MIMO demodulation approaches . . . . .	58

2.6	Summary . . . . .	59
<b>3</b>	<b>A multi-stack algorithm for soft MIMO demodulation</b>	<b>61</b>
3.1	Introduction . . . . .	62
3.2	System model . . . . .	65
3.3	Multi-stack algorithm . . . . .	69
3.3.1	Symbol and re-start orderings . . . . .	71
3.3.1.1	V-BLAST symbol and re-start orderings . . . . .	72
3.3.1.2	Symbol and re-start orderings based on a <i>pri-</i> <i>ori</i> information . . . . .	72
3.3.1.3	Re-start ordering based on a <i>priori</i> information	72
3.3.1.4	Natural re-start order . . . . .	73
3.3.2	Bounding the path metrics . . . . .	73
3.3.3	Bounding the complexity . . . . .	74
3.4	Likelihood computation . . . . .	74
3.5	Simulation results . . . . .	77
3.6	Conclusion . . . . .	87
<b>4</b>	<b>Efficient Soft Demodulation of MIMO QPSK via Semidefinite Re-</b> <b>laxation</b>	<b>89</b>
4.1	Introduction . . . . .	90
4.2	System model and iterative receiver . . . . .	93
4.3	Hard demodulation using SDR . . . . .	98
4.4	List-SDR method for soft demodulation . . . . .	101
4.5	Soft demodulation using single SDR . . . . .	102
4.6	List-free implementation . . . . .	106
4.7	Computational cost . . . . .	110

4.8	Simulations . . . . .	113
4.8.1	$8 \times 8$ system . . . . .	114
4.8.2	$4 \times 4$ system . . . . .	123
4.9	Conclusion . . . . .	128
<b>5</b>	<b>Efficient soft MIMO 16-QAM demodulation using SDR</b>	<b>129</b>
5.1	Maximum likelihood demodulation of MIMO 16-QAM using SDR . . . . .	131
5.1.1	Increased dimension relaxation . . . . .	134
5.1.2	Fixed dimension relaxation . . . . .	137
5.1.3	A fast interior point SDP solver for (5.15) . . . . .	138
5.1.4	Equality of increased dimension and fixed dimension relaxation techniques . . . . .	142
5.2	Maximum a posteriori probability MIMO 16-QAM demodulation using SDR . . . . .	147
5.2.1	Polynomial approximations of the cost function . . . . .	148
5.2.2	SDR formulation of MAP demodulation . . . . .	151
5.3	Soft MIMO 16-QAM demodulation using List-SDR method . . . . .	153
5.4	Soft MIMO 16-QAM demodulation using Single-SDR method . . . . .	156
5.5	Simulation results . . . . .	163
5.5.1	Simulations with turbo outer code . . . . .	164
5.5.2	Simulations with outer convolutional code . . . . .	178
5.6	Conclusion . . . . .	182
<b>6</b>	<b>Summary and future work</b>	<b>185</b>
6.1	Summary . . . . .	185
6.2	Directions for future work . . . . .	188

<b>A MIMO-IDD using MMSE-SIC</b>	<b>193</b>
<b>Bibliography</b>	<b>197</b>

# List of Tables

3.1	Proposed List Construction Algorithm . . . . .	75
3.2	Average size of the preliminary list, $\hat{\mathcal{L}}$ and the enriched list, $\hat{\mathcal{L}}'$ , and the average number of nodes visited, $N'$ , for the multistack algorithm with natural or VBLAST re-start orderings, and a limit $L$ on the preliminary list size, or a limit $N$ on the number of nodes visited. The scenario considered is that in Fig. 3.4. . . . .	83
3.3	Average size of the preliminary list, $\hat{\mathcal{L}}$ and the enriched list, $\hat{\mathcal{L}}'$ , and the average number of nodes visited, $N'$ , for the multistack algorithm with natural or VBLAST re-start orderings, and a limit $L$ on the preliminary list size, or a limit $N$ on the number of nodes visited. The scenario considered is that in Fig. 3.6. . . . .	83
4.1	List generation component of the List-SDR algorithm. . . . .	102
4.2	List generation component of the Single-SDR algorithm . . . . .	106
4.3	List-free implementation of Single-SDR algorithm for randomized soft demodulator . . . . .	109
4.4	Dominant computational cost per channel use of various MIMO soft demodulators for a system with $N_t$ transmit antennas and $T$ demodulation-decoding iterations. . . . .	112
5.1	Interior point algorithm for solving (5.15) . . . . .	141

5.2	List-free implementation of List-SDR algorithm for randomized soft MIMO 16-QAM demodulation . . . . .	155
5.3	List-free implementation of Single-SDR algorithm for randomized soft MIMO 16-QAM demodulation . . . . .	161
5.4	Dominant computational costs of various MIMO soft demodulators.	162

# List of Figures

1.1	MIMO BICM-IDD transceiver. . . . .	4
2.1	Block diagram of a BICM transmitter and a simple sub-optimal receiver. . . . .	21
2.2	Block diagram of BICM-IDD. The subscript “1” is associated with the inner code variables and the subscript “2” is associated with the outer code variables. . . . .	21
2.3	A typical multiple-input multiple-output system. . . . .	25
2.4	Some available BLAST space-time mapping schemes. . . . .	38
2.5	MIMO BICM-IDD transceiver. . . . .	39
2.6	MIMO BICM-IDD transceiver. . . . .	42
2.7	A pictorial representation of the received signal vector $\mathbf{y}$ , the receiver lattice $\Lambda(\mathbf{H})$ and the projection of the decoding sphere to the hyper-plane of the lattice. . . . .	48
2.8	Branch and bound tree search method. . . . .	50
3.1	MIMO BICM-IDD transceiver. . . . .	65
3.2	A snapshot of an instance of the stack algorithm for a system in which $K$ symbols from a 4-ary constellation are transmitted in each (block) channel use. . . . .	68

3.3	A snapshot of an instance of the proposed multistack algorithm for a system in which $K$ symbols from a 4-ary constellation are transmitted in each (block) channel use. . . . .	70
3.4	The trade-off between the SNR required for a BER of $10^{-4}$ and the average FLOPs per channel use for different algorithms for a $4 \times 4$ MIMO-BICM transmission scheme with 16-QAM symbols. . . . .	78
3.5	The trade-off between the SNR required for a BER of $10^{-4}$ and the worst-case FLOPs per channel use for different algorithms for a $4 \times 4$ MIMO-BICM transmission scheme with 16QAM symbols. . . . .	79
3.6	The trade-off between the SNR required for a BER of $10^{-4}$ and the average FLOPs per channel use for different algorithms for a $8 \times 8$ MIMO-BICM transmission scheme with QPSK symbols. . . . .	80
3.7	The trade-off between the SNR required for a BER of $10^{-4}$ and the worst-case FLOPs per channel use for different algorithms for a $8 \times 8$ MIMO-BICM transmission scheme with QPSK symbols. . . . .	81
3.8	Empirical probability density of FLOPS per channel use for different demodulation schemes for $4 \times 4$ MIMO 16-QAM transmission. . . . .	85
3.9	Empirical probability density of FLOPS per channel use for different demodulation schemes for $8 \times 8$ MIMO QPSK transmission. . . . .	86
4.1	MIMO BICM-IDD transceiver. . . . .	94
4.2	List generation scheme using the Single-SDR algorithm. . . . .	105
4.3	The proposed list-free soft demodulator. . . . .	107
4.4	Comparison of the BER performance of various demodulators for the $8 \times 8$ MIMO system with the turbo outer code. . . . .	116
4.5	Comparison of the BER performance of various demodulators for the $8 \times 8$ MIMO system with the convolutional outer code. . . . .	117

4.6 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  $8 \times 8$  system with the turbo outer code. For the SDR based methods results for several values for  $M$ , the number of randomizations, are provided. . . . . 118

4.7 Empirical probability density of the number of FLOPs per channel use in the  $8 \times 8$  system with the turbo outer code at an SNR of 2.75 dB. . . . . 119

4.8 BER performance of the  $8 \times 8$  MIMO system with the turbo outer code and the Single-SDR demodulator with  $M = 50$  randomizations and different accuracies to which the SDP is solved. . . . . 121

4.9 BER performance of the  $8 \times 8$  MIMO system with the turbo outer code and the Single-SDR demodulator with an SDP solution accuracy of  $\epsilon = 10^{-2}$ , and different numbers of randomizations,  $M$ . 122

4.10 Comparison of the BER performance of various demodulators for the  $4 \times 4$  MIMO system with the turbo outer code. . . . . 124

4.11 Comparison of the BER performance of various demodulators for the  $4 \times 4$  MIMO system with the convolutional outer code. . . . . 125

4.12 Comparison of average computational cost per channel use of the proposed demodulators and that of the Multi-SDR, full-list, sphere decoding and MMSE-SIC demodulators for the  $4 \times 4$  system with the turbo outer code. . . . . 126

4.13 Empirical probability density of the number of FLOPs per channel use in the  $4 \times 4$  system with the turbo outer code at an SNR of 2.75 dB. . . . . 127

5.1 The proposed list-free Single-SDR demodulator for MIMO 16-QAM transmission. . . . . 160

5.2	Comparison of the BER performance of various demodulators for the $4 \times 4$ MIMO 16-QAM system with the turbo outer code. . . .	167
5.3	Comparison of the average computational cost per channel use of the proposed demodulators and that of list sphere decoder, the LISS and MMSE-SIC demodulators for the $4 \times 4$ MIMO 16-QAM system with the turbo outer code. For the SDR based methods results for several values for $M$ , the number of randomizations, are provided. Here, the List-SDR method is the one with the quadratic approximation of the prior information. . . . .	168
5.4	Empirical probability density of the number of FLOPs per channel use in the $4 \times 4$ MIMO 16-QAM system with the turbo outer code at an SNR of 9.75 dB. Here, the List-SDR method is the one with the quadratic approximation of the prior information. . . . .	169
5.5	BER performance of a $4 \times 4$ MIMO 16-QAM system with the turbo outer code that employs the List-SDR demodulator with the quadratic approximation of the prior information, $M = 50$ randomizations, and different accuracies to which the SDP is solved.	173
5.6	BER performance of a $4 \times 4$ MIMO 16-QAM system with the turbo outer code that employs the Single-SDR demodulator with $M = 50$ randomizations and different accuracies to which the SDP is solved. . . . .	174

5.7	Comparison of the average computational cost per channel use of the proposed demodulators and that of list sphere decoder, the LISS and MMSE-SIC demodulators for the $4 \times 4$ MIMO 16-QAM system with the turbo outer code. For the SDR based methods results for different accuracies of solving the SDPs, $\epsilon$ , are provided. Here, the List-SDR method is the one with the quadratic approximation of the prior information. . . . .	175
5.8	BER performance of a $4 \times 4$ MIMO 16-QAM system with the turbo outer code that employs the List-SDR demodulator with the quadratic approximation of the prior information, an SDP solution accuracy of $\epsilon = 10^{-1}$ , and different numbers of randomizations, $M$ . . . . .	176
5.9	BER performance of a $4 \times 4$ MIMO 16-QAM system with the turbo outer code that employs the Single-SDR demodulator with an SDP solution accuracy of $\epsilon = 10^{-1}$ , and different numbers of randomizations, $M$ . . . . .	177
5.10	Comparison of the BER performance of various demodulators for the $4 \times 4$ MIMO 16-QAM system with the convolutional outer code. Here, the List-SDR method is the one with the quadratic approximation of the prior information. . . . .	179
5.11	BER performance of a $4 \times 4$ MIMO 16-QAM system with the convolutional outer code that employs the List-SDR demodulator with the quadratic approximation of the prior information, an SDP solution accuracy of $\epsilon = 10^{-1}$ , and different numbers of randomizations, $M$ . . . . .	180

5.12 BER performance of a $4 \times 4$ MIMO 16-QAM system with the convolutional outer code that employs the Single-SDR demodulator with an SDP solution accuracy of $\epsilon = 10^{-1}$ , and different numbers of randomizations, $M$ . . . . .	181
--	-----

# Chapter 1

## Introduction

'You see, wire telegraph is a kind of very, very long cat. You pull his tail in New York and his head is meowing in Los Angeles. Do you understand this? And radio operates in exactly the same way: you send signals here, they receive them there. The only difference is that there is no cat.'

---

ALBERT EINSTEIN

ONE OF THE advantages of wireless communication systems is the freedom of untethered communication. This offers great convenience to the user, but creates challenging issues for the designer. One of the current challenges in wireless communications is to design efficient wireless systems that can communicate at high data rates (of the order of Gigabits per second) in the presence of the multipath propagation that dominates most wireless communication environments [1]. One way to work towards this goal is to use multiple antennas at the transmitter and/or the receiver. The main focus of this thesis is on designing effective and efficient demodulation schemes for multiple antenna communication systems that provide reliable (point-to-point) communication at high data rates. In this chapter this focus will be

placed in context by providing an overview of some of the principles of multiple antenna communication systems, and then outlining the contributions of this thesis.

## **1.1 Multiple antenna communication systems**

In wireless communication systems messages are communicated by modulating the properties of electromagnetic waves. In most terrestrial applications, there will be more than one path along which these waves may propagate from the transmitter to the receiver. In some environments there will be a “line-of-sight” path, while in others some or all of the paths will involve reflections from electromagnetic scatterers in the environment. In general, each of these paths will incur a different attenuation, phase change and propagation delay. In this thesis, the focus will be on applications in which the bandwidth of the transmitted signal is not wide enough for the delays between the paths to be resolved at the receiver. As a result, the effective (complex) gain of the base-band equivalent channel can be modeled as the sum of the (complex) gains along each path. Depending on the phase relationships between the paths, this addition may have a constructive or a destructive effect on the received signal power [2]. Since even small changes in the position of the transmitter, receiver, or scatterers can change the phase relationships between the paths quite dramatically, received signal power in many wireless applications may change quite rapidly. This phenomenon, which is commonly known as fading, is one of the distinguishing features of wireless communication systems; and the mitigation (or exploitation) of fading is one of the key aspects of wireless system design.

In a communication environment with a large number of scatterers

with a reasonably uniform distribution, a small change in the position of the receive antenna tends to result in a substantial change in the channel characteristics, and the large number of scatterers makes this change difficult to predict. Indeed, the statistical dependence between the channel gains tends to disappear with position changes of the order of a wavelength of the electromagnetic wave (e.g., [3]). This observation encouraged designers to install multiple antennas at the receivers with spacing of the order of a wavelength. As the channel gains to each of these antennas are approximately independent (under the rich scattering assumption), the likelihood that all receive antennas experience a deep fade is substantially reduced, and by exploiting this diversity of the received signals, substantial performance improvements can be realized.

Signal diversity can also be introduced at the transmitter, by transmitting (variants of) the message signal from sufficiently well-spaced transmit antennas, and the use of multiple antennas at both the transmitter and receiver multiplies the potential performance gain that arises from diversity (e.g., [4]). Systems with multiple transmit and receive antennas, which are often called multiple-input multiple-output (MIMO) systems, also offer the potential for substantial increases in the achievable data rate (e.g., [5–8]) by sending independent information from each antenna. It is that aspect of MIMO systems that motivates much of the development in this thesis.

The desirable diversity properties of MIMO systems, and their potential for substantial increases in the achievable data rate have meant that they have become a key component in the development of the future wireless communication standards; e.g., [1, 9, 10]. For example, MIMO systems are becoming the basis for the current wireless systems such as WiFi wireless LAN (IEEE 802.11n standard), Mobile WiMAX (IEEE 802.16e standard),

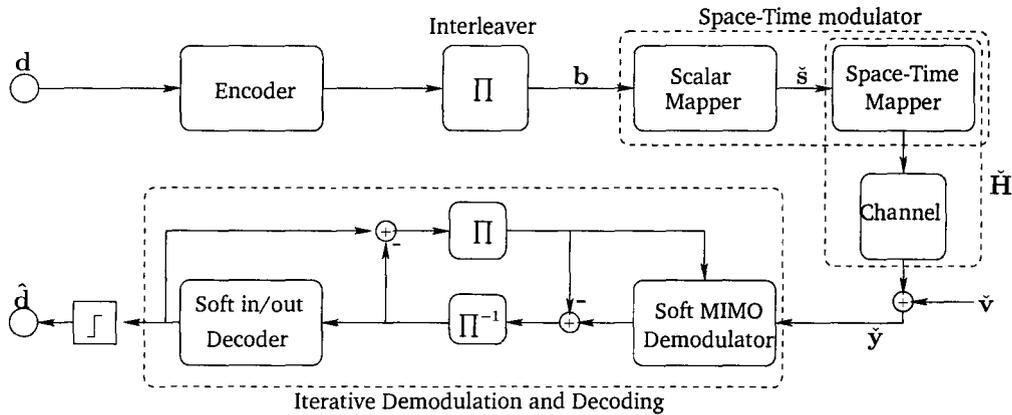


Figure 1.1: MIMO BICM-IDD transceiver.

LTE (Long Term Evolution) project and UMB (Ultra Mobile Broadband) systems [10]. However, the computational effort required for optimal detection of MIMO transmission schemes that operate at such high spectral efficiencies is beyond the capabilities of the envisioned communication devices [1], and hence there has been considerable interest in the development of transceivers that balance the competing demands of rate and computational efficiency at the receiver of such a MIMO communication scheme. One of the contributions of this thesis is the development of components for a MIMO receiver that achieve desirable trade offs between these two goals.

As will be discussed in Chapter 2, a popular communication scheme that enables considerable flexibility in the selection of an appropriate balance between rate and computational complexity in both single antenna and MIMO communication systems is bit interleaved coded modulation (BICM, e.g., [11]); see also Fig. 1.1. In a standard form of this strategy for coherent MIMO systems (e.g., [12]), the message bits,  $d$ , are encoded using a conventional binary encoder, interleaved, and then modulated into scalar symbols.

These scalar symbols are then mapped into space and time as space-time matrix symbols for transmission over the MIMO channel. Given the complexity of optimal decoding, the standard reception strategy is to adopt a bit-wise iterative “soft” demodulation and decoding (IDD) approach that attempts to maximize the *a posteriori* probability of each bit in the message. This strategy is often called the “Turbo principle” [13], and will be described in some detail in Chapter 2. Although this iterative demodulation and decoding strategy offers a substantial reduction in computational cost, the demodulation step, which involves the extraction of a sufficiently accurate approximation of the likelihood of each bit from the output of the MIMO channel, remains a substantial computational burden. Hence, the focus of this thesis is on designing low-complexity soft MIMO demodulators.

Most of the existing approaches to low-complexity soft demodulation are based on using the so-called “max-log” approximation of the soft information. These approaches can be divided into two major classes. One class is based on solving several hard demodulation problems using low-complexity schemes, such as tree search algorithms (e.g., [14, 15]) or semidefinite relaxation (SDR; e.g, [16]) techniques [17]. The other group is based on list demodulation schemes (e.g., [12, 18–20]) in which a list of bit-vectors that generate the dominant components of the soft information is carefully selected, and the max-log approximation is performed over that list. This thesis will focus on designing computationally efficient list-based soft demodulation schemes. The proposed schemes can, themselves, be divided into two classes. The first class, which will be introduced in Chapter 3, is based on tree search algorithms and, in particular, the stack algorithm (e.g., [21]). The second class, which will be introduced in Chapters 4 and 5, is based on semidefinite relaxation techniques. In the next section the contributions of

the upcoming chapters of this thesis will be reviewed in more detail.

## 1.2 Thesis outline

The rest of this thesis is divided into five chapters, of which Chapters 3, 4 and 5 contain the main technical contributions. Chapter 2 provides some introductory material as background for these chapters, and Chapter 6 will conclude the thesis and suggest some interesting avenues for future work.

Chapter 2 will review some of the principles of single antenna and multiple antenna wireless communications, along with some communication transceiver architectures that can be used to achieve reliable communication at high data rates in these systems. As was pointed out in the previous section, this chapter will show that among these communication schemes a popular one that enables considerable flexibility in the selection of an appropriate balance between rate and computational complexity in both single antenna and MIMO communication systems is the use of BICM scheme (cf., Fig. 1.1). Several existing “space-time” transmission schemes mapping the scalar symbols to be transmitted in a MIMO system to symbol matrices that span space and time will be reviewed. The receiver of the MIMO-BICM scheme shown in Fig. 1.1 uses a strategy which is often called the “Turbo principle” [13], and this principle will also be reviewed in some detail. As was discussed above, list-based techniques for reduced complexity soft MIMO demodulation schemes are the main focus of this thesis, and some of the available techniques for constructing this list in a computationally efficient manner will be reviewed. One group of these techniques involves the application of semidefinite relaxation (SDR), and since SDR techniques play a key role in the work in Chapters 4 and 5, this mathematical technique will

be discussed in some detail.

In Chapter 3 a multistack algorithm for soft demodulation will be introduced. The algorithm is based on the principles of the stack algorithm (e.g., [21]) for traversing the tree structure that is inherent in the MIMO demodulation problem. The stack algorithm stores a single stack of visited nodes in the tree, and expands the stack using the ‘best-first’ principle, as quantified by the (partial) likelihoods of the tree nodes. In the proposed multistack algorithm, the single stack is partitioned into a stack for each level of the tree, and the algorithm proceeds by performing one best-first search step in each of these stacks in the natural ordering of the tree. As will be shown in some simulation examples in this chapter, by assigning appropriate priorities to the level at which this ‘best-first search per level processing re-starts once a leaf node has been obtained, the proposed demodulators can achieve trade-offs between performance and complexity that dominate those of several existing methods, including the stack algorithm, in the low-complexity region.

Although soft MIMO demodulation schemes based on tree search algorithms, such as the list sphere decoder and the stack algorithm, have a lower computational cost than the optimal soft demodulator, their computational complexity is still exponential in the number of bits transmitted per channel use [22]. In contrast, the MIMO hard demodulation schemes that are based on semidefinite relaxation techniques have a computational complexity that is a (low order) polynomial of the number of bits transmitted per channel use [23]. The existing soft demodulation scheme that is based on the SDR technique takes the approach to soft demodulation that requires the solution of several hard demodulation problems per channel use [17]. However, before the developments in this thesis, no list-based soft demodulation

scheme that uses the SDR technique has been made available. Chapter 4 presents such a scheme for QPSK signaling. This soft demodulator has a computational cost that is greatly reduced compared to the existing SDR-based soft demodulator [17]. The proposed scheme exploits the randomization procedure inherent in the SDR technique to generate the list of candidate bit-vectors, and hence it will be called a List-SDR scheme. By approximating the randomization procedure using Bernoulli randomization trials, the computational cost of this demodulator can be further reduced. Indeed, the resulting demodulator has a computational cost that is lower than that of the existing low complexity demodulators that are based on minimum mean squared error with soft interference cancellation (MMSE-SIC) (e.g., [24]). The proposed soft demodulator achieves this low computational cost with only a small degradation in its performance compared to the existing tree search and SDR-based soft demodulators, and its performance is actually better than that of the MMSE-SIC demodulator. Since the proposed soft demodulator solves only one SDP per channel use, it will be called a Single-SDR scheme. Efficient implementations of these SDR-based algorithms with reduced memory requirements will also be provided in this chapter.

In the application of SDR techniques to demodulation problems, the structure of the corresponding SDP will change depending on the signal constellation that is transmitted. The developments in Chapter 4 are based on QPSK signaling and since their extension to higher order constellations is not straightforward, these schemes will be extended to be used for 16-QAM signaling in Chapter 5. It will be shown in this chapter that two of the existing SDR-based hard demodulation schemes for MIMO 16-QAM transmission [25, 26] are equivalent. Furthermore, an efficient interior point

method will be developed for the approach that has the lower dimensionality. The List-SDR scheme proposed in Chapter 4 solves a maximum *a posteriori* probability (MAP) hard demodulation problem in each demodulation-decoding iteration. Since the soft information component of the decision metric for 16-QAM signaling cannot be expressed in a polynomial format, the proposed List-SDR scheme cannot be applied directly. Therefore, two polynomial approximations for the metric will be developed in order to facilitate the extension of the List-SDR scheme to soft demodulation for a MIMO 16-QAM system. The extension of the Single-SDR scheme to 16-QAM signaling is performed by approximating the randomization procedure inherent in the SDR technique using independent random symbol generators with probability mass functions obtained from the SDP that is solved in the first demodulation-decoding iteration. This probability mass function is then updated in the subsequent iterations using the *a priori* information from the decoder. The simulation results in this chapter will show that the proposed Single-SDR and List-SDR schemes for 16-QAM signaling will provide complexity-performance trade-offs that dominate those of some existing soft demodulation algorithms. In particular, it will be shown that the Single-SDR algorithm has a computational cost that is lower than that of the MMSE-SIC demodulator and yet its performance is better than that of this soft demodulator and close to that of some more computationally costly tree search algorithms for soft demodulation.

Chapter 6 will conclude the thesis and will provide some suggestions for future research work.

A preliminary version of the work in Chapter 3 of this thesis appears in

- M. Nekuii and T. N. Davidson, “Reduced-complexity demodulation for MIMO-BICM-IDD using modified stack algorithms,” in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Honolulu, Apr. 2007, vol. 3, pp. 65–68,

and another version will appear as

- M. Nekuii and T. N. Davidson, “A multistack algorithm for soft MIMO demodulation”, to appear in *IEEE Trans. Veh. Technol.*, accepted in final form, August 2008.

Preliminary versions of the work in Chapter 4 of this thesis have appeared as

- M. Nekuii and T. N. Davidson, “List Based Soft Demodulation of MIMO QPSK via Semidefinite Relaxation”, in *Proc. IEEE Wkshp Signal Processing Advances in Wireless Commun.*, Helsinki, Finland, Jun. 2007. 5 pages.
- M. Nekuii, M. Kisiailiou, T. N. Davidson, and Z.-Q. Luo, “Efficient soft demodulation of MIMO QPSK via semidefinite relaxation”, in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Las Vegas, Apr. 2008, pp. 2665–2668,

and another version has been submitted as

- M. Nekuii, M. Kisiailiou, T. N. Davidson, and Z.-Q. Luo, “Efficient soft demodulation of MIMO QPSK via semidefinite relaxation”, submitted to *IEEE Trans. Signal Processing*, May 2008.

A version of the work in Chapter 5 has been submitted as

- M. Nekuii and T. N. Davidson, “Soft demodulation of MIMO M-ary QAM via semidefinite relaxation”, submitted to *IEEE Int. Conf. Commun.*, Jun. 2009.

and another version as

- M. Nekuii and T. N. Davidson, “Soft demodulation of MIMO M-ary QAM: An efficient semidefinite relaxation approach”, in preparation for submission to *IEEE Trans. Wireless Commun.*

# Chapter 2

## MIMO communication systems

'Calling all. This is our last cry before our eternal silence.'

---

LAST MESSAGE IN MORSE CODE BY THE FRENCH  
MARITIME SERVICE, JAN 31 1997

**T**HIS CHAPTER will provide some background discussion in preparation for Chapters 3, 4 and 5. In particular, the principles of several different modulation and coding schemes for single antenna and multiple antenna communication systems will be discussed.

### 2.1 Introduction

The provision of multiple antennas at both the transmitter and the receiver of a wireless communication system offers the potential for reliable communication at data rates substantially higher than those of the single antenna systems [4–6]. Several modulation, mapping and coding schemes have recently been developed in order to provide good performance at these high spectral efficiencies. For applications in which the frequency response of

the channel can be deemed to be flat across the transmission bandwidth, the transmitted codewords span space and time, and hence they are called “space-time” transmission schemes. 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. As mentioned in Chapter 1, one pragmatic approach that balances the competing demands for spectral and computational efficiencies in multiple antenna design is the use of space-time techniques in conjunction with bit-interleaved coded modulation (BICM) (e.g., [11, 27–30]) with iterative demodulation and decoding (IDD). A key computational bottleneck in these schemes is the demodulation step; that is, the extraction of soft information (in the form of log likelihood-ratio or an approximation thereof) of each of the bits transmitted in a given block from the corresponding output block of the MIMO channel. In this thesis we will provide several effective and computationally-efficient demodulation techniques for these transmission schemes.

In order to place the MIMO-BICM-IDD scheme in context, we will first discuss some coding and modulation schemes for single antenna communications systems. Many of the principles of these schemes generalize naturally to multiple antenna systems, and hence the discussion of these simple schemes will form a framework upon which the rest of this chapter will be built.

## 2.2 Single input single output communications systems and coding

Communication channels in which the transmitted signal is corrupted by additive white Gaussian noise only are often called AWGN channels. In many wireline applications, the gain of the AWGN channel can be assumed to be constant. However, in many wireless applications, the channel gain varies with time and this causes variations in the signal to noise ratio (SNR) at the receiver. These channels are usually called fading channels. In the rest of this section, we will review some aspects of modulation and coding for constant gain and fading AWGN channels.

### 2.2.1 AWGN channels

A simple single-input single-output communication system is the AWGN channel with a constant gain. Applying the conventional orthogonal basis representation [31], the channel output in response to the  $n$ th transmitted symbol can be written as

$$y_n = hs_n + v_n, \quad (2.1)$$

where  $s_n$  is the signal transmitted at the  $n$ th “channel use”,  $h$  is the constant (complex) channel gain which is assumed to be known at the receiver (coherent reception), and  $v_n$  is an independent and identically distributed (i.i.d.) Gaussian noise sample with variance  $\sigma^2$  per dimension. We then define the average signal to noise ratio (SNR) at the receiver as:

$$\text{SNR} = \frac{|h|^2 \mathbf{E}\{|s_n|^2\}}{\mathbf{E}\{|v_n|^2\}} = \frac{|h|^2 \mathbf{E}\{|s_n|^2\}}{2\sigma^2}. \quad (2.2)$$

In the next section we will review different modulation and coding schemes for reliable communications over AWGN channels, as these provide the framework for several modulation and coding schemes for multiple antenna systems. However, in order to assess the performance of those signaling schemes we need a fundamental benchmark. One such benchmark is the capacity of a channel, which is defined as the supremum possible data rate that can be transmitted through this channel with a vanishing small probability of error. Shannon proved that for an AWGN channel the capacity in bits per complex dimension per channel use is given by [32]

$$C = \log_2(1 + \text{SNR}). \quad (2.3)$$

He showed that reliable transmission is not possible at rates  $R$  greater than  $C$ , and that for  $R < C$  there exists a coding strategy under which the average probability of error at the receiver can be made arbitrarily small.

### 2.2.2 Fading channels

In wireless communication systems there are typically multiple propagation paths between the transmitter and the receiver. Furthermore motion of the transmitter, receiver or the scatterers on these paths means that the paths gains vary in time. We will consider communication applications in which the bandwidth of the transmitted signal is not wide enough for the relative delays between these paths to be resolved. In that case, the channel can be assumed to be flat in the frequency domain within the transmitted signal bandwidth. This channel is usually called a frequency-nonselective or narrow-band channel [2] and can be represented by a time varying gain, as

shown in the following model

$$y_n = h_n s_n + v_n. \quad (2.4)$$

In this model the (complex) fading coefficient  $h_n$  is the sum of the (complex) channel gains of the different paths from the transmitter to the receiver. One simple method to extract a statistical model for  $h_n$  is to assume that there are a large number of statistically independent reflected and scattered paths. Hence, according to the Central Limit Theorem,  $h_n$  can be modeled as a circularly symmetric complex Gaussian random variable. Because the squared magnitude  $|h_n|^2$  has a Rayleigh distribution, this channel is often called a Rayleigh fading channel.

In the applications that one considered in this thesis, the fading coefficient  $h_n$  is constant during a block of  $N$  channel uses, and it is assumed to have independent and identically distributed (i.i.d.) values between different blocks. That is, we consider a block fading channel model (e.g., [33]). If the application can tolerate the latency of using a long code-word that covers a large number of transmission blocks, the limit on the average rate that can be communicated is the ergodic channel capacity. For example, for applications in which the channel is not known to the transmitter but the receiver has perfect knowledge of the channel, the ergodic channel capacity is  $E_h \{\log_2(1 + |h|^2 \text{SNR})\}$  [2], where the expectation is over all possible channel realizations.

In some other applications like fixed point-to-point wireless telecommunications, the channel changes slowly. That is, the code-word length and the fading block length  $N$  are comparable to each other and the channel is

non-ergodic. In these applications there is a non-zero probability of encountering a realization of the channel for which the selected coding scheme, no matter how powerful it is, is incapable of reliable data transmission with the given data rate. In this case, the performance of the channel is typically measured by the outage probability, which is the probability that the maximum rate which can be transmitted over the given channel realization is smaller than the given transmission rate  $R$ . The outage probability provides a lower bound on the probability that the transmitted data will not be received correctly.

### **2.2.3 Modulation and coding for AWGN and fading channels**

In this section we review some modulation and coding schemes for reliable communication over single-input single-output channels. These schemes can be divided into two regimes [34], a low SNR regime in which the limitation on the transmission power dominates, and a high SNR regime in which the limit on the transmission bandwidth dominates. In the power-limited regime, reasonable performance can be achieved using binary codes, such as block codes and convolutional codes. Furthermore, low error rates can be achieved within fractions of a decibel of the SNR limit for the chosen rate using binary low-density parity-check (LDPC) codes [35] or turbo codes [36]. However, in the bandwidth limited regime, binary signaling is not sufficiently spectrally-efficient and we seek coded modulation schemes whose transmitted signal distributions better approximate the (optimal) Gaussian distribution. Many of the existing schemes combine binary coding

with bandwidth-efficient higher-order constellations, such as pulse amplitude modulation (PAM), phase-shift keying (PSK) and quadrature amplitude modulation (QAM), although other options, such as lattice codes [37], are also available.

In order to design codes with good performance an appropriate design criterion is needed. In AWGN channels, the high-SNR performance of a code is (strongly) dependent on the minimum Euclidean distance between pairs of the code-words [38]. One approach to increase the minimum distance is to combine modulation and coding. Trellis coded-modulation (TCM) [38] and multi-level coding [39] are two such examples. They use multi-level/phase constellations and simple convolutional codes with bit mapping rules that maximize the minimum Euclidean distance of the code.

The design criteria for codes for fading channels are quite different. In a Rayleigh fading channel, the coding performance is dominated by its minimum Hamming distance (e.g., [2, 40]) rather than its minimum Euclidean distance. In the case of binary codes, codes with a large Euclidean distance also have a large Hamming distance. However, in the bandwidth-limited regime, codes that are designed for good performance over AWGN channels might not provide good performance over a fading channel. For example, it was shown in [40] that a large Euclidean distance for a TCM code does not guarantee a large Hamming distance. An alternative approach to coded modulation was proposed by Zehavi [40]. He suggested a scheme based on a binary encoder (outer code) with a “bit-wise” interleaver at its output, followed by a conventional modulator. (This scheme is described in more detail in Section 2.2.3.1 below.) Zehavi showed that by using this method, which was later called bit-interleaved coded modulation (BICM) [11], Hamming distances up to the Hamming distance of the outer code can be obtained,

and hence the reliability of BICM over a Rayleigh fading channel can be greater than that of the corresponding TCM schemes [11].

### 2.2.3.1 Bit interleaved coded modulation (BICM)

In a BICM scheme the block of information bits  $\mathbf{d}$  is passed through an encoder before being interleaved, as shown in Fig. 2.1. The interleaved encoded bits are then modulated using a higher-order constellation and a pre-specified mapping strategy, such as Gray mapping. The optimum demodulation procedure would be to detect the maximum likelihood transmitted sequence  $\check{\mathbf{d}}$  given the channel output sequence  $\mathbf{y}$ ,

$$\check{\mathbf{d}} = \arg \max_{\mathbf{d}} P(\mathbf{y}|\mathbf{d}). \quad (2.5)$$

Unfortunately this demodulation scheme has a prohibitively high computational complexity, which increases exponentially with the length of the transmitted bit-sequence.

A sub-optimum demodulation scheme [11] is shown in Fig. 2.1. In this scheme the demodulator detects the maximum-likelihood transmitted symbol in each channel use, and passes the corresponding bits to the deinterleaver and then on to the binary-input decoder. It was shown in [40] that the performance of this BICM scheme over a Rayleigh fading channel is better than that of a TCM scheme, but its performance over an AWGN channel is not as good as that of a TCM scheme.

While the sub-optimum detector in Fig. 2.1 is relatively simple to implement, the fact that it makes independent “hard” decisions on each

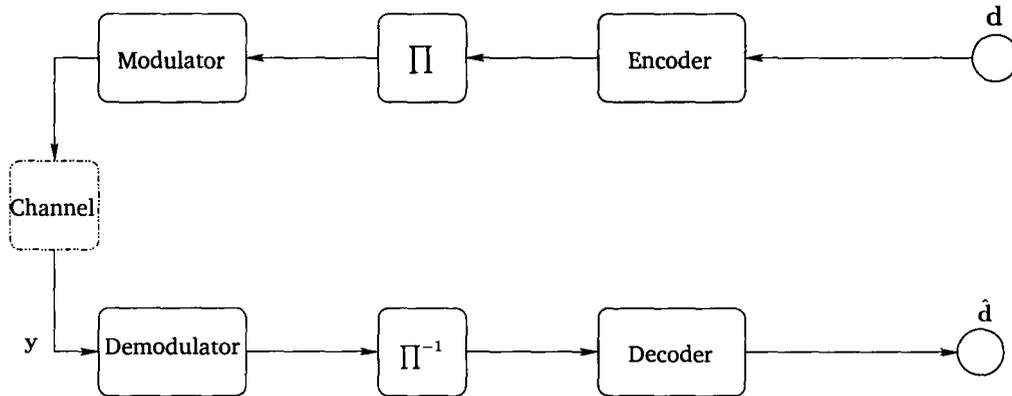


Figure 2.1: Block diagram of a BICM transmitter and a simple sub-optimal receiver.

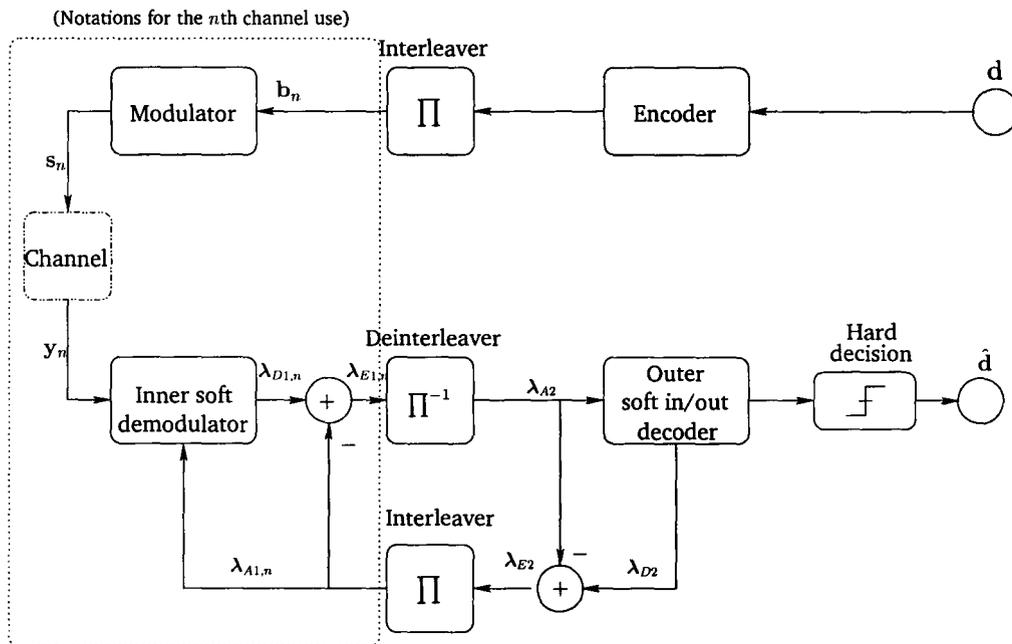


Figure 2.2: Block diagram of BICM-IDD. The subscript “1” is associated with the inner code variables and the subscript “2” is associated with the outer code variables.

transmitted symbol limits its performance. As a result, other iterative sub-optimum BICM detection strategies based on “soft” demodulation and decoding and the turbo principle [36] have been proposed [13, 27, 41–44]. While these strategies are also sub-optimal, they offer substantially better performance than schemes based on hard demodulation. In fact, their performance can come quite close to that of the optimum sequence detector in (2.5), while their computational cost is still quite reasonable. In these iterative receivers, the modulator block is viewed as an inner code, which is serially concatenated with the encoder (the outer code). Hence, at the receiver the turbo principle can be applied to construct an iterative demodulation and decoding (IDD) scheme, as shown in Fig. 2.2. (This receiver also has a factor-graph interpretation; e.g., [45–48].) It has been shown [43, 49] that by using optimized constellation mappings, BICM-IDD can result in improved performance-complexity trade-offs over those of TCM for AWGN channels, while significantly outperforming TCM over fading channels.

As a result of these favorable properties of BICM-IDD in single antenna communication systems (and, as we will see in the next section, in multiple antenna communication systems), we will choose BICM-IDD as the framework within which the performance of the MIMO soft demodulation schemes proposed in this thesis will be evaluated. Therefore, we will now discuss the iterative demodulation and decoding procedure in more detail. While we will consider a single antenna system at this stage, the principles extend directly to the multiple antenna case; cf. Section 2.4.2.2 and Chapters 3, 4 and 5.

Let us consider the transmission of the  $n$ th symbol,  $s_n$ , by the modulator in Fig. 2.2. Let  $\mathbf{b}_n \triangleq [b_{n,1}, \dots, b_{n,M}]^T \in \{\pm 1\}^M$  denote<sup>1</sup> the  $M$  bits selected from the encoded and interleaved bit-sequence  $\mathbf{b}$  to be modulated to  $s_n$ , and let  $\mathcal{M}(\cdot)$  denote the mapping strategy, so that  $s_n = \mathcal{M}(\mathbf{b}_n)$ . Let  $N$  denote the number of channel uses required to transmit the whole encoded code-word  $\mathbf{b}$ . At the receiver, the demodulator takes two inputs, the channel measurement  $y_n$  and the (extrinsic) information from the previous iteration of the outer encoder regarding the likelihood of each encoded bit in  $\mathbf{b}_n$ ; i.e., the *a priori* information. In a BICM-IDD scheme, the role of the (soft) demodulator is to compute (or approximate) the posterior probability of each bit  $b_{n,k}$ . Since  $b_{n,k}$  is binary, that information can be conveniently captured in the log likelihood-ratio (LLR)

$$\lambda_{n,k} = \log \frac{P(b_{n,k} = +1)}{P(b_{n,k} = -1)}, \quad k = 1, \dots, M. \quad (2.6)$$

Let  $\lambda_{A1,n}$  denote the vector of *a priori* information for each bit in  $\mathbf{b}_n$ , in LLR format. The demodulator in Fig. 2.2 takes this *a priori* information and the channel observation  $y_n$  and extracts the updated soft information  $\lambda_{D1,n} \triangleq [\lambda_{D1,n,1}, \dots, \lambda_{D1,n,M}]^T$ , where

$$\lambda_{D1,n,k} = \log \frac{P(b_{n,k} = +1|y_n)}{P(b_{n,k} = -1|y_n)} = \log \frac{\sum_{\mathbf{b}_n, b_{n,k}=+1} P(y_n|s_n)P(s_n)}{\sum_{\mathbf{b}_n, b_{n,k}=-1} P(y_n|s_n)P(s_n)}, \quad (2.7)$$

$s_n = \mathcal{M}(\mathbf{b}_n)$ , and the summations are over all possible bit-vectors  $\mathbf{b}_n$  for which  $b_{n,k}$  is fixed to the specified value. Assuming that the interleaver is

---

<sup>1</sup>In this thesis we will represent the two possible values for each bit by  $-1$  and  $+1$  rather than  $0$  and  $1$ , as this will simplify some of our derivations.

designed well enough for the assumption of independence between the interleaved encoded bits to be valid,  $P(s_n)$  in (2.7) can be computed from  $\lambda_{A1,n}$  using  $P(s_n) \simeq \prod_{k=1}^M P(b_{n,k})$ , where  $s_n = \mathcal{M}(\mathbf{b}_n)$ ,  $P(b_{n,k} = -1) = \frac{1}{1+\exp(\lambda_{A1,n,k})}$  and  $P(b_{n,k} = +1) = \frac{\exp(\lambda_{A1,n,k})}{1+\exp(\lambda_{A1,n,k})}$ , [13]. In order to prevent positive feedback in the iterative receiver, only the extrinsic component of the demodulator's output should be sent to the decoder. In log likelihood-ratio form, that extrinsic information can be written as

$$\lambda_{E1,n} = \lambda_{D1,n} - \lambda_{A1,n}. \quad (2.8)$$

After computing  $\lambda_{E1,n}$  for the  $N$  channel uses that are required to transmit the whole codeword  $\mathbf{b}$ , the extrinsic information is collected in a vector  $\lambda_{E1}$  and is deinterleaved to generate the *a priori* information  $\lambda_{A2}$  to be used in the soft-input soft-output outer decoder to extract  $\lambda_{D2}$ . For example, if we use a convolutional code at the transmitter, this decoder can use the Bahl-Cocke-Jelinek-Raviv (BCJR) algorithm [50, 51] to extract  $\lambda_{D2}$ . After computing  $\lambda_{D2}$  the extrinsic information regarding the encoded bits is extracted as  $\lambda_{E2} = \lambda_{D2} - \lambda_{A2}$ , and is then passed thorough the interleaver and onto the demodulator, where it is used as the *a priori* information  $\lambda_{A1}$  for the next iteration of the soft demodulator. This completes one iteration between the demodulator and the decoder. In the first demodulation-decoding iteration, no *a priori* information is available from decoder, and we will set  $\lambda_{A1} = 0$ . Typically, the bit-error rate (BER) decreases after each iteration, and hence a trade-off between performance and complexity can be set by constraining the number of iterations.

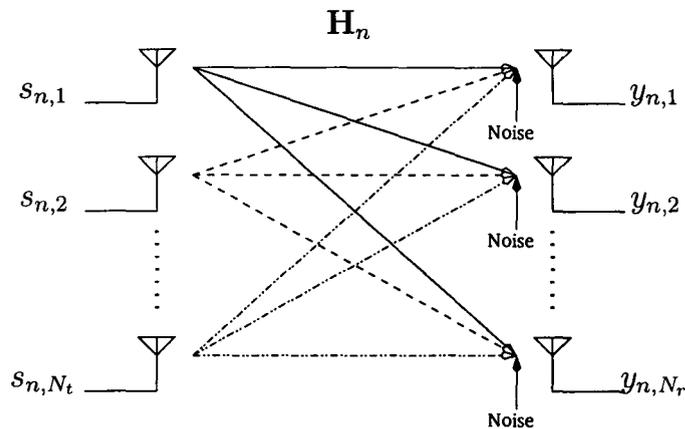


Figure 2.3: A typical multiple-input multiple-output system.

## 2.3 Multiple input multiple output communication systems

The use of multiple antennas at the transmitter and/or the receiver provides the potential for reliable (wireless) communications at significantly larger data rates than the corresponding single antenna system. In this section we will describe some theoretical and implementation aspects of MIMO communication systems, including some measures of the capacity of these systems, some space-time transmission and coding schemes, and the corresponding receiver structures that are designed to provide good performance at spectral efficiencies that approach the capacity limits of the MIMO channel.

### 2.3.1 Narrow-band MIMO system model

A diagram of a narrow-band multiple antenna communication system with  $N_t$  transmit antennas and  $N_r$  receive antennas is shown in Fig. 2.3. The discrete time model for the signal received by the  $i$ th antenna at the  $n$ th

channel use can be written as

$$y_{n,i} = \sum_{j=1}^{N_t} h_{n,ij} s_{n,j} + v_{n,i}, \quad n = 1, \dots, N, \quad i = 1, \dots, N_r, \quad (2.9)$$

where  $s_{n,j}$  is the transmitted signal from the  $j$ th antenna,  $v_{n,i}$  is an independent and identically distributed (i.i.d.) zero-mean Gaussian noise sample with variance  $\sigma^2$  per dimension at the  $i$ th receive antenna and  $h_{n,ij}$  represents the complex gain (fading coefficient) from the transmit antenna  $i$  to the receive antenna  $j$ . By collecting the observations at each antenna at the  $n$ th channel use into a vector  $\mathbf{y}_n$ , the discrete time model of this system can be written as

$$\mathbf{y}_n = \mathbf{H}_n \mathbf{s}_n + \mathbf{v}_n, \quad (2.10)$$

where

$$\mathbf{y}_n = \begin{bmatrix} y_{n,1} \\ \vdots \\ y_{n,N_r} \end{bmatrix}, \quad \mathbf{H}_n = \begin{bmatrix} h_{n,11} & \cdots & h_{n,1N_t} \\ \vdots & \ddots & \vdots \\ h_{n,N_r1} & \cdots & h_{n,N_r N_t} \end{bmatrix}, \quad \mathbf{s}_n = \begin{bmatrix} s_{n,1} \\ \vdots \\ s_{n,N_t} \end{bmatrix}, \quad \mathbf{v}_n = \begin{bmatrix} v_{n,1} \\ \vdots \\ v_{n,N_r} \end{bmatrix}.$$

We will impose the condition that the (average) transmit signal power is  $P$ ; that is

$$E \{ \mathbf{s}_n^H \mathbf{s}_n \} = P. \quad (2.11)$$

The signal to noise ratio (SNR) at each receive antenna is  $\rho = P/2\sigma^2$ .

Several other communication systems can be modeled using this general MIMO framework (e.g., [1, 52, 53]). For example, single or multiple-antenna systems that transmit over a (finite impulse response) dispersive channel using a block-based transmission scheme with guard sequences to

prevent inter-block interference (e.g., [1, 54]) can also be represented by this model. In that case, the matrix  $\mathbf{H}_n$  has a (block) Toeplitz structure when the zero padding guard sequence is used, and it has a (block) circulant structure when the cyclic prefix guard sequence is used. In this thesis, the focus will be on communication systems that can be represented using this general framework without assuming any special structure for the channel matrix  $\mathbf{H}_n$ . Indeed, in much of our performance analysis we will focus on the rich scattering scenario in which the elements of  $\mathbf{H}_n$  are modeled as i.i.d. zero-mean circularly symmetric complex Gaussian random variables.<sup>2</sup> This channel model is often referenced to as the i.i.d. Rayleigh channel model.

### 2.3.2 Performance measures for MIMO channels

As we discussed in Section 2.2.1, one of the benchmarks that can be used to assess the performance of a designed modulation and coding scheme for a particular communication system is the notion of capacity. In this section, we will discuss the ergodic capacity of a MIMO channel. We will also discuss one of the measures for characterizing the error performance on MIMO channels (the diversity gain), a measure of the rate of growth of the maximum achievable rate on these channels with SNR (the multiplexing gain) and the trade-off between these two gains.

#### 2.3.2.1 MIMO channel capacity

In this thesis, the focus will be on systems with delay constraints that are substantially longer than the coherence time of the channel, and hence the

---

<sup>2</sup>Measurements validating this model for rich scattering environments can be found in [55].

most relevant notion of capacity for these systems is the ergodic capacity.<sup>3</sup> The ergodic capacity of such a MIMO channel is the maximum achievable average rate that can be communicated over the channel realizations. It is shown that [6] in order to maximize the ergodic capacity for MIMO systems with Gaussian noise, the transmit signals should have a Gaussian distribution, and for systems where the channel is known to the receiver and unknown at the transmitter, the transmit power should be divided equally over all the transmit antennas. Hence, for these systems, the ergodic capacity in bits per transmit antenna per complex dimension can be obtained as [6, 57]

$$C = E_{\mathbf{H}} \left\{ \log_2 \det[\mathbf{I}_{N_r} + \frac{\rho}{N_t} \mathbf{H}\mathbf{H}^H] \right\}. \quad (2.12)$$

### 2.3.2.2 Multiplexing gain

For a general coded transmission, the slope at which the data rate  $R(\rho)$  of this system increases with  $\log_2 \rho$  is measured as

$$r \triangleq \lim_{\rho \rightarrow +\infty} \frac{R(\rho)}{\log_2 \rho}, \quad (2.13)$$

and is called the “multiplexing gain” [7] of the transmission scheme. In the asymptotic limit of high SNRs, the ergodic capacity of an i.i.d. Rayleigh model for the MIMO channel in (2.12) can be written as [5]

$$C = \tilde{N} \log_2 \frac{\rho}{N_t} + \sum_{i=1}^{\tilde{N}} E_{\mathbf{H}} \{ \log_2 \lambda_i^2 \}, \quad (2.14)$$

---

<sup>3</sup>For many practical communications applications, such as mobile communications in which the channel changes fast due to mobility, an encoded data block can experience many channel realizations, hence, the channel can be assumed ergodic [56]. In this thesis we will focus on these applications.

where  $\tilde{N} \triangleq \min(N_r, N_t)$  and  $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_{\tilde{N}}$  are the singular values of (the random matrix)  $\mathbf{H}$ . At high SNRs the ergodic capacity grows linearly with  $\log_2(\frac{\rho}{N_t})$  with a slope of  $\tilde{N}$ , and as a result, the capacity is roughly  $\tilde{N}$  times that of the equivalent single antenna communication channel. Indeed, from the perspective of the growth of the capacity at high SNRs, this MIMO channel behaves like  $\tilde{N}$  parallel independent spatial communication channels that carry independent messages. Hence, a MIMO communication channel has a maximum multiplexing gain of  $\tilde{N}$ , which corresponds to the total number of degrees of freedom to communicate over this channel [7].

### 2.3.2.3 Diversity gain

Another measure of performance in communication channels is to measure the probability of error at the receiver. If a transmission scheme at SNR  $\rho$  has an average error probability of  $P_e(\rho)$ , a notion that characterizes this error probability at high SNRs is the “diversity gain”  $d$  which is defined as

$$d \triangleq - \lim_{\rho \rightarrow +\infty} \frac{\log P_e(\rho)}{\log \rho}. \quad (2.15)$$

Hence, in systems with higher diversity gains, the error rate curve has a steeper slope with SNR. In a MIMO channel the diversity gain corresponds to the number of independently faded replicas of the transmitted signal at the receiver. Since a general  $N_t \times N_r$  MIMO communication system can have at most  $N_t N_r$  independent random fading paths, the maximum diversity gain provided by this MIMO channel is  $N_t N_r$  (e.g., [58]).

### 2.3.2.4 Diversity-multiplexing trade-off

MIMO transmission schemes can be designed to exploit both the diversity gain and the multiplexing gain of a MIMO channel. That is, the SNR gain can be used to increase the achievable data rate, or the diversity gain, or both of them. A fundamental trade-off between the reliability and the transmission rate exist [7, 8]. Zheng and Tse [7] derived this trade-off at high-SNRs, and they showed that in a rich scattering environment in order to have a multiplexing gain  $r$ , the optimum diversity gain  $d_{\text{opt}}(r)$  that can be achieved is

$$d_{\text{opt}}(r) = (N_t - r)(N_r - r), \quad 0 \leq r \leq \min(N_t, N_r). \quad (2.16)$$

Hence, both diversity gain and multiplexing gain can be achieved in a MIMO communication system, but the above fundamental trade-off exists between these two gains. Based on the definition of the multiplexing gain, the application of this trade-off is limited to those scenarios in which the channel throughput scales linearly with the logarithm of the SNR. Azarian and El Gamal [8] relaxed this constraint and derived a related trade-off between reliability and throughput that allows more general scenarios to be considered.

## 2.4 Space-time modulation and coding

Having studied some of the fundamental limits of MIMO communication systems in the previous section, in this section we will review some modulation and coding schemes that are designed to approach these limits. In these schemes the symbols are designed to span space and time and hence they are called “space-time” transmission schemes. For example, consider a

block of  $T$  symbol vectors to be transmitted in  $T$  channel uses over a MIMO channel in which the channel is fixed and equal to  $\mathbf{H}$  over a block-length of at least  $T$  channel uses. (Such channels are often called block-fading channels.) By defining  $\mathbf{S} = [\mathbf{s}_1, \dots, \mathbf{s}_T]$  as the  $N_t \times T$  matrix of transmitted signals,  $\mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_T]$  as the  $N_r \times T$  matrix of received signals and  $\mathbf{V} = [\mathbf{v}_1, \dots, \mathbf{v}_T]$  as the  $N_r \times T$  matrix of noise samples, the received signals over these  $T$  channel uses can be written as

$$\mathbf{Y} = \mathbf{H}\mathbf{S} + \mathbf{V}. \quad (2.17)$$

The space-time transmission schemes that we will present in this section differ in the way that they encode and map the data bits to the symbol matrix  $\mathbf{S}$ . In order to design good space-time codes some design criteria must be defined. One potential design criterion is to evaluate the pairwise probability of error at the receiver. If we assume the channel matrix is known to the receiver, then at high SNRs the pairwise error probability of the receiver mistaking a signal matrix  $\hat{\mathbf{S}}$  for the transmitted signal matrix  $\mathbf{S}$  is bounded by [56, 59, 60]

$$p(\mathbf{S} \rightarrow \hat{\mathbf{S}}) \leq \frac{1}{\left(\prod_{k=1}^{M_\Delta} \lambda_k(\Delta)\right)^{N_r}} \left(\frac{\rho}{4N_t}\right)^{-M_\Delta N_r}, \quad (2.18)$$

where  $\rho$  is, as in Section 2.3.1, the SNR per receive antenna,  $\Delta = (\mathbf{S} - \hat{\mathbf{S}})(\mathbf{S} - \hat{\mathbf{S}})^H$ ,  $M_\Delta$  is the rank of  $\Delta$  and  $\lambda_k(\Delta)$  is the  $k$ th non-zero eigenvalue of  $\Delta$ . Referring to the definition of the diversity gain in (2.15) we can conclude that the diversity gain of this pair of code-words is  $M_\Delta N_r$ . If we make sure that every pair of code-words satisfies this diversity order, the average error probability also satisfies it and this space-time transmission scheme is said

to have a diversity order of  $M_{\Delta}N_r$ . Hence, in order to reach to the maximum diversity gain of a MIMO communication channel ( $N_tN_r$ ) the space-time transmission matrix  $\mathbf{S}$  should be designed in a way that the difference matrix  $\Delta$  between any two of its codewords has full rank equal to  $N_t$ . This design criterion is called the *rank criterion* [59, 60].

A second design criterion is to maximize the coefficient of the signal to noise ratio in Equation (2.18),  $\left(\prod_{k=1}^{M_{\Delta}} \lambda_k(\Delta)\right)^{1/M_{\Delta}}$ , which is called the “coding gain”. In order to get a high coding gain we should maximize the minimum of the determinant of  $\Delta$  over all possible pair of codewords  $\mathbf{S}$ . This design criterion is called the *determinant criterion* [59, 60].

Given their impact on the error performance of MIMO systems, these two design criteria have been used widely to design several space-time transmission schemes; e.g., [56].<sup>4</sup> We will review some of these schemes in Section 2.4.1, below. In order to exploit the temporal diversity of the channel (in addition to its spatial diversity) several other space-time transmission schemes have been developed to combine channel coding schemes with space-time mapping schemes. We will review some of those schemes in Section 2.4.2.

## 2.4.1 Space-time mappings

In this section we will review some space-time mapping strategies that have been designed to achieve the potential multiplexing and diversity gains provided by a MIMO channel. We divide these mappings into two groups. The first group have short frame lengths  $T \approx N_t$  (space-time constellations or space-time block codes); e.g., [58], and they provide a simple and effective

---

<sup>4</sup>A general binary rank design criteria was later developed in [61] providing better construction methods for some other space-time transmission schemes.

way of achieving the diversity gains of the channel. The second group have larger frame lengths  $T \gg N_t$  (space-time trellis codes) with the advantage of better coding gains compared to the space-time block codes, but with the disadvantage of receivers with substantially higher computational complexities.

#### 2.4.1.1 Space-time constellations

One of the simplest space-time block codes was designed by Alamouti [62] for a MIMO communication system with two transmit antennas and one (or two) receive antenna. This scheme can provide full diversity gain, and for systems with one receive antenna it also achieves the full multiplexing gain of the system. In this scheme, two symbols are transmitted in every two channel uses and the receiver can optimally detect each transmitted symbol using only linear processing. Given the exponential cost of optimal vector detection for general transmission schemes (see Section 2.5.2), this motivated the search for similar schemes for higher numbers of transmit and receive antennas using the theory of orthogonal designs (e.g., [63]). In these schemes, the columns of the space-time code are designed to be orthogonal to each other in order for optimal detection to be achievable using linear processing. Some space-time block codes were developed based on this framework in [58, 64, 65]. They provide full spatial diversity, and have simple detection algorithms.

The orthogonality constraint on space-time block codes means that their maximum achievable data rate is less than the channel capacity (if the number of receive antennas is greater than one). To address this issue, Jafarkhani [66] proposed the quasi-orthogonal space-time block codes.

These codes are constructed using smaller orthogonal designs as their building blocks. The optimal detection procedure for these codes must be performed on pairs of symbols rather than on individual symbols. Hence, the receiver complexity is higher than that of orthogonal space-time block codes.

A broad design framework for space-time block codes was introduced by Hassibi and Hochwald in [67], where the orthogonality constraint was relaxed in order to enable higher achievable rates. In this coding scheme, the input symbol block is dispersed over space and time in a (widely) linear fashion and hence it is called a linear dispersion (LD) code. More specifically, linear dispersion codes transmit  $L$  symbols using  $N_t$  transmit antennas over  $T$  channel uses, using a space-time transmission matrix  $\mathbf{S}$  that has the following format:

$$\mathbf{S} = \sum_{\ell=1}^L (\alpha_{\ell} \mathbf{A}_{\ell} + j\beta_{\ell} \mathbf{B}_{\ell}), \quad (2.19)$$

where the  $L$  symbols are, respectively,  $s_{\ell} = \alpha_{\ell} + j\beta_{\ell}$ , and  $\mathbf{A}_{\ell}, \mathbf{B}_{\ell}$ ,  $\ell = 1, \dots, L$  are  $N_t \times T$  complex matrices. These matrices can be designed in a number of ways depending on the desired design criterion. In [67] the maximization of the ergodic capacity of the LD-coded transmission scheme was chosen as the design criterion. Other schemes like [68] used asymptotic guidelines to design the LD matrices. Full symbol-rate, full-diversity LD codes for  $2 \times 2$  systems were developed in [69] and later they were generalized to higher numbers of antennas in [70]. Information theoretic and detection error view points were used in [71] to design trace-orthonormal full diversity space-time codes using this framework. LD framework can also subsume several other existing space time transmission schemes including orthogonal space-time block codes (e.g., [58, 64, 65]) and some of the layered space-time transmission schemes that we will describe in Section 2.4.2.1.

In order to write the received signal of an LD-coded transmission scheme in a compact and convenient form, we consider the channel model (2.17), the LD code in (2.19), and we define the column vectors  $\check{\mathbf{s}} \triangleq [\alpha_1, \beta_1, \dots, \alpha_L, \beta_L]^T$  and  $\check{\mathbf{v}} \triangleq \text{vec}(\mathbf{V})$ , and the  $N_r T \times 2L$  complex matrix  $\check{\mathbf{H}} \triangleq [\text{vec}(\mathbf{H}\mathbf{A}_1), \text{vec}(j\mathbf{H}\mathbf{B}_1), \dots, \text{vec}(\mathbf{H}\mathbf{A}_L), \text{vec}(j\mathbf{H}\mathbf{B}_L)]$ . We can then write the received signal  $\check{\mathbf{y}} \triangleq \text{vec}(\mathbf{Y})$  in the following format

$$\check{\mathbf{y}} = \check{\mathbf{H}}\check{\mathbf{s}} + \check{\mathbf{v}}. \quad (2.20)$$

Hence, in the coherent detection perspective, linear dispersion codes can be applied to any combination of transmit and receive antennas to reshape the MIMO channel matrix into a new equivalent channel matrix  $\check{\mathbf{H}}$  that results in a model that is analogous to the general MIMO model of (2.10).

#### 2.4.1.2 Space-time trellis codes

The rank and determinant criteria (obtained from equation (2.18)) were originally applied to the design of space-time trellis codes (STTC) [56] over large code frame lengths  $T \gg N_t$  in order to improve their diversity and coding gains. In the process of designing these codes, it is assumed that the channel is fixed for this long frame length  $T$  (quasi static fading channel) [56] and hence these codes may not necessary be optimal for ergodic block fading channels (e.g., [59]). Space-time trellis codes are the extension of the conventional trellis coded modulation (TCM; e.g., [38]) schemes to MIMO channels. That is, they are defined on a trellis by combining modulation and coding with space-time mapping, and they can be decoded using a multi-dimensional version of the Viterbi algorithm [56]. These codes can have excellent diversity and coding gains, but the computational complexity of

their receiver increases exponentially with the transmission rate of the code [72].

## 2.4.2 Concatenated space-time codes

While the space-time transmission schemes described in Section 2.4.1.1 exploit the spatial diversity-multiplexing gains of MIMO channels, they are not designed to exploit the temporal diversity of the channel. However, channel coding can be combined with these space-time transmission schemes in order to exploit the temporal diversity. We will briefly review two popular schemes in this group: layered space-time codes, and space-time bit interleaved codes, which are the extension of the conventional bit-interleaved coded modulation (BICM) schemes to multiple antenna systems.

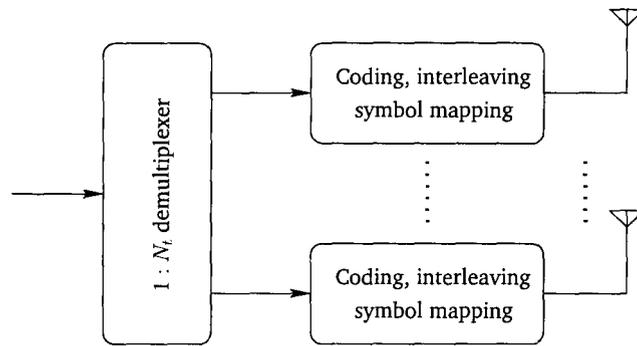
### 2.4.2.1 Layered space-time transmission

Layered space-time codes are rate-oriented codes that perform space-time mapping by grouping, encoding and modulating the data bits in layers of the space-time symbol matrix  $\mathbf{S}$  (cf. (2.17)). These schemes were first developed in [5] in which they were called Bell Labs Layered Space Time (BLAST) architectures. BLAST architectures have the feature that the two-dimensional (space-time) processing of modulating the data bits is performed in one-dimensional processing slots and that enables effective demodulation and decoding using conventional scalar approaches [73]. Several BLAST architectures exist, including Horizontal BLAST (H-BLAST), Vertical BLAST (V-BLAST) and Diagonal BLAST (D-BLAST). They have the common feature that they simultaneously transmit  $N_t$  independent substreams of data bits

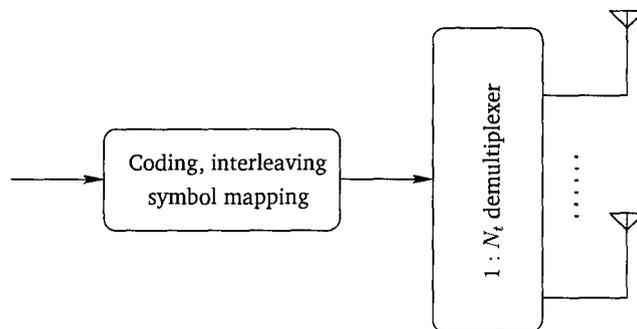
from the available  $N_t$  transmit antennas, but they use the different layering schemes depicted in Fig 2.4 to send them.

At the receiver, different demodulation schemes can be used to detect the transmitted signal (e.g., [74]), including the sub-optimum methods of interference suppression and cancellation (e.g., [73]), maximum likelihood (ML) demodulation methods (e.g., [75]) and iterative demodulation and decoding methods for coded BLAST schemes (e.g., [76, 77]). We will briefly review some of these methods in Section 2.5.

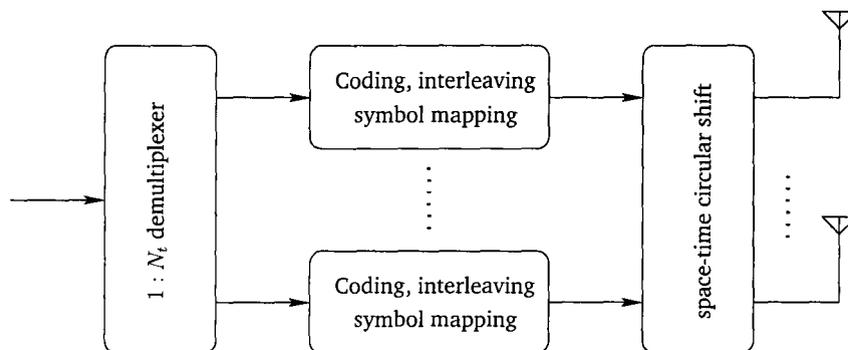
The BLAST architectures were conceived with ergodic rate objectives in mind and hence the focus is on exploiting the temporal diversity using channel coding. As will be described in the next section, one approach to doing so is to adopt a bit interleaved coded modulation architecture in which a space-time block code plays the role of the constellation and is chosen to exploit the available spatial diversity.



(a) Horizontal coding (H-BLAST)



(b) Vertical coding (V-BLAST)



(c) Diagonal coding (D-BLAST)

Figure 2.4: Some available BLAST space-time mapping schemes. (These diagrams are inspired by [9].)

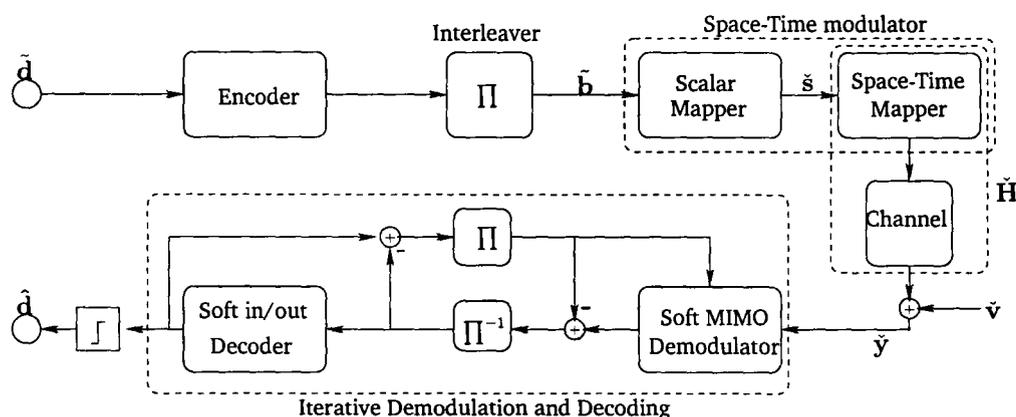


Figure 2.5: MIMO BICM-IDD transceiver.

### 2.4.2.2 Space-time bit-interleaved codes

In order to achieve both spatial and temporal diversity gains, channel codes can be used in conjunction with space-time transmission schemes in a variety of ways [12, 28–30, 76–78]. One of these is the space-time BICM (ST-BICM) scheme illustrated in Fig. 2.5. The structure of this scheme resembles the BICM structure [11, 40] that was described in Section 2.2.3.1 for single antenna communication systems. However, the scalar constellation has been replaced by a space-time constellation. In the ST-BICM scheme, the bit sequence  $\hat{\mathbf{d}}$  is binary encoded and interleaved to form the sequence  $\tilde{\mathbf{b}}$ , blocks of which are mapped to elements of the symbol sequence  $\tilde{\mathbf{s}}$ . The space-time transmitter maps blocks of this symbol sequence in space and time using a linear dispersion code [67]. Hence, the channel can be reshaped in the manner described by the equation (2.20) to form the equivalent channel matrix  $\tilde{\mathbf{H}}$ , as shown in Fig. 2.5.

It was shown in [79] that at low to intermediate SNRs the performance of space-time codes depends on the minimum Euclidean distance between any pair of the space-time codewords, in addition to the rank and

the determinant criteria. Hence, by choosing appropriate binary codes designed for AWGN single antenna communication systems, good performance can be achieved using ST-BICM transmission schemes [79, 80]. Using the iterative demodulation and decoding strategies presented in Section 2.2.3.1, ST-BICM schemes can effectively balance the computational costs and performance gains of the receiver, and can achieve good performance at SNRs quite close to the ergodic capacity limit of the channel [12, 28–30, 76–78].

Tonello [28] showed that in order to optimize the coding gain and fully exploit the temporal and spatial diversity gains offered by the ST-BICM architecture one should maximize the Hamming distance and the Euclidean distance of the codewords at the bit level rather than at the symbol level. Another important advantage of the ST-BICM schemes is that by choosing a full diversity space-time transmission scheme, these schemes can provide a robust performance over a variety of fading channels, including quasi-static and fast fading channels; cf. [30, 81].

While the optimal joint demodulation and decoding of ST-BICM schemes is computationally prohibitive, several suboptimal schemes have been developed based on the iterative soft MIMO demodulation and decoding structure illustrated in the receiver side of Fig. 2.5, [28–30, 78, 82–84]. As was discussed in Section 2.2.3.1, this receiver is not optimal, but it provides good performance at a substantially lower cost than the optimal receiver. Comparing the single antenna BICM iterative receiver structure of Fig. 2.2 with this iterative receiver structure reveals the fact that these two iterative schemes are similar in nature, with the soft scalar demodulator in Fig. 2.2 being replaced by a soft MIMO demodulator (e.g., [12]). Several ST-BICM schemes that use an optimal soft MIMO demodulator in this iterative receiver

structure have been proposed (e.g., [28–30, 78, 82–84]). However, the computational complexity of the optimal soft MIMO demodulator increases exponentially with the number of antennas and the data rate (e.g., [12]), and hence it is not a practical choice for applications that require high spectral efficiencies.

The core contribution of this thesis is to develop alternative soft MIMO demodulators that provide high performance while maintaining low computational cost. Since the emphasis in this thesis is on designing low complexity soft MIMO demodulators, we will adopt a conventional (and simple) transmission scheme, namely the V-BLAST space-time transmission scheme with Gray mapping. That is, we simply encode and interleave the data bits, modulate them to scalar symbols and then multiplex these symbols to the transmit antennas, see Fig. 2.6. (This MIMO transmission scheme is immediately applicable to the general linear dispersion code framework, as is evidenced by (2.20).) In order to make distinction between this simple structure and the general ST-BICM framework, we simply call this architecture a MIMO-BICM transceiver (e.g., [12]). We then use an iterative soft MIMO demodulation and decoding scheme at the receiver to detect the transmitted data bits. We will review some existing low complexity soft MIMO demodulation schemes for the MIMO-BICM transceiver of Fig. 2.6 in the following section.

## 2.5 Soft MIMO demodulation

We have seen in the previous section that among space-time transmission schemes that provide coding gain, ST-BICM schemes have some favorable properties. As a result, they have been widely proposed as pragmatic coded

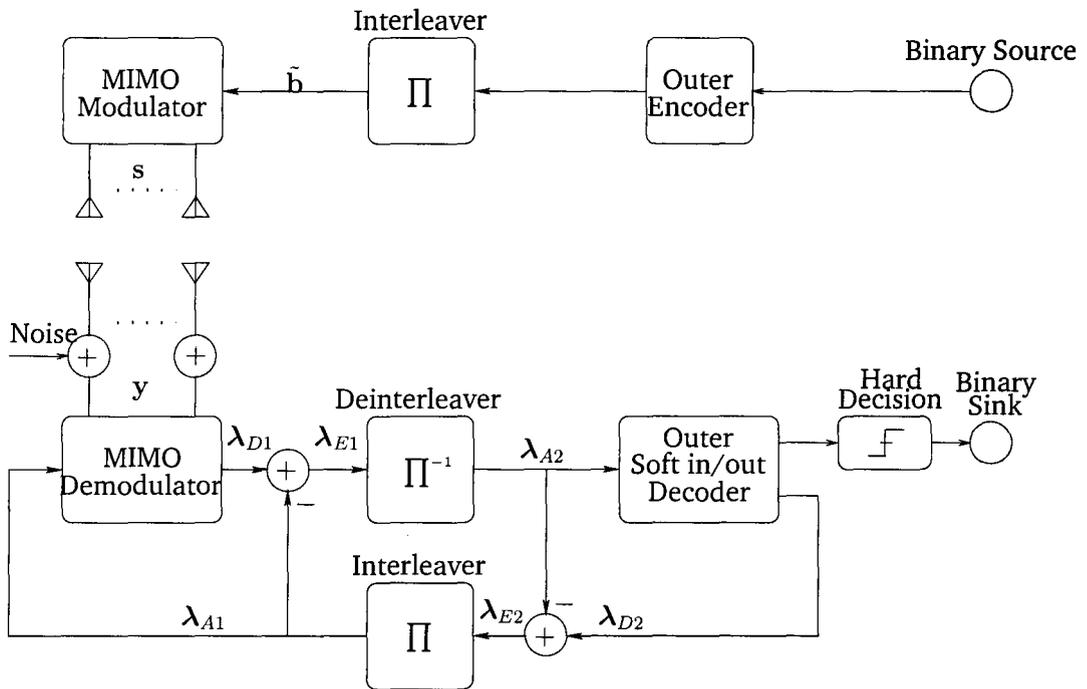


Figure 2.6: MIMO BICM-IDD transceiver.

transmission schemes for multiple antenna communication channels (e.g., [12, 14, 15, 17–20, 85–88]). The main detraction of these schemes is the computational complexity of the receiver, and in particular, the complexity of the soft MIMO demodulator, which increases exponentially with the spectral efficiency of the system. In this section we will first present the optimum demodulator for the MIMO-BICM transceiver with iterative demodulation and decoding (IDD). Then we will focus on existing approaches to efficiently approximating the output of the optimal demodulator.

## 2.5.1 Optimum soft MIMO demodulation for a MIMO-BICM-IDD scheme

We will consider the particular coherent narrow-band MIMO-BICM-IDD transceiver structure illustrated in Fig. 2.6. In this structure the space-time modulator is the concatenation of a scalar constellation mapper and a V-BLAST space-time transmission scheme; e.g., [12, 85], but extensions to the general linear dispersion code framework can be easily obtained, as described in Section 2.4.1.1. Since the emphasis of this thesis is on the demodulation step, we will allow any binary encoder as the outer encoder in Fig. 2.6, and we will adopt its corresponding soft-input soft-output decoder at the receiver. We will consider scalar constellations of size  $2^M$  and will transmit  $N_t$  such symbols per channel use, i.e., one symbol per antenna. We will let  $\mathbf{b}_n \in \{\pm 1\}^{MN_t}$  denote the vector of  $MN_t$  bits from the interleaved encoded bit stream that are mapped to the vector of transmitted symbols  $\mathbf{s}_n = \mathcal{M}(\mathbf{b}_n)$  at the  $n$ th channel use, where  $\mathcal{M}(\cdot)$  is the corresponding mapping strategy (e.g., independent Gray mapping of each symbol). Hence, the vector of received samples at the  $n$ th channel use can be written as

$$\mathbf{y}_n = \mathbf{H}_n \mathbf{s}_n + \mathbf{v}_n = \mathbf{H}_n \mathcal{M}(\mathbf{b}_n) + \mathbf{v}_n, \quad n = 1, \dots, N, \quad (2.21)$$

where, as it was defined in Section 2.3.1,  $\mathbf{H}_n$  is the matrix of channel gains at the  $n$ th channel use, which is assumed to be known at the receiver,  $\mathbf{v}_n$  is a vector of zero-mean additive white circularly symmetric complex Gaussian noise samples with variance  $\sigma^2$  per real scalar dimension, and  $N$  is the number of channel uses required to send the whole interleaved codeword at the transmitter.

The soft MIMO demodulator in Fig. 2.6 computes the soft information of each of the  $NMN_t$  interleaved encoded bits, based on the channel measurements and the extrinsic information from previous decoder iterations. The extrinsic component of this soft information is then passed to the outer soft decoder. The rest of the iterative demodulation and decoding process is the same as that for the single antenna BICM-IDD scheme described in Section 2.2.3.1.

As in Section 2.2.3.1, if the interleaver is designed well enough, it can be assumed that the interleaved encoded bits are locally independent. In that case, since the channel model in (2.21) is memoryless, the soft demodulator in an IDD scheme can operate on a block-by-block basis. Hence, for notational simplicity we will drop the subscript  $n$  in (2.21) and we consider a generic block channel use. As in Section 2.2.3.1, since the outer code is binary, the soft output from the demodulator can be in the form of the log likelihood-ratio (LLR) and the soft demodulator computes the (conditioned) LLR for each element  $b_i$  of  $\mathbf{b}$  as (e.g., [12]):

$$\lambda_{D1,i} \triangleq \log \frac{p(b_i = +1|\mathbf{y}, \mathbf{H})}{p(b_i = -1|\mathbf{y}, \mathbf{H})} = \log \frac{\sum_{\mathcal{L}_{i,+1}} p(\mathbf{y}|\mathbf{b}, \mathbf{H})p(\mathbf{b})}{\sum_{\mathcal{L}_{i,-1}} p(\mathbf{y}|\mathbf{b}, \mathbf{H})p(\mathbf{b})}, \quad i = 1, \dots, MN_t, \quad (2.22)$$

where  $\mathcal{L}$  is the list of all  $2^{MN_t}$  binary vectors  $\mathbf{b}$ ,  $\mathcal{L}_{i,b} \triangleq \{\mathbf{b} \in \mathcal{L} | b_i = b\}$ , and under the assumed AWGN noise model we have that:

$$p(\mathbf{y}|\mathbf{b}, \mathbf{H}) = \frac{1}{(2\pi)^{N_t/2}} \exp \left( -\frac{\|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2}{2\sigma^2} \right) \Big|_{\mathbf{s}=\mathcal{M}(\mathbf{b})}. \quad (2.23)$$

Since we have assumed local independence of the interleaved encoded bits, the conventional approximation  $p(\mathbf{b}) \approx \prod_{i=1}^{MN_t} p(b_i)$  holds and the LLR in

(2.22) can be written as (e.g., [85])

$$\lambda_{D1,i} \simeq \log \frac{\sum_{\mathcal{L}_{i,+1}} \exp(-D(\mathbf{b})/(2\sigma^2))}{\sum_{\mathcal{L}_{i,-1}} \exp(-D(\mathbf{b})/(2\sigma^2))}, \quad i = 1, \dots, MN_t, \quad (2.24)$$

where

$$D(\mathbf{b}) \triangleq \|\mathbf{y} - \mathbf{H}\mathcal{M}(\mathbf{b})\|_2^2 - 2\sigma^2 \sum_{i=1}^{MN_t} \log p(b_i). \quad (2.25)$$

Since each list  $\mathcal{L}_{i,b}$  in (2.24) contains  $2^{MN_t-1}$  terms, as the number of bits per channel use increases the computational cost of (2.22) increases exponentially. Hence, there has been considerable interest in schemes that enable the approximation of (2.22) with a reduction in complexity. Several popular approaches involve the use of one of the following approximations:

$$\lambda_{D1,i} \simeq \log \frac{\sum_{\hat{\mathcal{L}}_{i,+1}} \exp(-D(\mathbf{b})/(2\sigma^2))}{\sum_{\hat{\mathcal{L}}_{i,-1}} \exp(-D(\mathbf{b})/(2\sigma^2))} \quad (2.26)$$

$$\simeq \frac{1}{2\sigma^2} \left( \min_{\mathbf{b} \in \hat{\mathcal{L}}_{i,-1}} D(\mathbf{b}) - \min_{\mathbf{b} \in \hat{\mathcal{L}}_{i,+1}} D(\mathbf{b}) \right), \quad i = 1, \dots, MN_t, \quad (2.27)$$

where  $\hat{\mathcal{L}} \subseteq \mathcal{L}$ . Each of these equations reveals a general class of approximate soft MIMO demodulation as follows:

- List-based schemes, which are 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 using either marginalization over  $\hat{\mathcal{L}}_{i,\pm 1}$  in (2.26) (e.g., [85]), or by performing an exhaustive search over  $\hat{\mathcal{L}}_{i,\pm 1}$  to solve the minimization problems in (2.27), (e.g., [12, 18–20, 86–88]).
- Hard demodulation based schemes, which are based on selecting  $\hat{\mathcal{L}} = \mathcal{L}$  and solving the problems in (2.27) using the direct application of “hard” demodulation techniques. This requires the solution of each

of the two binary optimization problems in (2.27) for each of the  $MN_t$  bits,  $b_i$  in  $\mathbf{b}$ . Each of these solutions can be obtained (or approximated) using a tree search algorithm (e.g., [14, 15]), or can be approximated by other methods, such as semidefinite relaxation (e.g., [17]).

Given the importance of hard demodulation schemes, and the fact that most of the list-based schemes for soft demodulation use ideas from hard MIMO demodulation methods, we will first review some of the existing hard MIMO demodulation schemes before reviewing some of the existing soft MIMO demodulation methods.

## 2.5.2 Hard MIMO demodulation

For the MIMO communication system in (2.21), the maximum likelihood transmitted bit-vector  $\mathbf{b}_{\text{ML}}$  can be obtained by solving the following discrete optimization problem

$$\mathbf{b}_{\text{ML}} = \arg \min_{\mathbf{b} \in \mathcal{L}} \|\mathbf{y} - \mathbf{H}\mathcal{M}(\mathbf{b})\|_2^2, \quad (2.28)$$

where  $\mathcal{L}$  is the list of all possible  $2^{MN_t}$  transmitted bit-vectors  $\mathbf{b}$ . The computational complexity of solving this problem increases exponentially with the size of the transmitted bit-vector  $\mathbf{b}$  and hence there has been considerable effort to design demodulation schemes with ML or near ML performance with lower computational complexities (e.g., [21, 25, 26, 89–96]). Although demodulation schemes like (unordered) nulling and canceling using zero-forcing (ZF) or minimum mean squared error (MMSE) decision feedback

equalization (e.g., [5,73]) have cubic computational complexity in the problem size, they may incur considerable error performance degradation compared to maximum likelihood demodulation schemes (e.g., [90]). Hence, in this section we will focus on ML or near ML demodulation schemes for applications in which the channel matrix  $\mathbf{H}$  has full column rank, and we will review two groups of these schemes: tree search algorithms and semidefinite relaxation algorithms. (Extensions to the case of a “fat” channel matrix  $\mathbf{H}$  can be based on the principles outlined in [94].)

### 2.5.2.1 Tree search algorithms

In [89], Viterbo and Boutros presented a hard demodulation technique based on the algorithm of Fincke and Pohst [97] for binary quadratic optimization. This demodulator and the family of demodulators that it spanned, have become known as “sphere decoders”. These algorithms provide the ML solution to the hard demodulation problem (2.28). They aim to do so with a reduced computational complexity (in an average sense) and hence they have received quite a lot of attention in the design of hard demodulation algorithms (e.g., [89,91–94,98]).

Let us describe a general sphere decoding algorithm for solving the ML demodulation problem in (2.28) for a channel matrix  $\mathbf{H}$  that has full column rank. For simplicity we assume a mapping scheme  $\mathcal{M}(\cdot)$  that maps the bit-vectors  $\mathbf{b}$  to the symbol-vectors  $\mathbf{s}$  using V-BLAST transmission scheme with elements selected from a finite sub-set of  $\mathbb{Z}$ , the set of integer numbers. We call this finite sub-set,  $\hat{\mathbb{Z}}$ . The channel matrix  $\mathbf{H}$  generates a finite lattice  $\Lambda(\mathbf{H}) = \{\mathbf{x} = \mathbf{H}\mathbf{s} | \mathbf{s} \in \hat{\mathbb{Z}}^{N_t}\}$ , hence, the ML demodulation problem in (2.28) is equivalent to finding a bit-vector  $\mathbf{b}$  for which its equivalent lattice point

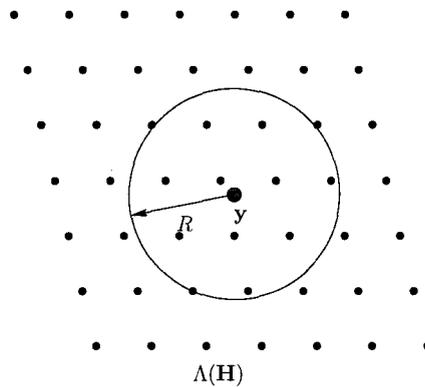


Figure 2.7: A pictorial representation of the received signal vector  $y$ , the receiver lattice  $\Lambda(\mathbf{H})$  and the projection of the decoding sphere to the hyper-plane of the lattice.

$\mathcal{H}\mathcal{M}(\mathbf{b})$  has the minimum Euclidean distance from  $y$ . As shown in Fig. 2.7, the sphere decoding algorithm finds the ML solution to this problem by examining those lattice points which are inside a hyper-sphere centered at  $y$  with a radius  $R$ . In order to do so, the sphere decoding algorithm should first find a proper radius  $R$  and then search the lattice points inside the hyper-sphere in an efficient manner; e.g, [89, 97].

Proper radius selection has a significant impact on the search complexity inside the sphere, as it will determine the number of lattice points inside the sphere. Some schemes select the radius based on the channel noise variance (e.g., [12, 99]), while others set the radius based on lattice properties defined by the channel matrix (e.g., [19, 91, 99]), or as the distance between the received signal  $y$  and the approximate solution to (2.28) based on a low-complexity linear or decision feedback demodulator [100]. In some cases, the selected radius may be so small that the sphere does not contain any of the lattice points. In that case, a trial and error scheme must be used to increase the size of the radius in order for the sphere to cover

some of the lattice points (e.g., [12]).

After selecting the radius of the sphere, the next step is to search for the lattice points inside the sphere. Most of the existing algorithms for doing so are based on branch and bound tree search algorithms (e.g., [95]) for searching the tree interpretation (e.g., [21]) of the cost function  $D_h(\mathbf{s}) = \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2^2$  in (2.28). This tree structure can be obtained by writing  $D_h(\mathbf{s})$  as

$$D_h(\mathbf{s}) = \sum_{i=1}^{N_t} w_i(\mathbf{s}_{1:i}), \quad (2.29)$$

in which,  $w_i(\mathbf{s}_{1:i})$  is a non-negative factor that depends only on the the first  $i$  elements of  $\mathbf{s}$ ,  $\mathbf{s}_{1:i}$ . A sample tree structure that includes some visited nodes is shown in Fig. 2.8. The tree search algorithm, which we will discuss later in this section, provides the ordering in which the branches of this tree are extended. After selecting a node for expansion, its branch metric is compared against the sphere radius (branch metric upper bound) and if it is larger than this upper bound, this branch and all its sub-branches are excluded from the search, reducing the search complexity. After finding a leaf node (which corresponds to a whole symbol-vector  $\mathbf{s}$ ) the sphere radius can be updated to tighten the search space. At the end of the search, the leaf node with the smallest branch metric is the ML solution to (2.28).

Tree search algorithms can be divided into different categories depending on the direction and order that they search the tree. Breadth-first search algorithms, extend the tree by extending the branches of all selected nodes in the current level of the tree before moving to the next level for further expansions (e.g., [97, 101, 102]). Depth-first search algorithms extend the branches of the currently selected node in the tree and move to the next level of the tree to extend the branches of the selected node in the next

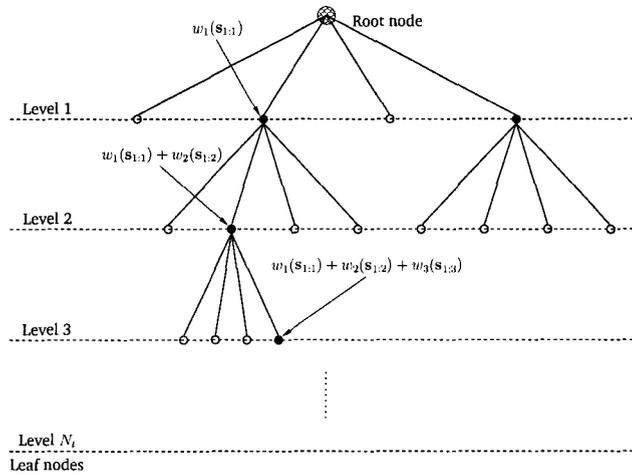


Figure 2.8: Branch and bound tree search method.

level of the tree (e.g., [90, 91]). Best-first search algorithms select the best extended branch of the tree (the branch with the smallest path metric) for expansion; e.g., [101, 103, 104]. A unified framework that encompasses all these tree search algorithms as special cases was recently presented in [21]. This unified framework subsumes the sequential decoding algorithms (including the stack algorithm [103, 104] and the Fano algorithm [105]) as special cases. These algorithms are shown to have excellent performance-complexity trade-offs compared to other tree search algorithms (e.g., [21]). In Chapter 3, soft demodulators based on a multi-stack algorithm will be proposed. We will also review the stack algorithm in more detail in that chapter.

Although the sphere decoding algorithms are computationally efficient for many practical scenarios, both the expected and worst-case computational complexities remain exponential in the problem size (the number of symbols to be detected) for a fixed SNR [22]. The rate of this exponential growth is dependent on the SNR, and can be a serious impediment to

implementation at low SNRs (i.e., near the Shannon capacity limit of the channel) and for large problem sizes [22]. Hence, other detection methods with polynomial complexities (such as semidefinite relaxation schemes) may need to be considered for systems that operate in these regimes. Several such schemes based on semidefinite relaxation will be developed in Chapters 4 and 5. The semidefinite relaxation approach to hard demodulation will be reviewed in Section 2.5.2.3, but first we will review a somewhat different search-based hard demodulation scheme.

### 2.5.2.2 Greedy algorithm

The algorithm in [106, 107] is an ordered search algorithm in which elements of the symbol-vector  $\mathbf{s}$  are evaluated both individually and pairwise, and at each stage of the algorithm the symbol or pair that optimizes the (partial) decision metric is chosen. (Hence the moniker “greedy”.) The order in which the evaluation is performed is in increasing order of the contribution to the metric. More precisely, let  $D_h(\mathbf{s}) = \|\mathbf{y} - \mathbf{H}\mathbf{s}\|_2^2$  denote the maximum likelihood metric. This metric can be written as

$$D_h(\mathbf{s}) = -2\text{Re}\{\mathbf{a}^H \mathbf{s}\} + \mathbf{s}^H \mathbf{B} \mathbf{s} + \mathbf{y}^H \mathbf{y}, \quad (2.30)$$

where  $\mathbf{a} = \mathbf{H}^H \mathbf{y}$  and  $\mathbf{B} = \mathbf{H}^H \mathbf{H}$ . Now let  $\hat{A}_i = |2a_i| + |B_{ii}|$  for  $i = 1, \dots, N_t$ , and let  $\hat{B}_{ij} = |2B_{ij}|$  for  $i = 1, \dots, N_t - 1$  and  $j = i + 1, \dots, N_t$ . These coefficients quantify the impact of each symbol and each pair of symbols, respectively, on the decision metric. The order of evaluation is based on arranging the set  $\{\hat{A}_i, \hat{B}_{ij}\}$  in descending order, and the partial metric is updated as decisions on the elements of  $\mathbf{s}$  are made. If the next element in the ordered set is  $\hat{A}_{i_0}$ , then the symbol  $s_{i_0}$  is evaluated, and the value that

produces the smallest increment in the metric is chosen. If the next element is  $\hat{B}_{i_0 j_0}$ , then the pair of symbols  $(s_{i_0}, s_{j_0})$  is evaluated, and the pair that results in the smallest partial metric is selected. In this basic formulation of the algorithm, only one vector of partial decisions is propagated to the next step. However, the performance of the algorithm can be significantly improved, at some computational cost, by propagating the  $L > 1$  best vectors of partial decisions to the next stage [106, 107]. If the greedy algorithm were only to consider the single symbol decisions corresponding to the elements of  $\{\hat{A}_i\}$ , then [107] when  $L = 1$  it becomes equivalent to certain successive interference cancellation (decision feedback) detectors, and when  $L > 1$  it becomes equivalent to certain constrained tree-search detectors.

### 2.5.2.3 Semidefinite relaxation

For many conventional constellation mappings, such as QPSK (e.g., [17, 96]),  $M$ -ary PSK (e.g., [108]), and for more general constellations in [109], the NP-hard optimization problem in (2.28) can be converted to a boolean quadratic programming (BQP) problem. For example, as we will show in Chapter 4 of this thesis, for QPSK or BPSK signaling, (2.28) can be converted to the following BQP (e.g., [96])

$$\min_{\mathbf{x} \in \{+1, -1\}^K} \mathbf{x}^T \mathbf{Q} \mathbf{x}, \quad (2.31)$$

where  $K$  is the problem size and  $\mathbf{Q}$  contains the channel information and the *a priori* information. By substituting  $\mathbf{X} = \mathbf{x}\mathbf{x}^T$  in (2.31), this problem

can be reformulated as

$$\min_{\mathbf{X}} \text{Trace}(\mathbf{X}\mathbf{Q}) \quad (2.32a)$$

$$\text{s.t. } \mathbf{X} \succeq \mathbf{0}, \quad \text{rank}(\mathbf{X}) = 1, \quad (2.32b)$$

$$[\mathbf{X}]_{ii} = 1, \quad i = 1, \dots, K, \quad (2.32c)$$

where the computational complexity of the problem arises from the rank-1 constraint. The semidefinite relaxation approach (e.g., [96]) to approximate the solution to (2.31) is to relax the rank-1 constraint on  $\mathbf{X}$  and solve the following semidefinite program (SDP):

$$\min_{\mathbf{X}} \text{Trace}(\mathbf{X}\mathbf{Q}) \quad (2.33a)$$

$$\text{s.t. } \mathbf{X} \succeq \mathbf{0}, \quad (2.33b)$$

$$[\mathbf{X}]_{ii} = 1, \quad i = 1, \dots, K. \quad (2.33c)$$

This problem can be solved in polynomial time using interior point methods (e.g., [16]). The solution to this semidefinite program is a matrix, whereas the solution to (2.31) is a vector. As will be described in Chapter 4, an efficient scheme to extract the approximate solution to the BQP in (2.31) solving (2.33) is to use the randomization procedure described in [110,111]. In this method the Cholesky factor of the solution to (2.33) is multiplied by random vectors selected from uniformly distributed vectors on the unit sphere. A candidate bit-vector  $\mathbf{x}$  can be obtained by quantizing the product. After a limited number of randomization iterations the best vector among these vectors is selected as the approximate solution to (2.31).

It has been shown (e.g., [96]) that the semidefinite relaxation scheme

is an efficient and accurate method for approximately solving optimization problems of the form in (2.31) and it has been used to develop several hard demodulators, including a near-ML multiuser demodulator with application to synchronous CDMA systems [96], a blind near-ML demodulator for orthogonal space-time block codes [112], a multiuser demodulator for CDMA systems with  $M$ -ary PSK signaling [108], near-ML demodulators for MIMO systems with 16-QAM signaling [25] and higher-order QAM signaling [26], and a near-ML demodulator for MIMO systems with any general constellation mapping [109]. In this thesis we will use the randomization procedure inherent in solving these problems to propose some low complexity soft MIMO demodulation schemes; cf. Chapters 4 and 5.

### 2.5.3 Approximate soft MIMO demodulation based on hard demodulation

In the previous section, several hard demodulation schemes were described, but the focus of the thesis will be on reduced complexity soft demodulation schemes. In this section and the next we will describe how these hard demodulation schemes can be exploited for soft demodulation.

As mentioned in Section 2.5.1, one of the standard approaches to reducing the computational cost of extracting the LLR (2.22) involves the use of the following approximation:

$$\lambda_{D1,i} \simeq \frac{1}{2\sigma^2} \left( \min_{\mathbf{b} \in \mathcal{L}_{i,-1}} D(\mathbf{b}) - \min_{\mathbf{b} \in \mathcal{L}_{i,+1}} D(\mathbf{b}) \right), \quad i = 1, \dots, MN_t. \quad (2.34)$$

Hence, for each transmitted bit  $b_i$  the following two hard demodulation problems should be solved in each iteration for each channel use

$$\mathbf{b}_{i,b} = \arg \min_{\mathbf{b} \in \mathcal{L}_{i,b}} D(\mathbf{b}), \quad i = 1, \dots, MN_t, b = +1, -1. \quad (2.35)$$

The hard demodulation schemes discussed in the previous section can be used to solve or approximate these hard demodulation problems. For example: variations of the hard sphere decoding algorithm were used in [14, 15]; a generalization of the Twin-stack tree-search algorithm [113] was used in [114] (for the case of a frequency-selective MIMO channel); the greedy algorithm was used in [106]; and the semidefinite relaxation technique was used in [17]. (The scheme in [17] is called the Multi-SDR scheme in this thesis.)

Since several hard demodulation problems need to be solved in each channel use, the computational complexity of this approach can be prohibitive for practical applications. Hence, in this thesis we will focus on list-based soft demodulation schemes (described in the next section) which can have lower computational complexities.

## 2.5.4 Approximate soft MIMO demodulation based on list approximations

The second approach to reduced complexity approximate soft demodulation schemes in Section 2.5.1 was based on approximating the LLR using

$$\lambda_{D1,i} \simeq \log \frac{\sum_{\mathbf{b} \in \hat{\mathcal{L}}_{i,+1}} \exp(-D(\mathbf{b})/(2\sigma^2))}{\sum_{\mathbf{b} \in \hat{\mathcal{L}}_{i,-1}} \exp(-D(\mathbf{b})/(2\sigma^2))} \quad (2.36)$$

$$\simeq \frac{1}{2\sigma^2} \left( \min_{\mathbf{b} \in \hat{\mathcal{L}}_{i,-1}} D(\mathbf{b}) - \min_{\mathbf{b} \in \hat{\mathcal{L}}_{i,+1}} D(\mathbf{b}) \right), \quad i = 1, \dots, MN_t, \quad (2.37)$$

where  $\hat{\mathcal{L}}$  is the list of carefully selected bit-vectors with dominant values for  $D(\mathbf{b})$  which replaces  $\mathcal{L}$  in (2.22).

In some applications, a suitable set of bit-vectors for the list is the set of vectors with metrics below a given threshold; i.e.,  $\hat{\mathcal{L}} = \{\mathbf{b} | D(\mathbf{b}) \leq R\}$ . Such a list can be constructed using sphere decoding algorithms by fixing the sphere radius to an appropriate value (or equivalently by fixing an upper bound in branch and bound methods) and collecting all the generated leaf nodes into  $\hat{\mathcal{L}}$ . Hochwald and ten Brink [12] used a sphere decoding algorithm to construct this list. In the first step of the method in [12], the sphere radius is determined. Then, all the bit-vectors which have metric values inside the defined sphere are generated via the implicit tree search in the sphere decoding algorithm. If the list size is constrained, the bit-vectors with the smallest metric values are selected for list membership. In the scheme in [12] the list is constructed only in the first demodulation-decoding iteration, and hence it is not adapted to the updated *a priori* information in subsequent iterations. That is, the list generated in the first iteration is stored in the memory to be used for soft information extraction in the subsequent

demodulation-decoding iterations. Since the cost function  $D(\mathbf{b})$  changes in each iteration, the desired sphere position changes and the list of bit-vectors constructed in the first demodulation-decoding iterations may not necessarily contain all the bit-vectors inside the sphere associated with the new cost function. This issue was addressed by Vikalo, et. al. [85], who modified the sphere decoding scheme of Hochwald and ten Brink [12] by updating the bit-vector candidates list in the second and subsequent iterations using a new sphere that is based on the updated cost function.

Some other list-based soft demodulation schemes, such as the LISS algorithm proposed in [18, 86], use the stack algorithm [103, 104], which is a best-first tree search algorithm, to generate the list. The schemes that will be proposed in Chapter 3 of this thesis, are based on a “multi-stack” algorithm, which generates the list of bit-vectors in an arguably more efficient way. Hence, the proposed algorithm can offer an improved trade-off between performance and computational complexity than some existing methods.

In Chapters 4 and 5 some alternative list-based soft demodulation algorithms based on semidefinite relaxation will be proposed. It will be shown that the proposed algorithms provide reduced computational costs compared to demodulators based on sphere decoding (e.g., [12]), demodulators based on minimum mean squared error with soft interference cancellation (MMSE-SIC) (e.g., [24]) and the method in [17], in which semidefinite relaxation is used in the context of the hard demodulation approach to soft demodulation in Section 2.5.3. It will also be shown that this reduction in computational cost is achieved without a significant degradation in performance.

## 2.5.5 Other soft MIMO demodulation approaches

There are several other approaches to soft MIMO demodulation that cannot be grouped in the hard demodulation based or list based approaches for soft demodulation, including the scheme based on minimum mean square error (spatial) equalization with soft interference cancellation (MMSE-SIC) (e.g., [24]), and a scheme in which the channel modeled as a graph and a low complexity message passing scheme is used to approximate the likelihoods [48]. These schemes are briefly reviewed below.

The MMSE-SIC scheme is a low-complexity soft MIMO demodulator that has been shown to have good performance (e.g., [24, 76, 115]). In this scheme the unbiased MMSE estimate of each symbol is obtained conditioned on the soft information about the other symbols provided by the decoder. By approximating the residual interference on each symbol as a Gaussian random variable, the soft information of each symbol is extracted as if the symbol were transmitted thorough a single-input single-output AWGN channel. The resulting computational cost is  $O(N_t^4) + O(N_t M 2^M)$  per demodulation-decoding iteration (e.g., [24, 76, 115]), where  $M$  is the number of bits transmitted per antenna per channel use.<sup>5</sup> Given the popularity of the MMSE-SIC scheme as a low complexity approach to soft MIMO demodulation that provides good performance, this scheme is described in more detail in Appendix A and the performance-complexity trade-offs of the demodulators proposed in Chapters 4 and 5 will be compared against that of the MMSE-SIC scheme.

---

<sup>5</sup>This can be compared to the cost using the Greedy detection algorithm described in Section 2.5.2.2 in the hard demodulation approach to soft demodulation, which is  $O(LM2^{2M}N_t^5)$ , where  $L \geq 1$  is a parameter of the Greedy algorithm.

A substantially different soft MIMO demodulation scheme was recently proposed in [48]. That scheme is based on a factor-graph representation of the channel and the application of the belief propagation (BP) algorithm [116] (see also [45, 117]) to that factor graph. As such, it is especially useful for systems with a large number of antennas and a channel matrix with a large fraction of relatively small gains. In its raw form, the BP algorithm remains computationally expensive, due to the cost of the operations at the nodes that represent the signal received by each antenna. However, a variety of schemes by which this cost can be reduced were proposed. These schemes involve the assessment of the impact of each edge in the graph. Edges that are deemed to have a minor impact are deleted and the interference represented by the edge is replaced by that of an independent Gaussian noise source of the same variance. In the simplest of these complexity reduction schemes,  $d_f$  edges are retained at each ‘receiver’ node, and the computational cost of the resulting BP algorithm is  $O(d_f N_r 2^{M d_f})$ .

## 2.6 Summary

In this chapter we have reviewed some of the principles of single antenna and multiple antenna wireless communication systems. We have discussed transmission and reception schemes for single antenna communication systems, including the trellis coded modulation (TCM) and the bit-interleaved coded modulation (BICM) schemes. We have argued that in fading channels BICM, when used in conjunction with iterative demodulation and decoding (IDD) schemes, provides good performance at SNRs quite close to capacity limits of the channel, and provides a means to balance the performance-complexity trade-off for these systems.

We have also reviewed some basic space-time transmission schemes, including the linear-dispersion codes and the BLAST architectures. These space-time schemes can exploit the diversity and multiplexing gains of multiple antenna systems, and when used within a BICM-IDD framework, they can provide good performance-complexity trade-offs. A key bottleneck in the implementation of such schemes is the complexity of soft demodulator, which increases exponentially with the number of bits transmitted per channel use. We reviewed some reduced complexity schemes for soft demodulation based on tree search algorithms and semidefinite relaxation. In the next chapter we will propose a multi-stack algorithm that can provide reduced computational complexities compared to some existing soft demodulation schemes with reasonable performance results.

# Chapter 3

## A multi-stack algorithm for soft MIMO demodulation

'There's an old story about the person who wished his computer were as easy to use as his telephone. That wish has come true, since I no longer know how to use my telephone.'

---

BJARNE STROUSTRUP

**I**N THE previous chapter we provided a broad review of some fundamental principles of single antenna and multiple antenna communication systems. We reviewed several available transmission schemes that exploit the diversity-multiplexing gains of the communication channel. One pragmatic approach that balances the computational complexity and performance is the MIMO-BICM-IDD scheme. These schemes require the design of a computationally feasible soft MIMO demodulator. In this chapter we propose a family of list-based soft demodulators for MIMO communication systems based on a multistack algorithm, proposed herein, for traversing the tree structure that is inherent in the MIMO demodulation problem. The existing

stack algorithm for MIMO soft demodulation stores a single stack of visited nodes in the tree, and expands the stack using the ‘best-first’ principle, as quantified by the (partial) likelihoods of the nodes. In the proposed multistack algorithm, the single stack is partitioned into a stack for each level of the tree, and the algorithm proceeds by performing one best-first search step in each of these stacks in the natural ordering of the tree. By assigning appropriate priorities to the level at which this ‘best-first search per level’ processing re-starts once a leaf node has been obtained, the proposed demodulators can achieve trade-offs between performance and complexity that dominate those of several existing methods, including the stack algorithm, in the low-complexity region.

### 3.1 Introduction

In Chapter 2 we mentioned that MIMO wireless communication systems are attractive because they provide the potential for reliable communication at substantially higher data rates than the corresponding single antenna system [6, 57]. However, the computational effort required to achieve these high spectral efficiencies is often beyond the capabilities of the envisioned communication devices, and hence there has been considerable interest in the development of transceivers that balance the competing demands of spectral and computational efficiency. As mentioned in Chapter 2 a popular transceiver architecture for balancing these demands is a MIMO version of bit interleaved coded modulation (BICM, e.g., [11]) with block-by-block transmission and (bit-wise) iterative ‘soft’ demodulation and decoding (IDD) at the receiver; e.g., [12]. In such MIMO-BICM-IDD schemes, a key computational bottleneck is the demodulation step; that is, the extraction of an

approximation of the log likelihood ratio (or an approximation thereof) of each of the bits transmitted in a given block from the corresponding output block of the MIMO channel. As explained in Chapter 2, the design of list-based techniques to manage this computational burden is the core topic of this thesis.

The goal of list-based soft demodulation for block-based MIMO transmission is to (efficiently) obtain a list of candidate bit-vectors<sup>1</sup> that generate the dominant components of the likelihoods for a given block, and then to approximate the log likelihood ratio of each bit transmitted in that block using the members of the list; e.g., [12, 18–20, 85–88]. A popular class of approaches to efficient list generation (e.g., [12, 18, 19, 85–88]) is based on the tree-search representation of the MIMO demodulation problem; e.g., [21, 95]. In that representation, the metric that is used to assess the significance of each bit-vector corresponds to an additive path metric in a tree, with non-negative branch metrics. The leaf nodes in the tree represent the complete bit-vectors, and the dominant bit-vectors correspond to leaf nodes with small path metrics. A feature of the tree-search representation is that in an IDD receiver, the extrinsic information provided by the previous iteration of the decoder can be easily incorporated into the branch metric; e.g., [18, 19, 85–88].

Once the list generation problem has been associated with the search for leaf nodes with small path metrics, a number of conventional tree search algorithms can be applied. Of particular interest is the extension of the stack algorithm for ‘hard’ demodulation (e.g., [21]) to the process of list generation; e.g., [18, 86, 88]. The stack algorithm adopts a ‘best-first’ search strategy in which the exposed nodes of the tree are stored in a global stack, and

---

<sup>1</sup>That is, candidates for the vector of bits transmitted in the given block.

the algorithm proceeds by expanding the node in the stack with the smallest path metric, until a leaf node is selected for expansion. (That leaf node is the best leaf node in the tree.) The natural extension of the stack algorithm to list generation simply involves continuing the search for the next best leaf node; e.g., [18, 86, 88]. Hence, the stack algorithm generates bit-vectors in order of their path metrics.

While the stack algorithm generates an ordered list, the rapid growth of the stack size and the consequent complexity of finding the next best list member are significant impediments to its implementation in list-based soft demodulation [88]. The goal of this chapter is to propose demodulators that provide greater control over the complexity-performance trade-off by constructing a tree-search algorithm that generates a sizeable collection of ‘good’ list members in the early stages (though not necessarily in order), so that good performance can be obtained even if the algorithm is terminated for reasons of complexity. The key aspect of the proposed multistack algorithm is that the (global) stack is partitioned into one stack per level in the tree. The algorithm then proceeds by performing one best-first search step per level of the tree in the natural ordering of the tree. We will show that by assigning appropriate priorities to the level at which this ‘best-first search per level’ processing re-starts, and by incorporating natural termination criteria, the proposed demodulators can achieve trade-offs between performance and complexity that dominate those of several existing methods in the low complexity region.

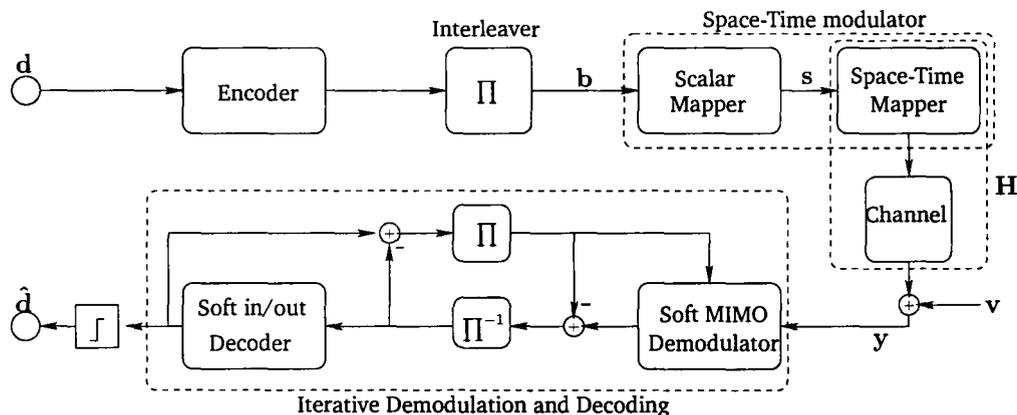


Figure 3.1: MIMO BICM-IDD transceiver.

## 3.2 System model

We will consider the coherent narrowband MIMO-BICM-IDD transceiver structure illustrated in Fig. 3.1 (e.g., [12]) which is the same as Fig. 2.2 in Chapter 2, where the space-time modulator is the concatenation of a scalar constellation mapper and any (widely) linear space-time block code; e.g., [67]. We will consider scalar constellations of size  $2^M$  and the space time block code will transmit  $K$  such symbols per block channel use. We will let  $\mathbf{b}_n$  denote the vector of  $MK$  bits from the interleaved encoded bit stream that are mapped to the  $K$  symbols at the  $n$ th channel use,  $\mathbf{s}_n = \mathcal{M}(\mathbf{b}_n)$ , where  $\mathcal{M}(\cdot)$  is the corresponding mapping. Hence, the vector of received samples at the  $n$ th channel use can be written as

$$\mathbf{y}_n = \mathbf{H}_n \mathbf{s}_n + \mathbf{v}_n = \mathbf{H}_n \mathcal{M}(\mathbf{b}_n) + \mathbf{v}_n, \quad (3.1)$$

where  $\mathbf{H}_n$  is the equivalent channel matrix (e.g., [67]) at the  $n$ th channel use, (and is assumed to be known at the receiver), and  $\mathbf{v}_n$  is a vector of noise

samples, which will be assumed to be from a zero mean, additive white circular Gaussian noise (AWGN) model with variance  $\sigma^2$  per real scalar dimension. We will focus on cases in which the space time block code is configured so that  $\mathbf{H}_n$  is square or tall.

As described in Chapter 2, since the emphasis of this thesis is on the demodulation step, the outer encoder in Fig. 3.1 can be any binary encoder, and we will adopt its corresponding soft-input soft-output decoder at the receiver. The role of the soft MIMO demodulator in Fig. 3.1 is to compute the log likelihood ratios of each of the  $NMK$  interleaved encoded bits, based on the channel measurements and the extrinsic information from previous decoder iterations. This ‘soft information’ is then passed to the outer soft decoder. Since the channel model in (3.1) is memoryless, the soft demodulator can operate on a block-by-block basis. For notational simplicity we will drop the subscript  $n$  in (3.1) and consider a generic block channel use. In that case, the soft demodulator computes (or approximates) the (conditioned) log likelihood ratio (LLR) for each element  $b_i$  of  $\mathbf{b}$  (e.g., [12]):

$$\lambda_i \triangleq \log \frac{p(b_i = +1|\mathbf{y}, \mathbf{H})}{p(b_i = -1|\mathbf{y}, \mathbf{H})} = \log \frac{\sum_{\mathcal{L}_{i,+1}} p(\mathbf{y}|\mathbf{b}, \mathbf{H})p(\mathbf{b})}{\sum_{\mathcal{L}_{i,-1}} p(\mathbf{y}|\mathbf{b}, \mathbf{H})p(\mathbf{b})}, \quad (3.2)$$

where  $\mathcal{L}$  is the list of all  $2^{MK}$  binary vectors  $\mathbf{b}$ ,  $\mathcal{L}_{i,b} \triangleq \{\mathbf{b} \in \mathcal{L} | b_i = b\}$ . As described in Chapter 2, under the assumed AWGN noise model and assuming that the interleaver is good enough for the conventional approximation  $p(\mathbf{b}) \approx \prod_{i=1}^{MK} p(b_i)$  to hold, the summands in (3.2) can be written as  $e^{-D(\mathbf{b})/(2\sigma^2)}$  (e.g., [85]), where

$$D(\mathbf{b}) \triangleq \|\mathbf{y} - \mathbf{H}\mathcal{M}(\mathbf{b})\|_2^2 - 2\sigma^2 \sum_{i=1}^{MK} \log p(b_i). \quad (3.3)$$

Since each list  $\mathcal{L}_{i,b}$  contains  $2^{MK-1}$  terms, there has been considerable interest in schemes that enable the approximation of (3.2) by replacing  $\mathcal{L}$  with a carefully selected reduced size list,  $\hat{\mathcal{L}}$ , that contains the dominant summands in (3.2), [12, 18–20, 85–88].

The dominant summands in (3.2) correspond to binary vectors  $\mathbf{b}$  that yield small values for  $D(\mathbf{b})$ . The search for such vectors is significantly simplified when the QR decomposition is used to make the inherent  $M$ -ary tree structure of the MIMO demodulation problem explicit; e.g., [12, 18, 21, 85–87, 95]. In particular, if we let  $\mathbf{HE} = \mathbf{QR}$  denote the QR decomposition<sup>2</sup> of  $\mathbf{HE}$ , where  $\mathbf{E}$  is a column permutation matrix that determines the arrangement of the symbols in the tree, and if we define  $\tilde{\mathbf{y}} \triangleq \mathbf{Q}^\dagger \mathbf{y}$ ,  $\tilde{\mathbf{v}} \triangleq \mathbf{Q}^\dagger \mathbf{v}$ ,  $\tilde{\mathbf{s}} \triangleq \mathbf{E}^\dagger \mathbf{s}$ , and  $\mathbf{R}_j$  to be the  $j$ th row of  $\mathbf{R}$ , then (3.3) can be rewritten as:

$$D(\mathbf{b}) = \sum_{j=0}^{K-1} |\tilde{y}_{K-j} - \mathbf{R}_{K-j} \tilde{\mathbf{s}}|^2 - 2\sigma^2 \log p(\tilde{s}_{K-j}). \quad (3.4)$$

Here,  $p(\tilde{s}_i) = \prod_{\ell=(i-1)M+1}^{iM} p(b_\ell)$ , where the product is over those bits that index the symbol  $\tilde{s}_i$ . In (3.4), the  $j$ th summand depends only on symbols  $K-j$  to  $K$ , and hence the inherent tree structure is exposed; e.g., [21]. In particular, we can assign the possible values for the  $j$ 'th summand to be the metrics of the branches emanating from the nodes at the  $j$ th level of the tree (with level 0 being the root). Since each  $\tilde{s}_{K-j}$  comes from an  $M$ -ary constellation, there will be  $M$  branches emanating from each node. For later convenience, we observe that the path metric for a node at level  $L$  in the

<sup>2</sup>We consider the conventional QR decomposition in which  $\mathbf{Q}^\dagger \mathbf{Q} = \mathbf{I}$ , where  $(\cdot)^\dagger$  denotes the (conjugate) transpose, and  $\mathbf{R}$  is an upper-triangular matrix with non-negative diagonal elements [118].

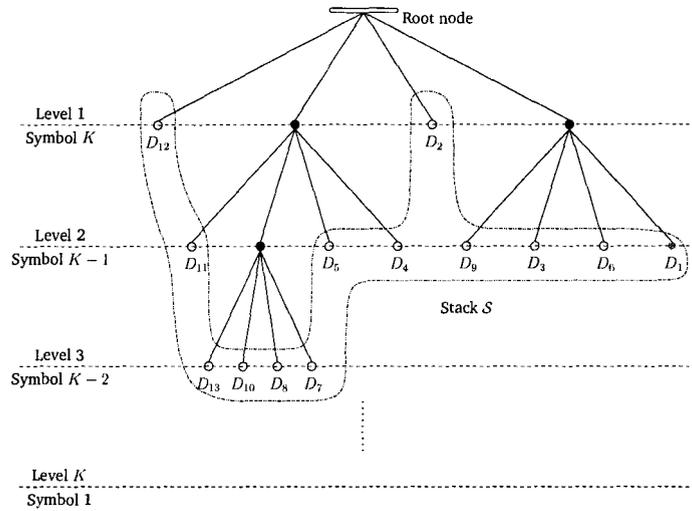


Figure 3.2: A snapshot of an instance of the stack algorithm for a system in which  $K$  symbols from a 4-ary constellation are transmitted in each (block) channel use.

tree is the sum of the first  $L$  terms in (3.4) and this path metric is additive, with non-negative branch metrics.

One approach to searching a tree for the leaf node with the smallest path metric is to employ the stack algorithm (e.g., [21]), in which all the exposed nodes of the tree are stored in a stack  $\mathcal{S}$ . This algorithm proceeds in a ‘best-first’ manner by selecting the node in the stack with the smallest metric and replacing it by its child nodes. The first leaf node that the algorithm selects for expansion corresponds to the bit-vector with the smallest value for  $D(\mathbf{b})$ . A snapshot of an instance of the stack algorithm is provided in Fig. 3.2, where the exposed nodes, shown with empty circles, have been labeled in increasing order of their path metrics. All these nodes are stored in the global stack  $\mathcal{S}$ , which is illustrated by the dashed closed curve. The next step in the algorithm would be to expand the node marked  $D_1$ .

If the stack algorithm is continued to search for the ‘next best’ leaf

nodes, it will produce leaf nodes in increasing order of  $D(\mathbf{b})$ , and the corresponding bit-vectors constitute a candidate list for demodulation purposes; e.g., [18, 86, 88]. However, such a scheme may explore many internal nodes in the tree before it reaches the leaf node with the next smallest value for  $D(\mathbf{b})$ , and hence may expend significant computational effort and memory resources to find only a few dominant leaf nodes. The multistack algorithm proposed in the following section provides greater control over the trade-off between performance and complexity, so that a large subset of the dominant leaf nodes can be obtained for lower computational cost.

### 3.3 Multi-stack algorithm

While the conventional stack algorithm (e.g., [21]) employs a single ordered stack of nodes,  $\mathcal{S}$ , for the whole tree, we propose to partition the stack into separate stacks,  $\mathcal{S}_k$ , for each level in the tree. In the conventional stack algorithm, the best node in the (global) stack is removed and is replaced by its child nodes. In the proposed multistack algorithm, when a stack is processed the node with the smallest path metric in the stack is removed and its child nodes are placed in the stack at the next lower level of the tree. A snapshot of an instance of the multistack algorithm is provided in Fig. 3.3, where the nodes in each stack are grouped together by a dashed closed curve and are labeled in increasing order of their path metrics within their stack. If the stack at level 3 is selected for processing, the node to be expanded will be that labeled  $D_1^3$ .

The partitioning of the (global) stack into a stack for each level naturally generates an additional degree of freedom: the order in which the

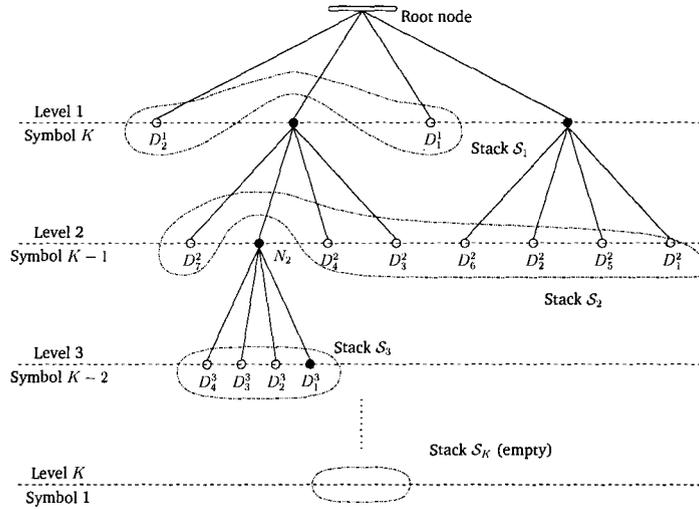


Figure 3.3: A snapshot of an instance of the proposed multistack algorithm for a system in which  $K$  symbols from a 4-ary constellation are transmitted in each (block) channel use.

stacks are processed. In an attempt to obtain ‘good’ leaf nodes without excessive processing, we will focus on orderings in which the next stack to be processed is the stack at the next lower level of the tree. As there are  $K$  levels in the tree, this guarantees that the next leaf node will be found in at most  $K$  steps.<sup>3</sup> Once a leaf node has been obtained, there is a degree of freedom in the level at which the search is re-started, and this choice provides some control over the way in which the tree is explored; see Section 3.3.1.

Although this degree of design freedom is a distinct advantage of the proposed algorithm, leaf nodes are no longer produced in increasing order of  $D(\mathbf{b})$ , and hence we need to ensure that the algorithm does not expend computational effort exploring paths with large metrics. To do so, we consider only those nodes with path metrics below a certain threshold,  $B$ , that

<sup>3</sup>Note that, by themselves, these  $K$  steps do not necessarily create a contiguous path to a leaf node. They merely expose the child nodes of the best node in the stack at each level of the tree.

is computed using a preliminary greedy depth first search of the tree; see Section 3.3.2.

A key feature of the proposed algorithm is that it generates a sizeable collection of good leaf nodes in the early stages of the algorithm, and we will exploit this feature by providing explicit termination criteria based on the size of the list, and/or the number of nodes visited in the tree. These termination criteria enable the algorithm to be tailored to the computational resources at hand; see Section 3.3.3.

A formal statement of the proposed algorithm is provided in Table 3.1, and in the following subsections we will discuss some of the features of the algorithm in more detail.

### 3.3.1 Symbol and re-start orderings

The conventional stack algorithm has a single degree of freedom, the ordering of the symbols in the tree. In our notation, this is controlled by the permutation matrix  $E$  that is implicit in (3.4). The multistack algorithm introduces an additional degree of freedom: the order in which we search for a non-empty stack (at which we re-start the core algorithm) after having obtained a leaf node.<sup>4</sup> That ordering will be denoted by  $t$ , and the best-first search per level processing is re-started from the first non-empty stack that  $t$  suggests. The variables  $E$  and  $t$  are set prior to each demodulation iteration, and enable the designer to exert some control over the way in which the tree is explored. Some candidate orderings are described below:

---

<sup>4</sup>or after having encountered a stack whose nodes all have path metrics greater than  $B$ .

### 3.3.1.1 V-BLAST symbol and re-start orderings

In the first iteration, no *a priori* information is available and, since the channel is known at the receiver, a natural choice for the symbol ordering,  $\mathbf{E}$ , is the V-BLAST ordering [73], in which the symbol to be expanded at the next level of the tree is the one with the largest SINR. In the generation of the list, it may be fruitful to examine those symbols with low SINR in the greatest detail. This suggests a choice of  $\mathbf{t} = [K, K - 1, \dots, 1]$ . In subsequent demodulation iterations we will retain the same orderings, which means that in this case the ordering of the search is determined by the channel and noise realization, and that the decoder exerts no influence over the ordering.

### 3.3.1.2 Symbol and re-start orderings based on *a priori* information

The principle of the V-BLAST ordering is to place the symbols about which we are most confident at the top of the tree. When we have *a priori* information (i.e., after the first demodulation iteration), we can choose to use the likelihoods provided by the decoder as the measure of confidence, instead of the SINR. In particular, if we let  $P(s_j^*)$  denote the largest of the prior probabilities for symbols at level  $j$ , we can arrange the symbols in descending order of  $P(s_j^*)$ . (We will use the V-BLAST ordering for the first iteration.) As the deep nodes in the tree represent the symbols about which we are least confident, we will use the re-start ordering  $\mathbf{t} = [K, K - 1, \dots, 1]$ .

### 3.3.1.3 Re-start ordering based on *a priori* information

A weakness of the previous ordering is that at each demodulation-decoding iteration the *a priori* information is updated and hence  $\mathbf{E}$  may change. If  $\mathbf{E}$  does change, then the QR decomposition of  $\mathbf{H}\mathbf{E}$  implicit in (3.4) will have

to be repeated, and this adds to the computational cost of the algorithm. An alternative is to sort the stacks, instead of the symbols. That is, we retain the V-BLAST symbol ordering, and, upon re-start, we examine the stacks in increasing order of  $P(s_j^*)$ .

### 3.3.1.4 Natural re-start order

In this approach, we order the symbols according to the V-BLAST ordering and, upon re-start, we examine the stacks in their natural order. That is,  $t = [1, 2, \dots, K]$ , and we examine the stacks starting from the top of the tree. The motivation for doing so is to provide a diverse collection of candidate paths in the stacks at each level of the tree.

## 3.3.2 Bounding the path metrics

One of the difficulties encountered in the direct application of best-first search strategies to MIMO soft demodulation is the breadth of the nodes that are visited, and the consequent computational cost and memory requirement. We address this issue by allowing the algorithm to cut from the tree, the sub-trees below nodes with a path metric greater than a prespecified bound,  $B$ . To ensure that this bound is adapted to the channel realization and the *a priori* information, it is determined by first performing a preliminary greedy depth-first search<sup>5</sup> that generates a single leaf node and then selecting that node's path metric as the threshold,  $B$ . (This preliminary search also populates the stacks at each level of the tree.) In the first demodulation iteration, when there is no *a priori* information, the resulting leaf

---

<sup>5</sup>In a greedy depth first search, the levels of the tree are expanded sequentially and the child node with the smallest branch metric is selected at each step.

node corresponds to the output of a zero-forcing decision feedback detector with the ordering prescribed by  $\mathbf{E}$ , [95]; the Babai point, e.g., [21].

### 3.3.3 Bounding the complexity

If we were to run the proposed algorithm until all leaf nodes with a path metric less than  $B$  were found, then our approach would be reminiscent of some adaptations of the Sphere Decoding (SD) algorithm to list-based soft demodulation; e.g., [12, 19, 85].<sup>6</sup> However, the goal of the proposed algorithm is to generate a sizeable collection of good leaf nodes in the early stages, so that good performance can be obtained even if the algorithm is terminated, before all leaf nodes with metrics less than  $B$  are found. Since the dominant operations in the tree search are those that are repeated at each node, and since the size of the list determines the complexity of computing the list approximation of the LLRs, the key factors in the computational cost of the algorithm are the number of nodes visited in the tree search and the size of the list. The proposed algorithm provides explicit control over both these terms, and we will show in Section 3.5 that these controls provide a convenient way to explore the performance-complexity trade-off.

## 3.4 Likelihood computation

The goal of the multistack algorithm in Section 3.3 (and that of the related algorithms [12, 18, 19, 85–88]) is to efficiently construct a reduced-sized list  $\hat{\mathcal{L}}$  with which the LLRs in (3.2) can be approximated. However, after generating  $\hat{\mathcal{L}}$  using the algorithm in Table 3.1 (or the other algorithms), there may

<sup>6</sup>That said, our simple choice for the threshold  $B$  avoids the “trial and error” methods in some approaches to list sphere decoding; e.g. [12].

Table 3.1: Proposed List Construction Algorithm

---

**Input data:**  $\mathbf{y}$ ;  $\mathbf{H}$ ;  $p(b_k)$ ;  $K$ ; a bound on the list size,  $L$ ; a bound on the number of nodes visited,  $N$ .

**Variables:**  $\mathbf{E}$ ; one stack per level,  $S_k$ ; the search order of the stacks,  $\mathbf{t}$ ; a bound on the path metric,  $B$ .

**Output:** the list,  $\hat{\mathcal{L}}$ .

**Preparatory computations:**

- 1) Using  $\mathbf{y}$ ,  $\mathbf{H}$ , and  $p(b_k)$ , select  $\mathbf{E}$  and  $\mathbf{t}$ . Perform the QR decomposition of  $\mathbf{HE}$ .

**Preliminary step: Greedy depth first search**

- 2) Generate the child nodes of the root node and place them in  $S_1$ . Set  $k = 1$ .
- 3) While  $k \leq K - 1$ , remove from  $S_k$  the node with the smallest metric. Generate all that node's child nodes and place them in  $S_{k+1}$ . Increment  $k$ .
- 4) ( $k = K$ ) Select the node from  $S_K$  with the smallest metric, and place it in the list  $\hat{\mathcal{L}}$ . Set  $B$  to the path metric of this node. Clear  $S_K$ .

**Bounded best first search per level**

- 5) Examine the stacks in the order imposed by  $\mathbf{t}$  and select the first non-empty stack. If all stacks are empty, terminate. Otherwise, set  $k$  to the index of the first non-empty stack.
  - 6) Select the node in  $S_k$  with the smallest path metric. If that metric is greater than  $B$ , clear  $S_k$ , and return to 5. Otherwise, remove this node from  $S_k$ , and generate all its child nodes.
    - a. If  $k < K$ , place the child nodes into  $S_{k+1}$ . If the number of nodes visited is  $< N$ , increment  $k$  and return to 6, otherwise, terminate.
    - b. If  $k = K$ , put those child nodes with metrics  $\leq B$  into  $\hat{\mathcal{L}}$ . If the number of nodes visited is  $< N$  and  $|\hat{\mathcal{L}}| < L$ , return to 5, otherwise, terminate.
-

be bit-positions for which  $\hat{\mathcal{L}}_{i,+1}$  or  $\hat{\mathcal{L}}_{i,-1}$  is empty, and such cases make the list approximation of the LLRs problematic. Therefore, we will generate an ‘enriched’ list  $\hat{\mathcal{L}}'$  by adding all those bit vectors that are within a Hamming distance of one of at least one member of  $\hat{\mathcal{L}}$ ; e.g., [20]. This enriched list can be generated by simply flipping one bit at a time of each list member.<sup>7</sup>

Once the enriched list has been constructed, the LLR in (3.2) can be approximated by performing the ‘max-log’ approximation (e.g., [12]) over the sub-lists  $\hat{\mathcal{L}}'_{i,+1}$  and  $\hat{\mathcal{L}}'_{i,-1}$ , respectively. That is, we will approximate the LLR using only the dominant vector in each sublist. To guard against severe over or under estimation of soft information caused by the list and max-log approximations (e.g., [87]), we will employ the common practice of clipping the approximated LLRs to a certain range (e.g., [87]); in our case, to the interval  $[-5, 5]$ .

---

<sup>7</sup>If the original list  $\hat{\mathcal{L}}$  has  $L$  members, then this enriched list has at most  $L(MK + 1)$  members, but simulation results indicate that many of the bit-flipped vectors are already members of  $\hat{\mathcal{L}}$ , and hence  $L' = |\hat{\mathcal{L}}'|$  is typically much smaller than  $L(MK + 1)$ .

### 3.5 Simulation results

In order to evaluate the performance and computational cost of various soft demodulation strategies, we consider MIMO systems with a narrowband Rayleigh block-fading channel with channel gains that are i.i.d. zero-mean circular complex Gaussian random variables of unit variance. To facilitate comparisons of our results with those in [12, 18], we employ the same transmitter and receiver components and parameters. That is, at the transmitter we use a rate-1/2 punctured parallel concatenated turbo code with block length 8,192 and (5,7) recursive systematic convolutional codes as the component codes, and the V-BLAST transmission scheme [73].<sup>8</sup> At the receiver, we use the conventional BCJR algorithm to decode the constituent codes of the turbo code. We perform 8 turbo decoding iterations before we pass the soft information back to the demodulator, and 4 demodulation-decoding iterations. We will consider two MIMO systems; a system with 4 transmit and 4 receive antennas with Gray-mapped 16-QAM symbols, and an  $8 \times 8$  MIMO system with Gray-mapped QPSK symbols. The size of the complete list for both of these systems is  $|\mathcal{L}| = 65,536$ .

In Figs. 3.4, 3.5, 3.6 and 3.7 we have provided performance-versus-complexity trade-off curves for a variety of soft MIMO demodulators for these two MIMO systems.<sup>9</sup> Performance is measured in terms of the SNR required to achieve a bit error rate (BER) of  $10^{-4}$  after four

---

<sup>8</sup>The (different) interleavers in the turbo code and in the BICM transmitter are selected from randomly generated candidates in each Monte-Carlo iteration.

<sup>9</sup>Using these trade-off curves, one can develop a notion of an ‘efficient frontier’ for soft MIMO demodulation that is similar in spirit to the efficient frontier for hard MIMO demodulation in [119].

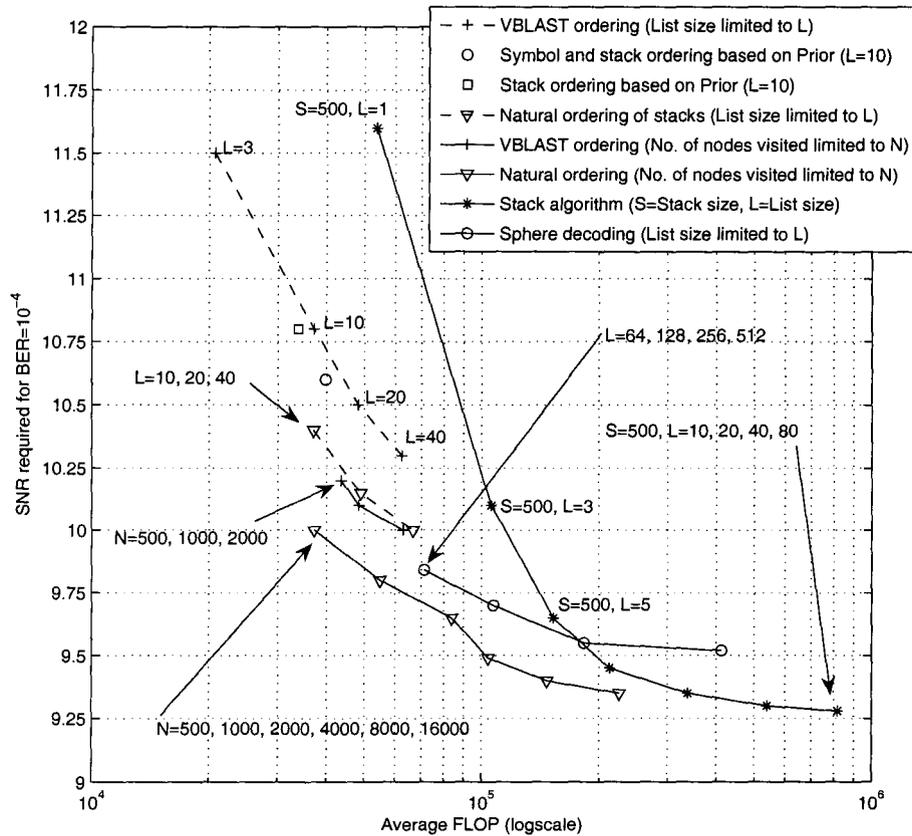


Figure 3.4: The trade-off between the SNR required for a BER of  $10^{-4}$  and the average FLOPs per channel use for different algorithms for a  $4 \times 4$  MIMO-BICM transmission scheme with 16-QAM symbols.

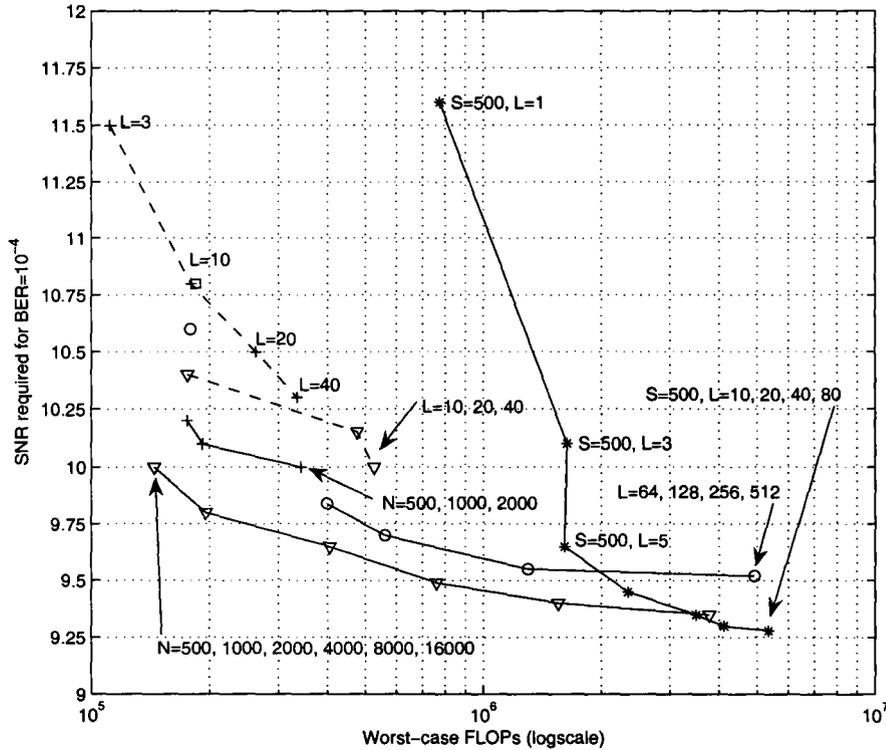


Figure 3.5: The trade-off between the SNR required for a BER of  $10^{-4}$  and the worst-case FLOPs per channel use for different algorithms for a  $4 \times 4$  MIMO-BICM transmission scheme with 16QAM symbols.

demodulation-decoding iterations, and complexity is measured by explicitly counting the number of floating point operations (FLOPs)<sup>10</sup> required to generate the (enriched) list  $\hat{\mathcal{L}}'$  and to compute the approximate LLRs. In Figs. 3.4 and 3.6 we measure the average number of FLOPs per channel use and in Figs. 3.5 and 3.7 we measure the peak FLOPs per channel use. In

<sup>10</sup>Although the demodulators compared here are all based on tree-search methods, there are significant differences in the computational effort required to process a given node, and counting the FLOPs enables us to take these differences into account. Counting the FLOPs also enables us to incorporate the computational cost of the QR decompositions that are performed, and the impact of the different list sizes on the cost of the list approximation of the LLRs.

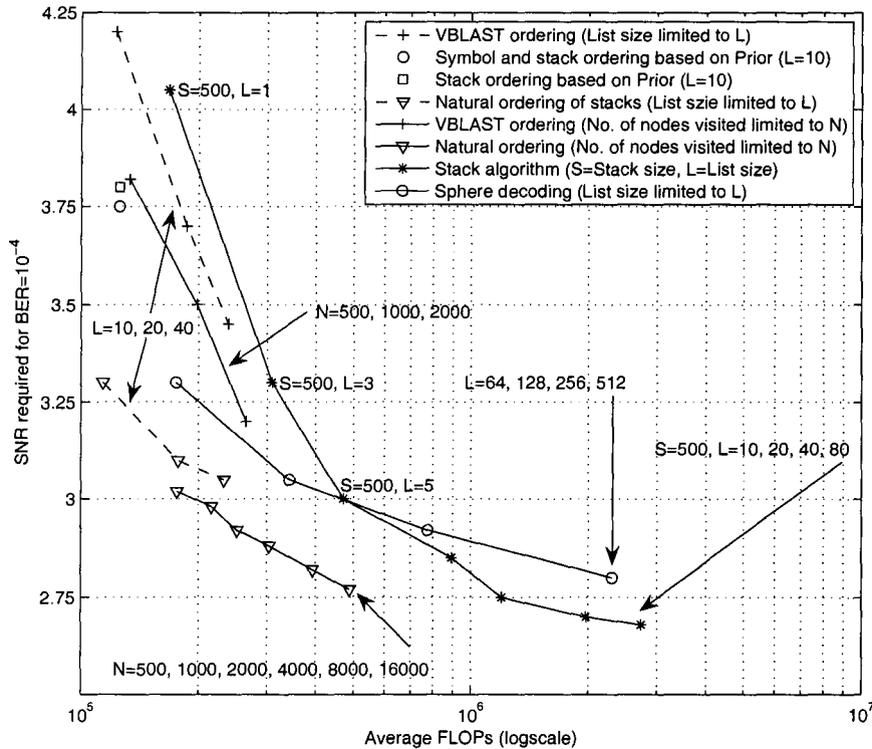


Figure 3.6: The trade-off between the SNR required for a BER of  $10^{-4}$  and the average FLOPs per channel use for different algorithms for a  $8 \times 8$  MIMO-BICM transmission scheme with QPSK symbols.

order to gauge the significance of the SNR gains in these figures, we used the method in [12] to compute the SNR threshold for the system under consideration. It is 6.9 dB for the  $4 \times 4$  system with 16-QAM symbols, and 1.6 dB for the  $8 \times 8$  system with QPSK symbols.

In Figs. 3.4–3.7 we consider the multistack algorithm with the four different symbol and re-start orderings mentioned in Section 3.3.1, and limits placed on either the list size (using  $L$ ), or on the number of nodes visited (using  $N$ ). We also consider performance-complexity trade-off of the stack

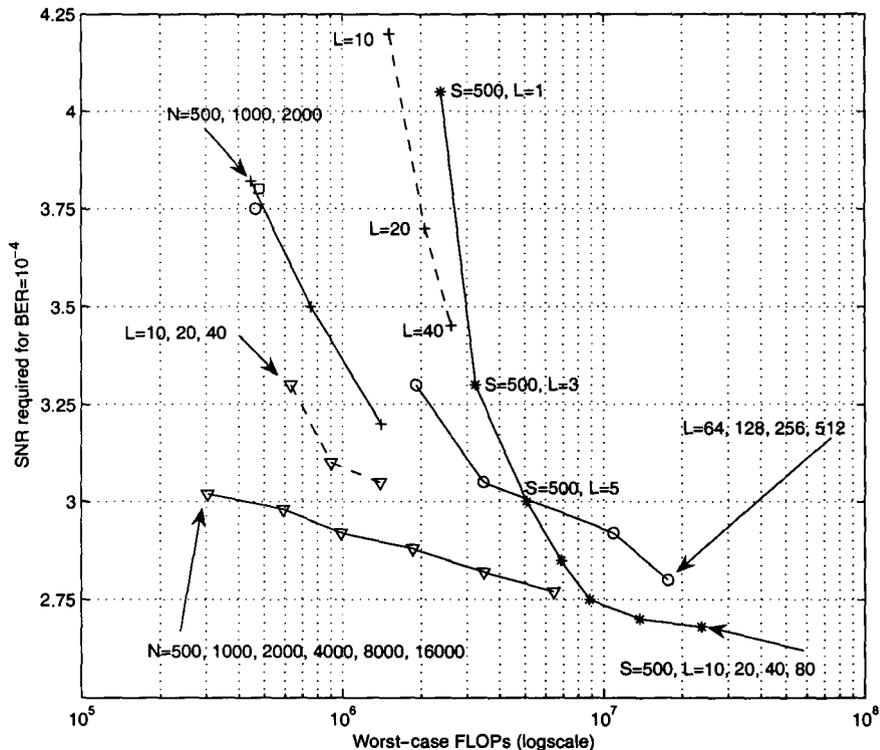


Figure 3.7: The trade-off between the SNR required for a BER of  $10^{-4}$  and the worst-case FLOPs per channel use for different algorithms for a  $8 \times 8$  MIMO-BICM transmission scheme with QPSK symbols.

algorithm and the sphere decoding algorithm. For the stack algorithm, we chose the LISS method in [18]. In that method, once an initial list of size  $L$  has been found, the LLRs are approximated using an augmented list that incorporates information from the incomplete paths of the tree. Following guidance from [18], in the first iteration we augmented the list using zero-forcing decision feedback detection of the incomplete paths, and in the subsequent iterations we used the *a priori* information. In our implementation of the LISS algorithm, we used a stack of size 500, different initial list sizes

$L$  (mentioned on the figures), and we augmented the list to a size of 100. For the sphere decoding algorithm, we used the method in [12], in which the list is generated only once. While that removes the computational load of list generation for the subsequent demodulation iterations, the list does not adapt to the updated *a priori* information from the decoder, and hence rather long lists are required for good performance. In order to obtain a target list size, the list sphere decoder in [12] employs a trial and error method to determine the appropriate search radius. However, we have excluded the FLOPs allocated to this task in the curves in Figs. 3.4, 3.5, 3.6 and 3.7.

Let us first make some observations regarding the performance of the different instances of the proposed algorithm. By comparing the dashed and solid curves with the same symbol, it is apparent that for a given computational cost, limiting the number of nodes visited provides better performance than limiting the size of the preliminary list, at least in the low computational cost region that we have examined. Also, by comparing the curves with different symbols, it is apparent that the natural re-start ordering provides better performance for a given computational cost. To provide insight into these comparisons, we selected four instances of the proposed algorithm that result in about the same average computational cost in Figs. 3.4 and 3.6; two that employ the natural re-start ordering, and two that employ the VBLAST ordering. For these demodulators, Tables 3.2 and 3.3 provide the average number of nodes visited, the average size of the preliminary list, and the average size of the enriched list, for the first, second and fourth demodulation-decoding iterations. For each ordering, the scheme with the limit on the number of nodes visited provides, on average, a larger enriched list in the first two iterations than the scheme with the bound on the size of the preliminary list, and it visits fewer nodes in generating these lists. The

Table 3.2: Average size of the preliminary list,  $\hat{\mathcal{L}}$  and the enriched list,  $\hat{\mathcal{L}}'$ , and the average number of nodes visited,  $N'$ , for the multistack algorithm with natural or VBLAST re-start orderings, and a limit  $L$  on the preliminary list size, or a limit  $N$  on the number of nodes visited. The scenario considered is that in Fig. 3.4.

System \ Iteration	1st			2nd			4th		
	$ \hat{\mathcal{L}} $	$ \hat{\mathcal{L}}' $	$N'$	$ \hat{\mathcal{L}} $	$ \hat{\mathcal{L}}' $	$N'$	$ \hat{\mathcal{L}} $	$ \hat{\mathcal{L}}' $	$N'$
Natural, $N = 500$	27.2	264	498	18.1	187	461	1.6	23	60
Natural, $L = 20$	14.3	191	987	11.6	152	705	2.4	34	122
VBLAST, $N = 1000$	38.3	237	959	25	191	792	1.5	22	58
VBLAST, $L = 20$	13.8	180	1132	12.2	158	836	4.1	56	235

Table 3.3: Average size of the preliminary list,  $\hat{\mathcal{L}}$  and the enriched list,  $\hat{\mathcal{L}}'$ , and the average number of nodes visited,  $N'$ , for the multistack algorithm with natural or VBLAST re-start orderings, and a limit  $L$  on the preliminary list size, or a limit  $N$  on the number of nodes visited. The scenario considered is that in Fig. 3.6.

System \ Iteration	1st			2nd			4th		
	$ \hat{\mathcal{L}} $	$ \hat{\mathcal{L}}' $	$N'$	$ \hat{\mathcal{L}} $	$ \hat{\mathcal{L}}' $	$N'$	$ \hat{\mathcal{L}} $	$ \hat{\mathcal{L}}' $	$N'$
Natural, $N = 500$	12.2	124	497	12.1	121	452	3.5	52	220
Natural, $L = 10$	5.6	80	886	5.1	75	571	7.5	78	310
VBLAST, $N = 500$	10.3	100	491	6.1	71	460	2.9	43	192
VBLAST, $L = 10$	5.2	74	844	4.8	69	560	5	60	254

richness of these lists enables the demodulator to leverage the power of the outer decoder more effectively. As shown in Tables 3.2 and 3.3, this results in a significant reduction in the number of nodes visited in the fourth iteration (and hence a reduction in the computational cost of that iteration) especially in the  $8 \times 8$  MIMO QPSK example, and, as shown in Figs. 3.4 and 3.6, it also results in a reduction of the SNR required to achieve the desired target error rate.

The trade-off curves in Figs. 3.4–3.7 for the stack (LISS) algorithm in [18] demonstrate the improved performance that can be obtained by searching for the best leaf nodes in sequence. However, these figures also demonstrate the significant computational cost of that search. In particular, by employing the proposed multistack algorithm with the natural restart ordering and a bound on the number of nodes visited, we obtain a performance-complexity trade-off that dominates that of the stack algorithm in the low-complexity region. Figs. 3.4–3.7 also show that the proposed method offers similar performance to the list sphere decoder [12] at a significantly lower computational cost.

The computational advantage of using the multistack algorithm over sphere decoding and stack algorithms tend to be greater when measuring the peak FLOPs required per channel use as shown in Figs. 3.5 and 3.7. This is particularly important in an implementational point of view. Since, the computational resources must accommodate the maximum FLOPs per channel use, algorithms with lower values for the maximum possible FLOPs are more favorable to be implemented. An alternative way in which the advantage of using the multistack algorithm can be visualized is by comparing its empirical computational cost distribution with those of the sphere decoding algorithm and the stack algorithm. For this purpose we picked three points in Figs. 3.4 and 3.6 at which each algorithm requires about the same SNR to reach a BER of  $10^{-4}$ . Figs. 3.8 and 3.9 show the corresponding computational cost distribution for the  $4 \times 4$  MIMO 16-QAM and the  $8 \times 8$  MIMO QPSK scenarios, respectively, where the stack algorithm has a stack size of 500 and a preliminary list size of 5 which is augmented to a list of size 100, and the sphere decoder has a list size of 128. We have compared these schemes to a multistack algorithm with natural stack ordering and a limit on the number

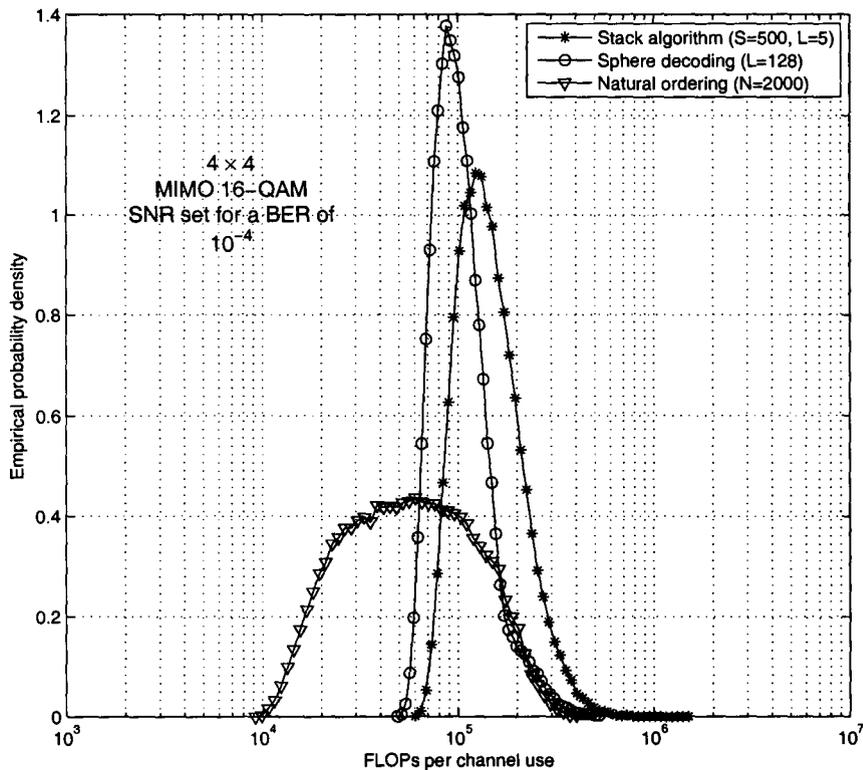


Figure 3.8: Empirical probability density of FLOPS per channel use for different demodulation schemes for  $4 \times 4$  MIMO 16-QAM transmission.

of nodes visited of 2000 in Fig 3.8 and 500 in Fig 3.9. Although all of these schemes have similar performance, the tail of the empirical computational cost distribution of the stack algorithm and the sphere decoding algorithm are extended more than that of the multistack algorithm in the high complexity region.

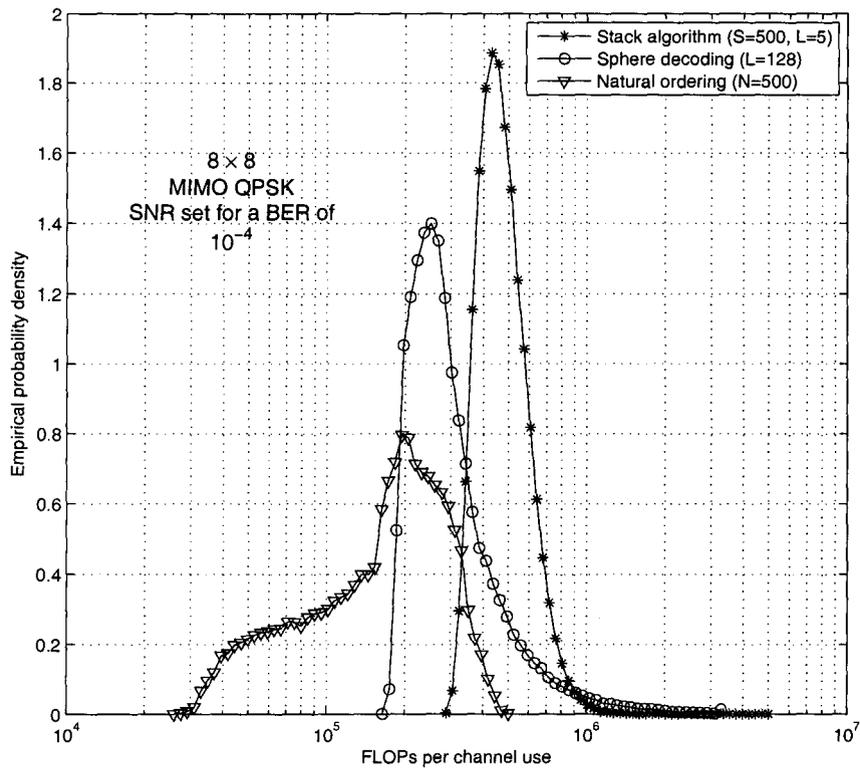


Figure 3.9: Empirical probability density of FLOPs per channel use for different demodulation schemes for  $8 \times 8$  MIMO QPSK transmission.

## 3.6 Conclusion

In this chapter we have proposed a tree-search algorithm for list-based MIMO soft demodulation that is based on the ‘best-first’ search principle used in the stack algorithm, but the search is performed on multiple stacks instead of a single (global) stack. In the proposed algorithm, the global stack is partitioned into a stack for each level of the tree, and the algorithm proceeds sequentially by performing one best-first search step in each of these stacks in the natural ordering of the tree. By assigning appropriate priorities to the level at which this best-first search per level processing re-starts once a leaf node has been obtained, we showed that the proposed approach can achieve a performance-complexity trade-off that dominates those of the stack (LISS) algorithm in [18] and the list sphere decoder [12] in the low-complexity region.

## Chapter 4

# Efficient Soft Demodulation of MIMO QPSK via Semidefinite Relaxation

'Well, if I called the wrong number, why did you  
answer the phone?'

---

JAMES THURBER

**I**N THIS chapter two efficient list-based 'soft' 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 of the proposed demodulators are also developed, and these implementations reduce the memory requirements of the demodulators. Analysis and simulation results show that the proposed demodulators offer an attractive trade-off between performance and computational cost. In particular, in the scenarios that we consider, the second of the proposed demodulators provides error rates that are lower than those of the minimum mean square error soft interference canceler (MMSE-SIC) and close to those of the list sphere decoder, and does so at a significantly lower computational cost.

## 4.1 Introduction

As was mentioned in the previous chapters, 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 [6]. 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., [12]; see also Fig. 4.1. As in the previous chapter, we will focus on narrowband multiple antenna systems, and when configured for such systems, the MIMO-BICM

transmitter encodes the information bits using an outer binary code, interleaves the encoder output and then maps blocks of the interleaved codeword to points on a (space-time) matrix constellation for transmission over the channel. At the receiver, the demodulator and outer decoder iteratively exchange the extrinsic components of their estimates of the posterior log likelihood-ratio (LLR) of each encoded bit, until a hard decision is taken. Although this IDD scheme 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., [12, 14, 15, 17, 18, 20, 85–87].

As described in Section 2.5.3, one approach to lower-complexity soft demodulation is to apply the so-called “max-log” approximation [51], under which the LLR of each bit is approximated by the difference between the optimal values of a pair of “hard” demodulation problems; e.g., [14, 15, 17]. However, each of these hard demodulation problems is also hard in the NP sense. Tree search methods (e.g., [21]), such as sphere decoding (e.g., [90, 91]), can be used to find optimal solutions to these problems (e.g., [14, 15]), but both the average and worst-case computational costs remain exponential in the problem size [22], and the “tail” of the distribution of the computational cost can be quite significant at low SNRs or for large problem sizes; e.g., [112]. As an alternative, semidefinite relaxation methods (e.g., [110, 111]) can be used to efficiently generate approximate solutions to the hard demodulation problems [17], and these methods have the advantage that the growth of the computational cost is bounded by a low-order polynomial in the problem size. However, 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.

As in Section 2.5.4, another approach to approximate 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 LLRs; e.g., [12, 18, 85–87]. The LLRs can then be approximated by marginalizing over the list. Most of the existing techniques are based on the use of tree search algorithms to identify members of the list (e.g., [12, 18, 85–87]), and hence can be rather computationally expensive, especially at SNRs close to the (ergodic capacity) threshold for the chosen rate. In some list demodulation schemes (e.g., [12]) the list for each channel use is generated once, in the first demodulation-decoding iteration, and is stored for use in the subsequent iterations. This may require substantial memory resources. In other schemes (e.g., [18, 85–87]), the list for each channel use is regenerated in each demodulation-decoding iteration, but the memory requirements can still be significant.

In this chapter we develop a semidefinite relaxation (SDR) approach to list-based soft demodulation, and propose two new demodulators, both of which regenerate the list in each demodulation-decoding iteration. The first demodulator applies the semidefinite relaxation technique once per demodulation-decoding iteration, and generates list members via the randomization procedure that is inherent in SDR techniques [110, 111]. 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. Furthermore, we develop a list-free implementation

of both demodulators that reduces the memory resources required for implementation. Our simulation results will show that the resulting computational and memory efficiencies are obtained without incurring a significant degradation in performance.

This chapter is organized as follows. In Section 4.2 we provide an overview of the MIMO-BICM-IDD system, and in Section 4.3 we review the SDR approach to hard demodulation of MIMO QPSK. Although the material in those sections does reiterate some of the concepts introduced in Chapter 2, this material is included here to provide a timely review of the context in which the proposed demodulators will be developed. In Sections 4.4 and 4.5 we develop the proposed demodulators, which we will call the List-SDR and Single-SDR demodulators, respectively. In Section 4.6 we describe the list-free implementation of the proposed methods, and in Section 4.7 the computational cost of implementing the 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 4.8.

## 4.2 System model and iterative receiver

We consider a narrowband multiple antenna system with  $N_t$  transmit antennas and  $N_r$  receive antennas. If we let  $\mathbf{s}_n$  denote the signal vector transmitted at the  $n$ th channel use, the corresponding received signal vector can be written as

$$\mathbf{y}_n = \mathbf{H}_n \mathbf{s}_n + \mathbf{v}_n, \quad (4.1)$$

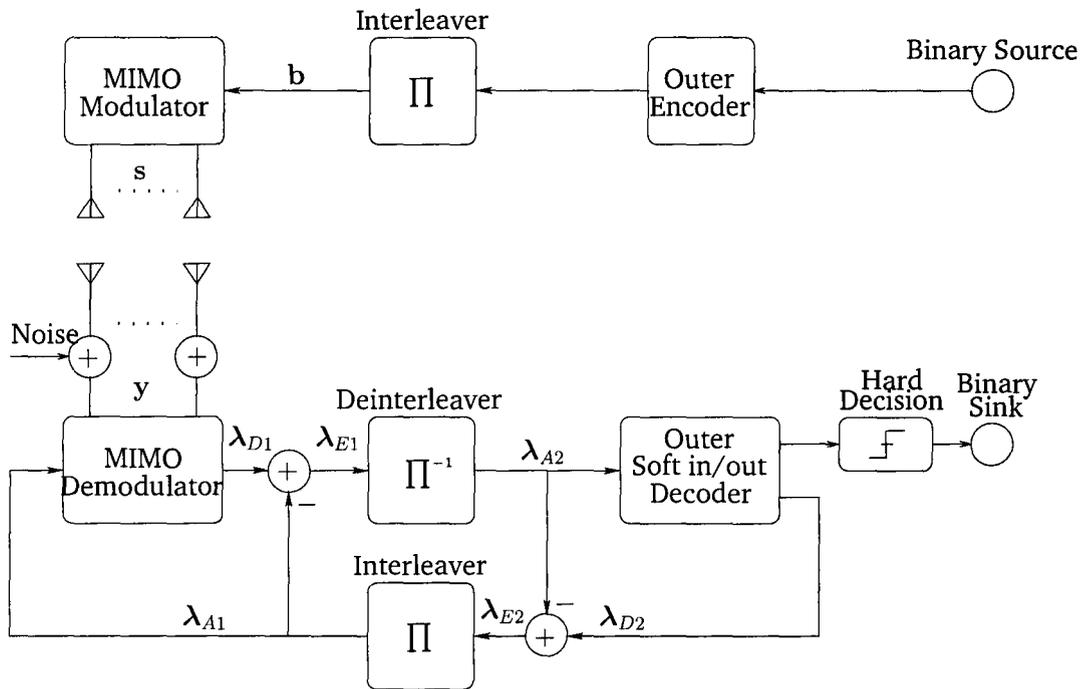


Figure 4.1: MIMO BICM-IDD transceiver.

where  $\mathbf{H}_n$  is the  $N_r \times N_t$  matrix of channel coefficients and 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  $\sigma^2$  per real dimension. We will consider a MIMO-BICM-IDD transceiver for this system, e.g., [12]; see Fig. 4.1. For simplicity, we will focus on V-BLAST transmission [73], but by using the equivalent channel concept in [67], the proposed demodulators extend directly to signaling schemes based on general linear dispersion codes. We will let  $\mathbf{b}_n$  denote the sub-block of the interleaved outer codeword that is to be transmitted in the  $n$ th channel use, and we will let  $\mathcal{M}(\mathbf{b})$  denote the mapping used by the MIMO modulator; i.e.,  $\mathbf{s}_n = \mathcal{M}(\mathbf{b}_n)$ . We will consider systems in which this mapping is to QPSK symbols, and we will let  $N$  denote the number of channel uses required to transmit one codeword of the outer

code.

At the receiver, the soft MIMO demodulator and the soft-input soft-output outer decoder iteratively exchange (extrinsic) information regarding the bit probabilities, using the log likelihood-ratio (LLR) format; see Fig. 4.1. This iterative process is based on the turbo principle [13], and since the channel in (4.1) is memoryless, the role of the soft demodulator is to compute (or approximate) the posterior LLR of each element of the block of the interleaved outer codeword that is transmitted in a given channel use, under the assumption that these blocks are independent from each other. That is, for the  $i$ th element of the bit-vector transmitted in the  $n$ th channel use,  $b_{n,i}$ , the soft demodulator computes (e.g., [12])

$$\lambda_{D1,n,i} = \log \frac{P\{b_{n,i} = +1 | \mathbf{y}_n\}}{P\{b_{n,i} = -1 | \mathbf{y}_n\}}, \quad (4.2)$$

where, for notational simplicity, we have not explicitly stated the conditioning on  $\mathbf{H}_n$  that is inherent in coherent demodulation. Once these LLRs have been computed for each of the  $N$  channel uses required to transmit a codeword of the outer code, they are passed through the de-interleaver to the outer decoder.

Since the focus of this chapter is on the soft demodulator, we will drop the subscript  $n$  in (4.1) and (4.2) and will consider a generic channel use. Applying Bayes' Rule enables us to rewrite (4.2) as

$$\lambda_{D1,i} = \log \frac{\sum_{\mathcal{L}_{i,+1}} p(\mathbf{y}|\mathbf{b})p(\mathbf{b})}{\sum_{\mathcal{L}_{i,-1}} p(\mathbf{y}|\mathbf{b})p(\mathbf{b})}, \quad (4.3)$$

where  $\mathcal{L} = \{\mathbf{b} \in \{-1, +1\}^{2N_i}\}$  denotes the (complete) list of possible transmitted bit-vectors and  $\mathcal{L}_{i,\pm 1} = \{\mathbf{b} \in \mathcal{L} | b_i = \pm 1\}$ . Given the model in (4.1)

and the fact that we are employing coherent demodulation, the likelihood function  $p(\mathbf{y}|\mathbf{b})$  in (4.3) satisfies

$$p(\mathbf{y}|\mathbf{b}) \propto \exp(-\|\mathbf{y} - \mathbf{H}\mathcal{M}(\mathbf{b})\|^2/(2\sigma^2)).$$

Furthermore, an estimate of the prior probability  $p(\mathbf{b})$  can be obtained from the output of the previous iteration of the decoder, by assuming that the elements of  $\mathbf{b}$  are independent; i.e.,  $p(\mathbf{b}) \approx \prod_i p(b_i)$ . If we define  $\lambda_{A1,i} = \log\left(\frac{p(b_i=+1)}{p(b_i=-1)}\right)$ , so that  $\boldsymbol{\lambda}_{A1}$  denotes the vector that represents these probabilities in LLR form, then  $\prod_i p(b_i) \propto \exp(\boldsymbol{\lambda}_{A1}^T \mathbf{b}/2)$ . Using these expressions, the LLR in (4.3) can be written as (e.g., [12])

$$\lambda_{D1,i} \simeq \log \frac{\sum_{\mathcal{L}_{i,+1}} \exp(-D(\mathbf{b})/(2\sigma^2))}{\sum_{\mathcal{L}_{i,-1}} \exp(-D(\mathbf{b})/(2\sigma^2))}, \quad (4.4)$$

where

$$D(\mathbf{b}) \triangleq \|\mathbf{y} - \mathbf{H}\mathcal{M}(\mathbf{b})\|_2^2 - \sigma^2 \boldsymbol{\lambda}_{A1}^T \mathbf{b}. \quad (4.5)$$

The number of elements in the lists  $\mathcal{L}_{i,\pm 1}$  in (4.4) grows exponentially as the number of (encoded) bits transmitted per channel use increases, and hence so does the computational cost of the demodulator. Many of the existing approaches to reducing this computational cost employ one of the following approximations:

$$\lambda_{D1,i} \approx \log \frac{\sum_{\hat{\mathcal{L}}_{i,+1}} \exp(-D(\mathbf{b})/(2\sigma^2))}{\sum_{\hat{\mathcal{L}}_{i,-1}} \exp(-D(\mathbf{b})/(2\sigma^2))} \quad (4.6)$$

$$\approx \frac{1}{2\sigma^2} \left( \min_{\mathbf{b} \in \hat{\mathcal{L}}_{i,-1}} D(\mathbf{b}) - \min_{\mathbf{b} \in \hat{\mathcal{L}}_{i,+1}} D(\mathbf{b}) \right), \quad (4.7)$$

where  $\hat{\mathcal{L}} \subseteq \mathcal{L}$ . Each of these equations reveals a class of approximate soft

MIMO demodulators.

The first class is based on selecting  $\hat{\mathcal{L}} = \mathcal{L}$  and solving the two binary quadratic optimization problems in (4.7) for each encoded bit. Solutions to these “hard” demodulation problems can be obtained using tree-search algorithms, as they are in [14, 15], but, as mentioned in the Introduction, these algorithms can be rather computationally expensive. Alternatively, approximate solutions can be found in polynomial time using the semidefinite relaxation technique [17].

The 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  $\hat{\mathcal{L}}_{i,\pm 1}$  in (4.6), e.g., [85], or by performing an exhaustive search over  $\hat{\mathcal{L}}_{i,\pm 1}$  to solve the minimization problems in (4.7); e.g., [12]. The key challenge in this class of methods is the efficient selection of the members of  $\hat{\mathcal{L}}$ . Most approaches are based on tree-search ideas, and hence are potentially rather computationally expensive; e.g., [12, 18, 85–87]. In addition, as we will explain in Section 4.6, these methods may require significant memory resources in order to store the generated list members.

One important approximate soft demodulator that does not fall into one of these two classes is that based on the minimum mean square error soft interference canceler (MMSE-SIC) in [24]; see also [46, 120, 121]. For each transmitted symbol, this demodulator first forms the unbiased conditional MMSE estimate of the symbol, where the conditioning is on the probabilities of the other symbols transmitted in the channel use of interest. The demodulator then approximates the residual interference by a Gaussian random variable, and computes the LLR of each bit in the symbol as if the channel was a scalar additive white Gaussian noise channel. The conditional

MMSE estimate has a canonical decomposition into two steps: subtraction of the mean of the interfering symbols, followed by (unbiased) linear MMSE estimation of the relevant symbol, and hence the moniker MMSE-SIC. The reader may refer to Appendix A for a more detailed overview of this approximate soft demodulation scheme.

The two demodulators proposed in this chapter fall into the second class of approximate soft demodulators, but they are based on semidefinite relaxation rather than a tree search, and hence their computational cost is bounded by a low-order polynomial of the problem size. Furthermore, the second of these demodulators requires the solution of only one semidefinite program per channel use, and both demodulators can be implemented without explicitly storing the list of candidate bit-vectors. Both the proposed demodulators exploit the properties of the randomization step that is inherent in the approximation of the solution of a binary quadratic problem by semidefinite relaxation, and before we introduce those demodulators we will provide a brief overview of the application of the semidefinite relaxation technique [110, 111] to hard demodulation [96, 122].

### 4.3 Hard demodulation using SDR

Consider the real-valued equivalent representation for (4.1) with QPSK signaling,

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\mathbf{b} + \tilde{\mathbf{v}}, \quad (4.8)$$

where  $\tilde{\mathbf{y}}$  and  $\tilde{\mathbf{v}}$  are the concatenations of the real and imaginary parts of  $\mathbf{y}$  and  $\mathbf{v}$ , respectively, and we have considered an arbitrary channel use. Given prior information on the bit probabilities in the form of  $\lambda_{A1}$  in (4.5), the

bit-vector  $\mathbf{b}$  that maximizes the *a posteriori* probability is the solution to the following binary optimization problem:

$$\min_{\mathbf{b} \in \{+1, -1\}^{2N_t}} D(\mathbf{b}) = \min_{\mathbf{b} \in \{+1, -1\}^{2N_t}} \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\mathbf{b}\|_2^2 - \sigma^2 \boldsymbol{\lambda}_{A1}^T \mathbf{b}. \quad (4.9)$$

Using the definitions [17, 96]

$$\tilde{\mathbf{b}} \triangleq \begin{bmatrix} \check{\mathbf{b}} \\ c \end{bmatrix}, \quad \check{\mathbf{b}} \triangleq c\mathbf{b}, \quad \mathbf{Q} \triangleq \begin{bmatrix} \tilde{\mathbf{H}}^T \tilde{\mathbf{H}} & \mathbf{a} \\ \mathbf{a}^T & 0 \end{bmatrix}, \quad \mathbf{a} \triangleq -\tilde{\mathbf{H}}^T \tilde{\mathbf{y}} - 0.5\sigma^2 \boldsymbol{\lambda}_{A1}, \quad (4.10)$$

in which  $c \in \{+1, -1\}$ , the problem in (4.9) can be stated as the following (NP-hard) binary quadratic programming (BQP) problem:

$$\min_{\tilde{\mathbf{b}} \in \{+1, -1\}^{2N_t+1}} \tilde{\mathbf{b}}^T \mathbf{Q} \tilde{\mathbf{b}}. \quad (4.11)$$

Using the substitution  $\mathbf{X} = \tilde{\mathbf{b}}\tilde{\mathbf{b}}^T$ , the problem in (4.11) can be reformulated as

$$\min_{\mathbf{X}} \text{Trace}(\mathbf{X}\mathbf{Q}) \quad (4.12a)$$

$$\text{s.t. } \mathbf{X} \succeq \mathbf{0}, \quad \text{rank}(\mathbf{X}) = 1, \quad (4.12b)$$

$$[\mathbf{X}]_{ii} = 1, \quad i = 1, \dots, 2N_t + 1, \quad (4.12c)$$

in which the computational difficulties manifest themselves in the rank-1 constraint. The semidefinite relaxation approach to approximating the solution to (4.11) is to relax the rank-1 constraint and solve the following

semidefinite program (SDP):

$$\min_{\mathbf{X}} \text{Trace}(\mathbf{X}\mathbf{Q}) \quad (4.13a)$$

$$\text{s.t. } \mathbf{X} \succeq 0, \quad (4.13b)$$

$$[\mathbf{X}]_{ii} = 1, \quad i = 1, \dots, 2N_t + 1. \quad (4.13c)$$

This problem is convex and can be efficiently solved using the interior point method in [23]; see Section 4.7. (See also [123] for some recent developments.) When  $\mathbf{X}_{\text{opt}}$ , the optimal solution to (4.13), is rank 1, its factorization generates an optimum solution to (4.11). In the more common event that the solution to (4.13) is not rank 1, a randomization procedure [110, 111] can be used to extract an approximation of the solution to (4.11) from  $\mathbf{X}_{\text{opt}}$ . 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. For each vector  $\mathbf{u}$  we compute  $\tilde{\mathbf{x}} = \text{sign}(\mathbf{V}^T \mathbf{u})$ , construct the vector  $\mathbf{x} = \tilde{x}_{2N_t+1} \times [\tilde{x}_1, \dots, \tilde{x}_{2N_t}]^T$ , and compute  $D(\mathbf{x})$  using (4.5). 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 solution to (4.11). A key feature of the SDR approach is that even for the worst-case channel, the expected value of  $D(\mathbf{x})$  over the randomizations is guaranteed to be within a (reasonably small) constant factor of the optimal value of (4.11), independent of the number of bits to be detected [111]. Furthermore, since each choice of the random vector  $\mathbf{u}$  is made independently, the probability that  $D(\mathbf{b}_{\text{sdr}})$  is higher than the expected value of  $D(\mathbf{x})$  decreases exponentially with the number of randomization iterations [111].

In [17], Steingrimsson *et al* developed a soft MIMO demodulator from the first class in Section 4.2 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 list-based soft demodulation scheme that requires the solution to just one SDP in each demodulation-decoding iteration for each channel use, and in Section 4.5 we will propose a scheme that requires the solution of only one SDP per channel use.

## 4.4 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 (4.9). 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 essence of the proposed approach. We will construct a preliminary list  $\hat{\mathcal{L}}'$  by simply storing each (unique) bit-vector generated by the randomization procedure. Since it is possible that there may be bit positions for which  $\hat{\mathcal{L}}'_{i,+1}$  or  $\hat{\mathcal{L}}'_{i,-1}$  is empty, once the randomizations have been completed we will construct an enriched list  $\hat{\mathcal{L}}$  consisting of  $\hat{\mathcal{L}}'$  plus all those bit-vectors with Hamming distance of 1 of the bit-vectors in  $\hat{\mathcal{L}}'$ . (This enrichment is based on ideas in [20] and can be implemented by “flipping” individual bits of each element of  $\hat{\mathcal{L}}'$ .) Once this enriched list has been constructed, we adopt the standard list-based approach to approximate the soft information using (4.6) or (4.7) over the constructed list  $\hat{\mathcal{L}}$ . Since the computational cost of this enumeration grows

Table 4.1: List generation component of the List-SDR algorithm.

- 
- Data:  $\mathbf{X}_{\text{opt}}$ , the solution to (4.13), or an approximation thereof.
  - Parameters:  $M$ , the number of randomization iterations;  $K$ , the maximum size of the preliminary list.
  - 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. Choose a random vector  $\mathbf{u}$  from the uniform distribution on the unit sphere.
  4. Compute  $\tilde{\mathbf{x}} = \text{sign}(\mathbf{V}^T \mathbf{u})$  and increment  $m$ .
  5. Construct  $\mathbf{x} = \tilde{x}_{2N_t+1} \times [\tilde{x}_1, \dots, \tilde{x}_{2N_t}]^T$ . If  $\mathbf{x}$  is not in  $\hat{\mathcal{L}}'$ , add it to  $\hat{\mathcal{L}}'$  and increment  $k$ .
  6. If  $k < K$  and  $m < M$ , return to 3.
  7. Construct  $\hat{\mathcal{L}}$  as the union of  $\hat{\mathcal{L}}'$  and all the single bit-flippings of the bit-vectors in  $\hat{\mathcal{L}}'$ .
- 

linearly with the cardinality of  $\hat{\mathcal{L}}$ , one may wish to bound the cardinality of  $\hat{\mathcal{L}}'$ , and to use this bound to enable early termination of the randomization procedure should  $\hat{\mathcal{L}}'$  be sufficiently rich. The resulting list generation algorithm is presented in Table 4.1. Since the *a priori* information  $\lambda_{A1}$  is updated in each demodulation-decoding iteration the cost function  $D(\mathbf{b})$  in (4.5) changes, and hence so does  $\mathbf{X}_{\text{opt}}$ . Therefore, for each iteration we regenerate the list using the procedure in Table 4.1.

## 4.5 Soft demodulation using single SDR

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 4.3. The mean value of the  $i$ th 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. [110,111]. 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

$$\mu_i = \frac{\pi - 2\theta_{i,2N_t+1}}{\pi}. \quad (4.14)$$

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 that the constraint  $[\mathbf{X}]_{ii} = 1$  in (4.13) ensures that all  $\|\mathbf{v}_i\| = 1$ , the mean value can be expressed directly in terms of the columns of  $\mathbf{V}$ ,

$$\mu_i = \frac{2}{\pi} \arcsin(\mathbf{v}_i^T \mathbf{v}_{2N_t+1}). \quad (4.15)$$

The first observation in the development of the proposed demodulator is that the expression in (4.15) 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  $\lambda_{A1}$ ). The expression for  $D(\mathbf{b})$  in (4.5) suggests that the demodulation procedure needs to be repeated at each iteration (as it is in [18, 85, 86] and in the List-SDR algorithm proposed in Section 4.4). 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. 4.2. 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: (i) the vector  $\boldsymbol{\lambda} = [\lambda_1, \dots, \lambda_{2N_t}]^T$  containing the mean values in (4.15) in LLR form, i.e.,

$$\lambda_i = \log\left(\frac{1 + \mu_i}{1 - \mu_i}\right); \quad (4.16)$$

and (ii) the vector  $\lambda_{A1}$  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 (4.18) below), 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 [13]. 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

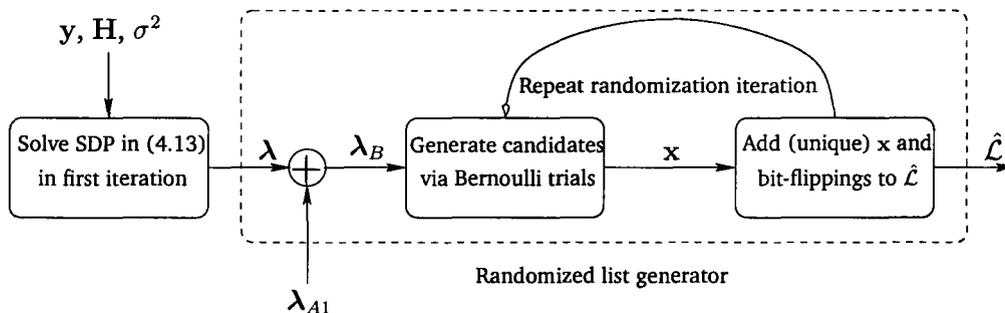


Figure 4.2: List generation scheme using the Single-SDR algorithm.

Bernoulli distribution should be

$$\lambda_B = \lambda + \lambda_{A1}. \quad (4.17)$$

The  $i$ -th entry of the corresponding mean vector  $\mu_B$  is

$$\mu_{B,i} = 1 - 2/(1 + \exp(\lambda_{B,i})). \quad (4.18)$$

Having computed  $\mu_B$ , the demodulator randomly generates the bit-vectors that will form the preliminary list,  $\hat{\mathcal{L}}'$ . The  $i$ th 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  $\hat{\mathcal{L}}'$  all the single bit-flippings of the bit vectors in  $\hat{\mathcal{L}}'$ . A formal statement of list generation using this algorithm is presented in Table 4.2. After construction of the list  $\hat{\mathcal{L}}$  the soft information from demodulator can be approximated using (4.6) or (4.7).

Table 4.2: List generation component of the Single-SDR algorithm

- 
- Data:  $\lambda$  in (4.16);  $\lambda_{A1}$ , 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.
  - Output:  $\hat{\mathcal{L}}$ , the enriched list.
1. Initialize  $\hat{\mathcal{L}}'$  and  $\hat{\mathcal{L}}$  empty,  $m = 0$ , and  $k = 0$ .
  2. Compute  $\lambda_B$  in (4.17) and subsequently  $\mu_B$ .
  3. Generate each element of  $\mathbf{x}$ ,  $x_i$ , independently according to the (antipodal) Bernoulli distribution with mean  $\mu_{B,i}$  and increment  $m$ .
  4. If  $\mathbf{x}$  is not in  $\hat{\mathcal{L}}'$ , add it to  $\hat{\mathcal{L}}'$  and increment  $k$ .
  5. If  $k < K$  and  $m < M$ , return to 3.
  6. Construct  $\hat{\mathcal{L}}$  as the union of  $\hat{\mathcal{L}}'$  and all the single bit-flippings of the bit-vectors in  $\hat{\mathcal{L}}'$ .
- 

## 4.6 List-free implementation

One of the bottlenecks in the implementation of list-based soft demodulators is the requirement of a significant amount of memory. This is an especially important issue in list demodulation schemes in which the list is generated in the first demodulation-decoding iteration and stored for use in the subsequent iterations. In those schemes (e.g., [12]), the system must provide enough memory to save the list associated with each of the  $N$  channel uses required to send a complete codeword. (Since these lists are based on information from the channel only, they need to be quite long.) In schemes that regenerate the list at each iteration, the receiver has only one list to store, but the resulting memory requirement can still be quite significant.

Another issue that arises in list-based demodulation schemes in which

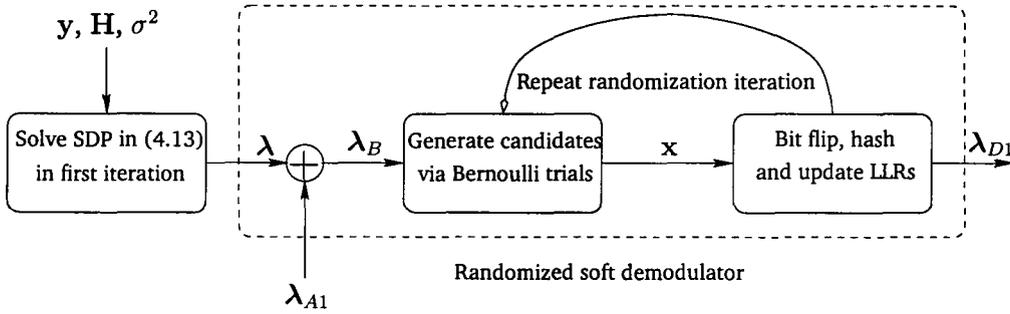


Figure 4.3: The proposed list-free soft demodulator.

the list members are generated via a randomization procedure, such as those proposed in Sections 4.4 and 4.5, is that candidate list members may be duplicated. In order to avoid redundant computation of the metric  $D(\cdot)$ , and in order to avoid storage of repeated list members, this duplication ought to be avoided; see Step 5 in Table 4.1 and Step 4 in Table 4.2. However, it is important that the computational cost of any scheme that is used to avoid this duplication is small.

In this section we will show that for list demodulators that regenerate the list at each iteration and use the max-log approximation on the list to approximate the LLR (cf. (4.7)), only two real vectors of size  $N_t$  need to be stored. We will also show that repeated generation of candidate list members can be efficiently detected via a hashing strategy. A block diagram of the proposed list-free implementation is provided in Fig. 4.3, and a formal statement of the algorithm is provided in Table 4.3. We point out that unlike the algorithms in the previous sections in which the output was the enriched list  $\hat{\mathcal{L}}$ , from which the soft information can be extracted using (4.6) or (4.7), in the list-free implementation the soft information is generated directly.

One of the features of many list demodulators, including the List-SDR and Single-SDR demodulators presented in the previous sections, is that the

candidate list members are generated one at a time. In such schemes we can update the optimal values of the  $2N_t$  optimization problems in (4.7) as each list member is generated, rather than waiting for the whole list to be constructed. We can then discard that list member and simply store the  $2N_t$  real values that are the current optimal values for the problems in (4.7). Doing so leads to a list-free implementation. In our demodulators, the list  $\hat{\mathcal{L}}$  is generated in two phases, the randomization phase and the enrichment phase (which is based on bit-flipping). The enrichment phase can be incorporated into the list-free implementation by simply evaluating the neighboring vectors at Hamming distance 1 of each randomized candidate just after it is generated.

This list-free implementation highlights the potential for redundant computation of  $D(\cdot)$ . One efficient way in which that can be avoided is via a hashing scheme [124]. The hashing scheme computes an integer ‘signature’ for each generated bit vector via a deterministic injective function, and then stores this signature in an ordered array. Before processing the next candidate bit-vector,  $\mathbf{x}$ , the demodulator computes its signature and checks if that value is in the signature array. The candidate is discarded if its signature is present, and if that value is not found, its signature is added to the array, and the candidate bit-vector is processed. In the simulations in Section 4.8 we chose the signature to be the decimal equivalent of the binary number  $(\mathbf{x}^T + \mathbf{1})/2$ , where  $\mathbf{1}$  is a conformally-sized vector of ones. As we will quantify in the next two sections, the computational costs of computing the signatures and searching the (ordered) signature array are small with respect to the overall computational cost of the demodulator. Implementing the hashing scheme does require additional memory. Although the maximum required memory,  $\min\{M, K\}$ , is, quite naturally, the same as that

Table 4.3: List-free implementation of Single-SDR algorithm for randomized soft demodulator

- 
- Data:  $\lambda$  in (4.16);  $\lambda_{A1}$ , 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 signature list.
  - Output:  $\lambda_{D1}$ , the vector of log likelihood-ratios
1. Compute  $\lambda_B$  in (4.17) and subsequently  $\mu_B$ .
  2. Initialize  $\mathbf{f}_{+1} = \{+\infty\}^{2N_t}$ ,  $\mathbf{f}_{-1} = \{+\infty\}^{2N_t}$ ,  $m = 0$ ,  $k = 0$
  3. Generate each  $x_i$  independently according to the (antipodal) Bernoulli distribution with mean  $\mu_{B,i}$ .
  4. Compute the signature of  $\mathbf{x}$ . If that value is not in the signature array, insert the value into the array and increment  $k$ , compute  $D(\mathbf{x})$ , and for each  $i = 1, 2, \dots, 2N_t$ , if  $x_i = +1$  then set  $[\mathbf{f}_{+1}]_i = \min\{[\mathbf{f}_{+1}]_i, D(\mathbf{x})\}$ , else set  $[\mathbf{f}_{-1}]_i = \min\{[\mathbf{f}_{-1}]_i, D(\mathbf{x})\}$ .
  5. For each  $i = 1, 2, \dots, 2N_t$ , set  $\check{\mathbf{x}}^{(i)} = \mathbf{x}$  and then  $\check{x}_i^{(i)} = -x_i$ . Repeat Step 4 for  $\check{\mathbf{x}}^{(i)}$ .
  6. Increment  $m$ . If  $m < M$  and  $k < K$  return to 3. Otherwise, return  $\lambda_{D1} = (\mathbf{f}_{+1} - \mathbf{f}_{-1}) / (2\sigma^2)$ .
- 

required to store the list, in practice the average length of the signature array will be significantly shorter, and the memory requirement will often be outweighed by the computational cost reduction that is obtained by (efficiently) avoiding redundant computation.

An additional advantage of the list-free implementation described above is that the parameters of the algorithm can be adapted dynamically. This offers the potential for the demodulator to dynamically adjust its operating point on its performance-complexity trade-off in response to changes in the characteristics of the channel or in the requirements of the application.

## 4.7 Computational cost

As mentioned in the Introduction, an advantage of SDR methods over tree search methods, such as sphere decoding, is their polynomial (worst-case) computational cost. An advantage of the particular (list-free) SDR-based soft demodulators that we have proposed herein is that their computational costs are lower than those of some existing soft demodulators. In this section we quantify that claim by evaluating the computational costs of these SDR-based algorithms. For convenience, we have summarized the outcomes of this analysis in Table 4.4. We will begin our analysis by stating the computational cost of each of the components of the algorithms, in terms of floating point operations.

If we let  $\epsilon$  denote the accuracy to which the SDP is solved, the worst-case computational cost of solving the SDP in (4.13) using the interior point method in [23] is  $O((2N_t + 1)^{3.5} \log \epsilon^{-1})$ . The computational cost of generating each bit-vector in the conventional randomization procedure used in the Multi-SDR [17] and List-SDR (cf. Section 4.4) methods is  $O((2N_t + 1)^2)$ . The computational cost of the simplified randomization step in the Single-SDR method (cf. Section 4.5) is  $O(2N_t)$ . Finally, since  $\mathbf{b}$  is binary, computing the metric  $D(\mathbf{b})$  requires  $2N_t(2N_t + 1)$  (signed) real additions.

The Multi-SDR method of [17] solves one SDP of size  $2N_t + 1$  and  $2N_t$  SDPs of size  $2N_t$  per demodulation-decoding iteration, and after solving each SDP it performs  $M$  randomization iterations and computes  $D(\mathbf{b})$  for all the generated bit-vectors. If we perform  $T$  demodulation-decoding iterations,

then the computational cost per channel use of the multi-SDR scheme is:

$$\begin{aligned}
& T \times [O((2N_t + 1)^{3.5} \log \epsilon^{-1}) + M \times O((2N_t + 1)^2) + M \times O((2N_t)^2)] \\
& + 2N_t T \times [O((2N_t)^{3.5} \log \epsilon^{-1}) + M \times O((2N_t)^2) + M \times O((2N_t)^2)] \\
& \sim O(TN_t^{4.5} \log \epsilon^{-1}) + O(TMN_t^3) + O(TMN_t^3), \quad (4.19)
\end{aligned}$$

where the first term corresponds to the cost of solving the SDPs, the second term corresponds to the cost of the randomization steps that follow each of these SDPs, and the last term denotes the cost of the metric computations.

The List-SDR approach proposed in Section 4.4 requires the solution of only one SDP of size  $2N_t + 1$  per demodulation-decoding iteration. If we perform  $M$  randomization steps (i.e., if  $K = M$  in Table 4.1), then the cardinality of the enriched list  $\hat{\mathcal{L}}$  is at most  $(2N_t + 1)M$ , and hence the enumeration approach to optimizing the terms on the right hand side of (4.7) requires at most  $(2N_t + 1)M$  evaluations of  $D(\mathbf{b})$  in each demodulation-decoding iteration. Therefore, the worst-case complexity of the proposed approach is

$$\begin{aligned}
& T \times [O((2N_t + 1)^{3.5} \log \epsilon^{-1}) + M \times O((2N_t + 1)^2) + M(2N_t + 1) \times O((2N_t)^2)] \\
& \sim O(TN_t^{3.5} \log \epsilon^{-1}) + O(TMN_t^2) + O(TMN_t^3), \quad (4.20)
\end{aligned}$$

where these terms correspond to the cost of solving the SDP, the cost of constructing the list  $\hat{\mathcal{L}}$  (via randomization), and the cost of the enumerations in (4.7) over  $\hat{\mathcal{L}}_{i,\pm 1}$ , respectively. This expression shows that the computational cost of the SDP component of the proposed list-based SDR scheme is one order lower than that of the multi-SDR scheme.

The computational cost per channel use of the Single-SDR approach

Table 4.4: Dominant computational cost per channel use of various MIMO soft demodulators for a system with  $N_t$  transmit antennas and  $T$  demodulation-decoding iterations.

Demodulator	Dominant Computational Cost
Multi-SDR	$O(TN_t^{4.5})$
List-SDR	$O(TN_T^{3.5})$
Single-SDR	$O(N_t^{3.5})$
MMSE-SIC	$O(TN_t^4)$

can be obtained in a similar way to those above, and is

$$O(N_t^{3.5} \log \epsilon^{-1}) + O(TMN_t) + O(TMN_t^3), \quad (4.21)$$

where the first term represents the complexity of solving the SDP, the second term represent the cost of the Bernoulli-based randomizations, and the third term represents the cost of computing  $D(\mathbf{b})$  for each bit-vector generated in the randomization step. (As in (4.20), we have assumed that  $K = M$  in Tables 4.2 and 4.3.) These expressions reveal the computational advantage of only having to solve one SDP per channel use, and the advantage of the Bernoulli-based randomizations. The list-free implementation of this algorithm requires a small amount of additional computation; at most  $O(TMN_t)$  integer additions per channel use to compute the signatures, and at most  $O(M \log M)$  operations to search for existing signatures in the (sorted) signature array.

As we will illustrate in Section 4.8, in most practical implementations, the cost of solving the SDPs will be the dominant component of the computational cost of the SDR-based demodulators. We have summarized those costs in Table 4.4, and to help place those costs in context, we have included the computational cost per channel use of the MMSE-SIC demodulator in [24],

which is  $TO((2N_t)^4) \sim O(TN_t^4)$ .

## 4.8 Simulations

In this section we will compare the performance and computational cost of the proposed demodulators with those of the Multi-SDR demodulator [17], the list sphere decoder in [12], and the MMSE-SIC demodulator in [24]. We consider MIMO BICM systems that employ V-BLAST transmission of QPSK symbols over an i.i.d. Rayleigh fading channel. The transceiver parameters, including those of the outer codes and the iterative demodulation and decoding algorithm, are chosen from those used in [12, 24]. For the proposed demodulators we will evaluate the performance of the list-free implementation described in Section 4.6, using the hashing strategy to avoid redundant computation. For all demodulation schemes based on list-decoding ideas, an insufficiently rich list can result in under or over estimation of the soft information. In our simulations of the proposed demodulators and the list sphere decoder in [12], we will take a common approach to mitigating this effect [87], and will clip the estimated log likelihood-ratios to the interval  $[-5, +5]$ .

We will consider two types of outer code: a turbo code with a reasonably long block length, and a convolutional code with a rather short block length. For the (scalar) binary-input additive white Gaussian noise channel, the turbo code provides good performance at SNRs close to the capacity threshold for the given rate. The convolutional code is a weaker code, but it can be implemented with significantly lower latency. Following [12], the turbo code was chosen to be 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. The (different) interleavers in the turbo code and in the BICM transmitter were selected from randomly generated candidates in each Monte-Carlo iteration. We used the conventional BCJR algorithm [50] to decode the constituent convolutional codes of the turbo code, and 8 turbo decoding iterations were performed before we passed extrinsic information back to the demodulator. Following [24], the convolutional code was chosen to be the rate  $1/2$  (23, 35) recursive systematic convolutional code with block length 256, and BCJR decoding was used. We have performed simulation experiments for  $8 \times 8$  and  $4 \times 4$  MIMO systems, and we will report some results from those experiments in the following sections, respectively.

#### 4.8.1 $8 \times 8$ system

In our first set of simulation experiments, we will consider a MIMO system with  $N_t = 8$  antennas at the transmitter and  $N_r = 8$  antennas at the receiver. In Figs. 4.4 and 4.5 we consider systems with the turbo outer code and the convolutional outer code, respectively, and we compare the BER performance of i) the Single-SDR demodulator, ii) the List-SDR demodulator of Section 4.4, iii) the Multi-SDR demodulator in [17], iv) the list sphere decoder in [12] (with a list size of  $L = 512$ ), and v) the MMSE-SIC demodulator in [24]. In this comparison we specify the Single-SDR demodulator with the SDP accuracy of  $\epsilon = 10^{-2}$  and  $M = 50$  randomizations. Later in this section, we will examine the impact of these parameters on the complexity and performance of the Single-SDR demodulator and will justify this choice. For fair comparison, the value of  $\epsilon$  in the List-SDR and Multi-SDR demodulators was also chosen to be  $10^{-2}$ . For reference, we have indicated the SNR at

which the mutual information for QPSK signals is equal to 8 bits per channel use (the data rate of the chosen scheme). That SNR is about 1.6 dB. From Fig. 4.4 it is apparent that the BER of the Single-SDR demodulator is better than that of the MMSE-SIC demodulator and is close to that of the other demodulators. The performance advantage of the Single-SDR demodulator over the MMSE-SIC demodulator is somewhat larger in the early iterations in the case of the convolutional outer code (see Fig. 4.5), but the relative weakness of this short code means that after four demodulation-decoding iterations the performance of all the considered demodulators is quite similar.

In order to show that the Single-SDR demodulator achieves the performance described above at low computational cost, we explicitly counted the number of floating point operations (FLOPs) 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 include the FLOPs required to solve the SDPs, to perform the randomization steps, and to compute the metrics and the hash functions. For the list sphere decoder we have included the FLOPs required to construct the list (which is only performed once per channel use), and those required to compute the metrics in each demodulation iteration. (For the list sphere decoder, we have excluded the computational cost of the trial and error scheme used to compute an appropriate radius for the sphere (cf. [12]), and hence the depicted results may be a little optimistic.) For the MMSE-SIC demodulator, for each bit 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. Our explicit counting of the number of required operations revealed that, as stated in

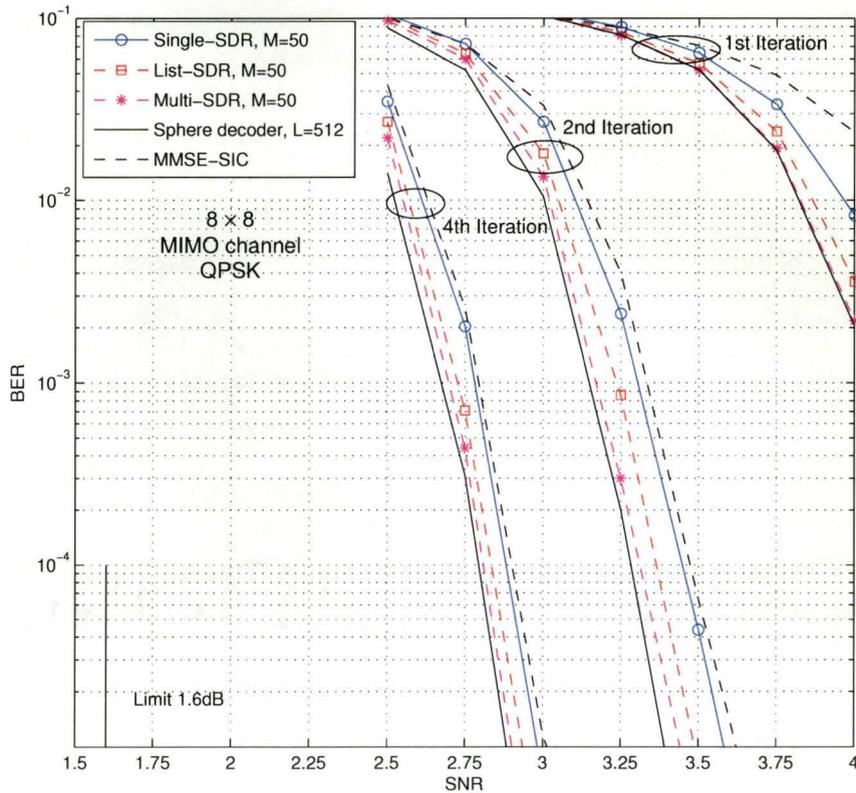


Figure 4.4: Comparison of the BER performance of various demodulators for the  $8 \times 8$  MIMO system with the turbo outer code.

Section 4.7, the dominant component of the computational cost of the SDR-based demodulators is the cost of solving the SDP or SDPs. This observation is quantified in Fig. 4.6, where we have plotted the average computational cost per channel use of each algorithm against the SNR, for the case of the turbo outer code, and we have included results for several values of  $M$ , the number of randomizations.

Fig. 4.6 also quantifies the computational advantages of the Single-SDR demodulator over the List-SDR and Multi-SDR demodulators, and its

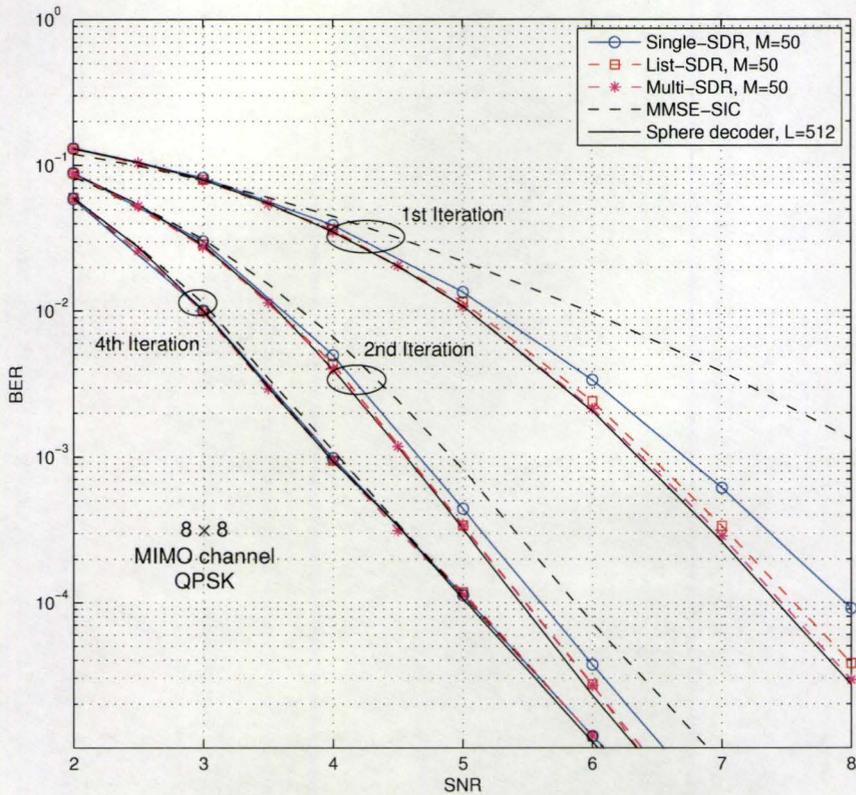


Figure 4.5: Comparison of the BER performance of various demodulators for the  $8 \times 8$  MIMO system with the convolutional outer code.

computational advantages over the MMSE-SIC demodulator and the list sphere decoder. In particular, in the ‘waterfall’ region of the BER curves in Fig. 4.4, the average computational cost per channel use of the Single-SDR demodulator is about half that of the MMSE-SIC demodulator and about a third of that of the list sphere decoder. 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. 4.7 the empirical probability density of the computational

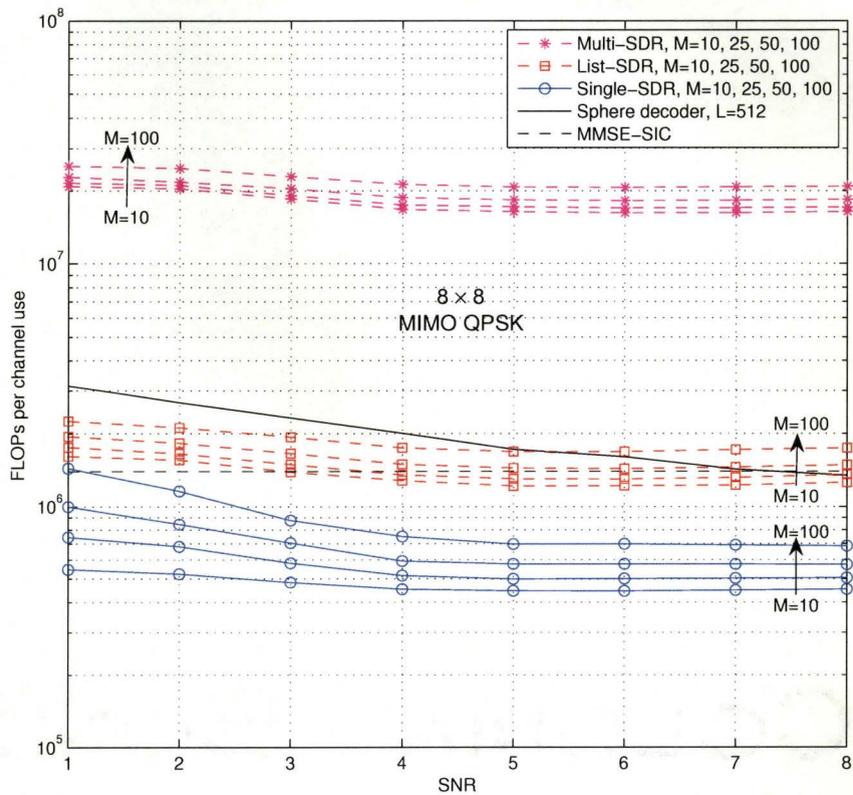


Figure 4.6: Comparison of the average computational cost per channel use of the proposed demodulators and that of the Multi-SDR, list sphere decoding, and MMSE-SIC demodulators for the  $8 \times 8$  system with the turbo outer code. For the SDR based methods results for several values for  $M$ , the number of randomizations, are provided.

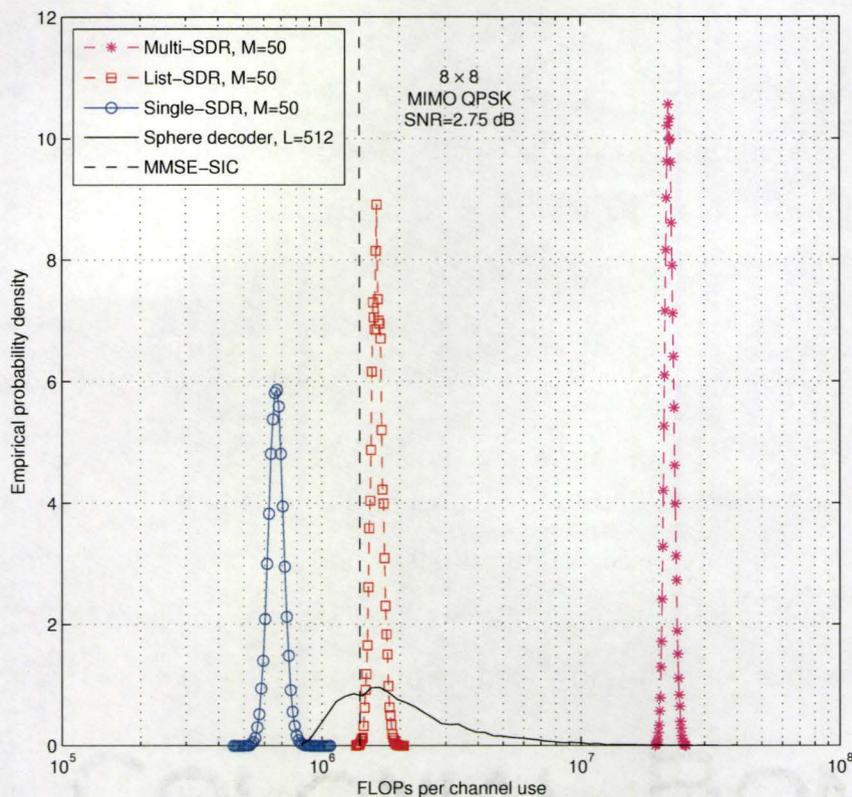


Figure 4.7: Empirical probability density of the number of FLOPs per channel use in the  $8 \times 8$  system with the turbo outer code at an SNR of 2.75 dB.

cost per channel use of several demodulators at an SNR of 2.75 dB. In this scenario, the whole empirical distribution of the computational cost of the Single-SDR demodulator lies below the computational cost of the MMSE-SIC demodulator. Fig. 4.7 also illustrates the rather ‘fat’ tail of the distribution of the computational cost of the list sphere decoder. This fat tail can make it rather awkward to provision an appropriate amount of computational resources for a list sphere decoder. The concentrated complexity distributions

of the SDR-based methods make that provisioning much more straightforward.

In order to justify the choices of an SDP accuracy of  $\epsilon = 10^{-2}$  and  $M = 50$  randomizations for the Single-SDR demodulator in Figs. 4.4 and 4.5, we now evaluate the impact of these parameters on the BER performance of the receiver. In Fig. 4.8 we plot the average BER of the Single-SDR demodulator with  $M = 50$  (and  $K = M$ ) and different values for  $\epsilon$ . Fig. 4.8 suggests that in order to extract the benefits of the Single-SDR demodulator, it is sufficient to solve the SDP to two digits of accuracy. This is an important observation because solving the SDP is the dominant computational task in the Single-SDR demodulator, and the cost of solving this SDR grows as  $O(N_t^{3.5} \log \epsilon^{-1})$ . Fig. 4.8 also suggests that solving the SDP more accurately does not necessarily lead to improved performance. This is a consequence of the fact that accurate list demodulation requires a rather rich set of list members. In the case that the SDP is solved accurately, the resulting randomization procedure may generate a rather narrowly-focused list, and hence the slightly degraded performance in the case of  $\epsilon = 10^{-4}$  in Fig. 4.8. In order to promote the generation of rich lists, and in order to reduce the computational cost, it is natural to consider demodulators in which the SDP is only coarsely solved.

In Fig. 4.9 we examine the BER performance of the Single-SDR demodulator for an SDP accuracy of  $\epsilon = 10^{-2}$  and various numbers of randomizations. Even though each randomization in the Single-SDR demodulator is rather cheap to implement (see Fig. 4.6), Fig. 4.9 suggests that the choice of  $M = 50$  is sufficient to demonstrate the potential performance of the Single-SDR demodulator.

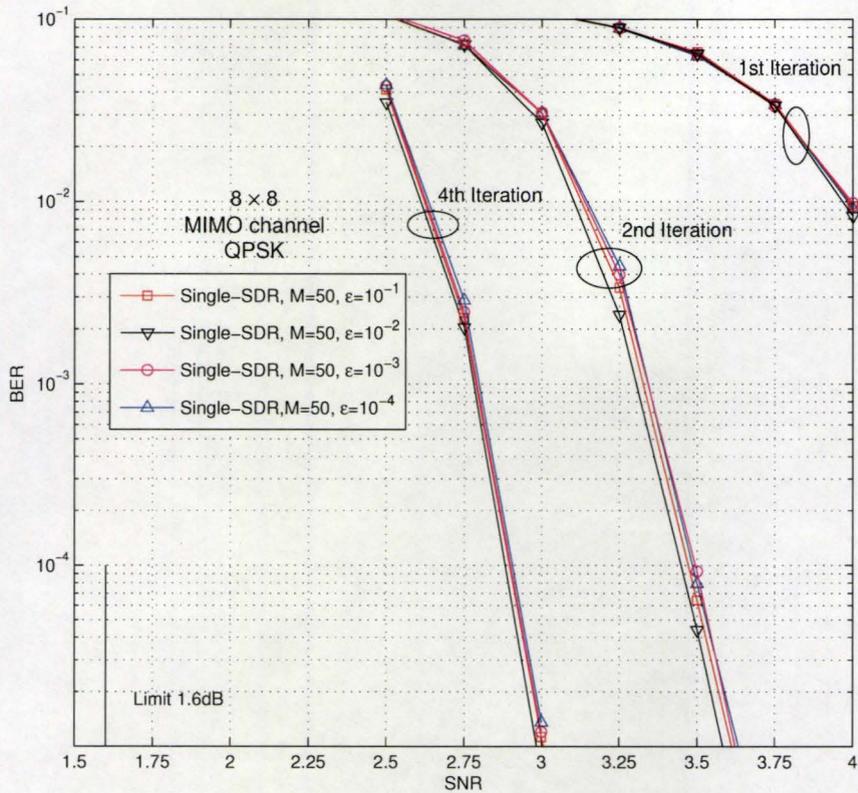


Figure 4.8: BER performance of the  $8 \times 8$  MIMO system with the turbo outer code and the Single-SDR demodulator with  $M = 50$  randomizations and different accuracies to which the SDP is solved.

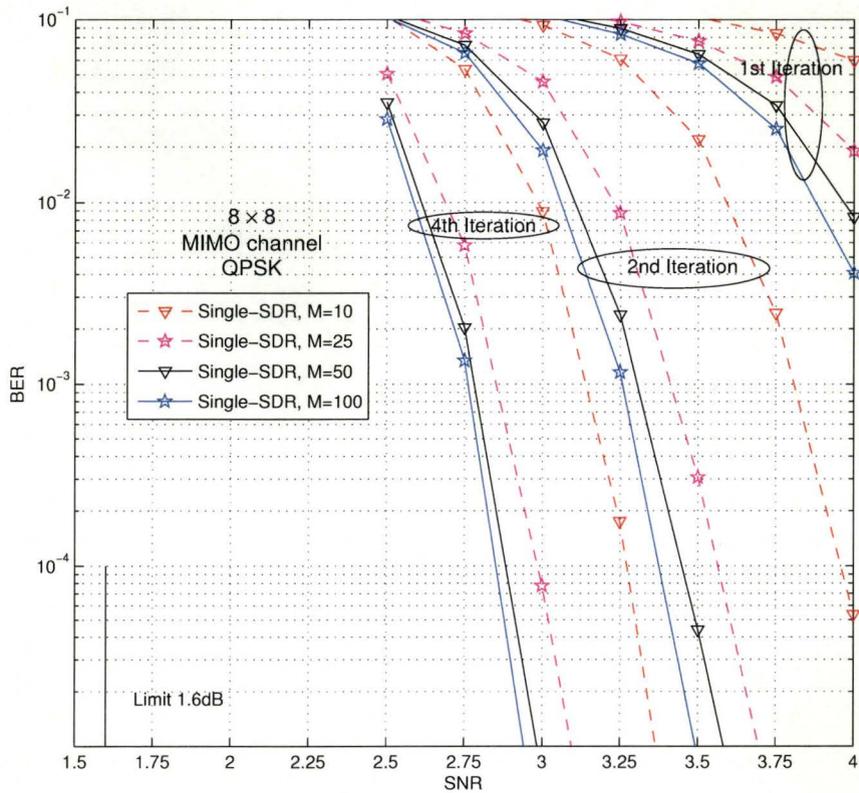


Figure 4.9: BER performance of the  $8 \times 8$  MIMO system with the turbo outer code and the Single-SDR demodulator with an SDP solution accuracy of  $\epsilon = 10^{-2}$ , and different numbers of randomizations,  $M$ .

## 4.8.2 $4 \times 4$ system

In the second set of simulation experiments, we repeated the above analysis for a MIMO system with  $N_t = N_r = 4$ . In this case, the full demodulation list has only 256 elements, and hence full-list demodulation is chosen as the performance benchmark. (A list sphere decoder with a list size of 128 is also considered.)

Following experiments analogous to those discussed in the previous section, we chose to solve the SDPs to an accuracy of  $\epsilon = 10^{-2}$  and to employ  $M = 25$  randomizations. The average BERs of the various demodulators in the system with the turbo outer code are plotted in Fig. 4.10. (The SNR threshold of this system is 1.6 dB.) As in the  $8 \times 8$  case, the performance of the proposed demodulators is close to that of the benchmark demodulator. However, in this  $4 \times 4$  case, the performance of the proposed demodulators is also substantially better than that of the MMSE-SIC demodulator. The relative degradation in the performance of the MMSE-SIC is due to the fact that there are fewer interfering symbols in the  $4 \times 4$  case, and hence the inherent approximation that the residual interference is Gaussian is less accurate in this case.

The average BERs of these demodulators in the case of the convolutional outer code are provided in Fig. 4.11. In the early iterations, the Single-SDR demodulator has a significant performance advantage over the MMSE-SIC demodulator, but as was the case for the  $8 \times 8$  system, after four iterations the performance of all demodulators is quite similar.

In Fig. 4.12 we plot the average computational cost per channel use of each demodulator in the system with the turbo outer code, and in Fig. 4.13 we plot the empirical density of the computational costs at an

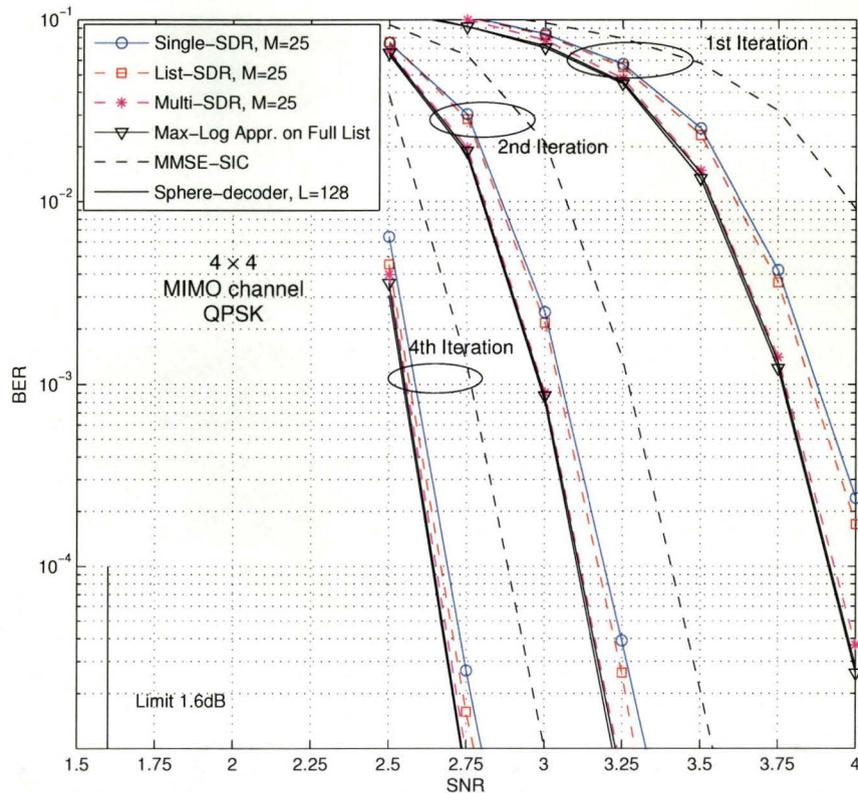


Figure 4.10: Comparison of the BER performance of various demodulators for the  $4 \times 4$  MIMO system with the turbo outer code.

SNR of 2.75 dB. These figures quantify the computational advantages of the Single-SDR demodulator. In particular, its average computational cost is just over two-thirds of that of the MMSE-SIC demodulator, and just over half of that of the full list demodulator. In this  $4 \times 4$  scenario, the computational advantages of the single-SDR demodulator over the MMSE-SIC demodulator are smaller than those in the  $8 \times 8$  case, but its performance advantage is significantly larger.

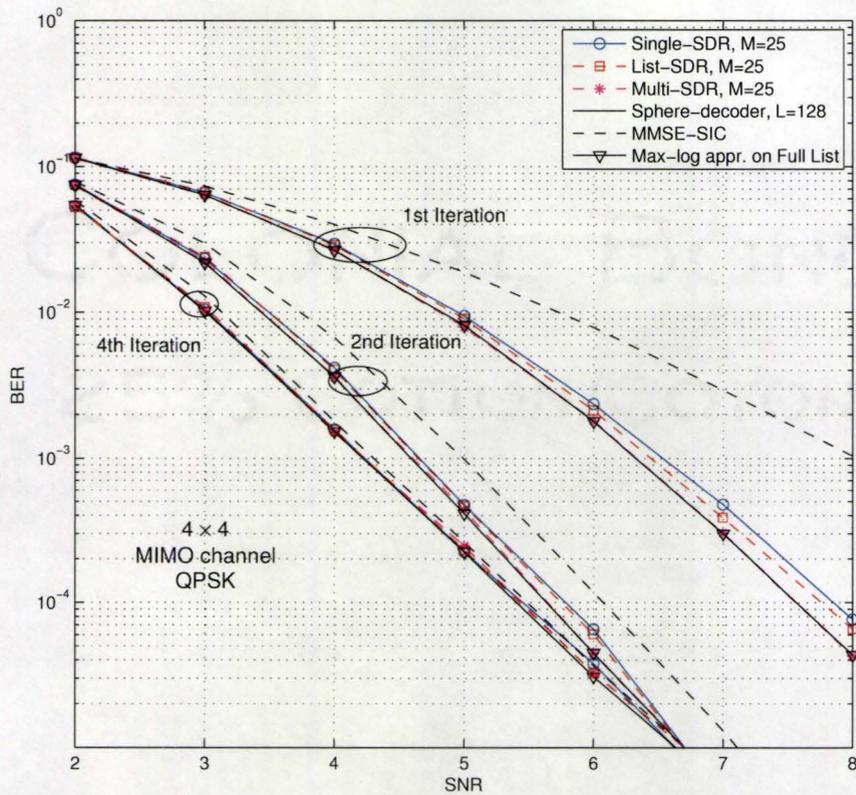


Figure 4.11: Comparison of the BER performance of various demodulators for the  $4 \times 4$  MIMO channel with the convolutional outer code.

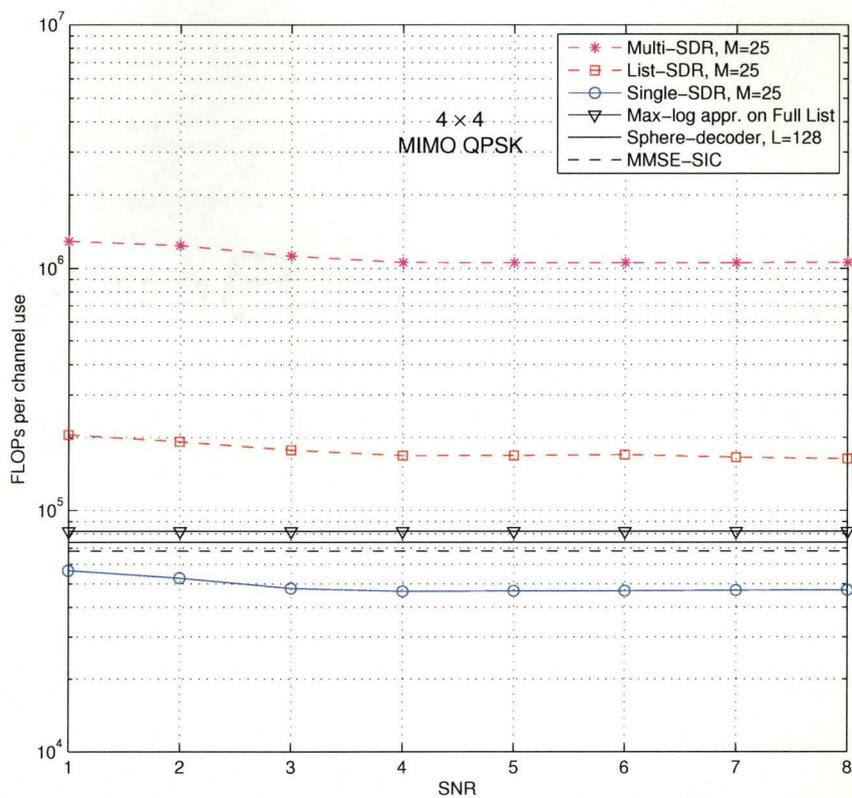


Figure 4.12: Comparison of average computational cost per channel use of the proposed demodulators and that of the Multi-SDR, full-list, sphere decoding and MMSE-SIC demodulators for the  $4 \times 4$  system with the turbo outer code.

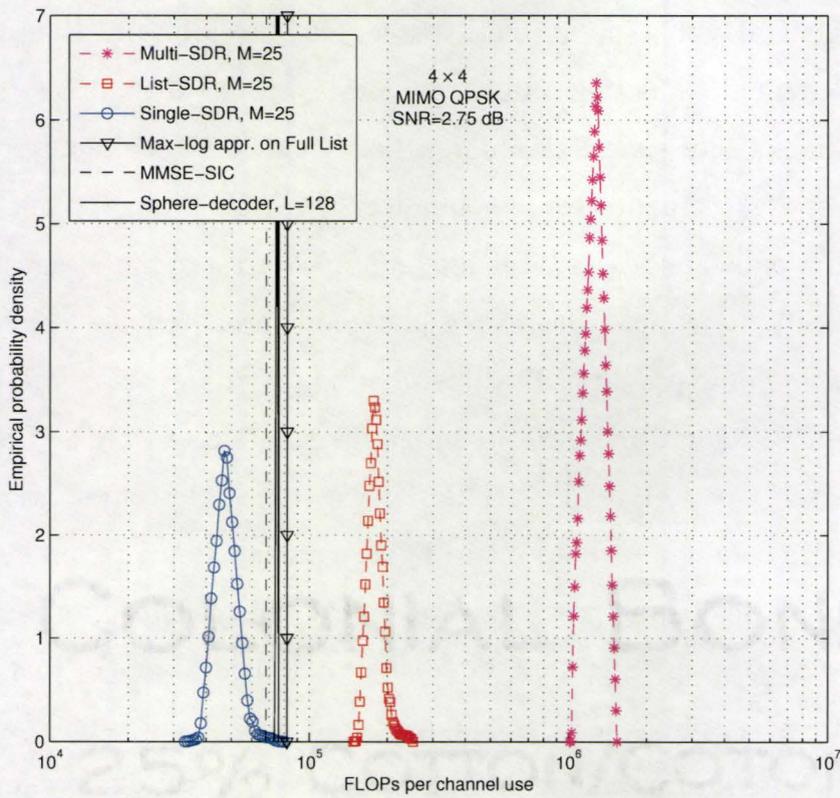


Figure 4.13: Empirical probability density of the number of FLOPs per channel use in the  $4 \times 4$  system with the turbo outer code at an SNR of 2.75 dB.

## 4.9 Conclusion

In this chapter, we have proposed two computationally-efficient soft MIMO demodulators based on an adaptation of the semidefinite relaxation (SDR) method for hard demodulation to list-based soft demodulation. We have also presented a list-free implementation of the proposed methods that can be implemented with a substantially smaller memory ‘footprint’ than conventional list demodulation algorithms. In contrast to list demodulators based on the principles of sphere decoding, 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 [17], 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. Our simulation results suggest that these computational advantages are obtained without incurring a significant degradation in performance. In particular, the proposed Single-SDR demodulator provides better performance than the MMSE-SIC demodulator and performance that is close to that of the list sphere decoder, and it does so at a substantially lower computational cost. In this chapter we have focussed on soft demodulators for MIMO-BICM systems that employ QPSK signalling. Extensions to systems that employ higher order QAM constellations are developed in Chapter 5, using the corresponding SDR approaches to hard demodulation in [25, 26].

# Chapter 5

## Efficient soft MIMO 16-QAM demodulation using SDR

'Transmission of documents via telephone wires is possible in principle, but the apparatus required is so expensive that it will never become a practical proposition.'

---

DENNIS GABOR, "INVENTING THE FUTURE", 1962

**I**N THE previous chapter some list-based soft demodulation schemes for MIMO QPSK demodulation were presented. Since the generalization of the methods presented in Chapter 4 to MIMO demodulation of higher-order QAM symbols is not straightforward, in this chapter we will present efficient soft demodulation schemes based on SDR for systems that transmit 16-QAM symbols. As in Chapter 4, we will first review some available schemes for hard demodulation of MIMO 16-QAM transmission. In particular we will review two existing SDR techniques for hard demodulation of 16-QAM. One of the SDR techniques requires the solution of an SDP that has a higher dimension than that for the QPSK case considered in Chapter 4, while the other

involves the solution of an SDP of the same size as that in the QPSK case. In the later technique, the SDP has a structure that resembles that for the QPSK case, and we will provide an efficient interior point algorithm for the solution of that SDP. We will then prove that these two existing SDR techniques for 16-QAM are equivalent, and hence the emphasis of this chapter will be on the technique for which we provided the efficient interior point algorithm.

The goal of this chapter is to apply the principles outlined in the previous chapter to develop soft MIMO demodulators for 16-QAM, based on the existing SDR-based hard demodulators. A feature of QPSK signals that enabled the development in the previous chapter is that the *a priori* information enters linearly into the metric  $D(\mathbf{b})$ . This is not the case for 16-QAM signaling, and one of the key steps in designing the proposed soft demodulators is the development of a representation of the *a priori* information, or an approximation thereof, that conforms to the SDR framework for 16-QAM. After providing such a representation and an approximation that offers lower computational cost, we then propose a List-SDR scheme for soft demodulation that generates list members via conventional randomization. Another key step in the developments of this chapter is the approximation of the conventional randomization step in the SDR framework for 16-QAM by independent scalar randomization. As in Chapter 4, this will enable us to develop a Single-SDR scheme that needs only to solve one SDP per channel use. As we will show in the simulations section, the proposed Single-SDR demodulator provides performance close to that of the list sphere decoding schemes and better performance than the MMSE-SIC scheme, and that it does so with significantly lower computational cost.<sup>1</sup>

---

<sup>1</sup>Since the computational cost of the Multi-SDR approach to soft demodulation is much

## 5.1 Maximum likelihood demodulation of MIMO 16-QAM using SDR

In this section we consider the maximum likelihood ('hard') demodulation problem for MIMO systems with 16-QAM signaling. As in Chapter 4, we consider a narrow-band MIMO system with  $N_t$  transmit antennas and  $N_r$  receive antennas, channel matrix  $\mathbf{H}_n$ , and V-BLAST transmission of 16-QAM symbols. (Again, the extension to more general space-time transmission schemes such as linear dispersion codes [67], is straightforward.) The received signal  $\mathbf{y}_n$  can be written as:

$$\mathbf{y}_n = \mathbf{H}_n \mathbf{s}_n + \mathbf{v}_n, \quad (5.1)$$

where  $\mathbf{v}_n$  is the vector of i.i.d. additive Gaussian noise samples with variance  $\sigma^2$  per real dimension. In this section we will consider the case of uncoded transmission, in which the symbol-vector  $\mathbf{s}_n$  is obtained by mapping a vector  $\mathbf{b}_n$  of uncoded (i.i.d. and equally-likely) bits to 16-QAM symbols using the (one-to-one) mapping  $\mathbf{s}_n = \mathcal{M}(\mathbf{b}_n)$ . We will consider the standard representation of 16-QAM symbols so that the real and imaginary parts of each element of  $\mathbf{s}_n$  can take on one of the values  $\{\pm 1, \pm 3\}$ . The real valued representation of (5.1) can be constructed by concatenating the real and imaginary parts of  $\mathbf{y}_n$ ,  $\mathbf{s}_n$  and  $\mathbf{v}_n$  in (5.1) as

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}} \tilde{\mathbf{s}} + \tilde{\mathbf{v}}, \quad (5.2)$$

---

higher than that of the List-SDR and Single-SDR approaches (cf. Chapter 4), in this chapter we will not provide performance nor complexity results for the Multi-SDR approach.

where, for simplicity, we have considered a general channel use and have dropped the subscript  $n$ . If we define

$$D(\tilde{\mathbf{s}}) = \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_2^2, \quad (5.3)$$

the maximum likelihood symbol-vector  $\tilde{\mathbf{s}}$  can be obtained by solving the following optimization problem

$$\min_{\tilde{\mathbf{s}} \in \{\pm 1, \pm 3\}^{2N_t}} D(\tilde{\mathbf{s}}). \quad (5.4)$$

Using the following definitions [96]

$$\mathbf{x} \triangleq \begin{bmatrix} \tilde{\mathbf{s}} \\ c \end{bmatrix}, \quad \check{\mathbf{s}} \triangleq c\tilde{\mathbf{s}}, \quad \tilde{\mathbf{Q}} \triangleq \begin{bmatrix} \tilde{\mathbf{H}}^T \tilde{\mathbf{H}} & -\tilde{\mathbf{H}}^T \tilde{\mathbf{y}} \\ -\tilde{\mathbf{y}}^T \tilde{\mathbf{H}} & 0 \end{bmatrix}, \quad (5.5)$$

in which  $c \in \{+1, -1\}$ , problem (5.4) can be stated as the following discrete optimization problem

$$\min_{\mathbf{x}} \mathbf{x}^T \tilde{\mathbf{Q}} \mathbf{x} \quad (5.6a)$$

$$\text{s.t. } \mathbf{x} = [x_1, \dots, x_{2N_t+1}]^T \quad (5.6b)$$

$$x_i \in \mathcal{A} = \{\pm 1, \pm 3\}, \quad i = 1, \dots, 2N_t, \quad (5.6c)$$

$$x_{2N_t+1} \in \{-1, +1\}, \quad (5.6d)$$

which is similar to the binary quadratic programming problem of (4.11) in Chapter 4, but the binary constraints for bits 1 to  $2N_t$  are replaced by the constraints in (5.6c). This problem is an NP-hard problem due to the constraints in (5.6c) and (5.6d). Denoting  $\mathbf{X} = \mathbf{x}\mathbf{x}^T$  and defining  $\mathcal{B} = \{L, U\}$

where  $L = +1, U = +9$ , we can rewrite (5.6) as

$$\min_{\mathbf{X}} \text{Trace}(\mathbf{X}\tilde{\mathbf{Q}}) \quad (5.7a)$$

$$\text{s.t. } \mathbf{X} \succeq 0, \quad \text{rank}(\mathbf{X}) = 1 \quad (5.7b)$$

$$[\mathbf{X}]_{ii} \in \mathcal{B}, \quad i = 1, \dots, 2N_t \quad (5.7c)$$

$$[\mathbf{X}]_{ii} = 1, \quad i = 2N_t + 1, \quad (5.7d)$$

which is still an NP-hard problem because of the rank-1 constraint and the constraints in (5.7c). There are several semidefinite relaxation techniques available in the literature (e.g., [25, 26, 109, 125–127]) to obtain approximate solutions to this ML detection problem by solving a semidefinite program. The approach used in most of these techniques is to formulate the problem in a higher dimension (e.g., [25, 109, 126, 127]) by adding slack variables and then relaxing the non-convex constraints to convert the problem to a semidefinite program. Interior point methods can be used to solve these SDPs but for most of them an interior point method with a computational cost analogous to that of the method of [23] that was used in Chapter 4 for MIMO QPSK demodulation does not exist. In addition, the increase in the dimensionality of these problems increases their computational complexity. In the technique proposed by Sidiropoulos and Luo [26], the non-convex constraints are relaxed directly, keeping the dimensionality of the problem unchanged. We will call this technique the “fixed dimension relaxation” technique, and in Section 5.1.3 we will provide an interior point method based on the method in [23] to solve the resulting SDP efficiently. Among other existing relaxation techniques, an interesting property of the method of Wiesel *et al.* [25] is that the resulting SDP is the Lagrangian bi-dual of

(5.6). Hence, the difference between the optimal value of this problem and the ML problem in (5.6) is exactly the duality gap. We will call this technique the “increased dimension relaxation” technique and we will review it in more detail in Section 5.1.1. In Section 5.1.4 we will prove the somewhat unexpected result that the optimal values of the fixed dimension [26] and the increased dimension [25] relaxation techniques are actually the same. We will also prove that the component of the solution to the increased dimension SDP that is used to generate the solution to (5.4) is equal to the solution of the SDP in the fixed dimension technique.

### 5.1.1 Increased dimension relaxation

In addition to relaxing the rank-1 constraint in both of the increased dimension and fixed dimension schemes, we also need to deal with the non-convex constraints in (5.7c). Wiesel *et al.* [25] proposed the increased dimension relaxation technique in which they used extra slack variables to increase the dimensionality of the problem in order to convert these non-convex constraints into equivalent linear equality constraints. This can be performed by defining a vector of slack variables  $\mathbf{t}^T \triangleq [t_1, \dots, t_{2N_t}]$  and observing that

$$\begin{aligned} \tilde{s}_i \in \mathcal{A} &\iff (\tilde{s}_i + 1)(\tilde{s}_i - 1)(\tilde{s}_i + 3)(\tilde{s}_i - 3) = 0, \quad i = 1, \dots, 2N_t \\ &\iff \left\{ \begin{array}{l} \tilde{s}_i^2 - t_i = 0, \text{ and} \\ t_i^2 - 10t_i + 9 = 0 \end{array} \right\}, \quad i = 1, \dots, 2N_t. \end{aligned}$$

If we define  $\mathbf{w}^T \triangleq [\tilde{\mathbf{s}}^T \quad 1 \quad \mathbf{t}^T]$  and  $\mathbf{W} \triangleq \mathbf{w}\mathbf{w}^T$  then

$$\mathbf{W} = \begin{bmatrix} \tilde{\mathbf{s}}\tilde{\mathbf{s}}^T & \tilde{\mathbf{s}} & \tilde{\mathbf{s}}\mathbf{t}^T \\ \tilde{\mathbf{s}}^T & 1 & \mathbf{t}^T \\ \mathbf{t}\tilde{\mathbf{s}}^T & \mathbf{t} & \mathbf{t}\mathbf{t}^T \end{bmatrix} \triangleq \begin{bmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} & \mathbf{W}_{13} \\ \mathbf{w}_{21} & w_{22} & \mathbf{w}_{23} \\ \mathbf{W}_{31} & \mathbf{w}_{32} & \mathbf{W}_{33} \end{bmatrix}.$$

Therefore, the optimization problem in (5.6) can be rewritten as [25]

$$\min_{\mathbf{W}} \text{Trace}(\mathbf{W}\mathbf{Q}) \quad (5.8a)$$

$$\text{s.t.} \quad \text{diag}(\mathbf{W}_{11}) - \mathbf{w}_{32} = \mathbf{0}, \quad (5.8b)$$

$$\text{diag}(\mathbf{W}_{33}) - (U + L)\text{diag}(\mathbf{W}_{11}) + UL\mathbf{1} = \mathbf{0}, \quad (5.8c)$$

$$w_{22} = 1, \quad (5.8d)$$

$$\mathbf{W} \succeq \mathbf{0}, \quad \text{rank}(\mathbf{W}) = 1, \quad (5.8e)$$

where

$$\mathbf{Q} \triangleq \begin{bmatrix} \tilde{\mathbf{Q}}_{(2N_t+1) \times (2N_t+1)} & \mathbf{0}_{(2N_t+1) \times 2N_t} \\ \mathbf{0}_{2N_t \times (2N_t+1)} & \mathbf{0}_{2N_t \times 2N_t} \end{bmatrix}, \quad \tilde{\mathbf{Q}} \triangleq \begin{bmatrix} \tilde{\mathbf{H}}^T \tilde{\mathbf{H}} & -\tilde{\mathbf{H}}^T \tilde{\mathbf{y}} \\ -\tilde{\mathbf{y}}^T \tilde{\mathbf{H}} & 0 \end{bmatrix}, \quad (5.9)$$

the operator  $\text{diag}(\cdot)$  constructs a vector of the diagonal elements of its matrix argument, and  $\mathbf{1}$  is a vector with all its elements equal to 1. In this optimization problem  $\text{diag}(\mathbf{W}_{11}) - \mathbf{w}_{32} = \mathbf{0}$  is equivalent to  $\tilde{s}_i^2 - t_i = 0$  for  $i = 1, \dots, 2N_t$ , and  $\text{diag}(\mathbf{W}_{33}) - (U + L)\text{diag}(\mathbf{W}_{11}) + UL\mathbf{1} = \mathbf{0}$  is equivalent to  $t_i^2 - 10t_i + 9 = 0$  for  $i = 1, \dots, 2N_t$ . By relaxing the  $\text{rank}(\mathbf{W}) = 1$  constraint

the following semidefinite relaxation of (5.8) is obtained [25]

$$\min_{\mathbf{W}} \text{Trace}(\mathbf{W}\mathbf{Q}) \quad (5.10a)$$

$$\text{s.t. } \text{diag}(\mathbf{W}_{11}) - \mathbf{w}_{32} = \mathbf{0}, \quad (5.10b)$$

$$\text{diag}(\mathbf{W}_{33}) - (U + L)\text{diag}(\mathbf{W}_{11}) + UL\mathbf{1} = \mathbf{0}, \quad (5.10c)$$

$$w_{22} = 1, \quad (5.10d)$$

$$\mathbf{W} \succeq \mathbf{0}. \quad (5.10e)$$

The computational cost of solving this SDP using general purpose interior point methods (e.g., using packages like SeDuMi [128]) is polynomial in the problem size and of order  $O((4N_t + 1)^{6.5} \log \epsilon^{-1})$  [16], where  $\epsilon$  is the accuracy of solving the SDP. Hence, in addition to the increase in the problem dimension, the computational cost is a factor of  $O(N_t^3)$  larger than the interior point methods that are available to solve the SDPs in Chapter 4 for MIMO QPSK demodulation.

As in Chapter 4, an approximate solution to (5.6) can be obtained by performing a randomization procedure on the solution of (5.10),  $\mathbf{W}_{\text{opt}}$ . For this purpose, we partition  $\mathbf{W}_{\text{opt}}$  using the following structure

$$\mathbf{W}_{\text{opt}} = \begin{bmatrix} \mathbf{W}_{\text{opt},11} & \mathbf{w}_{\text{opt},12} & \mathbf{W}_{\text{opt},13} \\ \mathbf{w}_{\text{opt},21} & w_{\text{opt},22} & \mathbf{w}_{\text{opt},23} \\ \mathbf{W}_{\text{opt},31} & \mathbf{w}_{\text{opt},32} & \mathbf{W}_{\text{opt},33} \end{bmatrix}, \quad (5.11)$$

where  $\mathbf{W}_{\text{opt},11}, \mathbf{W}_{\text{opt},33} \in \mathbb{R}^{2N_t \times 2N_t}$ , and we define the matrix

$$\tilde{\mathbf{W}}_{\text{opt}} \triangleq \begin{bmatrix} \mathbf{W}_{\text{opt},11} & \mathbf{w}_{\text{opt},12} \\ \mathbf{w}_{\text{opt},21} & w_{\text{opt},22} \end{bmatrix}. \quad (5.12)$$

We then compute its factorization  $\tilde{\mathbf{W}}_{\text{opt}} = \mathbf{V}^T \mathbf{V}$ , and choose a random vector  $\mathbf{u}$  from the uniform distribution on the unit sphere and compute

$$\check{\mathbf{s}}' = \mathcal{Q} \left( \frac{\mathbf{V}^T \mathbf{u}}{\mathbf{v}_{2N_t+1}^T \mathbf{u}} \right), \quad (5.13)$$

where  $\mathcal{Q}(\cdot)$  is a quantizer to the values in  $\mathcal{A} = \{\pm 1, \pm 3\}$ . We then repeat the randomization procedure and among all generated symbol-vectors, we pick  $\tilde{\mathbf{s}}' \triangleq [\check{s}'_1, \dots, \check{s}'_{2N_t}]^T$  with the smallest value for the optimization problem in (5.4), and we call it  $\tilde{\mathbf{s}}'_{\text{odr}}$ . In contrast to the randomization procedure for MIMO QPSK in Chapter 4, we are not aware of any analysis available in the literature with which to evaluate the quality of this randomization procedure, but the simulation results in [25] show that this procedure provides good bit error rate performance.

### 5.1.2 Fixed dimension relaxation

Adding the slack variables in the previous relaxation technique results in an increase in the problem dimension, and hence an increase in the computational cost of the SDP. An alternative technique was proposed by Sidiropoulos and Luo [26], where, in addition to relaxing the rank-1 constraint, they also relax the constraints in (5.7c) by replacing them with several inequality constraints. This is performed by defining

$$\mathbf{X} \triangleq \begin{bmatrix} \mathbf{X}_{11} & \mathbf{x}_{12} \\ \mathbf{x}_{21} & x_{22} \end{bmatrix},$$

and relaxing the constraints  $[\mathbf{X}]_{ii} \in \mathcal{B}$ ,  $i = 1, \dots, 2N_t$  to

$$L \leq \text{diag}(\mathbf{X}_{11}) \leq U. \quad (5.14)$$

As a result, in contrast to (5.7) the problem dimension will not increase. Another advantage of this method is that it can be used for generic  $M$ -ary QAM constellations by adjusting  $L$  and  $U$  appropriately. By relaxing the rank-1 constraint the following semidefinite program is obtained [26]

$$\min_{\mathbf{X}} \text{Trace}(\mathbf{X}\tilde{\mathbf{Q}}) \quad (5.15a)$$

$$\text{s.t. } L \leq \text{diag}(\mathbf{X}_{11}) \leq U, \quad (5.15b)$$

$$x_{22} = 1, \quad (5.15c)$$

$$\mathbf{X} \succeq 0. \quad (5.15d)$$

A fast interior point algorithm for a general class of SDPs which includes the SDP in (5.15) was proposed in [23]. In Section 5.1.3, we will specialize that algorithm to the problem in (5.15) in a way that exploits the specific structure of that SDP. Hence, (5.15) can be solved with a computational cost of the same order as that of the SDPs for the QPSK case in Chapter 4. As in Section 5.1.1, a randomization procedure can be applied to the factorization of the optimum solution to (5.15),  $\mathbf{X}_{\text{opt}} = \mathbf{V}^T \mathbf{V}$ , to obtain an approximate solution to (5.6); cf. (5.13) and the subsequent discussion.

### 5.1.3 A fast interior point SDP solver for (5.15)

Several fast interior point algorithms based on the principles outlined in [23] have been developed for multiuser and MIMO demodulation using SDR

(e.g., [96, 108, 112]). The work in [23] considered the rather large class of semidefinite programs with linear equality and inequality constraints. These specialized algorithms have been shown to be efficient and reliable for demodulation of BPSK and QPSK symbols (e.g., [96, 112]) and also for M-ary PSK symbols (e.g., [108]). In this section, following the standard primal-dual interior point methods in linear programming (e.g., [16]) and the principles developed in [23] we provide a fast reliable interior point algorithm specialized for solving the SDP in (5.15).<sup>2</sup> That is, using the specific structure of (5.15) we derive explicit expression for the search directions of the generic interior point method of [23].

By defining the dual variables  $\mathbf{p}_u \in \mathbb{R}^{2N_t}$ ,  $\mathbf{p}_l \in \mathbb{R}^{2N_t}$ ,  $v \in \mathbb{R}$ ,  $\mathbf{Z} \in \mathbb{S}_{2N_t+1}$ , where  $\mathbb{S}_n$  is the set of symmetric  $n \times n$  matrices, the dual problem associated to the primal semidefinite program in (5.15) can be written as

$$\max_{\mathbf{p}_u, \mathbf{p}_l, v, \mathbf{Z}} U\mathbf{1}^T \mathbf{p}_u - L\mathbf{1}^T \mathbf{p}_l + v \quad (5.16a)$$

$$\text{s.t. } \tilde{\mathbf{Q}} - \text{Diag}([\mathbf{p}_l^T - \mathbf{p}_u^T, -v]^T) = \mathbf{Z}, \quad (5.16b)$$

$$\mathbf{p}_u \geq 0, \mathbf{p}_l \geq 0, \mathbf{Z} \succeq 0. \quad (5.16c)$$

where the operator  $\text{Diag}(\cdot)$  constructs a diagonal matrix with its argument on the diagonal, and the first two inequalities in (5.16c) are defined elementwise. The primal-dual interior point method for jointly solving the primal and dual optimization problems in (5.15) and (5.16) involves solving the

---

<sup>2</sup>A concurrent and similar derivation of this interior point method to solve the SDP in (5.15) was provided in [129].

following KKT equations for a sequence of values of complementarity parameter  $\mu > 0$  that decays to zero,

$$\mathbf{Z} - \tilde{\mathbf{Q}} + \text{Diag}\{[\mathbf{p}_l^T - \mathbf{p}_u^T, -v]^T\} = 0, \quad (5.17a)$$

$$1 - x_{22} = 0, \quad (5.17b)$$

$$U\mathbf{1} - \mu\mathbf{p}_u^{-1} - \text{diag}(\mathbf{X}_{11}) = 0, \quad (5.17c)$$

$$\text{diag}(\mathbf{X}_{11}) - L\mathbf{1} - \mu\mathbf{p}_l^{-1} = 0, \quad (5.17d)$$

$$\mathbf{Z}\mathbf{X} - \mu\mathbf{I} = 0. \quad (5.17e)$$

Since some of these equations are nonlinear we use the Newton method (e.g., [130]) to find directions  $\Delta\mathbf{p}_u$ ,  $\Delta\mathbf{p}_l$ ,  $\Delta v$ ,  $\Delta\mathbf{Z}$  and  $\Delta\mathbf{X}$  toward the solution to (5.17), as given in step 2 of the algorithm given in Tab. 5.1. The key feature of this algorithm lies in the fact that we use the special structure of the optimization problem (5.15) to derive explicit expressions for the interior point search directions. As described in the algorithm provided in Tab. 5.1, in order to maintain the feasibility of the solution, we perform a line search to update  $\mathbf{p}_u$ ,  $\mathbf{p}_l$ ,  $v$ ,  $\mathbf{Z}$  and  $\mathbf{X}$ . After completing one Newton step, we update the complementarity parameter  $\mu$  and we take the next Newton step with this new complementarity parameter in order to reduce the duality gap to a prespecified value  $\epsilon$ . As was pointed out in [23], this algorithm requires at most  $O(\sqrt{N_t} \log \epsilon^{-1})$  iterations to reach to this duality gap and since in each iteration it performs operations of order  $O(N_t^3)$ , the total computational complexity of the algorithm is  $O(N_t^{3.5} \log \epsilon^{-1})$ .

Table 5.1: Interior point algorithm for solving (5.15)

- 
- Data:  $\tilde{\mathbf{Q}}, L, U$ , duality gap  $\epsilon > 0$ , feasible initial points for primal and dual variables  $\mathbf{X}, \mathbf{p}_l, \mathbf{p}_u, v$  and  $\mathbf{Z}$ .
  - Output: Primal and dual optimal solutions  $\mathbf{X}, \mathbf{p}_l, \mathbf{p}_u, v$  and  $\mathbf{Z}$ .
1. Compute  $\mu = (L\mathbf{1}^T \mathbf{p}_l - U\mathbf{1}^T \mathbf{p}_u - v - \text{Trace}(\tilde{\mathbf{Q}}\mathbf{X}))/2(6N_t + 1)$ .
  2. Find the search directions  $\Delta \mathbf{p}_u, \Delta \mathbf{p}_l$  and  $\Delta v$  by solving the linear equations

$$\begin{bmatrix} \mathbf{Y}_{11} + \mathbf{D}_u & -\mathbf{Y}_{11} & \mathbf{y} \\ -\mathbf{Y}_{11} & \mathbf{Y}_{11} + \mathbf{D}_l & -\mathbf{y} \\ \mathbf{y}^T & -\mathbf{y}^T & y_{22} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{p}_u \\ \Delta \mathbf{p}_l \\ \Delta v \end{bmatrix} = \begin{bmatrix} -U\mathbf{1} + \mu \mathbf{p}_u^{-1} + \mu \bar{\mathbf{Z}}_{11} \\ L\mathbf{1} + \mu \mathbf{p}_l^{-1} - \mu \bar{\mathbf{Z}}_{11} \\ -x_{22} + \mu \bar{z}_{22} \end{bmatrix}, \quad (5.18)$$

where

$$\mathbf{Y} \triangleq \begin{bmatrix} \mathbf{Y}_{11} & \mathbf{y} \\ \mathbf{y}^T & y_{22} \end{bmatrix} = \mathbf{Z}^{-1} \circ \mathbf{X}, \quad (5.19a)$$

$$\bar{\mathbf{Z}} \triangleq \begin{bmatrix} \bar{\mathbf{Z}}_{11} & \bar{\mathbf{z}} \\ \bar{\mathbf{z}}^T & \bar{z}_{22} \end{bmatrix} = \mathbf{Z}^{-1}, \quad (5.19b)$$

$$\mathbf{D}_u \triangleq \text{Diag}(\mathbf{p}_u^{-1} \circ (\text{diag}(\mathbf{X}_{11}) + U\mathbf{1})), \quad (5.19c)$$

$$\mathbf{D}_l \triangleq \text{Diag}(\mathbf{p}_l^{-1} \circ (\text{diag}(\mathbf{X}_{11}) - L\mathbf{1})), \quad (5.19d)$$

and  $\circ$  is the element-wise matrix multiplication. Then compute the search directions  $\Delta \mathbf{Z}$  and  $\Delta \mathbf{X}$  as

$$\Delta \mathbf{Z} = \text{Diag}\{[\Delta \mathbf{p}_l^T - \Delta \mathbf{p}_u^T, -\Delta v]^T\}, \quad (5.20)$$

$$\Delta \tilde{\mathbf{X}} = -\mathbf{X} + \mu \mathbf{Z}^{-1} - \mathbf{Z}^{-1} \Delta \mathbf{Z} \mathbf{X}. \quad (5.21)$$

$$\Delta \mathbf{X} = (\Delta \tilde{\mathbf{X}} + \Delta \tilde{\mathbf{X}}^T)/2. \quad (5.22)$$

3. Perform line search to find a primal step size  $0 < \alpha_p \leq 1$  such that  $\mathbf{X} + \alpha_p \Delta \mathbf{X} \succ 0$  and  $L \leq \text{diag}(\mathbf{X}_{11} + \alpha_p \Delta \mathbf{X}_{11}) \leq U$ .
  4. Perform line search to find a dual step size  $0 < \alpha_d \leq 1$  such that  $\mathbf{Z} + \alpha_d \Delta \mathbf{Z} \succ 0, \mathbf{p}_u + \alpha_d \Delta \mathbf{p}_u \geq 0$  and  $\mathbf{p}_l + \alpha_d \Delta \mathbf{p}_l \geq 0$ .
  5. Update  $\mathbf{X} = \mathbf{X} + \alpha_p \Delta \mathbf{X}, \mathbf{Z} = \mathbf{Z} + \alpha_d \Delta \mathbf{Z}, \mathbf{p}_u = \mathbf{p}_u + \alpha_d \Delta \mathbf{p}_u, \mathbf{p}_l = \mathbf{p}_l + \alpha_d \Delta \mathbf{p}_l$  and  $v = v + \alpha_d \Delta v$ .
  6. If the duality gap is less than  $\epsilon$ , that is if  $(L\mathbf{1}^T \mathbf{p}_l - U\mathbf{1}^T \mathbf{p}_u - v - \text{Trace}(\tilde{\mathbf{Q}}\mathbf{X})) < \epsilon$ , terminate the algorithm, otherwise go to 1.
-

### 5.1.4 Equality of increased dimension and fixed dimension relaxation techniques

In this section we will show that for the quadratic optimization problem in (5.6) the increased dimension SDR technique in (5.10) and the fixed dimension SDR technique in (5.15) are equivalent in the sense that they have the same optimal value and that the matrix that is factorized in the randomization procedure for generating approximate solution to (5.6) is also the same. We will begin with the following proposition.

**Proposition 5.1.** *The optimal values of the semidefinite programs in (5.10) and (5.15) are equal.*

*Proof:*<sup>3</sup> Let  $R_{\text{ID}}$  denote the optimum value of (5.10) and let  $R_{\text{FD}}$  denote the optimum value of (5.15). The proof has two parts. In the first part we show that  $R_{\text{FD}} \leq R_{\text{ID}}$  and in the second part we show that  $R_{\text{ID}} \leq R_{\text{FD}}$ . Hence  $R_{\text{FD}} = R_{\text{ID}}$ .

In order to show that  $R_{\text{FD}} \leq R_{\text{ID}}$ , we need to show that if a matrix

$$\mathbf{W} \triangleq \begin{bmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} & \mathbf{W}_{13} \\ \mathbf{w}_{21} & w_{22} & \mathbf{w}_{23} \\ \mathbf{W}_{31} & \mathbf{w}_{32} & \mathbf{W}_{33} \end{bmatrix} \quad (5.23)$$

is in the feasible region of (5.10), then the matrix

$$\bar{\mathbf{W}}_1 \triangleq \begin{bmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{21} & w_{22} \end{bmatrix} \quad (5.24)$$

---

<sup>3</sup>A different derivation of this proof was concurrently proposed in [129].

is in the feasible region of (5.15). In order to show this, we use the following lemma, in which  $\mathbb{M}_m$  denotes the set of  $m \times m$  matrices and the contraction  $\mathbf{C} \in \mathbb{M}_{m+n}$  is a matrix with a maximum singular value less than or equal to 1.

**Lemma 5.2.** [131, Lemma 3.5.12] *Given  $\mathbf{L} \in \mathbb{M}_m$  and  $\mathbf{M} \in \mathbb{M}_n$ , the matrix  $\begin{bmatrix} \mathbf{L} & \mathbf{X} \\ \mathbf{X}^T & \mathbf{M} \end{bmatrix} \in \mathbb{M}_{m+n}$  is positive semidefinite if and only if  $\mathbf{L}$  and  $\mathbf{M}$  are positive semidefinite and there is a contraction  $\mathbf{C} \in \mathbb{M}_{m+n}$  such that  $\mathbf{X} = \mathbf{L}^{1/2} \mathbf{C} \mathbf{M}^{1/2}$ .*

Since we know that  $w_{22} = 1$  (cf. (5.10d)), we define

$$\bar{\mathbf{W}}_2 \triangleq \begin{bmatrix} 1 & \mathbf{w}_{23} \\ \mathbf{w}_{32} & \mathbf{W}_{33} \end{bmatrix}. \quad (5.25)$$

According to Lemma 5.2, since  $\mathbf{W} \succeq 0$  we have  $\bar{\mathbf{W}}_2 \succeq 0$ . Multiplying  $\bar{\mathbf{W}}_2$  by permutation matrices to interchange its  $i$ th row and column ( $2 \leq i \leq 2N_t+1$ ) with the second row and column, respectively, the  $2 \times 2$  matrix in the north-west corner of this permuted matrix is

$$\begin{bmatrix} 1 & [\mathbf{w}_{23}]_{i-1} \\ [\mathbf{w}_{32}]_{i-1} & [\mathbf{W}_{33}]_{i-1,i-1} \end{bmatrix}. \quad (5.26)$$

Applying Lemma 5.2 to that matrix, we can write  $[\mathbf{W}_{33}]_{i-1,i-1} \geq ([\mathbf{w}_{32}]_{i-1})^2$ . By defining  $[\cdot]^2$  as the element-wise square operator on a vector, we can write:

$$\text{diag}(\mathbf{W}_{33}) \geq [\mathbf{w}_{32}]^2. \quad (5.27)$$

Using (5.10b), (5.10c) and (5.27) we have:

$$(U + L)\text{diag}(\mathbf{W}_{11}) - UL\mathbf{1} \geq [\text{diag}(\mathbf{W}_{11})]^2 \quad (5.28)$$

$$\Rightarrow (\text{diag}(\mathbf{W}_{11}) - U)(\text{diag}(\mathbf{W}_{11}) - L) \leq 0 \quad (5.29)$$

$$\Rightarrow L \leq \text{diag}(\mathbf{W}_{11}) \leq U. \quad (5.30)$$

Applying Lemma 5.2 to a different partition of  $\mathbf{W}$  we also have  $\mathbf{W}_{11} \succeq 0$ . Hence,  $\bar{\mathbf{W}}_1$  is in the feasible region of problem (5.15), and therefore  $R_{\text{FD}} \leq R_{\text{ID}}$ .

In order to prove  $R_{\text{ID}} \leq R_{\text{FD}}$ , we need to find a matrix  $\mathbf{W}$  of the form in (5.23) that lies in the feasible region of the problem in (5.10) and is such that the matrix  $\begin{bmatrix} \mathbf{W}_{11} & \mathbf{w}_{12} \\ \mathbf{w}_{12}^T & 1 \end{bmatrix}$  lies in the feasible region of the problem in

(5.15). In order to find such a matrix, let  $\mathbf{X} \triangleq \begin{bmatrix} \mathbf{X}_{11} & \mathbf{x}_{12} \\ \mathbf{x}_{12}^T & 1 \end{bmatrix}$  denote a generic matrix in the feasible region of (5.15) and define the following matrices and vectors

$$\mathbf{z} \triangleq \text{diag}(\mathbf{X}_{11}), \quad (5.31a)$$

$$\tilde{\mathbf{x}} \triangleq [\mathbf{x}_{12}^T \quad 1 \quad \mathbf{0}_{1 \times 2N_t}]^T, \quad (5.31b)$$

$$\mathbf{y} \triangleq [\mathbf{x}_{12}^T \quad 1 \quad \mathbf{z}^T]^T, \quad (5.31c)$$

$$\bar{\mathbf{X}} \triangleq \begin{bmatrix} \mathbf{X}_{11} & \mathbf{x}_{12} & \mathbf{0}_{2N_t \times 2N_t} \\ \mathbf{x}_{12}^T & 1 & \mathbf{0}_{1 \times 2N_t} \\ \mathbf{0}_{2N_t \times 2N_t} & \mathbf{0}_{2N_t \times 1} & \mathbf{0}_{2N_t \times 2N_t} \end{bmatrix} \quad (5.31d)$$

Using the constraints of problem (5.15) we can write

$$L \leq z_i \leq U \implies z_i^2 - (U + L)z_i + UL \leq 0, \quad i = 1, \dots, 2N_t.$$

This implies that there exists a  $c_i \geq 0$  such that

$$c_i + z_i^2 - (U + L)z_i + UL = 0, \quad i = 1, \dots, 2N_t.$$

Hence, if we construct a matrix  $\tilde{\mathbf{W}}$  as

$$\tilde{\mathbf{W}} \triangleq \bar{\mathbf{X}} + \mathbf{y}\mathbf{y}^T - \tilde{\mathbf{x}}\tilde{\mathbf{x}}^T + \text{Diag}([\mathbf{0}_{1 \times (2N_t+1)}, c_1, \dots, c_{2N_t}]) \quad (5.32)$$

and if we show that  $\tilde{\mathbf{W}} \succeq 0$ ,  $\tilde{\mathbf{W}}$  will satisfy all the constraints of problem (5.10). Since  $\mathbf{y}\mathbf{y}^T$  and  $\text{Diag}([\mathbf{0}_{1 \times (2N_t+1)}, c_1, \dots, c_{2N_t}])$  are positive semidefinite (PSD), it remains to prove that  $\bar{\mathbf{X}} - \tilde{\mathbf{x}}\tilde{\mathbf{x}}^T \succeq 0$ . For this purpose, we use the following theorem from [118].

**Theorem 5.3.** (*[118, Theorem 7.7.7]*) Let  $\mathbf{A} \in \mathbb{M}_n$  and  $\mathbf{C} \in \mathbb{M}_m$  be positive definite and let  $\mathbf{B} \in \mathbb{M}_{m+n}$ . The following are equivalent:

- a)  $(\mathbf{x}^T \mathbf{A} \mathbf{x})(\mathbf{y}^T \mathbf{C} \mathbf{y}) \geq |\mathbf{x}^T \mathbf{B} \mathbf{y}|^2$  for all  $\mathbf{x} \in \mathbb{R}^n$  and for all  $\mathbf{y} \in \mathbb{R}^m$ ,
- b)  $\mathbf{x}^T \mathbf{A} \mathbf{x} + \mathbf{y}^T \mathbf{C} \mathbf{y} \geq 2|\mathbf{x}^T \mathbf{B} \mathbf{y}|$  for all  $\mathbf{x} \in \mathbb{R}^n$  and for all  $\mathbf{y} \in \mathbb{R}^m$ ,
- c)  $\rho(\mathbf{B}^T \mathbf{A}^{-1} \mathbf{B} \mathbf{C}^{-1}) \leq 1$ ,
- d)  $\begin{bmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{C} \end{bmatrix} \succeq 0$ ,

in which  $\rho(\mathbf{A})$  is the maximum eigenvalue of matrix  $\mathbf{A}$ .

According to parts a) and d) of this theorem we can write

$$\begin{aligned}
\bar{\mathbf{X}} \succeq 0 &\implies \mathbf{a}^T \mathbf{X}_{11} \mathbf{a} \geq |\mathbf{a}^T \mathbf{x}_{12}|^2, \quad \forall \mathbf{a} \in \mathbb{R}^{2N_t+1} \\
&\implies \mathbf{a}^T \mathbf{X}_{11} \mathbf{a} - \mathbf{a}^T \mathbf{x}_{12} \mathbf{x}_{12}^T \mathbf{a} \geq 0, \quad \forall \mathbf{a} \in \mathbb{R}^{2N_t+1} \\
&\implies \mathbf{X}_{11} - \mathbf{x}_{12} \mathbf{x}_{12}^T \succeq 0 \\
&\implies \bar{\mathbf{X}} - \tilde{\mathbf{x}} \tilde{\mathbf{x}}^T \succeq 0.
\end{aligned}$$

So,  $\tilde{\mathbf{W}} \succeq 0$  and the proof of Proposition 5.1 is complete.  $\square$

Now we provide a proof of the following proposition.

**Proposition 5.4.** *If the optimum solutions to problems (5.10) and (5.15) are denoted by  $\hat{\mathbf{W}}$  and  $\hat{\mathbf{X}}$ , respectively, then  $\hat{\mathbf{X}} = \hat{\mathbf{W}}_1$ , where  $\hat{\mathbf{W}}_1 \triangleq \begin{bmatrix} \hat{\mathbf{W}}_{11} & \hat{\mathbf{w}}_{12} \\ \hat{\mathbf{w}}_{21} & \hat{w}_{22} \end{bmatrix}$ .*

*Proof:* The proof is by contradiction. If the matrix  $\hat{\mathbf{W}}_1$  is not an optimum solution to (5.15) and  $\text{Trace}(\hat{\mathbf{X}}\tilde{\mathbf{Q}})$  is smaller than  $\text{Trace}(\hat{\mathbf{W}}_1\tilde{\mathbf{Q}})$ , we can construct a matrix  $\hat{\mathbf{W}}'$  in the feasible region of (5.10) using  $\hat{\mathbf{X}}$  and the method described in the proof of Proposition 5.1. However, since the optimal values of (5.10) and (5.15) are the same, such a matrix would result in  $\text{Trace}(\hat{\mathbf{W}}'\tilde{\mathbf{Q}}) < \text{Trace}(\hat{\mathbf{W}}\tilde{\mathbf{Q}})$ , which is a contradiction, and hence the proof.  $\square$

As an aside, we would like to point out that although the equivalence of the optimal solutions to (5.10) and (5.15) has been shown, these problems have different dimensions and the structure of the constraints are quite different. Therefore, the trajectory of the interior point iterations in the solution of these problems may be different, and hence if we terminate the interior point iterations early for reasons of computational cost, the resulting solutions may only be approximately equivalent.

In this section we have shown that for the quadratic objective function in (5.6) the increased dimension relaxation in (5.10) and the fixed dimension relaxation in (5.15) are equivalent. However, an advantage of the increased dimension relaxation is that it naturally accommodates a broader class of quadratic objective functions. We will exploit that advantage in the next section.

## 5.2 Maximum a posteriori probability MIMO

### 16-QAM demodulation using SDR

In Section 5.1 we considered the problem of maximum likelihood detection of i.i.d. symbols. In this section we assume the availability of *a priori* information  $p(\tilde{\mathbf{s}})$  for each transmitted symbol-vector  $\tilde{\mathbf{s}} = [\tilde{s}_1, \dots, \tilde{s}_{2N_t}]^T$  in the channel use of interest. In that case, the maximum *a posteriori* (MAP) estimate of the transmitted symbol-vector can be expressed as

$$\tilde{\mathbf{s}}_{\text{MAP}} = \arg \max_{\tilde{\mathbf{s}} \in \mathcal{A}^{2N_t}} p(\tilde{\mathbf{y}}|\tilde{\mathbf{s}})p(\tilde{\mathbf{s}}). \quad (5.33)$$

Since the transmission is over a Gaussian MIMO channel, this MAP (hard) decision can be performed by solving the following optimization problem (e.g., [85])

$$\tilde{\mathbf{s}}_{\text{MAP}} = \arg \min_{\tilde{\mathbf{s}} \in \mathcal{A}^{2N_t}} D(\tilde{\mathbf{s}}), \quad (5.34)$$

where, in place of (5.3) we have

$$D(\tilde{\mathbf{s}}) = \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_2^2 - 2\sigma^2 \log p(\tilde{\mathbf{s}}). \quad (5.35)$$

If we were to use tree search algorithms to solve this optimization problem, the tree structure of the cost function (5.35) can be exploited in a straight forward manner, as was described in Chapter 3. However, since  $\log p(\tilde{\mathbf{s}})$  in (5.35) is a non-polynomial expression, we can not use the semidefinite relaxation approaches reviewed in Sections 5.1.1 and 5.1.2 to approximate the solution to (5.34) in its current format. If the semidefinite relaxation technique in (5.10) is to be used to find the approximate solution to (5.34), cost functions with polynomial expressions of up to a degree of 4 of the elements in  $\tilde{\mathbf{s}}$  can be accommodated. In contrast, the application of the semidefinite relaxation technique in (5.15) is limited to cost functions  $D(\tilde{\mathbf{s}})$  with second-order polynomial expressions in the elements of  $\tilde{\mathbf{s}}$ . In the following subsection we will provide two methods to obtain polynomial approximations of  $\log p(\tilde{\mathbf{s}})$ . To construct these approximations, we first make the assumption that the elements of  $\tilde{\mathbf{s}}$ ,  $\tilde{s}_i$ , are independent (e.g., [12]). We then provide an exact express for  $\log p(\tilde{s}_i)$  as a polynomial of up to degree 3 and an approximation for  $\log p(\tilde{s}_i)$  in the form of a second order polynomial. Then we will provide two SDPs to approximate the solution to (5.34) based on these polynomial expressions for the *a priori* information component of the cost function.

### 5.2.1 Polynomial approximations of the cost function

In this section we will derive polynomial approximations for the *a priori* information expression  $\log p(\tilde{\mathbf{s}})$  in the cost function of the MAP demodulation

problem (5.34). The first step is to make the approximation that the elements of  $\tilde{\mathbf{s}}$  are independent, and hence to approximate  $\log p(\tilde{\mathbf{s}})$  as<sup>4</sup>

$$\log p(\tilde{\mathbf{s}}) \approx f(\tilde{\mathbf{s}}) = \sum_{i=1}^{2N_t} \log p_i(\tilde{s}_i). \quad (5.36)$$

The second step is to determine a polynomial expression for an approximation of each of the summands in (5.36),  $\log p_i(\tilde{s}_i)$ ,  $i = 1, \dots, 2N_t$ , for the case of 16-QAM signaling. We will first derive an exact third-order expression for  $\log p_i(\tilde{s}_i)$  and subsequently a second-order approximation.

We will derive the third-order expression by showing that for  $\mathcal{A} = \{\pm 1, \pm 3\}$  there exists a quadruple  $(a_i, b_i, c_i, d_i)$  such that

$$\log p_i(\tilde{s}_i) = a_i \tilde{s}_i^3 + b_i \tilde{s}_i^2 + c_i \tilde{s}_i + d_i \text{ for all } \tilde{s}_i \in \mathcal{A}. \quad (5.37)$$

The values for  $a_i, b_i, c_i, d_i$  can be found simply by enforcing the equality in (5.37) for each  $\tilde{s}_i \in \mathcal{A}$ . If we define  $\bar{s}_1 = -3$ ,  $\bar{s}_2 = -1$ ,  $\bar{s}_3 = +1$  and  $\bar{s}_4 = +3$ , the values for  $a_i, b_i, c_i, d_i$  can be found by solving the following set of linear equations

$$\underbrace{\begin{bmatrix} \bar{s}_1^3 & \bar{s}_1^2 & \bar{s}_1 & 1 \\ \bar{s}_2^3 & \bar{s}_2^2 & \bar{s}_2 & 1 \\ \bar{s}_3^3 & \bar{s}_3^2 & \bar{s}_3 & 1 \\ \bar{s}_4^3 & \bar{s}_4^2 & \bar{s}_4 & 1 \end{bmatrix}}_{\triangleq \mathbf{C}_{3\text{rd}}} \begin{bmatrix} a_i \\ b_i \\ c_i \\ d_i \end{bmatrix} = \begin{bmatrix} \log p_i(\tilde{s}_i = \bar{s}_1) \\ \log p_i(\tilde{s}_i = \bar{s}_2) \\ \log p_i(\tilde{s}_i = \bar{s}_3) \\ \log p_i(\tilde{s}_i = \bar{s}_4) \end{bmatrix}. \quad (5.38)$$

By solving (5.38) for each  $i \in \{1, \dots, 2N_t\}$  and collecting the solutions  $a_i$ ,

---

<sup>4</sup>Note that in iterative demodulation and decoding schemes, the prior information that is provided to the demodulator by the previous iteration of the decoder is based on an assumption of independence, and hence can be interpreted as an approximation of  $\log p(\tilde{\mathbf{s}})$  in (5.36).

$b_i$ ,  $c_i$  and  $d_i$  in the vectors  $\mathbf{a}$ ,  $\mathbf{b}$ ,  $\mathbf{c}$  and  $\mathbf{d}$ , respectively, we obtain the following third-order polynomial expression for (5.36)

$$\log p(\tilde{\mathbf{s}}) \simeq f(\tilde{\mathbf{s}}) = \mathbf{a}^T \tilde{\mathbf{s}}^3 + \mathbf{b}^T \tilde{\mathbf{s}}^2 + \mathbf{c}^T \tilde{\mathbf{s}} + \mathbf{d}^T \mathbf{1}, \quad (5.39)$$

where the only approximation here is due to the independence assumption that lead to (5.36).

Now we will derive a second-order approximation for  $\log p(\tilde{\mathbf{s}})$ , based on the independence assumption and a second-order approximation of  $\log p_i(\tilde{s}_i)$ . Since  $\tilde{s}_i$  takes four possible values, it is not possible to exactly interpolate  $\log p_i(\tilde{s}_i)$  with the second-order polynomial  $b_i \tilde{s}_i^2 + c_i \tilde{s}_i + d_i$ . Instead, we will choose  $b_i$ ,  $c_i$  and  $d_i$  to minimize the mean squared error between  $\log p_i(\tilde{s}_i)$  and  $b_i \tilde{s}_i^2 + c_i \tilde{s}_i + d_i$  over  $\tilde{s}_i \in \mathcal{A}$ . That is,  $b_i$ ,  $c_i$  and  $d_i$  will be chosen as the solution of the following set of linear equations

$$\underbrace{\begin{bmatrix} \sum_{\bar{s} \in \mathcal{A}} \bar{s}^4 & \sum_{\bar{s} \in \mathcal{A}} \bar{s}^3 & \sum_{\bar{s} \in \mathcal{A}} \bar{s}^2 \\ \sum_{\bar{s} \in \mathcal{A}} \bar{s}^3 & \sum_{\bar{s} \in \mathcal{A}} \bar{s}^2 & \sum_{\bar{s} \in \mathcal{A}} \bar{s} \\ \sum_{\bar{s} \in \mathcal{A}} \bar{s}^2 & \sum_{\bar{s} \in \mathcal{A}} \bar{s} & 4 \end{bmatrix}}_{\triangleq \mathbf{C}_{2\text{nd}}} \begin{bmatrix} b_i \\ c_i \\ d_i \end{bmatrix} = \begin{bmatrix} \sum_{\bar{s} \in \mathcal{A}} \bar{s}^2 \log p_i(\tilde{s}_i = \bar{s}) \\ \sum_{\bar{s} \in \mathcal{A}} \bar{s} \log p_i(\tilde{s}_i = \bar{s}) \\ \sum_{\bar{s} \in \mathcal{A}} \log p_i(\tilde{s}_i = \bar{s}) \end{bmatrix}. \quad (5.40)$$

By solving this set of equations for each  $i \in \{1, \dots, 2N_t\}$  and collecting the solutions  $b_i$ ,  $c_i$  and  $d_i$  in the vectors  $\mathbf{b}$ ,  $\mathbf{c}$  and  $\mathbf{d}$ , respectively, the second-order approximation of (5.36) can be written as

$$\log p(\tilde{\mathbf{s}}) \simeq f(\tilde{\mathbf{s}}) \simeq \hat{f}(\tilde{\mathbf{s}}) = \mathbf{b}^T \tilde{\mathbf{s}}^2 + \mathbf{c}^T \tilde{\mathbf{s}} + \mathbf{d}^T \mathbf{1}. \quad (5.41)$$

We should note that since  $\mathbf{C}_{2\text{nd}}$  and  $\mathbf{C}_{3\text{rd}}$  are not dependent on the channel information nor the *a priori* information, the factorizations of these

matrices can be computed and stored in memory in order to reduce the computational cost of solving the linear equations in (5.38) and (5.40) for each  $i \in \{1, \dots, 2N_t\}$  in each demodulation-decoding iteration for each channel use.

## 5.2.2 SDR formulation of MAP demodulation

By using the polynomial approximations developed in the previous section, the *a priori* component of the cost function of the MAP demodulation problem in (5.34) can be approximated using a third-order polynomial that is compatible with the increased dimension SDR approach in (5.10), or a second-order polynomial that is compatible with both the increased and the fixed dimension SDR approaches in (5.10) and (5.15).

Using the third-order approximation in (5.39), the cost function can be written as

$$D(\tilde{\mathbf{s}}) \simeq \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_2^2 - 2\sigma^2(\mathbf{a}^T\tilde{\mathbf{s}}^3 + \mathbf{b}^T\tilde{\mathbf{s}}^2 + \mathbf{c}^T\tilde{\mathbf{s}} + \mathbf{d}^T\mathbf{1}). \quad (5.42)$$

If we define  $\mathbf{w}^T \triangleq [\tilde{\mathbf{s}}^T \quad 1 \quad \mathbf{t}^T]$  where  $\mathbf{t}$  contains the slack variables  $t_i = \tilde{s}_i^2$ ,  $i = 1, \dots, 2N_t$ , (5.42) can be rewritten as  $\mathbf{w}^T \mathbf{Q} \mathbf{w}$ , where

$$\mathbf{Q} \triangleq \begin{bmatrix} \tilde{\mathbf{Q}}_{2N_t+1 \times 2N_t+1} & -\sigma^2 \mathbf{Diag}(\mathbf{a}) \\ -\sigma^2 \mathbf{Diag}(\mathbf{a}) & \mathbf{0}_{2N_t \times 2N_t} \end{bmatrix} \quad (5.43)$$

and

$$\tilde{\mathbf{Q}} \triangleq \begin{bmatrix} \tilde{\mathbf{H}}^T \tilde{\mathbf{H}} - 2\sigma^2 \mathbf{Diag}(\mathbf{b}) & -\tilde{\mathbf{H}}^T \tilde{\mathbf{y}} - \sigma^2 \mathbf{c} \\ -\tilde{\mathbf{y}}^T \tilde{\mathbf{H}} - \sigma^2 \mathbf{c}^T & 0 \end{bmatrix}. \quad (5.44)$$

Hence, using the increased order relaxation approach, the MAP decision in (5.34) can be approximated by solving the following semidefinite program

$$\min_{\mathbf{W}} \text{Trace}(\mathbf{W}\mathbf{Q}) \quad (5.45a)$$

$$\text{s.t. } \text{diag}(\mathbf{W}_{11}) - \mathbf{w}_{32} = \mathbf{0}, \quad (5.45b)$$

$$\text{diag}(\mathbf{W}_{33}) - (U + L)\text{diag}(\mathbf{W}_{11}) + UL\mathbf{1} = \mathbf{0}, \quad (5.45c)$$

$$w_{22} = 1, \quad (5.45d)$$

$$\mathbf{W} \succeq \mathbf{0}, \quad (5.45e)$$

and performing the randomization procedure described in Section 5.1.1. (The SDP in (5.45) is analogous to that in (5.10), but with the matrix  $\mathbf{Q}$  in (5.43) replacing the matrix  $\mathbf{Q}$  in (5.9).)

As explained in Section 5.1.2, the fixed dimension relaxation (5.15) of the ML demodulation problem (5.6) has the advantages that its dimensionality is less than that in (5.10) and that it can be solved with the efficient interior point method developed in Section 5.1.2. In order to exploit these advantages we need to approximate the cost function of the MAP optimization problem by a second-order polynomial in the elements of  $\tilde{\mathbf{s}}$ . For this purpose, we use the second-order approximation (5.41) and we approximate the metric in (5.35) as

$$D(\tilde{\mathbf{s}}) \simeq \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_2^2 - 2\sigma^2(\mathbf{b}^T\tilde{\mathbf{s}}^2 + \mathbf{c}^T\tilde{\mathbf{s}} + \mathbf{d}^T\mathbf{1}). \quad (5.46)$$

Defining a vector  $\mathbf{x}^T \triangleq [\tilde{\mathbf{s}}^T \quad 1]$ , this cost function can be written as  $\mathbf{x}^T \tilde{\mathbf{Q}} \mathbf{x}$ , where

$$\tilde{\mathbf{Q}} = \begin{bmatrix} \tilde{\mathbf{H}}^T \tilde{\mathbf{H}} - 2\sigma^2 \text{Diag}(\mathbf{b}) & -\tilde{\mathbf{H}}^T \tilde{\mathbf{y}} - \sigma^2 \mathbf{c} \\ -\tilde{\mathbf{y}}^T \tilde{\mathbf{H}} - \sigma^2 \mathbf{c}^T & 0 \end{bmatrix}. \quad (5.47)$$

Since this cost function is a second-order polynomial in the elements of  $\mathbf{x}$ , the following fixed dimension semidefinite program

$$\min_{\mathbf{X}} \text{Trace}(\mathbf{X}\tilde{\mathbf{Q}}) \quad (5.48a)$$

$$\text{s.t. } L \leq \text{diag}(\mathbf{X}_{11}) \leq U, \quad (5.48b)$$

$$x_{22} = 1, \quad (5.48c)$$

$$\mathbf{X} \succeq 0. \quad (5.48d)$$

can be used in conjunction with a randomization procedure to extract an approximate solution to this optimization problem in an efficient way, using the interior point method developed in Section 5.1.3. (This SDP is analogous to that in (5.15), but with the matrix  $\tilde{\mathbf{Q}}$  in (5.47) replacing the matrix  $\tilde{\mathbf{Q}}$  in (5.9).)

### 5.3 Soft MIMO 16-QAM demodulation using List-SDR method

In this section we implement the principles of the List-SDR method proposed in Section 4.4 for soft demodulation of MIMO 16-QAM. That is, at each demodulation iteration we use the randomization step inherent in the extraction of an approximate solution to (5.34) from the solution to the semidefinite programs (5.45) or (5.48), to extract a list of candidate bit-vectors to approximate the soft information of each transmitted bit. Similar to the equations (4.6) and (4.7) in Chapter 4, the soft information for each

transmitted bit can be approximated as

$$\lambda_{D1,i} \simeq \log \frac{\sum_{\mathbf{b} \in \hat{\mathcal{L}}_{i,+1}} \exp(-D(\mathcal{M}(\mathbf{b}))/2\sigma^2)}{\sum_{\mathbf{b} \in \hat{\mathcal{L}}_{i,-1}} \exp(-D(\mathcal{M}(\mathbf{b}))/2\sigma^2)} \quad (5.49)$$

$$\simeq \frac{1}{2\sigma^2} \left( \min_{\mathbf{b} \in \hat{\mathcal{L}}_{i,-1}} D(\mathcal{M}(\mathbf{b})) - \min_{\mathbf{b} \in \hat{\mathcal{L}}_{i,+1}} D(\mathcal{M}(\mathbf{b})) \right), \quad (5.50)$$

where  $\hat{\mathcal{L}}$  is the enriched list. In each demodulation-decoding iteration the extrinsic information provided by the decoder is updated and hence we need to solve a new SDP to update the list of candidate bit-vectors. This new SDP is obtained by updating the matrix  $\tilde{\mathbf{Q}}$  in (5.43) or  $\mathbf{Q}$  in (5.47) in order to solve (5.45) or (5.48), respectively. Since a third-order approximation of soft information can be used in conjunction with (5.45) and this approximation is more accurate than the second order approximation of soft information used in conjunction with (5.48), the approach based on the increased dimension relaxation has a better performance. However, good performance can also be obtained using the second-order approach in (5.48), and that approach has the advantage that (5.48) can be solved using the computationally efficient interior point method presented in Section 5.1.2. Hence, the choice between (5.45) or (5.48) is made depending on the available computational resources and the required soft demodulation performance. After obtaining the matrix solution to (5.45) or (5.48) in each demodulation-decoding iteration, the randomization procedure is used to extract a preliminary list of candidate bit-vectors  $\hat{\mathcal{L}}'$ . The list  $\hat{\mathcal{L}}$  that is used in soft demodulation approximations (5.49) or (5.50), is then constructed by adding to  $\hat{\mathcal{L}}'$  all the single bit-flippings of its elements. As in Section 4.6, a list-free implementation is possible for the List-SDR schemes for soft MIMO 16-QAM demodulation described in this section. For convenience, this list-free implementation is

Table 5.2: List-free implementation of List-SDR algorithm for randomized soft MIMO 16-QAM demodulation

- 
- Data:  $\mathbf{X}_{\text{opt}}$  the solution to (5.48) (or  $\tilde{\mathbf{W}}_{\text{opt}}$  if using (5.45))
  - Parameters:  $M$ , the number of randomization iterations;  $K$ , the maximum size of the preliminary list.
  - Output:  $\lambda_{D1}$ , the vector of log likelihood-ratios
1. Initialize  $\mathbf{f}_{+1} = \{+\infty\}^{4N_t}$ ,  $\mathbf{f}_{-1} = \{+\infty\}^{4N_t}$ ,  $m = 0$ ,  $k = 0$ .
  2. Compute a (Cholesky) factor  $\mathbf{V}$  of  $\mathbf{X}_{\text{opt}}$  such that  $\mathbf{X}_{\text{opt}} = \mathbf{V}^T \mathbf{V}$ .
  3. Choose a random vector  $\mathbf{u}$  from the uniform distribution on the unit sphere.
  4. Construct  $\tilde{\mathbf{s}} = \mathcal{Q}\left(\frac{\mathbf{V}^T \mathbf{u}}{\sqrt{2N_t+1} \mathbf{u}}\right)$ ,  $\tilde{\mathbf{s}} = [\tilde{s}_1, \dots, \tilde{s}_{2N_t}]^T$  and increment  $m$ .
  5. Find the bit-vector  $\mathbf{b}$  corresponding to  $\tilde{\mathbf{s}}$ .
  6. Compute the signature of  $\mathbf{b}$ . If that value is not in the signature array, insert the value into the array and increment  $k$ , compute  $D(\tilde{\mathbf{s}}) = \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\tilde{\mathbf{s}}\|_2^2 - 2\sigma^2 \log p(\tilde{\mathbf{s}})$ , and for each  $i = 1, 2, \dots, 4N_t$ , if  $b_i = +1$  then set  $f_{+1}(b_i) = \min\{f_{+1}(b_i), D(\tilde{\mathbf{s}})\}$ , else set  $f_{-1}(b_i) = \min\{f_{-1}(b_i), D(\tilde{\mathbf{s}})\}$ .
  7. For each  $i = 1, 2, \dots, 4N_t$ , set  $\check{\mathbf{b}}^{(i)} = \mathbf{b}$  and then  $\check{b}_i^{(i)} = -b_i$ . Repeat Step 6 for  $\tilde{\mathbf{s}} = \mathcal{M}(\check{\mathbf{b}}^{(i)})$ .
  8. Increment  $m$ . If  $m < M$  and  $k < K$  return to 3. Otherwise, return  $\lambda_{D1} = (\mathbf{f}_{+1} - \mathbf{f}_{-1})/(2\sigma^2)$ .
- 

presented in Tab. 5.2.

The worst-case computational complexity of the above List-SDR demodulators can be computed by slightly modifying the approach used in Section 4.7. The computational complexity of List-SDR demodulator using the increased dimension SDR in (5.45) can be written as

$$O(TN_t^{6.5} \log \epsilon^{-1}) + O(TMN_t^2) + O(TMN_t^3), \quad (5.51)$$

where  $T$  is the number of demodulation-decoding iterations,  $O(N_t^{6.5} \log \epsilon^{-1})$  is the computational cost of solving each SDP (using a general purpose interior point method, e.g., [16]),  $O(TMN_t^2)$  is the computational cost of the

randomization iterations, and  $O(TMN_t^3)$  is the computational cost of computing the metrics. Using the fixed dimension SDR in (5.48) in conjunction with the efficient interior point solver described in Section 5.1.2, the computational cost of List-SDR method can be reduced to

$$O(TN_t^{3.5} \log \epsilon^{-1}) + O(TMN_t^2) + O(TMN_t^3). \quad (5.52)$$

As in Section 4.7

## 5.4 Soft MIMO 16-QAM demodulation using Single-SDR method

In this section we develop a Single-SDR demodulator for MIMO 16-QAM by extending the principles used for QPSK in Section 4.5 to the 16-QAM case. As in the Single-SDR demodulator for QPSK in Chapter 4, we will derive an analytic expression for the probability that each element of the candidate symbol-vector  $\tilde{\mathbf{s}}$  at the output of the randomization procedure (5.13) takes each of the symbol values in the set  $\mathcal{A} = \{-3, -1, +1, +3\}$ . For each symbol element  $\tilde{s}_i$ ,  $i = 1, \dots, 2N_t$ , of a candidate symbol-vector  $\tilde{\mathbf{s}}$ , the randomization procedure generates

$$\tilde{s}_i = \mathcal{Q} \left( \frac{\mathbf{v}_i^T \mathbf{u}}{\mathbf{v}_{2N_t+1}^T \mathbf{u}} \right), \quad (5.53)$$

where  $\mathcal{Q}(\cdot)$  is a quantizer to the values in  $\mathcal{A} = \{\pm 1, \pm 3\}$  and  $\mathbf{v}_i$  is the  $i$ th column of the Cholesky factor  $\mathbf{V}$  in the factorizations  $\mathbf{X}_{\text{opt}} = \mathbf{V}^T \mathbf{V}$  to the solutions of (5.48) or  $\tilde{\mathbf{W}}_{\text{opt}} = \mathbf{V}^T \mathbf{V}$  using (5.12) from the solution of (5.45). We define  $p_i^r(\tilde{s}_i)$ ,  $i = 1, \dots, 2N_t$ , to be the probability that  $\tilde{s}_i$  takes one of the

values in  $\mathcal{A} = \{-3, -1, +1, +3\}$  at the output of the randomization procedure. (We have added the superscript  $r$  to  $p_i^r(\tilde{s}_i)$  to distinguish between the randomization probability and the *a priori* probability  $p_i(\tilde{s}_i)$  provided by the decoder.) Since  $\mathbf{u}$  is uniformly distributed on the unit sphere, we can compute  $p_i^r(\tilde{s}_i)$ ,  $i = 1, \dots, 2N_t$  by evaluating the probability of  $\frac{\mathbf{v}_i^T \mathbf{u}}{\mathbf{v}_{2N_t+1}^T \mathbf{u}}$  being in the corresponding interval for the set  $\{(-\infty, -2], [-2, 0], [0, +2], [+2, +\infty)\}$ . Since  $\|\mathbf{v}_{2N_t+1}\|_2 = 1$  and  $\|\mathbf{u}\|_2 = 1$ , for each  $i \in \{1, \dots, 2N_t\}$  these probabilities depend on the norm of the  $i$ th column of  $\mathbf{V}$ ,  $\|\mathbf{v}_i\|_2$ , and the angle between  $\mathbf{v}_i$  and  $\mathbf{v}_{2N_t+1}$ ,  $\theta_i$ . For example, in order to obtain  $p_i^r(\tilde{s}_i = +3)$  we need to obtain the range of values for  $\gamma$ , the angle between  $\mathbf{u}$  and  $\mathbf{v}_{2N_t+1}$ , for which  $\frac{\mathbf{v}_i^T \mathbf{u}}{\mathbf{v}_{2N_t+1}^T \mathbf{u}} \geq 2$ . To do so, we observe that

$$\begin{aligned} \frac{\mathbf{v}_i^T \mathbf{u}}{\mathbf{v}_{2N_t+1}^T \mathbf{u}} &\geq 2 \\ \Leftrightarrow \|\mathbf{v}_i\|_2 \frac{\cos(\theta_i - \gamma)}{\cos(\gamma)} &\geq 2 \\ \Leftrightarrow \|\mathbf{v}_i\|_2 \cos(\theta_i - \gamma) &\geq 2 \cos(\gamma) \\ \Leftrightarrow \|\mathbf{v}_i\|_2 \cos(\theta_i) \cos(\gamma) + \|\mathbf{v}_i\|_2 \sin(\theta_i) \sin(\gamma) &\geq 2 \cos(\gamma) \\ \Leftrightarrow \tan(\gamma) &\geq \frac{2 - \|\mathbf{v}_i\|_2 \cos \theta_i}{\|\mathbf{v}_i\|_2 \sin \theta_i}, \end{aligned}$$

for  $\gamma \in [-\pi/2, +\pi/2]$ . In the same way, it can be shown that the angles  $\gamma$  for which  $\tilde{s}_i = -3$ , or equivalently  $\frac{\mathbf{v}_i^T \mathbf{u}}{\mathbf{v}_{2N_t+1}^T \mathbf{u}} \leq -2$ , are those for which

$$\tan(\gamma) \leq \frac{2 + \|\mathbf{v}_i\|_2 \cos \theta_i}{\|\mathbf{v}_i\|_2 \sin \theta_i}.$$

Similarly, the values for  $\gamma$  for which  $\frac{\mathbf{v}_i^T \mathbf{u}}{\mathbf{v}_{2N_i+1}^T \mathbf{u}} \geq 0$  are those for which

$$-\pi/2 \leq \theta_i - \gamma \leq \pi/2.$$

Therefore, if we define

$$\alpha_i = \tan^{-1} \left( \frac{2 - \|\mathbf{v}_i\|_2 \cos \theta_i}{\|\mathbf{v}_i\|_2 \sin \theta_i} \right), \quad (5.54a)$$

$$\beta_i = \tan^{-1} \left( \frac{2 + \|\mathbf{v}_i\|_2 \cos \theta_i}{\|\mathbf{v}_i\|_2 \sin \theta_i} \right), \quad (5.54b)$$

the probabilities that a given symbol will be generated by the conventional randomization procedure can be written as

$$p_i^r(\tilde{s}_i = -3) = \frac{\pi/2 - \beta_i}{\pi}, \quad (5.55a)$$

$$p_i^r(\tilde{s}_i = -1) = \frac{\theta_i - \pi/2 + \beta_i}{\pi}, \quad (5.55b)$$

$$p_i^r(\tilde{s}_i = +1) = \frac{\alpha_i + \pi/2 - \theta_i}{\pi}, \quad (5.55c)$$

$$p_i^r(\tilde{s}_i = +3) = \frac{\pi/2 - \alpha_i}{\pi}. \quad (5.55d)$$

Combining these expressions with the principles of Section 4.5 suggests that we can approximate the randomization procedure by assuming independence between each element of  $\tilde{\mathbf{s}}$ , and generating the candidate symbol-vectors using independent discrete random number generators with the probabilities computed in (5.55). As stated in Chapter 4, one advantage of this approach is that it avoids the computation of  $\mathbf{V}^T \mathbf{u}$  in each randomization iteration. More importantly, it provides the opportunity to separate the processing of the *a priori* information and the channel information in the second and subsequent demodulation-decoding iterations. This results in a

substantial reduction in the computational complexity, as we need to solve only one SDP per channel use.

The architecture of this list generation scheme is shown in Fig. 5.1, which is analogous to the Single-SDR list generator structure of Fig. 4.2 for MIMO QPSK. The difference between these two schemes is that the soft information from the channel and the extrinsic information from the decoder are represented using symbol probabilities instead of the LLR representation. Since the extrinsic information from the decoder and the channel information are independent by construction [13], if the randomized demodulator is to generate candidate bit-vectors via random symbol generations that reflect both of these sources of soft information, the symbol distribution probabilities of the  $i$ th symbol at the output of the randomized symbol-vector generator can be approximated as

$$p_{B,i}(\tilde{s}_i = \bar{s}) = \kappa_i p_i^r(\tilde{s}_i = \bar{s}) p_{A1,i}(\tilde{s}_i = \bar{s}), \quad (5.56)$$

where  $\bar{s} \in \mathcal{A} = \{\pm 1, \pm 3\}$ ,  $p_{A1,i}(\tilde{s}_i)$  are the symbol probabilities computed using their corresponding bit probabilities obtained from  $\lambda_{A1}$  (the vector of *a priori* information for each bit in LLR format), and  $\kappa_i$  are computed in such a way that  $\sum_{\bar{s} \in \mathcal{A}} p_{B,i}(\tilde{s}_i = \bar{s}) = 1$ . We should note that (5.56) corresponds to (4.17) in Chapter 4, which represents the Bernoulli distribution of randomization procedure for binary symbols using the LLR representation of the soft information. In that representation the LLRs from the decoder and the soft information from the channel are added together to compute the mean of the Bernoulli distribution for the randomization iterations in LLR format.

At each demodulation-decoding iteration, the Single-SDR demodulator generates a sequence of  $M$  symbol-vectors using scalar random number

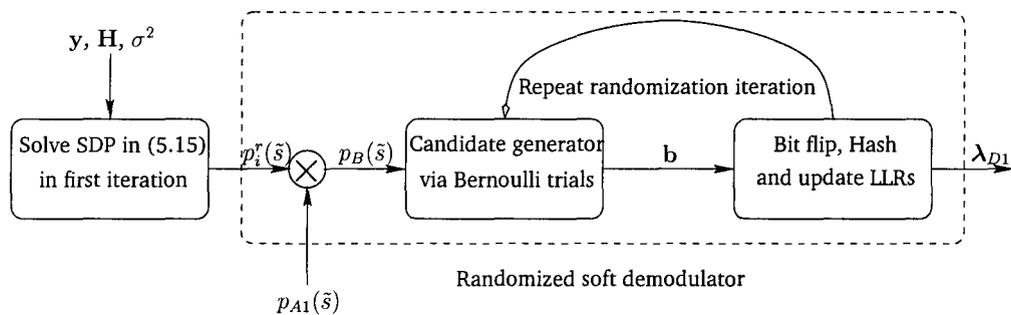


Figure 5.1: The proposed list-free Single-SDR demodulator for MIMO 16-QAM transmission.

generator with probability mass function as given in (5.56). The bit-vector representation of the uniquely generated symbol-vectors are then (notionally) stored in a preliminary list  $\hat{\mathcal{L}}'$ . As in Chapter 4, this list is then enriched by adding all the single bit-flippings of the bit-vectors in  $\hat{\mathcal{L}}'$  to form  $\hat{\mathcal{L}}$ . The soft information for each transmitted bit is then approximated using (5.49) or (5.50).

An implementational advantage of using the Single-SDR scheme is that instead of storing large lists of candidate bit-vectors generated in the first demodulation-decoding iterations, it requires the storage of vectors of randomization probability distributions computed in (5.55). These probability vectors in conjunction with the updated extrinsic information from the decoder are then used to generate a new list of bit-vectors in each demodulation-decoding iteration. The list-free implementation of this Single-SDR algorithm is presented in Tab. 5.3.

In the Single-SDR scheme, the SDP is solved only in the first demodulation-decoding iteration where no *a priori* information is available. Therefore, the objective is quadratic and hence we can use the fixed-dimension SDR approach in (5.15) directly. The computational cost per

Table 5.3: List-free implementation of Single-SDR algorithm for randomized soft MIMO 16-QAM demodulation

- 
- Data:  $p_i^r(\bar{s}_i)$ ,  $i = 1, \dots, 2N_t$  computed in (5.55),  $p_{A1,i}(\bar{s}_i)$ ,  $i = 1, \dots, 2N_t$  computed using the *a priori* information  $\lambda_{A1}$ .
  - $M$ , number of randomization iterations;  $K$ , the maximum size of the signature list.
  - Output:  $\lambda_{D1}$ , the vector of log likelihood-ratios.
1. Initialize  $\mathbf{f}_{+1} = \{+\infty\}^{4N_t}$ ,  $\mathbf{f}_{-1} = \{+\infty\}^{4N_t}$ ,  $m = 0$
  2. Compute  $p_{B,i}(\bar{s}_i)$ ,  $i = 1 \dots, 2N_t$  as in (5.56).
  3. Generate each  $\bar{s}_i$  independently according to the probability distributions with probabilities computed in step 2.
  4. Find the bit-vector  $\mathbf{b}$  corresponding to  $\bar{\mathbf{s}}$ .
  5. Compute the signature of  $\bar{\mathbf{s}}$ . If that value is not in the signature array, insert the value into the array and increment  $k$ , compute  $D(\bar{\mathbf{s}}) = \|\tilde{\mathbf{y}} - \tilde{\mathbf{H}}\bar{\mathbf{s}}\|_2^2 - 2\sigma^2 \log p(\bar{\mathbf{s}})$  and for each  $i = 1, 2, \dots, 4N_t$ , if  $b_i = +1$  then set  $f_{+1}(b_i) = \min\{f_{+1}(b_i), D(\bar{\mathbf{s}})\}$ , else set  $f_{-1}(b_i) = \min\{f_{-1}(b_i), D(\bar{\mathbf{s}})\}$ .
  6. For each  $i = 1, 2, \dots, 4N_t$ , set  $\check{\mathbf{b}}^{(i)} = \mathbf{b}$  and then  $\check{b}_i^{(i)} = -b_i$ . Repeat Step 5 for  $\bar{\mathbf{s}} = \mathcal{M}(\check{\mathbf{b}}^{(i)})$ .
  7. Increment  $m$ . If  $m < M$  and  $k < K$  return to 3. Otherwise, return  $\lambda_{D1} = (\mathbf{f}_{+1} - \mathbf{f}_{-1})/(2\sigma^2)$ .
- 

channel use of the resulting Single-SDR soft demodulation algorithm is, therefore,

$$O(N_t^{3.5} \log \epsilon^{-1}) + O(TMN_t) + O(TMN_t^3), \quad (5.57)$$

where  $T$  is the number of demodulation-decoding iterations. In this expression the first term represents the complexity of solving the SDP, the second term represents the complexity of the randomization step, and the third term represents the computational cost of computing the metrics.

As in Section 4.7 we have summarized the dominant computational cost of different SDR based schemes presented in this chapter in Tab. 5.4, and we also included the computational cost per channel use of the MMSE-SIC demodulator in [24], which is  $TO(T(4N_t)^4) \sim O(TN_t^4)$ .

Table 5.4: Dominant computational costs of various MIMO soft demodulators.

Demodulator	Dominant Computational Cost
List-SDR (increased dimension)	$O(TN_t^{6.5})$
List-SDR (fixed dimension)	$O(TN_t^{3.5})$
Single-SDR (fixed dimension)	$O(N_t^{3.5})$
MMSE-SIC	$O(TN_t^4)$

## 5.5 Simulation results

In this section we will explore the performance and complexity of the proposed List-SDR and Single-SDR demodulators. In particular we will compare the performance and complexity of these demodulators with those of some of the existing demodulators, such as the MMSE-SIC demodulator (e.g., [24, 115]), the list sphere decoder [12], and the LISS demodulator [86]. As in the previous chapters, we consider a MIMO BICM system that employs V-BLAST transmission over an i.i.d. Rayleigh fading channel, and will consider the transmission of 16-QAM symbols. We will consider a MIMO channel with  $N_t = 4$  transmit antennas and  $N_r = 4$  receive antennas. In order to prevent over or under estimation of the soft information we clip the log-likelihood ratios provided by the soft demodulators to the interval  $[-5, +5]$ . As in Chapter 4, in the first set of simulations we explore the performance of the proposed schemes using a turbo outer code, and in the second set we use a simple convolutional code as the outer code. In the following subsections we will consider two sets of simulations and in each set we will use a different type of outer code. In the first set we use a turbo code with a reasonably long block length and in the second set we use a weaker low latency convolutional code. In the Single-SDR scheme, and in the List-SDR scheme with the quadratic approximation of the prior information, the interior point algorithm developed in Section 5.1.3 was used to solve the SDPs. In the List-SDR scheme with the cubic approximation of the prior information, SeDuMi [128] was used to solve the SDPs.

### 5.5.1 Simulations with turbo outer code

As in Chapters 3 and 4 the outer code in our first set of simulations was chosen to be a turbo code with the (5, 7) recursive systematic convolutional code as the component codes and an (input) block length of 8192. The (different) interleavers in the turbo code and in the BICM transmitter are chosen from randomly generated candidates in each Monte-Carlo iteration. The conventional BCJR algorithm [50] is then used to decode the constituent convolutional codes of the turbo code and 8 turbo decoding iterations are performed before we pass the extrinsic information back to the demodulator.

First we compare the BER performance of the proposed schemes with those of the LISS demodulator [86], the list sphere decoder [12] and the MMSE-SIC demodulator (e.g., [24, 115]). For the LISS demodulator we considered a stack size of  $S = 500$  and a list size of  $L = 80$ . The list size of the sphere decoding scheme was set to  $L = 512$ . In this comparison we specify the Single-SDR and List-SDR demodulators with the SDP accuracy  $\epsilon = 10^{-1}$  and we performed  $M = 50$  randomization iterations before adding the bit-flippings of all the generated bit-vectors to the final list (i.e., we chose  $K = M$ ). Later in this section, we will justify this choice by examining the impact of these parameters on the complexity and performance of the Single-SDR and List-SDR demodulators

Fig. 5.2 compares the BER performance of the considered demodulators after 4 demodulation-decoding iterations. For reference, the SNR at which the mutual information for 16-QAM signals is 8 bits per channel use is approximately 6.9 dB. From Fig. 5.2 it is apparent that the performance of the List-SDR demodulator with cubic approximation of the soft information (the increased dimension relaxation scheme) is better than that of the

List-SDR scheme with the quadratic approximation of the soft information (the fixed dimension relaxation scheme). This figure also illustrates that the performance of both of these two methods is better than that of the list sphere decoder and the MMSE-SIC demodulator and is close to that of the LISS demodulator. In order to better illustrate the performance of the Single-SDR scheme, in addition to the case of  $M = 50$  randomization iterations we have added simulation results with  $M = 100$  and  $M = 200$  randomization iterations to Fig. 5.2. From this figure it is apparent that the BER performance of the Single-SDR demodulator with  $M = 50$  is better than that of the MMSE-SIC demodulator, and that by increasing the number of randomization iterations the Single-SDR demodulator can achieve performance close to that of the other schemes.

In order to show that the proposed demodulators achieve this performance at a low computational cost, in Fig. 5.3 we compare the average FLOPs of these schemes with those of some other existing schemes. As in Chapter 4, for the list sphere decoder we counted the FLOPs required to construct the list (which is performed once per channel use) and the FLOPs required to compute the metrics. We should note that in this graph we didn't provide the computational cost of the List-SDR scheme with cubic approximation of the soft information. The reason for this is that our FLOP measure involves explicit measurements of each operation, and the fact that this List-SDR scheme uses SeDuMi to solve the SDPs means that such an explicit count is not readily available. Nevertheless, it is apparent from Tab. 5.4 that the computational cost of the List-SDR scheme with the cubic approximation will be significantly higher than that of the List-SDR scheme with quadratic approximation of the soft information and that of the Single-SDR scheme,

since we use the developed low complexity interior point scheme in Section 5.1.3 to solve the SDPs of those schemes. For the LISS demodulator the list generation is repeated in each channel use and we included the required FLOPs for all these iterations. For the MMSE-SIC demodulator we counted, for each symbol, the FLOPs required to compute and subtract the mean of the interfering symbols, and those of the unbiased linear MMSE estimator of the resulted zero-mean signal. Fig. 5.3 quantifies the computational advantage of the List-SDR and Single-SDR demodulators over the list sphere decoder and the LISS demodulator. It also quantifies the computational advantage of the Single-SDR scheme over MMSE-SIC demodulator where the number of randomization iterations is less than  $M = 100$ . In particular, in the ‘waterfall’ region of the BER curves of Fig. 5.2, the average computational cost of the Single-SDR scheme with  $M = 50$  randomization iterations is about  $5/8$  of that of the MMSE-SIC demodulator and about  $1/8$  and  $1/10$  of the computational complexity of the list sphere decoder and the LISS demodulators, respectively. Furthermore, the computational cost distribution of the List-SDR and Single-SDR schemes is concentrated around the mean whereas the distributions of the list sphere decoder and the LISS demodulator have quite long tails. To illustrate that fact, we have plotted in Fig. 5.4 the empirical probability density of the computational cost per channel use of several demodulators at an SNR of 9.75 dB in a logarithmic scale. Fig. 5.4 also illustrates that most of the empirical computational cost distribution of the Single-SDR demodulator lies below the cost of the MMSE-SIC demodulator.

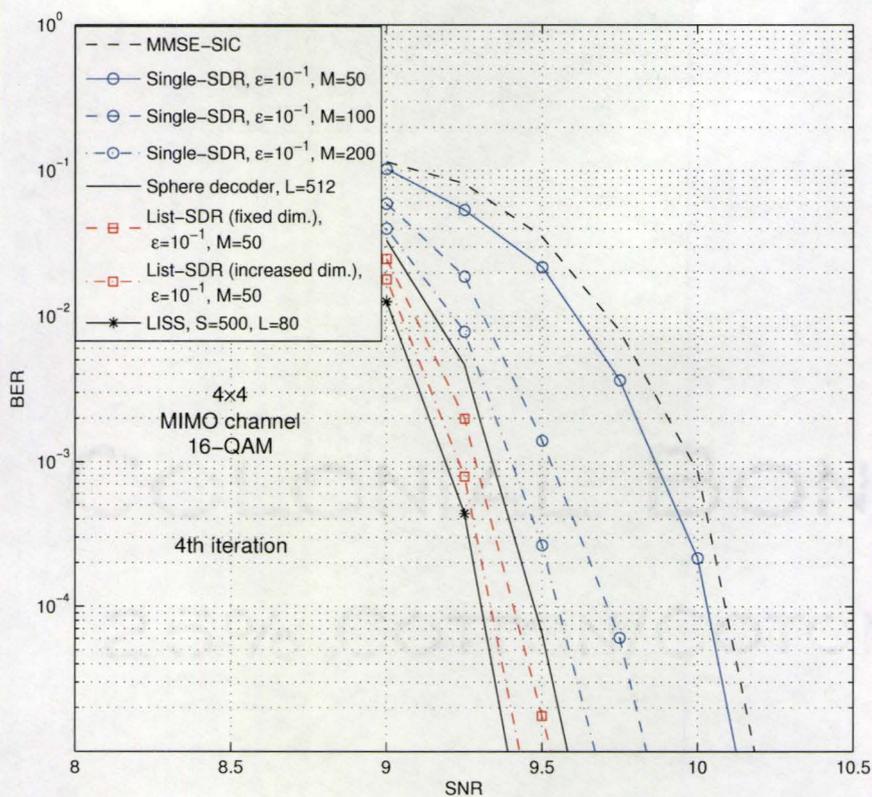


Figure 5.2: Comparison of the BER performance of various demodulators for the  $4 \times 4$  MIMO 16-QAM system with the turbo outer code.

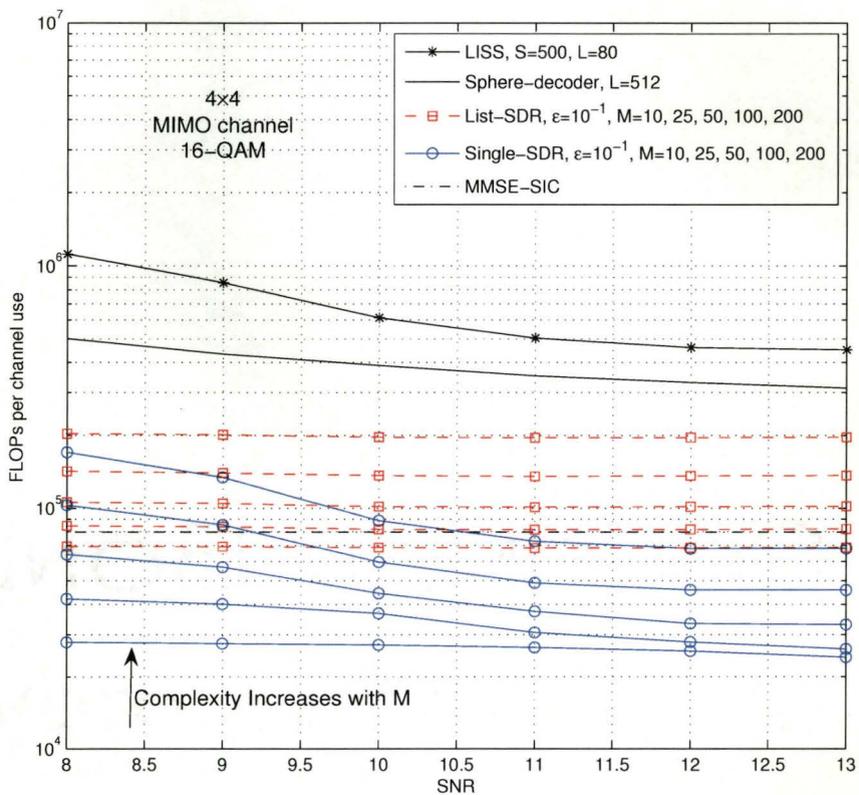


Figure 5.3: Comparison of the average computational cost per channel use of the proposed demodulators and that of list sphere decoder, the LISS and MMSE-SIC demodulators for the  $4 \times 4$  MIMO 16-QAM system with the turbo outer code. For the SDR based methods results for several values for  $M$ , the number of randomizations, are provided. Here, the List-SDR method is the one with the quadratic approximation of the prior information.

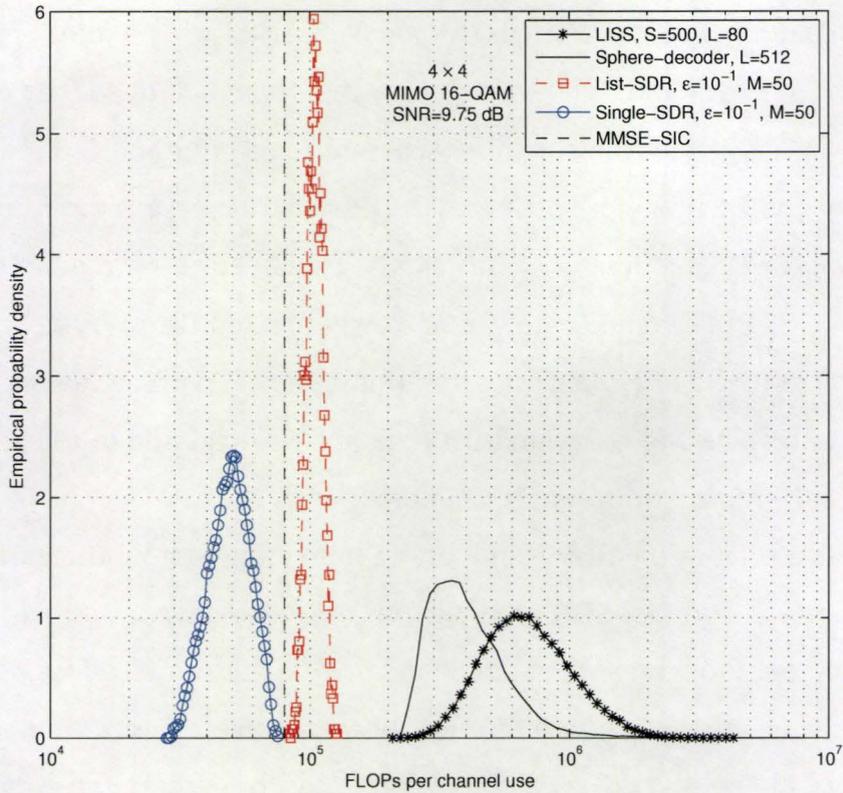


Figure 5.4: Empirical probability density of the number of FLOPs per channel use in the  $4 \times 4$  MIMO 16-QAM system with the turbo outer code at an SNR of 9.75 dB. Here, the List-SDR method is the one with the quadratic approximation of the prior information.

In order to justify the choices of an SDP accuracy of  $\epsilon = 10^{-1}$  and  $M = 50$  randomizations for the Single-SDR and List-SDR demodulators in Fig 5.2, we now evaluate the impact of these parameters on the BER performance and complexity of the receiver. In Figs. 5.5 and 5.6 we plot the average BER at the end of the 4th demodulation-decoding iteration for the List-SDR and Single-SDR schemes, respectively, with  $M = 50$  randomization iterations ( $M = K$ ) and different accuracy requirements for solving the SDP involved in list generation for these schemes. Figs. 5.5 and 5.6 show that increasing the accuracy of solving the SDPs will decrease the performance of the proposed demodulators. This can be explained using the fact that by solving the SDPs more accurately, a narrowly focused list of bit-vectors will be generated in the randomization iterations. Hence, with higher accuracies in solving the SDP, the generated list may not be rich enough to extract an accurate soft information from the channel.

Since solving the SDP is one of the dominant computational tasks in the Single-SDR and List-SDR schemes this will also have an impact on the total complexity of the soft demodulator. We explore the effect of different accuracies for the solution of the SDP on the average soft demodulator complexity in Fig. 5.7. As in Chapter 4, in order to measure the complexity, we explicitly counted the number of floating point operations (FLOPs) required for solving the SDPs, generating the list, and computing the metrics for all demodulation-decoding iterations per channel use. As demonstrated in Fig. 5.7, since the List-SDR scheme requires the solution of one SDP in each demodulation-decoding iteration, solving the SDP is the main computational task in this scheme and increasing the required accuracy of its solution will increase the average computational cost of the receiver over a wide range of SNR values.

The Single-SDR scheme requires the solution of only one SDP per channel use, and hence depending on the size of the generated list, solving the SDP might not be the main computational task in this scheme. However, since the reliability of the extrinsic information from the decoder is higher at high SNRs, the number of unique bit-vectors generated by the randomization procedure is smaller. Hence, at high SNR the SDP computational cost remains the dominant computational task and increasing its solution accuracy will increase the overall computational cost of the receiver. This is not the case at low SNRs. Since the size of the generated list is larger at low SNRs, computing the metrics becomes the dominant computational task. Increasing the accuracy of the SDP will reduce the list size, and hence will reduce the total receiver computational cost at low SNRs.

Based on the performance and complexity comparisons in Figs. 5.5, 5.6 and 5.7, it appears that an accuracy  $\epsilon = 10^{-1}$  for the solution of the SDPs seems to be a reasonable choice for the rest of our simulations. It provides better BER performance for the List-SDR and Single-SDR schemes compared to higher accuracies, and also the computational cost of the List-SDR scheme is lower with this choice, as is the cost of the Single-SDR scheme at higher SNRs. Furthermore, the increase in computational cost of the Single-SDR scheme at low SNRs for this choice of accuracy is negligible for most practical implementations.

We examine the effect of choosing different numbers of randomization iterations  $M$  on the performance of the List-SDR and Single-SDR demodulators in Figs. 5.8 and 5.9, respectively, where the SDPs are solved with an accuracy of  $\epsilon = 10^{-1}$ . These figures show the BER performance of the proposed schemes at the end of the 4th demodulation-decoding iteration and it is apparent that increasing  $M$  will improve the BER performance of

these schemes. Although each randomization iteration is a relatively cheap operation to implement (cf. Fig. 5.3), we chose  $M = 50$  in the simulations of Fig 5.2. This is a reasonable choice for the List-SDR scheme, because increasing the number of randomization iterations above  $M = 50$  has a very small effect on the performance of the soft demodulator. In the case of the Single-SDR scheme, increasing the number of randomization iterations above  $M = 50$  can significantly improve the BER performance. Therefore, depending on the computational cost that the implementations allow, one can get better performance by increasing the number of randomization iterations, even up to a value of  $M = 200$ .

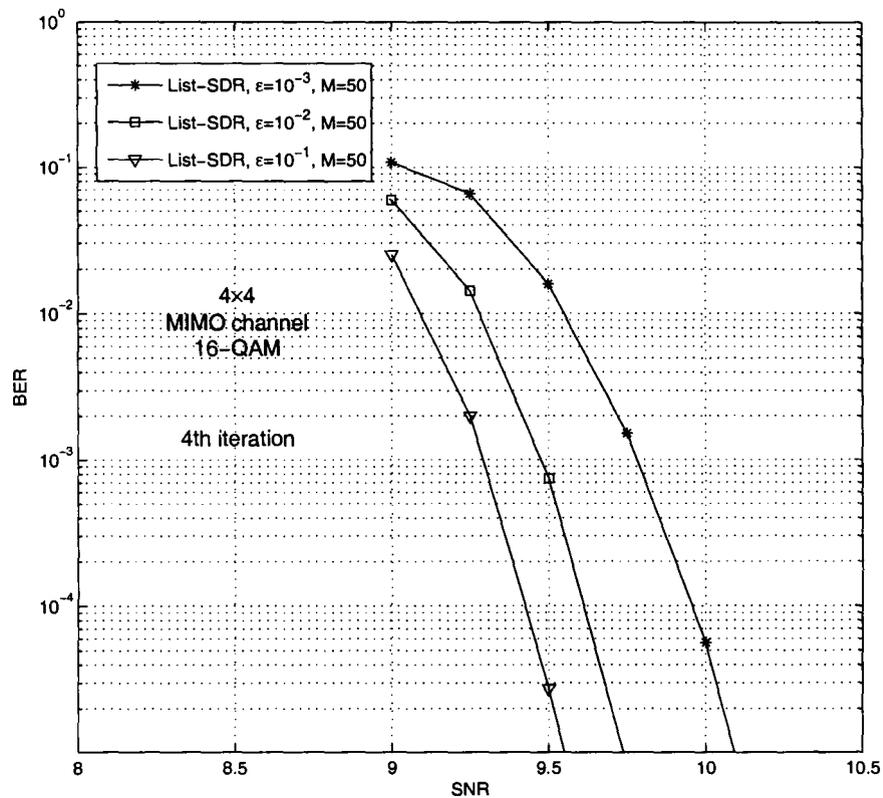


Figure 5.5: BER performance of a  $4 \times 4$  MIMO 16-QAM system with the turbo outer code that employs the List-SDR demodulator with the quadratic approximation of the prior information,  $M = 50$  randomizations, and different accuracies to which the SDP is solved.

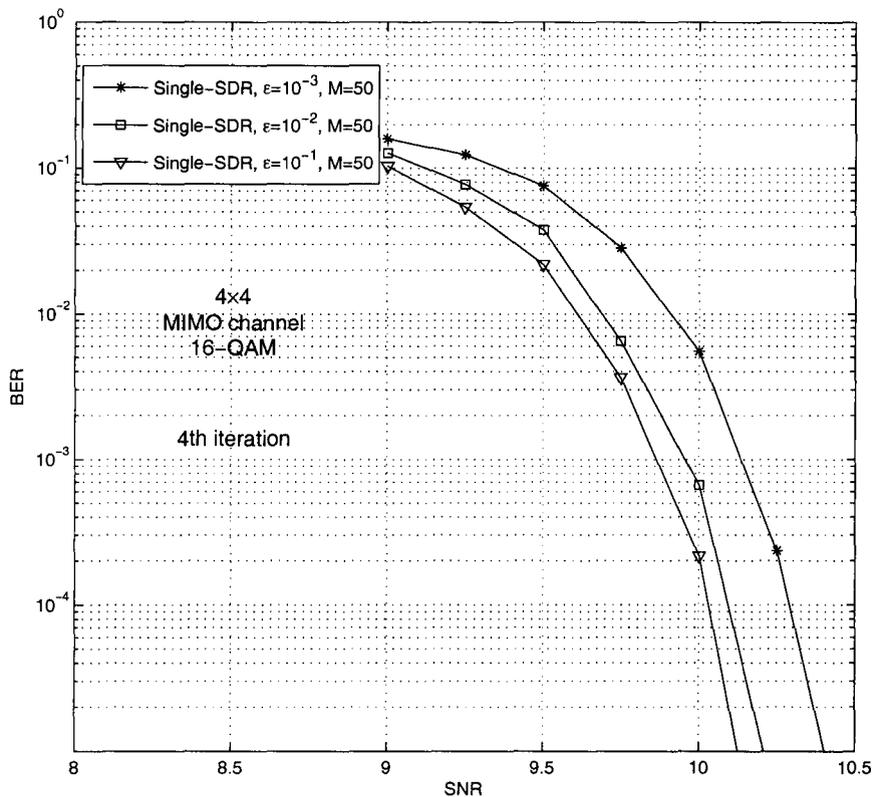


Figure 5.6: BER performance of a  $4 \times 4$  MIMO 16-QAM system with the turbo outer code that employs the Single-SDR demodulator with  $M = 50$  randomizations and different accuracies to which the SDP is solved.

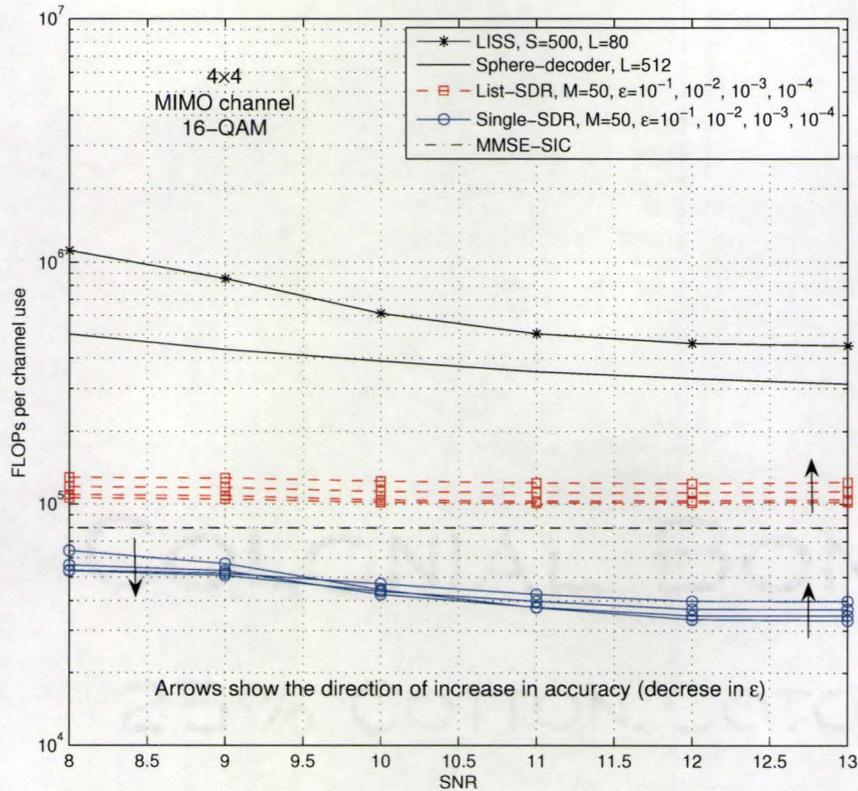


Figure 5.7: Comparison of the average computational cost per channel use of the proposed demodulators and that of list sphere decoder, the LISS and MMSE-SIC demodulators for the  $4 \times 4$  MIMO 16-QAM system with the turbo outer code. For the SDR based methods results for different accuracies of solving the SDPs,  $\epsilon$ , are provided. Here, the List-SDR method is the one with the quadratic approximation of the prior information.

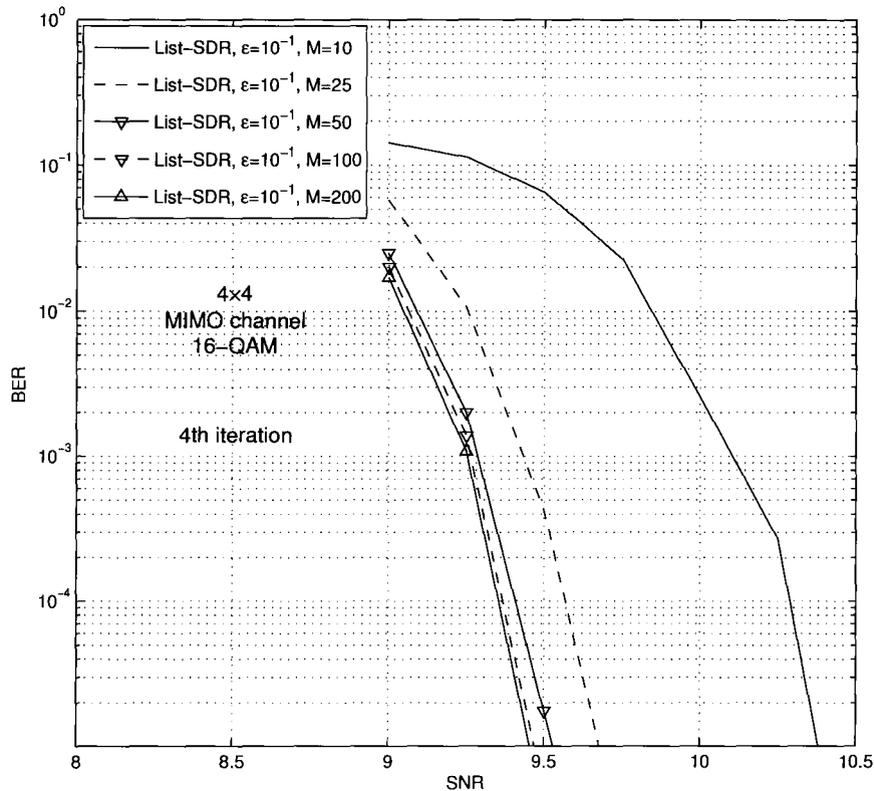


Figure 5.8: BER performance of a  $4 \times 4$  MIMO 16-QAM system with the turbo outer code that employs the List-SDR demodulator with the quadratic approximation of the prior information, an SDP solution accuracy of  $\epsilon = 10^{-1}$ , and different numbers of randomizations,  $M$ .

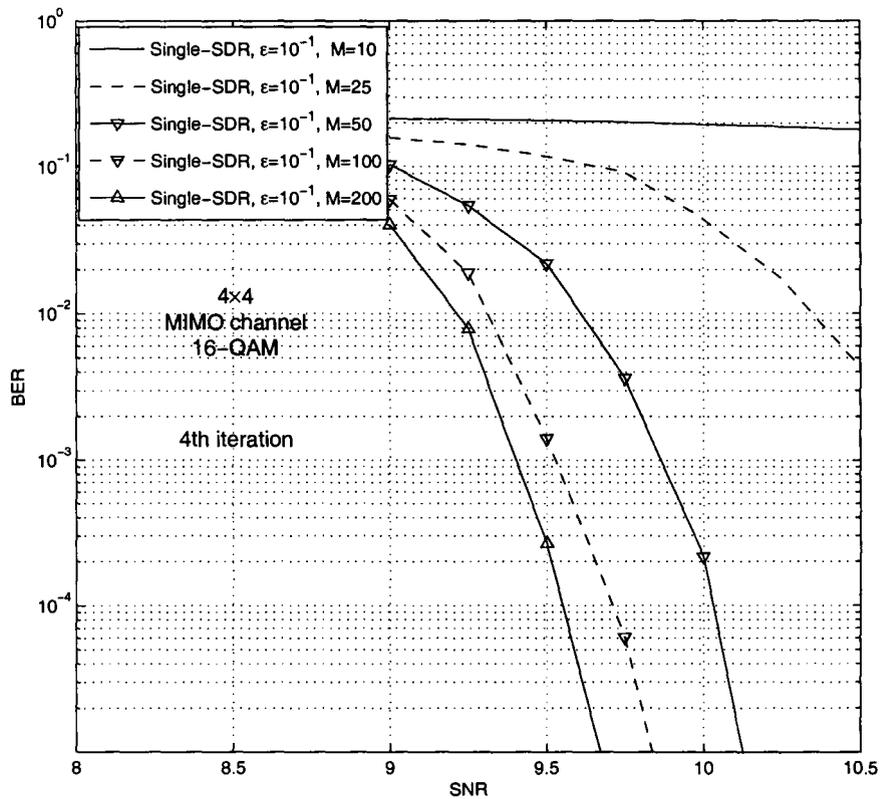


Figure 5.9: BER performance of a  $4 \times 4$  MIMO 16-QAM system with the turbo outer code that employs the Single-SDR demodulator with an SDP solution accuracy of  $\epsilon = 10^{-1}$ , and different numbers of randomizations,  $M$ .

## 5.5.2 Simulations with outer convolutional code

Now we consider a rather short convolutional code which incurs a significantly lower latency than the turbo code used for the previous simulations. Following [24], we use a rate  $1/2$   $(23, 35)$  recursive systematic convolutional code with block length 256 and a BCJR decoder at the receiver. This outer code is rather weak when compared to the turbo code that was used in the previous simulations.

As in the previous simulations we will solve the SDPs to an accuracy of  $\epsilon = 10^{-1}$ . Fig. 5.10 compares the performance of the proposed schemes with that of other soft demodulation schemes such as the LISS demodulator, the list sphere decoder and the MMSE-SIC demodulator, for the case of  $M = 100$  randomization iterations ( $K = M$ ) and we will justify this choice of  $M$  later. From this figure it is apparent that the BER performance of the Single-SDR scheme is better than that of the MMSE-SIC demodulator and close to that of the LISS and list sphere decoder. It is also apparent that the performance of the List-SDR algorithm is very close to the performance of the LISS and list sphere decoder in all demodulation-decoding iterations.

In Figs. 5.11 and 5.12 we examined the effect of the number of randomization iterations on the performance of the proposed schemes. Fig. 5.11 presents this performance comparison for the List-SDR scheme and Fig. 5.12 presents it for the Single-SDR scheme. (Here, as earlier, we have chosen  $K = M$ ). These simulations suggest that increasing  $M$  above 50 has a small effect on the performance of the List-SDR and Single-SDR schemes.

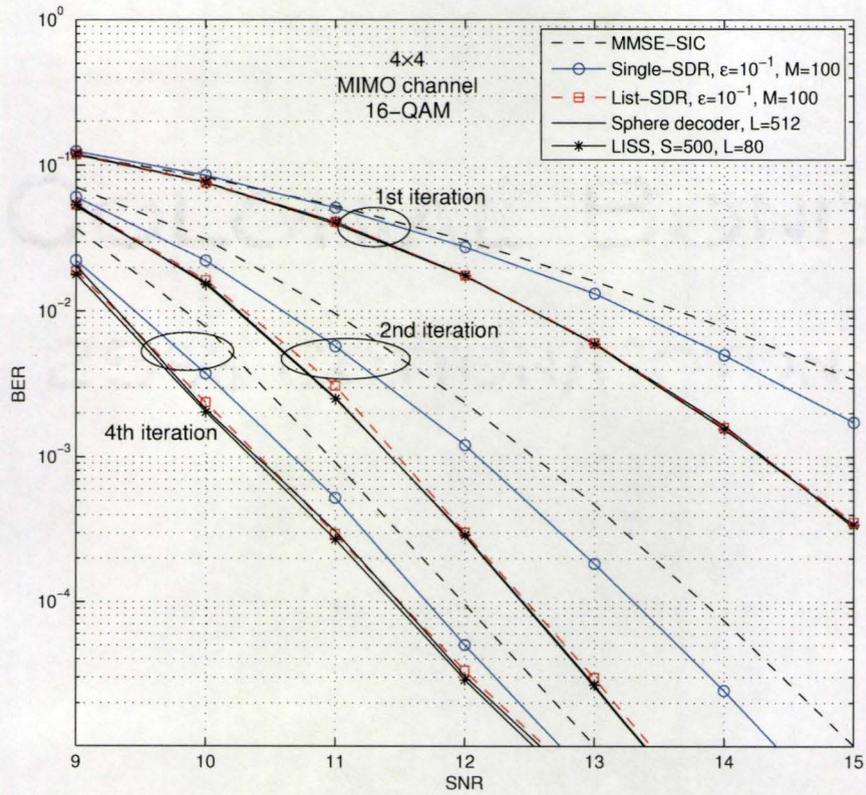


Figure 5.10: Comparison of the BER performance of various demodulators for the  $4 \times 4$  MIMO 16-QAM system with the convolutional outer code. Here, the List-SDR method is the one with the quadratic approximation of the prior information.

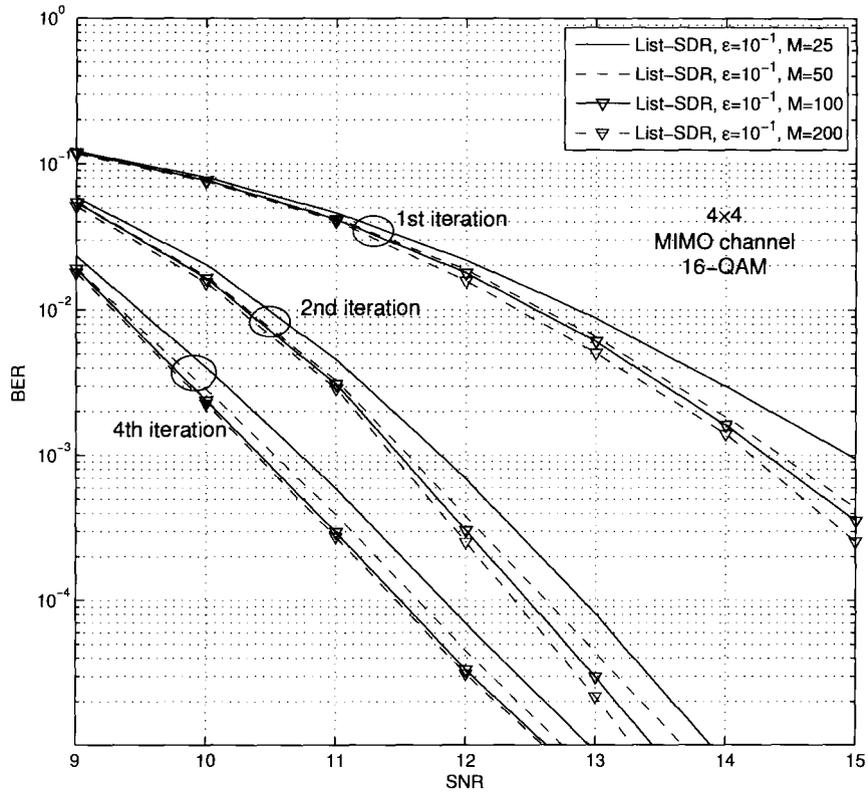


Figure 5.11: BER performance of a  $4 \times 4$  MIMO 16-QAM system with the convolutional outer code that employs the List-SDR demodulator with the quadratic approximation of the prior information, an SDP solution accuracy of  $\epsilon = 10^{-1}$ , and different numbers of randomizations,  $M$ .

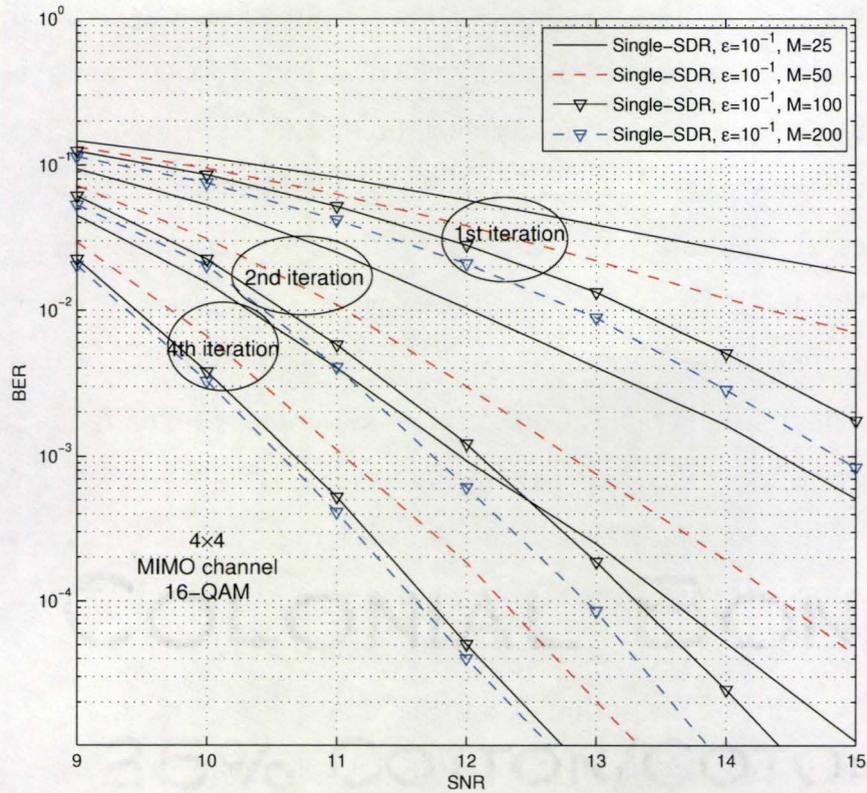


Figure 5.12: BER performance of a  $4 \times 4$  MIMO 16-QAM system with the convolutional outer code that employs the Single-SDR demodulator with an SDP solution accuracy of  $\epsilon = 10^{-1}$ , and different numbers of randomizations,  $M$ .

## 5.6 Conclusion

In this chapter we presented the extension of the two computationally-efficient soft demodulation schemes proposed in the previous chapter, namely the List-SDR and Single-SDR schemes, to MIMO systems with 16-QAM signaling. In contrast to the existing soft demodulators based on tree search principles, the proposed schemes have a (low-order) polynomial complexity in the number of bits to be transmitted. The extension to 16-QAM of the schemes in Chapter 4 was based on using semidefinite relaxation (SDR) techniques to approximate the solution to the hard demodulation problem for a MIMO 16-QAM system. There are several SDR-base hard demodulation algorithms in the literature. In this chapter we proved that two of these existing schemes that result in substantially different semidefinite programs (of different dimensions) are actually equivalent. We then developed a low complexity interior point algorithm for extracting the solution to the SDP corresponding to the existing SDR technique with the lower dimension.

The List-SDR schemes that was proposed for QPSK modulation in Chapter 4 requires the solution of one SDP in each demodulation-decoding iteration, and the objective of that SDP had a polynomial expression. However, in the case of a system with 16-QAM modulation, the logarithm of the *a posteriori* probability (Log-APP) has a non-polynomial expression and hence the List-SDR technique is not directly applicable. In order to develop a List-SDR scheme for 16-QAM systems we employed interpolation and curve fitting techniques to obtain cubic and quadratic approximations of the Log-APP, respectively. The extension of the Single-SDR scheme to the case of 16-QAM did not require these approximations, but did require the development of expressions for the probability mass function for each symbol under

the conventional randomization procedure.

Simulation results illustrated that the computational advantage of the proposed demodulators is obtained without incurring a substantial degradation in their performance. In particular, the BER performance of the Single-SDR scheme is better than that of the MMSE-SIC demodulator and close to that of the list sphere decoder and LISS demodulator, and this performance is achieved at a lower computational cost. The List-SDR scheme provides better performance than the Single-SDR scheme, and, in the case of a turbo outer code, provided better performance than the list sphere decoder. The List-SDR scheme achieves this performance at a computational cost that is lower than that of the list sphere decoder and the LISS demodulator and is close to that of the MMSE-SIC demodulator.

# Chapter 6

## Summary and future work

### 6.1 Summary

This thesis considered the design of efficient soft MIMO demodulators for communication systems that deploy iterative MIMO demodulation and decoding schemes. Since the main computational bottleneck in these schemes is the MIMO soft demodulator, the proposed demodulators have the potential to have a significant impact in practical applications. Most of the existing soft demodulators are based on the max-log approximation of the soft information, and those demodulators can be divided into two classes. One class is based on (approximately) solving several hard demodulation problems using reduced complexity schemes like tree search methods or semidefinite relaxation (SDR) techniques. The other class is based on list soft demodulation techniques, in which a list of candidate bit-vectors is efficiently selected and the max-log approximation is performed over the generated list. This thesis was focused on design of several computationally-efficient list-based soft MIMO demodulators. The proposed demodulators can be classified into two

classes, one based on tree search ideas, and in particular the stack algorithm, and the other based on semidefinite relaxation techniques.

The scheme based on the tree search interpretation was called the multi-stack algorithm, because it assigns a unique stack to each level of tree structure. The exposed nodes of each level of the tree are stored in their corresponding stacks and when each stack is examined the best node in the stack (i.e., the one with the smallest metric) is chosen for expansion. By assigning appropriate priorities to the stack from which the search re-starts once it finds a leaf node or an empty stack, the proposed algorithm is able to extract a collection of good bit-vectors in the early stages of the algorithm. Hence, if the algorithm is terminated early for reasons of computational cost, it is likely able to maintain good performance. In particular, it was shown that the performance-complexity trade-offs achieved by instances of the proposed algorithm can dominate those of several existing list based algorithms, especially in the low complexity region.

In contrast to tree search algorithms for hard demodulation, such as the stack algorithm and the sphere decoding algorithm, which have computational costs that are exponential in the number of bits transmitted per channel use, the computational cost of semidefinite relaxation techniques for hard demodulation is a (low-order) polynomial of that number of bits. The existing soft demodulation scheme based on the SDR techniques (the Multi-SDR scheme) approximates the soft information by solving several hard demodulation problems per demodulation-decoding iteration in each channel use. However, before the research reported in the second part of the thesis, there was no SDR-based technique for list demodulation. In the second part of the thesis, it was first shown that the randomization procedure that is inherent in the SDR technique can be exploited to construct a list of bit-vectors

for a list-based demodulator. This List-SDR demodulator has a substantial computational advantage over the existing SDR-based hard demodulation approach to soft demodulation, in that it only requires one SDP to be solved per demodulation-decoding iteration in each channel use. By approximating the randomization procedure by independent Bernoulli trials, and by exploiting the presumption of independence between the likelihoods obtained from the channel and the prior information from the previous decoding iteration, the single-SDR demodulator was developed. The computational cost of this demodulator is significantly smaller than that of the List-SDR demodulator, as it only requires the solution of one SDP per channel use (the SDP that arises in the first demodulation-decoding iteration).

Since the structure of the corresponding SDP changes depending on the signal constellation that is transmitted, QPSK signaling was first considered in the development of the proposed SDR-based list demodulators. Then, in Chapter 5, the development of the List-SDR and Single-SDR schemes for 16-QAM signaling was considered. In this development, it was shown that two existing hard demodulation schemes for MIMO 16-QAM (one with a higher dimensionality and the other with a conformal dimensionality) are equivalent. Furthermore, an efficient interior point algorithm for solving the lower dimension SDP was developed. Since the metric function corresponding to the MIMO hard (MAP) demodulation for this system does not have a polynomial structure, the List-SDR scheme was extended for use with 16-QAM signaling by approximating the *a priori* information part of the metric function using second order and third order polynomials. The extension of the Single-SDR scheme to 16-QAM signaling was performed by approximating the randomization procedure using independent random

symbol generators with probability mass functions obtained from the solution of the SDP that is solved in the first demodulation-decoding iteration. These probabilities are then updated in the subsequent iterations using the available updated *a priori* information from the decoder.

Simulation results confirm that the proposed Single-SDR and List-SDR schemes provide performance-complexity trade-offs that are superior to those of some existing soft demodulation algorithms. In particular, the Single-SDR algorithm is a favorable choice for practical implementations since it requires fewer operations than one of the widely known low complexity soft demodulation algorithms, the minimum mean square error with soft interference cancellation (MMSE-SIC) scheme, and yet it provides better performance than that scheme.

## 6.2 Directions for future work

The studies conducted in this thesis revealed the potential of multi-stack algorithm and semidefinite relaxation techniques in the design of low-complexity high-performance soft demodulation schemes, and hence they have opened several interesting directions for future work. In particular, one may consider the following directions and questions as possible future research.

- The List-SDR and Single-SDR schemes can be easily extended to other higher dimension  $M$ -ary QAM constellations using the guidelines provided in Chapter 5. One possible direction for future work is to extend the List-SDR and Single-SDR schemes for use in those systems, and to evaluate the performance and computational cost of the resulting

schemes. In particular, the extension of the fixed dimension relaxation approach to 16-QAM signaling to higher order ( $M$ -ary QAM) constellations simply involves changing the upper and lower bounds of the linear inequality constraints in the semidefinite program, and hence the extension should be quite straightforward in the case of the Single-SDR approach. However, the List-SDR approach will require the development of an appropriate approximation of the prior information provided by the previous iteration of the decoder.

- Some constellations like  $M$ -ary PSK (with  $M > 4$ ) do not fit in the general SDR framework for  $M$ -ary QAM signaling that was provided in Chapters 4 and 5. However, there are some available SDR techniques in the literature for hard demodulation using these signaling schemes (such as hard demodulation of  $M$ -ary PSK multiuser systems in [108]). It would be interesting to develop the List-SDR and Single-SDR approaches to be used in these systems, using (or developing) a low-complexity interior point method to solve the corresponding SDP and also developing the corresponding approximate randomization techniques for a Single-SDR approach to soft demodulation.
- Geomans and Williamson [110] and Nesterov [111] provided bounds to measure the performance of the conventional randomization procedure for extracting the vector solution of the original binary quadratic program from the matrix solution of the corresponding semidefinite relaxation. In the communications context, those results are restricted to the case of BPSK and QPSK constellations. It would be interesting to see whether a performance bound can be derived for applications of SDR techniques to higher dimensional constellations, such as 16-QAM.

- In the Single-SDR approach, the conventional randomization procedure was approximated by a set of independent Bernoulli trials. It would be interesting to see whether a performance bound can be derived for this approximate randomization procedure.
- In this thesis it was shown that two existing semidefinite relaxation techniques to approximate the solution to a hard MIMO 16-QAM demodulation problem are equivalent. These schemes are the increased dimension relaxation technique of [25] and the fixed dimension relaxation technique of [26]. There are several other relaxation techniques in the literature (e.g., [109,125–127]), and some of them claim to have a smaller feasible region due to the constraints selected, and hence claim to have a better performance. They prove their claims by simply referring to the extra constraints added to the corresponding SDR. It would be interesting to take a closer look at the feasibility regions of these problems, using the ideas that were used to prove the equality of the techniques in [25] and [26] in Chapter 5, and to attempt to quantify the extent to which the feasibility region is reduced.
- The structure of the received signal in a MIMO system,  $\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}$ , is quite similar to that of (synchronous) multiuser CDMA systems and that of (imperfectly synchronized) OFDM systems, and hence the proposed demodulators can be used in such schemes. However, in such cases, the matrix  $\mathbf{H}$  has some structure that can be exploited to reduce the computational cost of demodulation. The extent to which SDR-based techniques can also exploit that structure is an interesting topic for future work.

- In order to enrich the final list to be used in list-based soft demodulation schemes the bit-vectors with a Hamming distance 1 of the originally generated bit-vectors were added to the list in both the SDR based schemes and the multi-stack schemes. That approach seems to be well matched to the case of Gray mapped constellations, but it would be interesting to investigate its performance under other mapping strategies, and to determine whether a simple bit-flipping provides sufficient enrichment of the list.
- The simple V-BLAST transmission scheme was used throughout the simulations of this thesis to map the encoded data bits in space and time. As it was noted in Chapter 2, the proposed soft demodulators can also be used in conjunction with any general linear dispersion (LD) code [67]. As the LD code family includes codes that have desirable diversity properties, the simulations ought to be extended to measure the performance of the proposed soft demodulation schemes for LD coded transmission over spatially and temporally correlated channels.

# Appendix A

## MIMO-IDD using MMSE-SIC

MMSE demodulation with soft interference cancellation (MMSE-SIC, e.g., [24, 76, 115]) is a popular low-complexity soft demodulator for MIMO systems. The MMSE-SIC first forms the unbiased conditional MMSE estimate of each transmitted symbol, where the conditioning is on the soft information of the other symbols that is provided by the decoder. The LLR of each transmitted bit is then obtained by approximating the residual interference of each symbol as a Gaussian random variable and computing the soft information as if the channel was a single-input single-output AWGN channel.

In order to review this scheme we consider the following channel model

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{v}, \quad (\text{A.1})$$

where, as in Chapters 3, 4 and 5,  $\mathbf{y}$  is the vector of received signal,  $\mathbf{H}$  is the  $N_r \times N_t$  matrix of channel coefficients known to the receiver,  $\mathbf{s}$  is the vector of transmitted signal with power  $\sigma_s^2$  per dimension and  $\mathbf{v}$  is the vector of additive white circular Gaussian noise samples with variance  $\sigma_n^2$  per dimension. For convenience, we will use the real-valued equivalent representation

of this channel model (e.g., [85]),

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{s}} + \tilde{\mathbf{v}}, \quad (\text{A.2})$$

where  $\tilde{\mathbf{y}}$ ,  $\tilde{\mathbf{s}}$  and  $\tilde{\mathbf{v}}$  are the concatenations of the real and imaginary parts of  $\mathbf{y}$ ,  $\mathbf{s}$  and  $\mathbf{v}$ , respectively and  $\tilde{\mathbf{H}}$  is the real-valued decomposition of the channel matrix  $\mathbf{H}$  and has a size of  $2N_t \times 2N_r$ . In each channel use  $2N_t M_c$  bits are transmitted using this channel model and the  $2^{M_c}$  possible (real-valued) symbols that can be transmitted from the  $k$ 'th transmit antenna can be obtained using the mapping  $\hat{s}_k^\ell = \mathcal{M}(\mathbf{b}_k^\ell)$ ,  $\ell = 1, \dots, 2^{M_c}$ ,  $k = 1, \dots, 2N_t$ , where  $\mathbf{b}_k^\ell \triangleq [b_{(k-1)M_c+1}^\ell, \dots, b_{kM_c}^\ell]^T$  is the vector of transmitted bits from the  $k$ 'th antenna. In the second and subsequent iterations an estimate of the *a priori* probability of each transmitted bit  $p(b_i = +1)$  and  $p(b_i = -1)$ ,  $i = 1, \dots, 2N_t M_c$ , is available from the decoder. Using the assumption of independence between the transmitted bits, the *a priori* probability of transmitting each symbol can be approximated by  $p(\tilde{s}_k = \hat{s}_k^\ell) \approx \prod_{i=(k-1)M_c+1}^{kM_c} p(b_i = b_i^\ell)$ . Hence the *a priori* expected value of each transmitted symbol from each transmit antenna can be written as

$$\bar{s}_k = \sum_{\ell=1}^{2^{M_c}} \hat{s}_k^\ell p(\tilde{s}_k = \hat{s}_k^\ell), \quad k = 1, \dots, 2N_t. \quad (\text{A.3})$$

Let us define  $\tilde{\mathbf{H}}_k = [\tilde{\mathbf{h}}_1, \dots, \tilde{\mathbf{h}}_{k-1}, \tilde{\mathbf{h}}_{k+1}, \dots, \tilde{\mathbf{h}}_{2N_t}]$ , where  $\tilde{\mathbf{h}}_i$  is the  $i$ 'th column of the channel matrix  $\tilde{\mathbf{H}}$ , and  $\bar{\mathbf{s}}_k = [\bar{s}_1, \dots, \bar{s}_{k-1}, \bar{s}_{k+1}, \dots, \bar{s}_{2N_t}]^T$ . For each value of  $k$ ,  $1 \leq k \leq 2N_t$ , the MMSE-SIC demodulator performs soft interference cancellation with respect to  $\bar{\mathbf{s}}_k$  on the received signal. That is, it computes the  $2N_t$  vectors

$$\tilde{\mathbf{y}}_k = \tilde{\mathbf{y}} - \tilde{\mathbf{H}}_k \bar{\mathbf{s}}_k. \quad (\text{A.4})$$

The MMSE-SIC then applies a linear MMSE filter to each of these signal vectors. The filtered output can be written as

$$z_k = \mathbf{w}_k^T \tilde{\mathbf{y}}_k, \quad (\text{A.5})$$

where, for each value of  $k$ , the weight vector of the linear MMSE filter can be computed as [24, 115]

$$\mathbf{w}_k = [\tilde{\mathbf{H}}\mathbf{C}_k\tilde{\mathbf{H}}^T + \sigma_n^2\mathbf{I}_{2N_t}]^{-1}\tilde{\mathbf{h}}_k\sigma_s^2, \quad (\text{A.6})$$

where  $\mathbf{C}_k = \text{Diag}([\sigma_{s_1}^2, \dots, \sigma_{s_{k-1}}^2, \sigma_s^2, \sigma_{s_{k+1}}^2, \dots, \sigma_{s_{2N_t}}^2])$  and  $\sigma_{s_i}^2 = \text{E}\{(\tilde{s}_i - \bar{s}_i)^2\}$ . The linear MMSE filtered output is then approximated by a Gaussian distributed random variable [24, 115]. That is,  $z_k$  can be approximated using the following AWGN channel model

$$z_k = \mu_k \tilde{s}_k + \eta_k, \quad (\text{A.7})$$

where

$$\mu_k = \mathbf{w}_k^T \tilde{\mathbf{h}}_k, \quad (\text{A.8})$$

and  $\eta_k$  is an additive Gaussian noise with variance

$$\sigma_{\eta_k}^2 = \mathbf{w}_k^T [\tilde{\mathbf{H}}\mathbf{C}_k\tilde{\mathbf{H}}^T - \sigma_s^2\tilde{\mathbf{h}}_k\tilde{\mathbf{h}}_k^T + \sigma_n^2\mathbf{I}_{2N_t}] \mathbf{w}_k. \quad (\text{A.9})$$

Using this approximate model, the soft information for each transmitted bit at the output of the MMSE-SIC demodulator can be approximated as [24,

115]

$$\lambda_{D_k}^i = \log \frac{\sum_{\tilde{s}_k^\ell \in S_{b_i=+1}} \exp(\zeta(\tilde{s}_k^\ell))}{\sum_{\tilde{s}_k^\ell \in S_{b_i=-1}} \exp(\zeta(\tilde{s}_k^\ell))}, \quad k = 1, \dots, 2N_t, \quad i = 1, \dots, M_c. \quad (\text{A.10})$$

where

$$\zeta(\tilde{s}_k^\ell) = -\frac{(z_k - \mu_k \tilde{s}_k^\ell)^2}{2\sigma_{\eta_k}^2} + \frac{1}{2} \boldsymbol{\lambda}_{A_k}^T \mathbf{b}_k^\ell, \quad (\text{A.11})$$

$S_{b_i=\pm 1}$  is the list of all symbols  $\hat{s}_k^\ell = \mathcal{M}(\mathbf{b}_k^\ell)$ ,  $\ell = 1, \dots, 2^{M_c}$  with the  $i$ th bit of  $\mathbf{b}_k^\ell$  fixed to  $\pm 1$ , and  $\boldsymbol{\lambda}_{A_k}$  is the (extrinsic) soft information in LLR form, corresponding to the bit-vector  $\mathbf{b}_k^\ell$  that indexes the symbol  $s_k^\ell$  transmitted from the  $k$ th transmit antenna. The extrinsic component of the extracted soft information is then fed back to the decoder for another round of iteration between the demodulator and the decoder.

# Bibliography

- [1] S. N. Diggavi, N. Al-Dhahir, A. Stamoulis, and A. R. Calderbank, "Great expectations: The value of spatial diversity in wireless networks," *Proc. IEEE*, vol. 92, no. 2, pp. 219–270, Feb. 2004.
- [2] E. Biglieri, J. Proakis, and S. Shamai, "Fading channels: Information-theoretic and communications aspects," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2619–2692, Oct. 1998.
- [3] D. Tse and P. Viswanath, *Fundamentals of wireless communication*, Cambridge University Press, New York, NY, 2005.
- [4] A. Goldsmith, S. A. Jafar, N. Jindal, and S. Vishwanath, "Capacity limits of MIMO channels," *IEEE J. Select. Areas Commun.*, vol. 21, no. 5, pp. 684–702, June 2003.
- [5] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Lab. Sys. Tech. J.*, vol. 1, no. 2, pp. 41–59, Autumn 1996.
- [6] I. E. Telatar, "Capacity of multiple antenna Gaussian channels," *Eur. Trans. Telecom.*, vol. 10, pp. 585–595, Nov. 1999.

- [7] L. Zheng and D. N. C. Tse, "Diversity and multiplexing: A fundamental tradeoff in multiple-antenna channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 5, pp. 1073–1096, May 2003.
- [8] K. Azarian and H. El Gamal, "The throughput-reliability tradeoff in block-fading MIMO channels," *IEEE Trans. Inform. Theory*, vol. 53, no. 2, pp. 488–501, Feb. 2007.
- [9] S. Barbarossa, *Multiantenna wireless communication systems*, Artech House, Norwood, MA, 2005.
- [10] A. Goldsmith, *Wireless communications*, Cambridge University Press, New York, NY, 2005.
- [11] G. Caire, G. Taricco, and E. Biglieri, "Bit-interleaved coded modulation," *IEEE Trans. Inform. Theory*, vol. 44, no. 3, pp. 927–946, May 1998.
- [12] 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.
- [13] 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.
- [14] J. Jaldén and B. Ottersten, "Parallel implementation of a soft output sphere decoder," in *Proc. Asilomar Conf. Sig. Sys. Comp.*, Monterey, CA, Oct. 2005, pp. 581–585.

- [15] 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.
- [16] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM Rev.*, vol. 38, pp. 49–95, 1996.
- [17] 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.
- [18] 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, AK, May 2003, vol. 4, pp. 2653–2657.
- [19] J. Boutros, N. Gresset, L. Brunel, and M. Fossorier, "Soft-input soft-output lattice sphere decoder for linear channels," in *Proc. IEEE Global Telecomm. Conf.*, San Francisco, CA, Dec. 2003, vol. 3, pp. 1583–1587.
- [20] 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.
- [21] 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, no. 3, pp. 933–953, Mar. 2006.
- [22] 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.

- [23] 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.
- [24] 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, July 1999.
- [25] 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.
- [26] 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.
- [27] X. Li, A. Chindapol, and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding and 8 PSK signaling," *IEEE Trans. Commun.*, vol. 50, no. 8, pp. 1250–1257, Aug. 2002.
- [28] A. M. Tonello, "Space-time bit-interleaved coded modulation with an iterative decoding strategy," in *Proc. IEEE Vehicular Technology Conf.*, Boston, MA, Sept. 2000, vol. I, pp. 473–478.
- [29] E. Biglieri, G. Taricco, and E. Viterbo, "Bit-interleaved time-space codes for fading channels," in *Proc. Conf. Inform. Sci. Syst.*, Princeton, NJ, Mar. 2000, pp. WA4.1–WA4.6.
- [30] Z. Hong and B. L. Hughes, "Bit-interleaved space-time coded modulation with iterative decoding," *IEEE Trans. Wireless Commun.*, vol. 3, no. 6, pp. 1912–1917, Nov. 2004.

- [31] J. Wozencraft and I. M. Jacobs, *Principles of communication engineering*, Wiley, New York, NY, 1967.
- [32] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IRE*, vol. 37, no. 1, pp. 10–21, Jan. 1949.
- [33] G. Caire and K. R. Kumar, "Information theoretic foundations of adaptive coded modulation," *Proc. IEEE*, vol. 95, no. 12, pp. 2274–2298, Dec. 2007.
- [34] G. D. Forney, Jr. and G. Ungerboeck, "Modulation and coding for linear gaussian channels," *IEEE Trans. Inform. Theory*, vol. 44, no. 6, pp. 2384–2415, Oct. 1998.
- [35] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electron. Lett.*, vol. 33, no. 6, pp. 457–458, Mar. 1997.
- [36] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes," in *Proc. IEEE Int. Conf. Commun.*, Geneva, May 1993, vol. 2, pp. 1064–1070.
- [37] J. H. Conway and N. J. A. Sloane, *Sphere packings, lattices and groups*, Springer-Verlag, New York, NY, 1988.
- [38] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Trans. Inform. Theory*, vol. 28, no. 1, pp. 55–67, Jan. 1982.
- [39] H. Imai and S. Hirakawa, "A new multilevel coding method using error-correcting codes," *IEEE Trans. Inform. Theory*, vol. 23, no. 3, pp. 371–377, May 1977.

- [40] E. Zehavi, "8-PSK trellis codes for a Rayleigh channel," *IEEE Trans. Commun.*, vol. 40, no. 5, pp. 873–884, May 1992.
- [41] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Trans. Inform. Theory*, vol. 42, no. 2, pp. 429–445, Mar. 1996.
- [42] X. Li and J. A. Ritcey, "Bit-interleaved coded modulation with iterative decoding," *IEEE Commun. Lett.*, vol. 1, no. 6, pp. 169–171, Nov. 1997.
- [43] X. Li and J. A. Ritcey, "Trellis-coded modulation with bit interleaving and iterative decoding," *IEEE Commun. Lett.*, vol. 17, no. 4, pp. 715–724, Apr. 1999.
- [44] S. ten Brink, J. Speidel, and R. Yan, "Iterative demapping and decoding for multilevel modulation," in *Proc. IEEE Global Telecomm. Conf.*, Sydney, Australia, Nov. 1998, vol. I, pp. 579–584.
- [45] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, "Factor graphs and the sum-product algorithm," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.
- [46] J. Boutros and G. Caire, "Iterative multiuser joint decoding: Unified framework and asymptotic analysis," *IEEE Trans. Inform. Theory*, vol. 48, no. 7, pp. 1772–1793, July 2002.
- [47] A. P. Worthen and W. E. Stark, "Unified design of iterative receivers using factor graphs," *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 843–849, Feb. 2001.

- [48] J. Hu and T. M. Duman, "Graph-based detection algorithms for layered space-time architectures," *IEEE J. Select. Areas Commun.*, vol. 26, no. 2, pp. 269–280, Feb. 2008.
- [49] A. Gorokhov and M. van Dijk, "Optimised labeling maps for bit-interleaved transmission with turbo demodulation," in *Proc. IEEE Vehicular Technology Conf.*, Rhodes, Greece, May 2001, vol. II, pp. 1459–1463.
- [50] L. Bahl, J. Coke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inform. Theory*, vol. 20, no. 3, pp. 284–287, Mar. 1974.
- [51] 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, June 1995, vol. 2, pp. 1009–1013.
- [52] G. L. Stüber, J. R. Barry, S. W. McLaughlin, Y. Li, M. A. Ingram, and T. G. Pratt, "Broadband MIMO-OFDM wireless communications," *Proc. IEEE*, vol. 92, no. 2, pp. 271–294, Feb. 2004.
- [53] A. Scaglione, P. Stoica, S. Barbarossa, G. B. Giannakis, and H. Sampath, "Optimal designs for space-time linear precoders and decoders," *IEEE Trans. Signal Processing*, vol. 50, no. 5, pp. 1051–1064, May 2002.
- [54] N. Al-Dhahir and S. N. Diggavi, "Guard sequence optimization for block transmission over linear frequency-selective channels," *IEEE Trans. Commun.*, vol. 50, no. 6, pp. 938–946, June 2002.

- [55] D. Chizhik, J. Ling, P. W. Wolniansky, R. A. Valenzuela, N. Costa, and K. Huber, "Multiple-input-multiple-output measurements and modeling in Manhattan," *IEEE J. Select. Areas Commun.*, vol. 21, no. 3, pp. 321–331, Apr. 2003.
- [56] V. Tarokh, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criterion and code construction," *IEEE Trans. Inform. Theory*, vol. 44, no. 2, pp. 744–765, Mar. 1998.
- [57] G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," *Wireless Personal Comm.*, vol. 6, no. 3, pp. 311–335, Mar. 1998.
- [58] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1456–1467, July 1999.
- [59] V. Tarokh, A. Naguib, N. Seshadri, and A. R. Calderbank, "Space-time codes for high data rate wireless communication: Performance criteria in the presence of channel estimation errors, mobility, and multiple paths," *IEEE Trans. Commun.*, vol. 47, no. 2, pp. 199–207, Feb. 1999.
- [60] J.-C. Guey, M. P. Fitz, M. R. Bell, and W.-Y. Kuo, "Signal design for transmitter diversity wireless communication systems over Rayleigh fading channels," *IEEE Trans. Commun.*, vol. 47, no. 4, pp. 527–537, Apr. 1999.

- [61] H. El Gamal and A. R. Hammons, Jr., "On the design of algebraic space-time codes for MIMO block-fading channels," *IEEE Trans. Inform. Theory*, vol. 49, no. 1, pp. 151–163, Jan. 2003.
- [62] S. M. Alamouti, "A simple transmit diversity technique for wireless communications," *IEEE J. Select. Areas Commun.*, vol. 16, no. 8, pp. 1451–1458, Oct. 1998.
- [63] A. V. Geramita and J. Seberry, *Orthogonal designs, quadratic forms and Hadamard matrices*, Marcel Dekker, New York, NY, 1979.
- [64] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block coding for wireless communications: Performance results," *IEEE J. Select. Areas Commun.*, vol. 17, no. 3, pp. 451–460, Mar. 1999.
- [65] Xue-Bin Liang, "Orthogonal designs with maximal rates," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2468–2503, Oct. 2003.
- [66] H. Jafarkhani, "A quasi-orthogonal space-time block code," *IEEE Trans. Commun.*, vol. 49, no. 1, pp. 1–4, Jan. 2001.
- [67] 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.
- [68] R. H. Gohary and T. N. Davidson, "Design of linear dispersion codes: Asymptotic guidelines and their implementation," *IEEE Trans. Wireless Commun.*, vol. 4, no. 6, pp. 2892–2906, Nov. 2005.
- [69] J.-C. Belfiore, G. Rekaya, and E. Viterbo, "The Golden code: A  $2 \times 2$  full-rate space-time code with nonvanishing determinants," *IEEE Trans. Inform. Theory*, vol. 51, no. 4, pp. 1432–1436, Apr. 2005.

- [70] F. Oggier, G. Rekaya, J.-C. Belfiore, and E. Viterbo, "Perfect space-time block codes," *IEEE Trans. Inform. Theory*, vol. 52, no. 9, pp. 3885–3902, Sept. 2006.
- [71] J.-K. Zhang, J. Liu, and K. M. Wong, "Trace-orthonormal full-diversity cyclotomic space-time codes," *IEEE Trans. Signal Processing*, vol. 55, no. 2, pp. 618–630, Feb. 2007.
- [72] A. F. Naguib, N. Seshadri, and A. R. Calderbank, "Increasing data rate over wireless channels," *IEEE Signal Processing Mag.*, vol. 17, no. 3, pp. 76–92, May 2000.
- [73] G. J. Foschini, G.D. Golden, R.A. Valenzuela, and P.W. Wolniansky, "Simplified processing for high spectral efficiency wireless communication employing multi-element arrays," *IEEE J. Select. Areas Commun.*, vol. 17, no. 11, pp. 1841–1852, Nov. 1999.
- [74] G. J. Foschini, D. Chizhik, M. J. Gans, C. Papadias, and R. A. Valenzuela, "Analysis and performance of some basic space-time architectures," *IEEE J. Select. Areas Commun.*, vol. 21, no. 3, pp. 303–320, Apr. 2003.
- [75] P. W. Wonianski, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over rich scattering wireless channels," in *Proc. URSI Int. Symp. Sign. Syst. Electr.*, Pisa, Italy, Oct. 1998, pp. 295–300.
- [76] M. Sellathurai and S. Haykin, "Turbo-BLAST for wireless communications: Theory and experiments," *IEEE Trans. Signal Processing*, vol. 50, no. 10, pp. 2538–2546, Oct. 2002.

- [77] A. Stefanov and T. Duman, "Turbo-coded modulation for systems with transmit and receive antenna diversity over block fading channels: System model, decoding approaches, and practical considerations," *IEEE J. Select. Areas Commun.*, vol. 19, no. 5, pp. 958–968, May 2001.
- [78] S. L. Ariyavisitakul, "Turbo space-time processing to improve wireless channel capacity," *IEEE Trans. Commun.*, vol. 48, no. 8, pp. 1347–1359, Aug. 2000.
- [79] E. Biglieri, G. Taricco, and A. Tulino, "Performance of space-time codes for a large number of antennas," *IEEE Trans. Inform. Theory*, vol. 48, no. 7, pp. 1794–1803, July 2002.
- [80] E. Biglieri, A. Nardio, and G. Taricco, "Doubly iterative decoding of space-time turbo codes with a large number of antennas," *IEEE Trans. Commun.*, vol. 53, no. 5, pp. 773–779, May 2005.
- [81] Z. Hong and B. L. Hughes, "Robust space-time trellis codes based on bit-interleaved coded modulation," in *Proc. Conf. Inform. Sci. Syst.*, Baltimore, MD, Mar. 2001, vol. 2, pp. 665–670.
- [82] A. van Zelst, R. van Nee, and G. Awater, "Turbo-BLAST and its performance," in *Proc. IEEE Vehicular Technology Conf.*, Rhodes, Greece, May 2001, vol. II, pp. 1282–1386.
- [83] H.-J. Su and E. Geraniotis, "Space-time turbo codes with full antenna diversity," *IEEE Trans. Commun.*, vol. 49, no. 1, pp. 47–57, Jan. 2001.
- [84] C. Schlegel and A. Grant, "Concatenated space-time coding," in *Proc. IEEE Symp. Personal, Indoor and Mobile Radio Commun.*, San Diego, CA, Sept. 2001, vol. I, pp. 139–143.

- [85] 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.
- [86] 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.
- [87] 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, June 2005.
- [88] S. Bittner, E. Zimmermann, W. Rave, and G. Fettweis, "List sequential MIMO detection: Noise bias term and partial path augmentation," in *Proc. IEEE Int. Conf. Commun.*, Istanbul, Turkey, June 2006, vol. 3, pp. 1300–1305.
- [89] E. Viterbo and J. Bouros, "A universal lattice code decoder for fading channels," *IEEE Trans. Inform. Theory*, vol. 45, no. 5, pp. 1639–1642, July 1999.
- [90] M. O. Damen, H. El Gamal, and G. Caire, "On maximum-likelihood detection and the search for the closest lattice point," *IEEE Trans. Inform. Theory*, vol. 49, no. 10, pp. 2389–2402, Oct. 2003.
- [91] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

- [92] W. H. Mow, "Maximum likelihood sequence estimation from the lattice viewpoint," *IEEE Trans. Inform. Theory*, vol. 40, no. 5, pp. 1591–1600, Sept. 1994.
- [93] W. H. Mow, "Universal lattice decoding: Principle and recent advances," *Wireless Commun. and Mobile Computing*, vol. 3, no. 5, pp. 553–569, Aug. 2003.
- [94] O. Damen, A. Chkeif, and J. C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Commun. Lett.*, vol. 4, no. 5, pp. 161–163, May 2000.
- [95] J. Luo, K. R. Pattipati, P. Willett, and G. M. Levchuk, "Fast optimal and suboptimal any-time algorithms for CDMA multiuser detection based on branch and bound," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 632–642, Apr. 2004.
- [96] 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 Processing*, vol. 50, no. 4, pp. 912–922, Apr. 2002.
- [97] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Math. Comput.*, vol. 44, pp. 463–471, Apr. 1985.
- [98] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, and H. Bolcskei, "VLSI implementation of MIMO detection using the sphere decoding algorithm," *IEEE J. Solid-State Circuits*, vol. 40, no. 7, pp. 1566–1577, July 2007.

- [99] B. Hassibi and H. Vikalo, "On the sphere-decoding algorithm I. Expected complexity," *IEEE Trans. Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug. 2005.
- [100] P. Marsch, E. Zimmermann, and G. Fettweis, "Improved methods for search radius estimation in sphere detection based MIMO receivers," in *Proc. IST Mobile & Wireless Commun. Summit*, Dresden, Germany, June 2005.
- [101] J. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Trans. Commun.*, vol. 32, no. 2, pp. 169–176, Feb. 1984.
- [102] M. Kokkonen and K. Kalliojarvi, "Soft-decision decoding of binary linear block codes using reduced breadth-first search algorithms," *IEEE Trans. Commun.*, vol. 48, no. 6, pp. 905–907, June 2000.
- [103] K. S. Zigangirov, "Some sequential decoding procedures," *Probl. Pered. Inform.*, pp. 19–35, Nov. 1966.
- [104] M. Kokkonen and K. Kalliojarvi, "A fast sequential decoding algorithm using a stack," *IBM J. Res. Devel.*, vol. 13, pp. 675–685, Nov. 1969.
- [105] R. Fano, "A heuristic discussion of probabilistic decoding," *IEEE Trans. Inform. Theory*, vol. 9, no. 2, pp. 64–74, Apr. 1963.
- [106] A. AlRustamani, B. R. Vojcic, and A. Stefanov, "Greedy detection," *J. VLSI Signal Processing*, vol. 30, no. 1-3, pp. 179–195, 2002.
- [107] A. AlRustamani and B. R. Vojcic, "A new approach to greedy multiuser detection," *IEEE Trans. Commun.*, vol. 50, no. 8, pp. 1326–1336, Aug. 2002.

- [108] W. K. Ma, P. C. Ching, Z. Ding, and J. Choi, "Semidefinite relaxation based multiuser detection for M-ary PSK multiuser systems," *IEEE Trans. Signal Processing*, vol. 52, no. 10, pp. 2862–2872, Oct. 2004.
- [109] A. Mobasher, M. Taherzadeh, R. Sotirov, and A. K. Khandani, "A near-maximum-likelihood decoding algorithm for MIMO systems based on semi-definite programming," *IEEE Trans. Inform. Theory*, vol. 53, no. 11, pp. 3869–3886, Nov. 2007.
- [110] 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.
- [111] Y. E. Nesterov, "Quality of semidefinite relaxation for nonconvex quadratic optimization," Tech. Rep. CORE, Universite Catholique de Louvain, Brussels, Belgium, Mar. 1997.
- [112] 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 Processing*, vol. 54, no. 2, pp. 738–751, Feb. 2006.
- [113] R. Sivasankaran and S. W. McLaughlin, "Twin-stack decoding of recursive systematic convolutional codes," *IEEE Trans. Commun.*, vol. 49, no. 7, pp. 1158–1167, July 2001.
- [114] T. Gucluoglu and T. M. Duman, "Soft input soft output stack equalization for MIMO frequency selective fading channels," in *Proc. IEEE Int. Conf. Commun.*, Seoul, May 2005, vol. 1, pp. 510–514.

- [115] Y. Li and X.-G. Xia, "Iterative demodulation/decoding methods based on Gaussian approximations for lattice based space-time coded systems," *IEEE Trans. Wireless Commun.*, vol. 5, no. 8, pp. 1976–1983, Aug. 2006.
- [116] J. Pearl, *Probabilistic reasoning in intelligent systems: Network of plausible inference*, Morgan Kaufmann, San Francisco, CA, 1988.
- [117] H.-A. Loeliger, J. Dauwels, J. Hu, S. Korl, L. Ping, and F. R. Kschischang, "The factor graph approach to model-based signal processing," *Proc. IEEE*, vol. 95, pp. 1295–1322, June 2007.
- [118] R. A. Horn and C. R. Johnson, *Matrix analysis*, Cambridge University Press, New York, NY, 1985.
- [119] F. Hasegawa, J. Luo, K. R. Pattipati, P. Willett, and D. Pham, "Speed and accuracy comparison of techniques for multiuser detection in synchronous CDMA," *IEEE Trans. Commun.*, vol. 52, no. 4, pp. 540–545, Apr. 2004.
- [120] H. El-Gamal and E. Geraniotis, "Iterative multiuser detection for coded CDMA signals in AWGN and fading channels," *IEEE J. Select. Areas Commun.*, vol. 18, no. 1, pp. 30–41, Jan. 2000.
- [121] G. Caire, R. R. Müller, and T. Tanaka, "Iterative multiuser joint decoding: Optimal power allocation and low-complexity implementation," *IEEE Trans. Inform. Theory*, vol. 50, no. 9, pp. 1950–1973, Sept. 2004.
- [122] P. H. Tan and L. K. Rasmussen, "The application of semidefinite programming for detection in CDMA," *IEEE J. Select. Areas Commun.*, vol. 19, no. 8, pp. 1442–1449, Aug. 2001.

- [123] 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, HI, Apr. 2007, vol. IV, pp. 1329–1332.
- [124] D. Knuth, *The art of computer programming: Volume 3*, Addison Wesley, Reading, MA, 1973.
- [125] Y. Yang, C. Zhao, P. Zhou, and W. Xu, "MIMO detection of 16-QAM signaling based on semidefinite relaxation," *IEEE Signal Processing Lett.*, vol. 14, no. 11, pp. 797–800, Nov. 2007.
- [126] Z. Mao, X. Wang, and X. Wang, "Semidefinite programming relaxation approach for multiuser detection of QAM signals," *IEEE Trans. Wireless Commun.*, vol. 6, no. 12, pp. 4275–4279, Dec. 2007.
- [127] Z. Li, C. Pan, Y. Cai, and Y. Xu, "A novel quadratic programming model for soft-input soft-output MIMO detection," *IEEE Signal Processing Lett.*, vol. 14, no. 12, pp. 924–927, Dec. 2007.
- [128] J. F. Sturm, "Using SeDuMi 1.02, A Matlab toolbox for optimization over symmetric cones," *Optim. Meth. Soft.*, vol. 11–12, 1999.
- [129] W.-K. Ma, C.-C. Su, J. Jaldén, and C.-Y. Chi, "Some results on 16-QAM MIMO detection using semidefinite relaxation," in *Proc. IEEE Int. Conf. Acoustics, Speech, Signal Processing*, Las Vegas, NV, Apr. 2008, pp. 2673–2676.
- [130] S. Boyd and L. Vandenberghe, *Convex optimization*, Cambridge University Press, New York, NY, 2004.

- [131] R. A. Horn and C. R. Johnson, *Topics in matrix analysis*, Cambridge University Press, New York, NY, 1994.