N° d'ordre :

THESE

Pour obtenir le grade de :

DOCTEUR DE L'UNIVERSITE DE BRETAGNE SUD

Spécialité : Sciences de l'ingénieur

Mention : Technologies de l'information et des communications

AMOR NAFKHA

A geometrical approach detector for solving the combinatorial optimisation problem : Application in wireless communication systems

Soutenance prévue pour le 23 Mars 2006

COMPOSITION DE JURY

Rapporteurs	M. Jean-Francois Helard
	Mme Marie-Laure Boucheret
Examinateurs	M. Maurice Bellanger
	M. Emmanuel Boutillon
	M. Christian Roland
	M. Michel Jezequel

Laboratoire d'Electronique des Systèmes Temps Réel
Université de Bretagne Sud

Al-hamdoulillah and thanks to my mother and father,

Amor

Contents

abstract **11**

Acknowledgements **12**

Notation **13**

Abbreviations **14**

1 Introduction **16**

1.1 Purpose and Requirements of Research 16

1.2 Thesis Outline 17

2 Background **19**

2.1 Wireless channels 20

2.1.1 Large scale path loss 21

2.1.2 Small scale fading 22

2.1.2.1 Flat fading and frequency selective fading 23

2.1.2.2 Fast and slow fading 23

2.2 Diversity 23

2.3 Linear Channel Model 25

2.3.1 Probability distribution of $\tilde{\mathbf{w}}$ 26

2.3.2 Probability distribution of $\tilde{h_{ij}}$ 26

2.3.3 Equivalent Real-Valued Model 27

2.4 Signal Constellations 27

2.5 Channel model examples 28

2.5.1 Synchronous Code Division Multiple Access 29

2.5.2 Multicarrier Code Division Multiple Access 31

2.5.3 Multiple-Input Multiple-Output Arrays 33

2.5.4 The FIR Channel 36

2.6 Computational Complexity 37

2.7 Chapter Summary 38

3 Detection Fundamentals 39

3.1 Maximum Likelihood Detection 39

3.2 Branch and Bound 42

3.3 Sphere Decoding 44

3.4 Linear detector 48

3.4.1 Zero Forcing detector 49

3.4.2 Generalized MMSE detector 50

3.5 Interference cancellation detectors 51

3.5.1 Successive Interference Canceller 51

3.5.2 Parallel Interference Canceller 52

3.6 The Semidefinite Relaxation 53

3.7 Performances and Complexity Evaluation 55

3.7.1 Data Model 55

3.7.2 Complexity study 57

3.7.3 Performance 60

3.8 Conclusion 62

4 Proposed detection method 63

4.1 Singular Value Decomposition 64

4.1.1 Definition 65

4.1.2 Properties of the SVD 65

4.1.3 Detection performances and ill-conditioned channel occurs 66

4.1.4 Geometrical interpretation of ill-conditioned channel . . . 69

4.2 The intensification step 70

4.2.1 Intensification step study 73

4.2.2 computation complexity of the intensification step 74

4.3 The diversification step 75

4.3.1 Repeated random start local search 76

4.3.1.1 Definition 77

4.3.2 BCH diversification 78

4.3.3 Geometrical diversification 79

4.3.3.1 Hypercube Intersection and Selection method . . 81

4.3.3.2 Canonical Basis Intersection and Selection method 81
4.3.3.3 Plane Intersection and Selection method 83
4.4 GISD detector flow chart 84
4.5 Computation complexity of the GISD 88
4.6 Simulations and Discussions 89
4.6.1 Simulations 90
4.6.1.1 *Experiment 1*: MIMO 90
4.6.1.2 *Experiment 2*: MC-CDMA 90
4.7 Channel Estimation Errors 92
4.7.1 Error Model 93
4.7.2 Simulation Results 93
4.8 Conclusion 93

5 Extended GISD detector 95
5.1 Pre-processing complexity reduced 96
5.1.1 Fixed-step gradient descent method 96
5.1.2 An extended of fixed-step gradient descent method 98
5.1.2.1 Deflation method 99
5.1.2.2 Rayleigh method 99
5.1.2.3 Method summary/complexity 99
5.1.3 Performances of the extended method 100
5.2 16-QAM GISD detection 101
5.2.1 Extended GISD 102
5.2.2 Simulation results 103
5.3 Soft-output detection 104
5.3.1 Simulation results 107
5.4 Conclusion 109

6 VLSI implementation of GISD detection 110
6.1 Simplified norm algorithm 111
6.2 GISD Parameters Impact 114
6.3 Architecture description 115
6.3.1 Key requirements 116
6.4 Finite word length analysis 117
6.5 GISD Architecture 121
6.6 Globally Asynchronous Locally Synchronous 122

6.7 Implementation Results 122
6.8 Conclusion 123

7 Conclusions 125
7.1 Contributions 125
7.2 Topics for Future Work 126

A Properties of the Real-Valued Model 128

B Extraction of smallest singular vectors 130
B.1 The power method 130
B.2 Iterative methods and Krylov space 132
B.2.1 Lanczos method 133
B.2.2 Arnoldi method 134

C Algorithms 136
C.1 Intensification step 136
C.1.1 Greedy Search function using l_2-norm 136
C.1.2 Greedy Search function l_1-norm 137
C.1.3 Greedy Search function using l_∞-norm 138
C.2 Diversification step 139
C.2.1 HIS function using l_2-norm 139
C.2.2 HIS function using l_1-norm 140
C.2.3 HIS function using l_∞-norm 140
C.3 fixed step gradient descent algorithm 141

D Branch and Bound detector 143

Bibliography 145

List of Tables

3.1 Complexity computation of ZF and MMSE 58
3.2 Complexity computation of ML 59
3.3 Complexity computation of pre-processing step for SD and BBD . 59

4.1 Complexity computation of processing step of the 1^{st}-order and the 2^{sd}-order greedy search methods when θ is the iteration number 75
4.2 This table give all point in $sign(\mathcal{I}_1)$ when we used the HISD detector. 87
4.3 Application of greedy search in the case of HISD algorithm. . . . 87
4.4 This table give all point in $sign(\mathcal{I}_1)$ when we used the PISD detector. 88
4.5 Application of greedy search in the case of PISD algorithm. 88
4.6 Complexity computation of HIS, PIS and CBIS techniques for diversification step. 88
4.7 Complexity of processing part of Geometrical Intersection and Selection Detector. 89

6.1 Complexity of greedy search function using different norms when $n = 2N$, and θ is the iteration number. 113
6.2 Complexity of diversification step using different norms when $n = 2N$. 113
6.3 Relative Cost per Operation. 114
6.4 Quantization schemes summary. 121
6.5 Virtex2 synthesis results. 123

B.1 Krylov subspace methods 132

List of Figures

2.1 The wireless propagation environment. 21

2.2 Approximate behaviour of a diversity system. 24

2.3 The received vector model. 25

2.4 QAM signal constellations $\tilde{\mathcal{A}}$ used for transmission: 4,16-QAM (top), and their projections onto the real axis, $\mathcal{A}$, 2,4-ASK (bottom). 28

2.5 Multiple Access Communication System. 29

2.6 Multiple Access Communication System. 30

2.7 MC-CDMA transmitter and OFDM receiver $L_c = N_p$. 32

2.8 MIMO System Model. 34

2.9 Spatial Multiplexing System. 34

2.10 The transmission diversity of 2 for the rotated constellation. . . . 36

3.1 General Detection Setup: transmitted symbol vector $\mathbf{x} \in \mathcal{A}^n$, channel matrix $\mathbf{H} \in \mathbb{R}^{m \times n}$, additive noise vector $\mathbf{w} \in \mathbb{R}^m$ and detected symbol $\hat{\mathbf{x}} \in \mathbb{R}^n$. 40

3.2 (a)Bounded lattice representing the uncoded set of vectors $\mathcal{A}^2$. (b)Corresponding decision regions for the AWGN channel. 41

3.3 (a) Bounded lattice representing all possible vectors $\mathbf{Hx}$ for an interference channel. (b)Corresponding decision regions. 42

3.4 Example of the Branch and Bound method. 43

3.5 Geometric representation of the sphere decoding algorithm.. . . . 45

3.6 Example of the tree search in SD. 48

3.7 Hypotheses and decision quadrants in two dimensional case for decorrelating (ZF) detector. 50

3.8 Computation complexity comparison of the processing step (SD and BBD) with $n = m = 10$ and 4-QAM modulation in the worst-case, $r = \sqrt{\alpha n}\sigma$ where α is chosen so that [HV02]. 60

3.9 Average bit error rates achieved for linear zero-forcing (ZF) and linear minimum-mean-square error (MMSE) $N = M = 5$, $4-QAM$ uncoded system. 61
3.10 Performance comparison on Rayleigh fading channel for an uncoded systems using $4-QAM$ modulation 61

4.1 Distribution of the condition number for a (10×10) real iid random channel. 67
4.2 Cumulative distribution function of the condition number for (10×10) real iid random channel. 68
4.3 Performances of detectors in presence of channel $C1$ or channel $C2$. 68
4.4 Sections of the surface $f(\mathbf{x}) = const$ and ZF regions in the ZF-detection domain for good conditioned real (2;2) channel and BPSK modulation, SNR=0 dB, and $cond(\mathbf{H}) = 1,3428$. 70
4.5 Sections of the surface $f(\mathbf{x}) = const$ and ZF regions in the ZF-detection domain for ill-conditioned real (2;2) channel and BPSK modulation, SNR=0 dB, and $cond(\mathbf{H}) = 5,6284$. 71
4.6 Three dimensional visualization of a $Q = 1$ and $Q = 2$ neighborhood. In each case, all neighbors of the point $\mathbf{x}$ are encircled. . . . 72
4.7 Performances comparison of 1^{st}-order and 2^{sd}-order greedy search methods in case of uncoded BPSK modulation and $\mathbf{H} \in \mathbb{R}^{10\times10}$. . 73
4.8 The cumulative distribution function of local minima (number of subset K), when SNR=10 dB. 74
4.9 The effect of iteration number θ in BER performance when $m = n = 10$ and $\xi \triangleq \{-1,+1\}^n$ 75
4.10 Average BER of RRS-GS based detectors with different values of parameter p for $n = m = 10$ system using uncoded BPSK. 78
4.11 Shift Register Encoding using BCH Codes. 79
4.12 Performance Comparison between RRS-GS based detectors and GS using extended BCH code for $n = m = 16$ system. 80
4.13 Hypercube Intersection and Selection method : One to one mapping from $\{\beta_k^{s,i}\}_{i=1..n}^{s=-1,1}$ to $\{\bar{\beta}_k^{s,i}\}_{i=1..n}^{s=-1,1}$ for $n = 2$. 82

4.14 Canonical basis intersection method : One to one mapping from $\{\rho_{zf}, \beta_k^1, .., \beta_k^n\}$ to $\{\hat{\mathbf{x}}_{zf}, \bar{\beta}_k^1, .., \bar{\beta}_k^n\}$ for $n = 2$. Each intersection point β_k^i is of equal distance from its two neighboring candidate points. $\bar{\beta}_k^i$ is chosen to be one of these two candidate points that is on the opposite side of the i^{th} coordinate hyper-plane with respect to $\hat{\mathbf{x}}_{zf}$. 83

4.15 Plane Intersection and Selection method: One to one mapping from $\{\beta_k^t\}_{t=-l,-l+2,...,l-2,l}$ to $\{\bar{\beta}_k^t\}_{t=-l,-l+2,...,l-2,l}$ for $n = 2$. 85

4.16 Flowchart of the proposed algorithm. 86

4.17 BER versus SNR for $N = 5$ and $M = 5$ MIMO system, comparison of GISD variants, SDP and MMSE detectors. 91

4.18 BER versus SNR for $N = 5$ and $M = 6$ MIMO system, comparison of GISD variants, SDP and MMSE detectors. 91

4.19 Comparison between SD detection, GISD variants ($D = C = 4$) and others sub-optimum detector in a Rayleigh fading channel ($N_u = 16$ users) 92

4.20 BER versus SNR. Comparison of different detector methods (SD, PISD, HISD, SDSID, MMSE and SDP) with the same channel estimation error for a 5 × 5 uncoded MIMO system, iid channel matrix assumption. 94

5.1 BER versus SNR, comparison between sphere decoder ,EHISD ($D = 2$ and $C = 4$), and HISD ($D = 2$ and $C = 4$) 101

5.2 Diversification using hypercube intersection and selection variant for 16-QAM modulation and n=2 103

5.3 Two dimensional visualization of a $Q = 1$ neighborhood in case of 16-QAM modulation 103

5.4 Performance comparison for $M \times N$ uncoded MIMO systems with $N = M = 5$, $C = 4$ and $D = 2$. 104

5.5 Diagram of a MIMO system. 108

5.6 Diagram of a BLAST transmitter employing error control code (ECC). 108

5.7 Diagram of a BLAST receiver employing turbo decoder. 108

5.8 BER curves as a function of SNR for $m \times n$ channel transmitting 4-QAM with $R = 1/2$ memory 2 turbo code. 109

6.1 Performance comparison for 5×5 MIMO systems with uncoded 4-QAM modulation, $C = 4$ and $D = 2$. 113
6.2 BER/Complexity of HISD detector for 5×5 MIMO systems with uncoded 4-QAM modulation, $C = 4$ and $D \in \{1, 2, 3\}$. 115
6.3 BER/Complexity of HISD detector for $M \times N$ MIMO systems with uncoded 4-QAM modulation, $N = M = 5$, $n = 2N$, $D = 2$ and $C \in \{1, 2, 3, 4\}$. 116
6.4 Sundance architecture description. 117
6.5 Block diagram of the HISD testbed. 118
6.6 Performance comparison of various quantization schemes for $\mathbf{H}$ and $\mathbf{v}_k$: 5×5 MIMO system with uncoded 4-QAM modulation, $C = 4$ and $D = 2$. 120
6.7 Performance comparison of various quantization schemes for $\mathbf{y}$ and ρ_{zf}: 5×5 MIMO system with uncoded 4-QAM modulation, $C = 4$ and $D = 2$. 120
6.8 Block diagram of the GISD detector. 121
6.9 General GALS module. 123

abstract

Cette thèse s'intéresse à la résolution du problème classique de décodage d'un mélange linéaire entaché d'un bruit additif gaussien. A partir d'une observation bruitée: $\mathbf{y} = \mathbf{Hx}+\mathbf{b}$, d'un signal $\mathbf{x} \in \{\pm 1\}^n$ mélangé linéairement par une matrice $\mathbf{H}$ connue, $\mathbf{b}$ étant un vecteur de bruit, on cherche le vecteur $\mathbf{x}$ minimisant la distance Euclidienne entre $\mathbf{y}$ et le vecteur $\mathbf{Hx}$. Ce problème est réputé NP-complet. Il intervient dans un grand nombre de systèmes de télécommunications (CDMA, MIMO, MC-CDMA, etc.). Nous proposons dans cette thèse un algorithme de résolution quasi optimal de ce problème et bien adapté à une implémentation matérielle.

Notre démarche s'appuie sur l'utilisation des méthodes classiques de recherche opérationnelle : trouver des points initiaux répartis sur l'espace des solutions possibles et potentiellement proches de la solution optimale (diversification) et effectuer une recherche locale au voisinage des ces points (intensification). Dans ce travail, la diversification est basée sur une approche géométrique utilisant les axes dominants de concentration du bruit (vecteurs singuliers associés aux valeurs singulires minimales de la matrice $\mathbf{H}$). Les performances en terme de taux d'erreur par bit de la méthode proposée sont proches de l'optimum tout en gardant une complexité constante et un degré de parallélisme important (même pour des matrices H de taille très importantes, de l'ordre de 100). Nous avons étendu cette méthode à la constellation MAQ-16 d'une part, et à la génération d'une décision souple d'autre part.

Nous avons étudié l'algorithme proposé du point de vue implémentation matérielle. La sensibilité à l'utilisation de la précision finie et des normes sous optimales est décrite. Une étude de complexité de l'algorithme est présentée ainsi que les effets d'une mauvaise estimation de la matrice $\mathbf{H}$.

L'algorithme proposé présente d'une part une nouvelle alternative pour le

décodage quasi optimal du mélange linéaire bruité et d'autre part un important degré de parallélisme permettant une implémentation matérielle efficace et rapide.

Acknowledgements

It is time to draw a period to my Ph.D studies. I would like to thank all wonderful people around me. The years with all these people will be cherished forever.

I would like to express my sincerest thanks to my supervisor, Prof. Emmanuel. Boutillon. His constant encouragement, together with friendly guidance, has been invaluable. Through his insight, I have learned many lessons about communication engineering and the necessary skills to improve my research methods. I thank him for guiding me to learn the art of research and the philosophy behind it. Although there were times when I did not feel this way, I am now thankful that he did not tell me exactly what to do and allowed me to develop my own ideas. I have enjoyed the personal interactions. I feel very lucky to have an adviser as humorous as him.

I would also like to thank M. Christian Roland for his help and advice. I have appreciated his guidance in the area of telecommunication theory.

I am privileged to have been a member of the LESTER UBS Laboratory, which is a very positive environment for learning and working. To this end, I am thankful to all the lab members for the knowledge and friendship shared.

I would like to acknowledge the members of my thesis committee, Prof. Maurice Bellanger, Prof. Michel Jezequel, Prof. Jean-Francois Helard and Prof. Marie-Laure Boucheret for their invaluable suggestions and comments with respect to this thesis.

Finally, I would like to express my deepest gratitude to my parents Tahar and Daklia for giving me the opportunity to build a successful career. I wish also to thank my brothers and sisters Samir, Ramzy, Sami, Amel, Hayet and Fatma, for their constant and unconditional support.

Notation

Through out this report, small letters are used to denote scaler, complex or real variables. In order to denote real, complex or integer vectors we use small boldface letters and for real or complex matrices we use capital boldface letters. We use the notation presented in following table throughout this thesis:

Symbol	Description
$\Re(.)$	real part of complex variable or matrix
$\Im(.)$	imaginary part of complex variable or matrix
$\mathbf{I}_n$	identity matrix of size n
$\mathbf{A}^{-1}$	inverse of matrix $\mathbf{A}$
$\mathbf{A}^{+}$	pseudo inverse of matrix $\mathbf{A}$
$\mathbf{A}^{T}$	transpose of matrix $\mathbf{A}$
$\mathbf{A}_{i,j}$	element (i,j) of matrix $\mathbf{A}$
$\mathbf{A}(i,:)$	i^{th} row of matrix $\mathbf{A}$
$\mathbf{A}(:,i)$	i^{th} column of matrix $\mathbf{A}$
$tr(\mathbf{A})$	trace of matrix $\mathbf{A}$
$\|.\|_2^2$	Euclidean norm
$\|.\|_1^2$	Manhattan norm
$\|.\|_\infty^2$	Maximum norm
$\lceil a \rceil$	smallest integer, greater or equal than $a \in \mathbb{R}$
$\lfloor a \rfloor$	largest integer, lower or equal than $a \in \mathbb{R}$

Abbreviations

The following list summarizes the acronyms used in this thesis

APP	A Posteriori Probability
AWGN	Additive White Gaussian Noise
BBD	Branch and Bound Detector
BCH	Bose-Chaudhuri-Hocquenghem codes
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
BLAST	Bell Laboratories Layered Space Time
CBIS	Canonical Basis Intersection and Selection
CDMA	Code division multiple access
CSI	Channel State Information
DS-CDMA	Direct sequence code division multiple access
FDMA	Frequency Division Multiple Access
FIR	Finite Impulse Response
GISD	Geometrical Intersection and Selection Detector
GALS	Globally Asynchronous and Locally Synchronous
HIS	Hypercube Intersection and Selection
i.i.d.	independent identically distributed
ISI	Intersymbol Interference
LOS	Line-Of-Sight
MAI	Multiple-Access Interference
MC-CDMA	Multicarrier Code Division Multiple Access
MIMO	Multiple-Input Multiple-Output

ML	Maximum Likelihood
MMSE	Minimum Mean Squares Error
MRC	Maximum Ratio Combining
PIC	Parallel Interference Cancellation
PIS	Plane Intersection and Selection
OFDM	Orthogonal Frequency Division Multiplex
QAM	Quadratic Amplitude Modulation
QPSK	Quaternary Phase Shift Keying
SD	Sphere Decoder
SDP	Semi-Definite Programming
SDR	Semi-Definite Relaxation
SIC	Successive Interference Cancellation
SNR	Signal-to-Noise Ratio
STBC	Space Time Block coding
STTC	Space Time Trellis Coding
SVD	Singular Value Decomposition
TDMA	Time-Division Multiple Access
V-BLAST	Vertical Bell Labs layered space-time (detection algorithm)
VHDL	Very High Speed Integrated Circuit Hardware Description
ZF	Zero-forcing

Chapter 1

Introduction

The use of radio waves to transmit information from one point to another was discovered over a century ago. While commercial and military radio communication systems have been deployed for many decades,the last decade has seen an unprecedented surge in demand for personal wireless devices. Extensive penetration of the end user market is a direct result of advances in circuit design and chip manufacturing technologies that have enabled a complete wireless transmitter and receiver to be packaged in a pocket-sized device.

To achieve improved performance at high data rates, we require the implementation of highly sophisticated detection algorithm. However, existing detection methods involving in general matrix multiplications and inversions which increase significantly the computation complexity of the receiver. Moreover, many optimum and suboptimum detection techniques have been published but unfortunately most of these methods have inherent structure disadvantages which make them difficult to implement, or provide very limited performance improvement in a "*real world*" communications.

1.1 Purpose and Requirements of Research

The problem of finding the least-squares solution to a system of linear equations $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}$, where $\mathbf{y}$ is the received vector, $\mathbf{H}$ is a channel matrix, $\mathbf{x}$ is the transmitted vector of data symbols chosen from a finite set, and $\mathbf{w}$ is a noise vector, arises in many communication contexts: the equalization of intersymbol interference (ISI) channels, the cancellation of multiple-access interference (MAI) in code-division multiple-access (CDMA) systems, the decoding of multiple-input multiple-output (MIMO) systems in fading environments, the decoding of multi-carrier code-division multiple-access (MC-CDMA) systems, to name is more. The objective at the receiver is to detect the most likely vector $\mathbf{x}$ that was transmitted based on knowledge of $\mathbf{y}$, $\mathbf{H}$, and the statistics of $\mathbf{w}$.

The maximum-likelihood (ML) detector is well known to exhibit better bit-error-rate (BER) performance than many other existing detectors. Unfortunately, ML detection (MLD) is a non-deterministic polynomial-time hard ($\mathcal{NP}$-hard) problem, for which there is no known algorithm that can find the optimal solution with polynomial-time complexity (in the dimension of the search space).

The purpose of our research is to develop a sophisticated suboptimal ML detector satisfying the following three requirements:

- A near-maximum likelihood performance.
- A polynomial computational complexity.
- An inherent parallel structure for suitable hardware implementation.

To this end we adopt an approach based on "*real time*" operational research methods. In fact, the developed method is comprised of the following two complementary techniques:

- *Intensification*: local search method is used to find good solution to the receiver detection problem. Throughout this work, we use the term local search as a synonym of neighbourhood search. This technique has a main weaknesses: it may sometimes be trapped in a very poor local optimum. In order to overcome this difficulty, an efficient diversification technique has been required.
- *Diversification*: the idea is to create a reduced subset of good solutions in order to reduce risk that the intensification step gives local minimas for all starting solutions. The main principe of this step is: "*dont put all you eggs in one basket*".

1.2 Thesis Outline

This section outlines the chapters of this thesis. It should be noted that the focus of this thesis is not just the introduction of new strategy to solve the ML detection problem but also the analysis of methods which have been described in the literature.

- *Chapter 2*: This chapter provides a common framework for the rest of the thesis. The real model for the linear wireless channel is presented. A few motivating examples of systems which have previously been studied in the literature and which may be modeled as linear channels are given. Also, formal definitions of the concepts of polynomial and exponential complexity are given.

- *Chapter 3*: Based on the given channel model in chapter 2, the mathematical formulation of the ML detector is derived. Also, this chapter provides a review of most popular detection methods and discuss their performance and computational complexity.

- *Chapter 4*: Develop a new suboptimal detection algorithm based on an intensification/diversification strategy. The intensification algorithm (greedy search method) is described and its convergence properties are analyzed. Moreover, different diversification methods are investigated. The simulations presented in this chapter show that the proposed technique provides a good approximation to the ML detector with a computational complexity of $\mathcal{O}(n^3)$. Finally, we investigate the impact on the performance due to the channel estimation errors.

- *Chapter 5*: An extended of the proposed detection technique to 16-QAM constellation and a new soft-output detector based on the proposed detection technique (given in this work) are derived. Based on the recently work in [SG01], we present a low complex method to reduce the pre-processing complexity.

- *Chapter 6*: An implementation on a FPGAs/DSPs multiprocessor motherboard of the proposed technique is discussed. This discussion is mainly focussed on using different norm that can reduce the computation complexity and the study of proposed method parameters.

- *Chapter 7*: This chapter presents the conclusions and future work that can be drawn from the research presented in this thesis.

Chapter 2

Background

Wireless devices such as mobile phones have been gaining more and more popularity mainly because of their mobility. Though voice was the only service available on early phones, text service has now been added, and more recently multimedia services, such as pictures and videos have started to emerge. These services are not widespread, but the demand for them is increasing. At the same time wireless local area networks still have to compete with their wireline counterparts mainly because of their high data rates. Wireless local area networks are attractive for their mobility, but the high data rates available on the wireline network still seem to be unreachable in wireless networks. A requirement for high data rates directly imposes a wider bandwidth requirement which is not feasible because of the limited radio spectrum. Nevertheless, digital wireless systems are slowly replacing ordinary analog ones. Examples include the new standards for radio and television broadcast, the digital audio broadcasting and digital video broadcasting.

The increasing adoption of multimedia and demand for mobility in computer networks have resulted in a huge wireless research effort in recent years. Given the limited radio spectrum and unfriendly propagation conditions, designing reliable high data rate wireless networks requires solving many problems. The maximum capacity of a radio channel with a given bandwidth is limited by the well known Shannon [Sha48] formula. The Shannon limit gives the maximum limit on the capacity of a channel but does not say anything about the way to achieve that limit. Various techniques have been proposed to counter the problem of propagation conditions, and achieved data rates are now very close to the Shannon limit. Data transmission at rates higher than the Shannon limit have never been thought possible until very recently.

This chapter serves the purpose of introducing some central elements of this work, *i.e.* the linear wireless channel model and the mathematical formulation of the corresponding received vector. The definitions given in this chapter will play a fundamental role in the analysis of the existing detection algorithms in Chapter 2. Additionally, the concept of algorithm complexity will be introduced.

2.1 Wireless channels

The wireless channel in mobile radio poses a great challenge as a medium for reliable high speed communications. When a radio signal is transmitted through a wireless channel, the wave propagates through a physical medium and interacts with physical objects and structures, such as buildings, hills, trees, moving vehicles, *etc* [Rap96]. The propagation of radio waves through such an environment is a complicated process that involves diffraction, refraction, and multiple reflections. Also, the speed of the mobile impacts how rapidly the received signal level varies as the mobile terminal moves in space. Modeling all these phenomenon effectively has been one of the most challenging tasks related to wireless system design. Typically it is necessary to use statistical models that reasonably approximate the environment, based on measurements made in the field. A reliable communication system tries to overcome or take advantage of these channel perturbations.

A typical mobile radio communication scenario in an urban area usually involves an elevated fixed base-station antenna (or multiple antennas), a mobile handest, a line-of-sight (LOS) propagation path followed by many reflected paths due to the presence of natural and man-made objects between the mobile and the base station. The figure 2.1 illustrates such an environment. The different propagation paths (LOS as well as reflected paths) change with the movement of the mobile unit or the movement of its surroundings [Rap96].

Radio propagation models usually focus on predicting the average signal strength based upon the separation between the transmitter and the receiver, and also the rapid fluctuations in the instantaneous signal level that may be observed over short distances. The variation of the average signal strength over large distances (typically several hundreds of meters) is called *large scale path loss*. The rapid fluctuations over short travel distances (typically a few wavelengths)

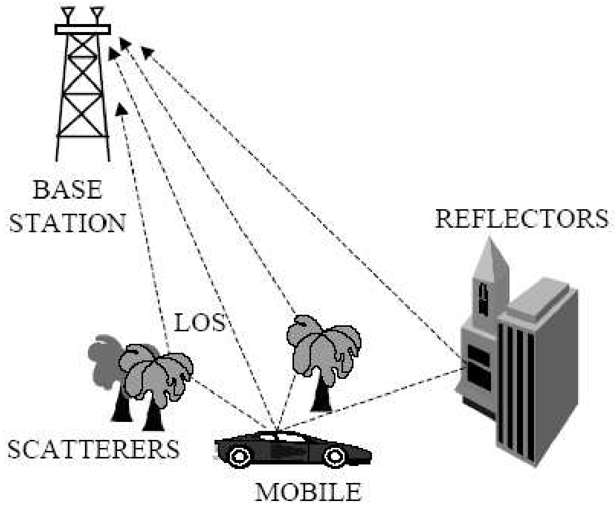


Figure 2.1: The wireless propagation environment.

is called *small scale fading*.

2.1.1 Large scale path loss

Both theoretical analysis and experimental measurements indicate that the average large scale path loss ($PL(d)$) or decrease in signal strength at the receiver is proportional to some power of the distance between the transmitter and the receiver

$$PL(d) \propto = (\frac{d}{d_0})^a \tag{2.1}$$

or

$$PL(d)[dB] = \bar{PL}(d_0)[dB] + 10 \cdot a \cdot log(d/d_0) \; d \geq d_0 \tag{2.2}$$

where d is the separation between the transmitter and receiver, d_0 is a reference distance which is determined from measurements close to the transmitter, and a is the path loss exponent. The path loss exponent determines the rate at which the path loss increases with the separation d, and its value depends on the specific propagation environment.

The path loss model stated above does not consider the fact that the dynamics of the wireless environment may be quite different at two different locations with the same transmitter-receiver separation. In order to predict the instantaneous signal level, the following model is widely used: the path loss $PL(d)$ at a location is randomly distributed around the average value:

$$PL(d)[dB] = \bar{PL}(d_0)[dB] + 10 \cdot a \cdot log(d/d_0) + \tilde{\mathbf{X}}_\sigma,\ d \geq d_0 \tag{2.3}$$

where $\tilde{\mathbf{X}}$ is a zero-mean Gaussian distributed random variable with standard deviation σ. This effect is known as *log-normal shadowing.*

2.1.2 Small scale fading

Small-scale fading refers to the rapid variations in the amplitude of the received signal, in a wireless environment, over short distances or time intervals, such that the effect of large-scale path loss may be ignored. There are a number of physical factors causing these rapid fluctuations [Rap96]:

- **Multipath propagation**: Due to the presence of a number of reflectors and scatterers between the transmitter and the receiver, in most instances, there may exist more than one path between the transmitter and receiver. These paths may add up either constructively or destructively due to the randomly time varying delays, phases, and attenuation of the various paths. So the amplitude of the composite received signal consisting of various path components may vary over time and give rise to the phenomenon of fading. The parameter of interest when dealing with multiple paths is the *delay spread.* The maximum delay spread is defined as the time delay during which the multipath energy falls to a pre-specified level below the maximum.

- **Doppler spread**: The relative motion between the base station and mobile as well as the movements of the surrounding objects results in random frequency modulation due to different Doppler shifts of each multipath component. The Doppler shift will be positive or negative depending on the direction of relative motion between the mobile and the base station. The parameter of interest here is the *Doppler spread* or the spectral broadening caused by this phenomenon. Again, it is defined as the range of frequencies over which the received Doppler spectrum is non-zero or above a certain threshold.

Depending upon the relationship between the signal parameters (bandwidth, symbol period) and the channel parameters (delay spread and Doppler spread), different transmitted signals will undergo different types of small scale fading in different environments.

2.1.2.1 Flat fading and frequency selective fading

If the mobile radio channel has a constant gain and linear phase response over a bandwidth which is greater than the signal bandwidth, the received signal undergoes flat fading and in the opposite case it is said to undergo frequency selective fading [Rap96].

In flat fading, the delay spread is much less than the symbol period and hence the spectral characteristics of the transmitted signal are preserved at the receiver. However the strength of the received signal varies with time. In frequency selective fading, the received signal includes multiple copies of the transmitted waveforms, attenuated and delayed in time, and hence the received signal is distorted. Modeling frequency selective fading is more difficult as each multipath signal should be modeled and the channel should be modeled as a linear filter. A typical model is the Rayleigh fading model which considers the channel impulse response to be made up of a number of delta functions which independently fade and have sufficient time delay between them to induce frequency selective fading.

2.1.2.2 Fast and slow fading

Depending upon the relative rate of change of the transmitted signal and the channel characteristics, a channel may be fast fading or slow fading [Rap96]. In a fast fading channel the channel impulse response varies rapidly within the symbol duration, that is, the Doppler spread is large relative to the symbol bandwidth. In slow fading, the channel may be assumed to be static over several symbol periods.

2.2 Diversity

The basic idea in diversity techniques is to use several independently fading channels to transmit the data. Then the receiver would pick up several independent replicas of the same signal. The probability that all these channels fade simultaneously is very low. In other words, there is higher probability that at least one high quality copy of the signal is present at the receiver. In this way, diversity reduces the bit error rate substantially by preventing most of the error bursts that usually happen in deep fades.

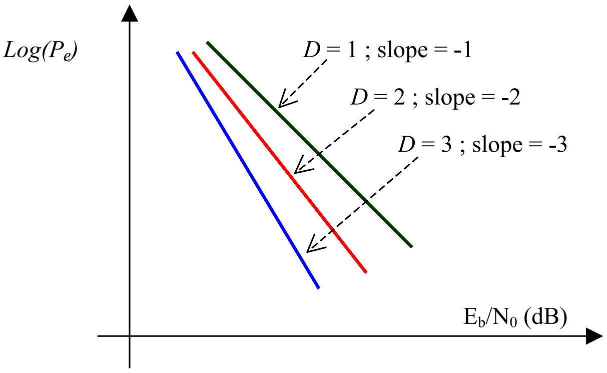


Figure 2.2: Approximate behaviour of a diversity system.

Without diversity, the probability of error P_e, decreases only as $SNR(E_b/N_0)^{-1}$ but if we have D independent channels, the probability of all of them failing would be $(P_e)^D$. Figure 2.2 shows the approximate behaviour of a diversity system. In this case D is called the diversity order of the system and it decreases when the channels are correlated or suboptimal detection methods are used. Several different methods can be used to achieve diversity.

One way is to employ frequency diversity. In this way several copies of the signal will be sent via different uncorrelated channels. In other words, the separation between these channels should be higher than coherence bandwidth to ensure independent fading. Another way is to use time diversity and send the signal several times over different time frames. Separation between these time frames also has to be long enough to make sure that they fade independently. These methods waste the bandwidth and energy because of the repetitive transmissions. Another more commonly used method is space diversity that employs multiple antennas. Different antennas can obtain independent fading if they have different polarizations or directionality, or if they are far enough spatially. The required separation can be determined by spatial correlation function and is usually a few wavelengths. The extra antennas can be either in the receiver or the transmitter. The advantage of antenna diversity is gaining extra quality or capacity without using extra spectrum.

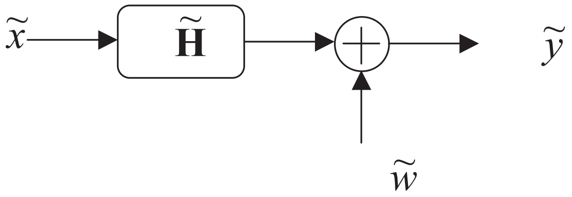


Figure 2.3: The received vector model.

2.3 Linear Channel Model

In this thesis, we focus on discrete-time baseband channel models, which abstract the channel impairments and hide the specific implementational details of the digital communication system. In doing so, we can talk about different digital communication systems with different kinds of channel interference in one common signal space framework. Let us now describe the channel model that we use in this thesis. The $N \times 1$ vector $\tilde{\mathbf{x}}$ contains the data to be transported over the channel, and is chosen from a finite equiprobable set $\tilde{\mathcal{A}}$. Depending on the underlying communication system, the components of $\tilde{\mathbf{x}}$ may correspond either to distinct time instants, distinct carrier frequencies, distinct physical locations, etc. The channel interference is modeled as linear interference, which is represented by multiplication of $\tilde{\mathbf{x}}$ with a $M \times N$ channel matrix $\tilde{\mathbf{H}}$. With channel noise being composed of the superposition of many independent actions, the central limit theorem suggests that we can model the noise as a zero-mean, complex-valued, additive white Gaussian noise (AWGN) vector $\tilde{\mathbf{w}}$ with circularly symmetric components of variance N_0 per dimension. The $M \times 1$ vector $\tilde{\mathbf{y}}$ that is obtained at the receiver, as illustrated in figure 2.3, is thus

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{w}} \tag{2.4}$$

In this thesis, we are primarily concerned with detection at the receiver of the transmit vector $\tilde{\mathbf{x}}$ based on knowledge of $\tilde{\mathbf{y}}$, $\tilde{\mathbf{H}}$, and the statistics of $\tilde{\mathbf{w}}$. The parameters of $\tilde{\mathbf{H}}$ can be learned at the receiver via techniques collectively known as *training*, in which $\tilde{\mathbf{H}}$ is estimated by sending vectors jointly known to the transmitter and receiver across the channel. If the channel changes with time, then the estimate of $\tilde{\mathbf{H}}$ can be updated using the detection decisions. Sometimes it is also useful to periodically perform training in case tracking becomes unsuccessful. We assume in most of the thesis that $\tilde{\mathbf{H}}$ and the statistics of $\tilde{\mathbf{w}}$ are explicitly known at the receiver.

2.3.1 Probability distribution of $\tilde{\mathbf{w}}$

A complex random vector $\tilde{\mathbf{w}}$ is said to be Gaussian if the real random vector

$$\mathbf{w} = \begin{bmatrix} \Re(\tilde{\mathbf{w}}) \\ \Im(\tilde{\mathbf{w}}) \end{bmatrix}$$

is Gaussian, where $\Re(\tilde{\mathbf{w}})$ and $\Im(\tilde{\mathbf{w}})$ are the real and imaginary parts of $\tilde{\mathbf{w}}$, respectively.

To determine the distribution of vector $\mathbf{w}$, its expectation and covariance matrix must be specified. Let $\mathbf{A}^\dagger$ denote the conjugate transpose matrix of $\mathbf{A}$, and $\varepsilon[.]$ denote the expected value. If the covariance matrix of $\mathbf{w}$ has the form

$$\varepsilon[(\mathbf{w} - \varepsilon[\mathbf{w}])(\mathbf{w} - \varepsilon[\mathbf{w}])^\dagger] = \frac{1}{2} \begin{bmatrix} \Re(\mathbf{Q}) & -\Im(\mathbf{Q}) \\ \Im(\mathbf{Q}) & \Re(\mathbf{Q}) \end{bmatrix}$$

where $\mathbf{Q} \in \mathbb{C}^{M \times M}$ is a Hermitian non-negative definite matrix, then $\tilde{\mathbf{w}}$ is said to be circularly symmetric. In this case, the covariance matrix of $\tilde{\mathbf{w}}$ is given by $\mathbf{Q}$.

Since each element of $\mathbf{w}$ is independent of the others, then its covariance matrix has the form:

$$\mathbf{Q}_{\mathbf{w}} = \mathbf{I}_{2M} \cdot N_0$$

where $\mathbf{I}_{2M} \in \mathbb{R}^{2M \times 2M}$ is the identity matrix; in consequence, the noise in an interference channel model defined above is circularly symmetric. Its mean value is the same as that of $\mathbf{w}$, and its covariance matrix is given by

$$\mathbf{Q}_{\tilde{\mathbf{w}}} = \mathbf{I}_M \cdot 2N_0$$

2.3.2 Probability distribution of $\tilde{h_{ij}}$

The probability density function of a random complex variable can be specified as the joint density function of its real and imaginary parts. In the case of the elements of $\tilde{\mathbf{H}}$, $\tilde{h_{ij}}$, $1 \leq i \leq M$, $1 \leq j \leq N$, both its real and imaginary parts are independent Gaussian random variables of zero mean and variance 0.5 per dimension. Let $h_R = \Re(\tilde{h_{ij}})$ and $h_I = \Im(\tilde{h_{ij}})$. The probability density function of $\tilde{h_{ij}}$ is then given by

$$\begin{aligned} p(\tilde{h_{ij}}) &= p(h_R) \cdot p(h_I) \\ &= \frac{exp(-h_R^2)}{\sqrt{\pi}} \cdot \frac{exp(-h_I^2)}{\sqrt{\pi}} \\ &= \frac{exp(-|\tilde{h_{ij}}|^2)}{\pi} \end{aligned}$$

Each element $\tilde{\mathbf{y}}[i], \quad i = 1, 2, ..., M$ of the received vector $\tilde{\mathbf{y}}$ is a different linear combination of the transmitted vector $\tilde{\mathbf{x}}$ plus noise. The coefficients of the linear combinations are determined by the rows of $\tilde{\mathbf{H}}$.

2.3.3 Equivalent Real-Valued Model

For complex-valued channel models it will turn out to be useful to work with an equivalent real-valued transmission model. Taking the complex-valued model (4.20) by separating real and imaginary parts we can equivalently write [Tel95].

$$\begin{bmatrix} \Re(\tilde{\mathbf{y}}) \\ \Im(\tilde{\mathbf{y}}) \end{bmatrix} = \begin{bmatrix} \Re(\tilde{\mathbf{H}}) & -\Im\tilde{\mathbf{H}} \\ \Im(\tilde{\mathbf{H}}) & \Re(\tilde{\mathbf{H}}) \end{bmatrix} \begin{bmatrix} \Re(\tilde{\mathbf{x}}) \\ \Im(\tilde{\mathbf{x}}) \end{bmatrix} + \begin{bmatrix} \Re(\tilde{\mathbf{w}}) \\ \Im(\tilde{\mathbf{w}}) \end{bmatrix} \tag{2.5}$$

which gives an equivalent $n = 2N$-dimensional real model of the form

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{2.6}$$

with the obvious definitions of $\mathbf{y}$, *etc.* Some useful properties of this mapping from complex to real matrices and vectors are collected in Appendix A. One reason to use this description is that, if the components of $\mathbf{x}$ are taken from some set of evenly spaced points on the real line, the noiseless received signal $\mathbf{Hx}$ from (2.6) can be interpreted as points in a lattice described by the basis $\mathbf{x}$, and the detection problem to be considered as an instance of the lattice decoding problem.

2.4 Signal Constellations

In this work, the signal sets that we consider for the components of the symbols to be transmitted will be called $\tilde{\mathbf{a}}$, and the corresponding real-valued vector $\mathbf{a}$. The channel input vector is denoted as $\tilde{\mathbf{x}}$, and $\tilde{\mathbf{x}} = \tilde{\mathbf{a}}$ if no transmitter processing is performed. Traditional communication systems use constellations among which

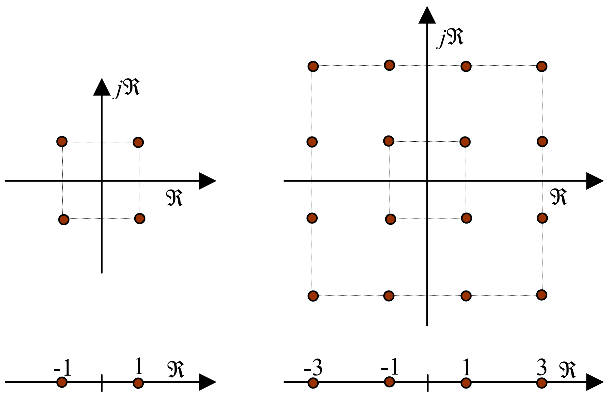


Figure 2.4: QAM signal constellations $\tilde{\mathcal{A}}$ used for transmission: 4,16-QAM (top), and their projections onto the real axis, $\mathcal{A}$, 2,4-ASK (bottom).

QPSK and M-QAM are popular, illustrated in figure 2.4. A QPSK constellation uses two quadrature carriers each of which is BPSK modulated. In M-QAM the phase as well as the amplitudes of a pair of quadrature carriers are varied according to the binary data. Whereas QPSK can transmit a maximum two bits per symbol, M-QAM can send $log_2(M)$ bits per symbol. However, higher level constellations have higher probabilities of error thus requiring higher SNRs to achieve a given bit error rate (BER).

Since in our schemes the dominant errors will be symbols distorted to the nearest neighbors of the transmit symbol, we use Gray labeling [Pro00] to map the information bits to the constellation points in order to minimize the effect of symbol errors on the bit error rate. Since the nearest neighbors in the complex plane are situated either along a purely real or purely imaginary offset, Gray labeling can be independently applied to real and imaginary part of the constellation.

2.5 Channel model examples

Though we focus specifically on applications of the vector detection model (4.20) to digital communication systems, the detection schemes we develop in this thesis are applicable to any scenario in which (4.20) applies. We now give some applications of this model in digital communication.

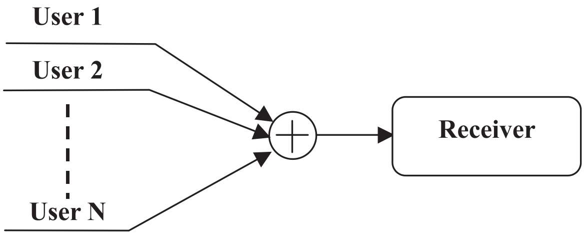


Figure 2.5: Multiple Access Communication System.

2.5.1 Synchronous Code Division Multiple Access

The channel model (4.20) can be applied in the uplink scenario of a N-user discrete-time synchronous code division multiple access (CDMA) system [PSM82]. The term of multiple access communication system is used for a system that uses a communication channel to enable several transmitters to send information at the same time. Multiple access communication is used widely in different communication systems, especially in mobile and satellite communications. The signal sources in a multiple access channel are referred to as users. The multiple access communication scenario is depicted in Figure 2.5. Several multiple access techniques have been implemented in current wireless systems, such as frequency-division multiple access (FDMA), Time-division multiple access (TDMA) and CDMA.

In a CDMA system, users are assigned distinct *signature sequences* or *spreading* codes. Each transmitter sends its data stream by modulating it with its own scrambling code. Since the scrambling codes have fairly low mutual cross correlation, a CDMA receiver can detect its own data using the corresponding scrambling code, although the multiple users' signals overlap both in frequency and in time. The cost of this is that the spectrum of the transmitted wave will be spread by M times, where M is the "spreading factor". The scrambling code contains M chips in a data symbol period. The scrambling codes should be carefully designed to achieve low cross correlations between users. For ease of generation and synchronization, a scrambling code is pseudo random, meaning that it can be generated by mathematically precise rules, but statistically it satisfies the requirements of a truly random sequence. A CDMA transmitter spreads the data by multiplying it with a pseudo noise (PN) sequence (scrambling code). The receiver then despreads the desired signal by multiplying it with a synchronized replica

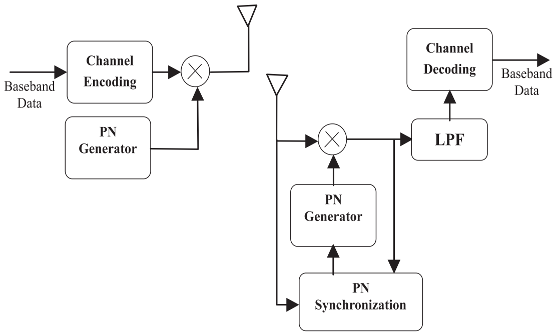


Figure 2.6: Multiple Access Communication System.

of the original PN sequence. The baseband model of a direct sequence CDMA (DS-CDMA) transmitter and receiver is shown in Figure 2.6. The PN sequence is called a "short code" if it is the same for every data symbol period (that is, its repetitive period is equal to the symbol period). Short-code scrambling can be used to model the uplink CDMA system (it is an option in UMTS uplink). The well known short codes are Gold and Kasami [DJ98]. However, if the period of the PN sequence is larger than the symbol period, the data symbols will be modulated by different portions of the sequence. This kind of PN sequence is called a "long sequence". Most CDMA systems (such as UMTS) employ long-code scrambling in downlink. Long sequences can support more users than short sequences. For a synchronous CDMA system operating in Additive White Gaussian Noise (AWGN) channel, the equivalent low-pass received waveform can be expressed as [Ver98]:

$$y(t) = \sum_{k=1}^{N} \sqrt{E_k} s_k(t) x_k + n(t), \quad t = [0, T] \tag{2.7}$$

where N is the number of users, E_k, $s_k(t)$ and $x_k \in \{-1, 1\}$ represent energy per bit, unit-energy signature waveform and bit value of the k^{th} user, respectively; T is the bit interval and $n(t)$ is the noise. The receiver consists of a bank of filters matched to the signature waveforms assigned to the users and a multiuser detector. The output of the filter matched to the signature waveform of user k and sampled at T is achieved by the following equation.

$$y_k = \int_0^T y(t) s_k(t) dt = \sqrt{E_k} x_k + \sum_{i=1, i \neq k}^{N} \sqrt{E_i} \rho_{ik} x_i + n_k \tag{2.8}$$

where,

$$n_k = \int_0^T n(t)s_k(t)dt, \quad and \quad \rho_{ik} = \int_0^T s_i(t)s_k(t)dt$$

where ρ_{ik} denotes the cross correlation of the signature waveforms of users i and k and n_k denotes the noise at the output of the k^{th} matched filter. The matched filter outputs are sufficient statistics for optimal multiuser detection and can be expressed in vector form as follows:

$$y_k = [y_1, y_2, .., y_N]^T = \mathbf{REx} + \mathbf{n} \tag{2.9}$$

where $\mathbf{R}$ is the normalized cross correlation matrix of the signature waveforms, $\mathbf{R}_{ij} = \rho ij$, $\mathbf{E} = diag(\sqrt{E_1}, \sqrt{E_2}, .., \sqrt{E_N})$, the noise vector is $\mathbf{n}$ with autocorrelation matrix $\frac{\sigma^2\mathbf{R}}{2}$ and σ^2 is the one-sided noise power spectral density of a zero-mean AWGN source.

2.5.2 Multicarrier Code Division Multiple Access

In order to obtain multiple access transmission systems with high bandwidth efficiency, Multi-Carrier Code Division Multiple Access (MC-CDMA) combines Orthogonal Frequency Division Multiplex (OFDM) modulation and CDMA [MCH01, CBJ93, YLF93]. The OFDM modulation is robust against multipath and ensures good spectral efficiency. The CDMA allows simultaneous communications between different transceivers by allocating to each transmission link a distinct signature (or spreading sequence) that has good orthogonal properties with the other used signatures. Instead of spreading the binary information in the time domain as in the Direct Sequence CDMA technique, the MC-CDMA spreading is performed in the frequency domain. Therefore, the orthogonality among transceivers signals has to be ensured in the frequency domain.

Let us consider a synchronous MC-CDMA system with N_u users as described in figure 2.7. At time i and for user k, the transmitted symbol $\tilde{\mathbf{x}}_i(k)$, taken from a modulation alphabet $\tilde{\mathcal{A}}$ of cardinality $|\tilde{\mathcal{A}}|$, is spread by a signature $\tilde{c}_k = (\tilde{c}_{k1}, .., \tilde{c}_{kL_c})$, which has good cross-correlation properties with other user signatures. In this thesis, signatures belong to an orthogonal Walsh-Hadamard set of size L_c. After spreading of $\tilde{\mathbf{x}}_i(k)$, the L_c obtained chips are transmitted with signal amplitude $\tilde{a}(k)$ on the N_p different sub-carriers of an OFDM modulation symbol. We assumed that $L_c = N_p$ and we denote $\tilde{\mathbf{s}}_i(k)$ the modulated signal

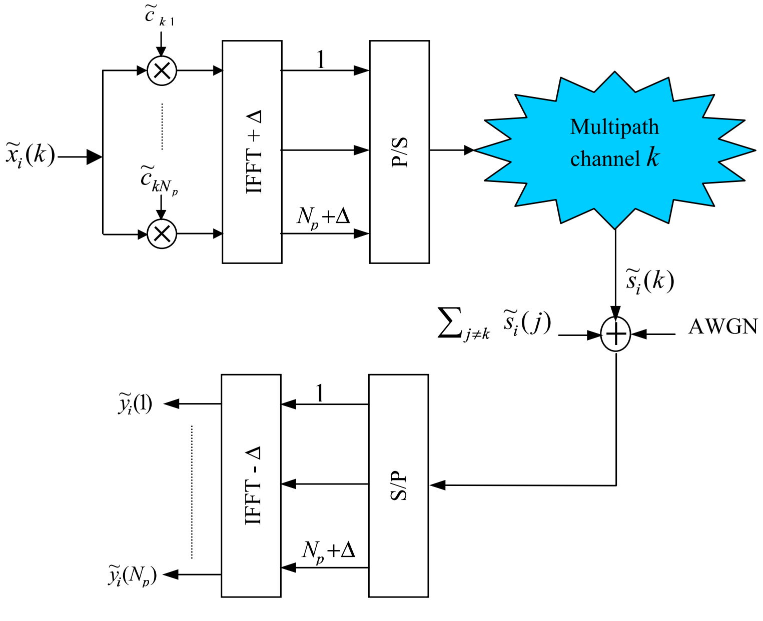


Figure 2.7: MC-CDMA transmitter and OFDM receiver $L_c = N_p$.

filtered by a frequency selective multipath channel. After addition of interfering user signals $\sum_{j\neq k} \tilde{\mathbf{s}}_i(j)$ and Additive White Gaussian Noise (AWGN), OFDM demodulation is performed. The channel is assumed non frequency selective on the sub-carrier bandwidth and is thus described by a single complex coefficient $\tilde{h}^i_{kp}$ for each user k and each sub-carrier p. We denote $\tilde{\mathbf{C}}^i$ the $N_u \times N_p$ matrix combining spreading and channel effects for all users:

$$\tilde{\mathbf{C}}^i = \begin{bmatrix} \tilde{c}_{11}\tilde{h}^i_{11} & \tilde{c}_{12}\tilde{h}^i_{12} & \dots & \tilde{c}_{1N_p}\tilde{h}^i_{1N_p} \\ \tilde{c}_{21}\tilde{h}^i_{21} & \tilde{c}_{22}\tilde{h}^i_{22} & \dots & \tilde{c}_{2N_p}\tilde{h}^i_{1N_p} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{c}_{N_u1}\tilde{h}^i_{N_u1} & \tilde{c}_{N_u2}\tilde{h}^i_{N_u2} & \dots & \tilde{c}_{N_uN_p}\tilde{h}^i_{N_uN_p} \end{bmatrix}$$

At time i, the received vector $\tilde{\mathbf{y}}_i = (\tilde{\mathbf{y}}_i(1), , \tilde{\mathbf{y}}_i(N_p))^T$ may be expressed as

$$\tilde{\mathbf{y}}_i = \tilde{\mathbf{C}}_i \tilde{\mathbf{A}} \tilde{\mathbf{x}}_i + \tilde{\mathbf{w}}_i \tag{2.10}$$

where vector $\tilde{\mathbf{x}}_i = (\tilde{\mathbf{x}}_i(1), ..., \tilde{\mathbf{x}}_i(N_u))^T$ contains the N_u transmitted symbols, diagonal matrix $\tilde{\mathbf{A}} = diag(\tilde{a}(1), ..., \tilde{a}(N_u))$ contains the amplitudes of the different users and $\tilde{\mathbf{w}}_i = (\tilde{\mathbf{w}}_i(1), ..., \tilde{\mathbf{w}}_i(N_p))^T$ is the AWGN vector. In downlink, all users share the same channel, defined by $\tilde{\mathbf{H}}_i = diag(\tilde{h}_i(1), ..., \tilde{h}_i(N_p))$ Thus, $\tilde{\mathbf{C}}_i = \tilde{\mathbf{C}}\tilde{\mathbf{H}}_i$

where all user signatures are placed in the $N_u \times N_p$ matrix defined as

$$\tilde{\mathbf{C}} = \begin{bmatrix} \tilde{c}_{11} & \tilde{c}_{12} & \dots & \tilde{c}_{1N_p} \\ \tilde{c}_{21} & \tilde{c}_{22} & \dots & \tilde{c}_{2N_p} \\ \vdots & \vdots & \vdots & \vdots \\ \tilde{c}_{N_u1} & \tilde{c}_{N_u2} & \dots & \tilde{c}_{N_uN_p} \end{bmatrix}$$

2.5.3 Multiple-Input Multiple-Output Arrays

In this section we consider a multiple-input multiple-output (MIMO) scenario where multiple antenna arrays are at both ends. This configuration has many degrees of freedom and is expected to provide us with increased capacity and diversity with no increase in required bandwidth.

Pioneering work by Winters [Win87], Foschini [FG98] and Telatar [Tel95] has predicted a significant capacity increase associated with the use of multiple transmit and multiple receive antenna systems. This is under the assumptions that the channel can be accurately tracked at the receiver and exhibits rich scattering in order to provide independent transmission paths from each transmit antenna to each receive antenna. A key feature of MIMO systems is the ability to turn multipath propagation, traditionally seen as a major drawback of wireless transmission, into a benefit. This discovery resulted in a explosion of research activity in the realm of MIMO wireless channels for both single user and multiple user communications. In fact, this technology seems to be one of the recent technical advances with a chance of resolving the traffic capacity bottleneck in future Internet-intensive wireless networks. It is surprising to see that just a few years after its invention, MIMO technology already seems poised to be integrated in large-scale commercial wireless products and applications. A MIMO system model can be depicted as in Figure 2.8

The MIMO transmitter first demultiplexes the data stream onto the multiple antennas using an appropriate algorithm, then transmits the substreams through the antennas in parallel. A suitable algorithm is applied after the receiver antenna array to multiplex the multiple observations and recover the original data stream. Assuming the transmitter and receiver antenna arrays have N and M elements respectively, and the channel model is Rayleigh flat fading, the channel matrix

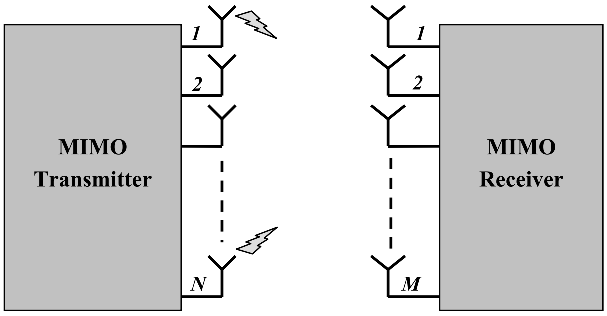


Figure 2.8: MIMO System Model.

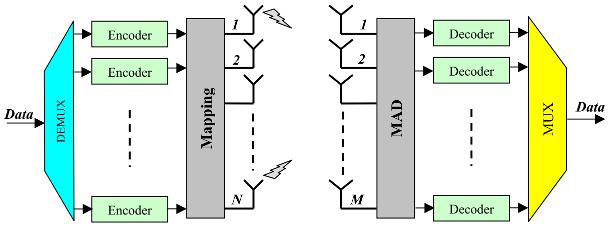


Figure 2.9: Spatial Multiplexing System.

for a MIMO system can be expressed as

$$\tilde{\mathbf{H}} = \begin{pmatrix} \tilde{h}_{11} & \tilde{h}_{21} & \cdots & \tilde{h}_{N1} \\ \tilde{h}_{12} & \tilde{h}_{22} & \cdots & \tilde{h}_{N2} \\ \vdots & \vdots & \ddots & \vdots \\ \tilde{h}_{1M} & \tilde{h}_{2M} & \cdots & \tilde{h}_{NM} \end{pmatrix} \tag{2.11}$$

where $\tilde{h}_{mn}$ $(n = 1, ..., N \quad m = 1, ..., M)$ is the fading factor from the n^{th} transmitter to the m^{th} receiver antenna. The received vector is written as:

$$\tilde{\mathbf{y}} = [\tilde{y}_1, \tilde{y}_1, ..., \tilde{y}_M]^T = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{w}} \tag{2.12}$$

where $\tilde{\mathbf{x}} = [\tilde{x}_1, \tilde{x}_2, ..., \tilde{x}_N]^T$ is the N-dimensional transmit signal vector, and $\tilde{\mathbf{w}}$ stands for the M-dimensional additive i.i.d. circularly symmetric complex Gaussian noise vector.

MIMO techniques are commonly divided into two classes, space-time codes (STC) and spatial multiplexing (SM). The former includes space-time trellis codes [TSC98] and space-time block codes [TSC98], while the best-known example of the latter is BLAST (Bell labs LAyered Space Time architecture). These two approaches have different motivations. The former is derived from earlier transmit diversity schemes. Hence its main motivation is to increase diversity, and thus improve the robustness of a communications link. The latter's main objective is to increase the capacity of a link. Its multiple transmitter antennas can equivalently be regarded as multiple users which consists in fact of data from the same user.

The basic principle of SM is to transmit essentially independent data from each antenna. Then at the receiver the multi-antenna signal is separated with appropriate detection techniques. A SM system can be illustrated as in Figure 2.9. In its simplest form the demultiplexed data is simply transmitted on the separate transmit antennas, and received using a multi-antenna detector, which is similar to a multiuser detector and treats separate streams as separate users of a multiuser channel.

Much effort has gone into developing space-time codes that posses specific properties. The first space time code was proposed by Alamouti [Ala98] over two transmit antenna and two time periods. This code is orthogonal and has linear decoding complexity. Tarokh *et al* [TJC99a, TJC99b] proved that orthogonal codes do not exist for more than two transmit antennas over complex constellations. Orthogonal codes have lower decoding compexity but are rate deficient. A set of codes called linear dispersion codes were proposed by Hassibi [HH02] that achieve capacity of the channel. These codes achieve both rate and diversity and hence give us a handle for design required for an application. Decoding of space time codes using maximum likelihood rule leads to exponential complexity in number of antennas making their usage prohibitive.

Among the existing techniques to build Space-Time block codes we can mention a powerful approach using algebraic number theory to construct full diversity ST codes. This theory has been used to construct appropriate modulation schemes well adapted to Rayleigh Fading channel based on rotated constellations. This idea has been first proposed by K. Boulle and J.-C. Belfiore in [BB92]. In this work they have proved that an n-dimensional constellation where all pairs of distinct symbols have all their coordinates distinct, leads to n^{th} order diversity. In other words a two-dimensional rotated QAM constellation as shown in figure 2.10 has a diversity order of 2 comparing with a classical QAM constellation with a transmit diversity of 1.

At the beginning, these constellations were only used to improve the performance of SISO system over Fading channels. Damen was the first who applied

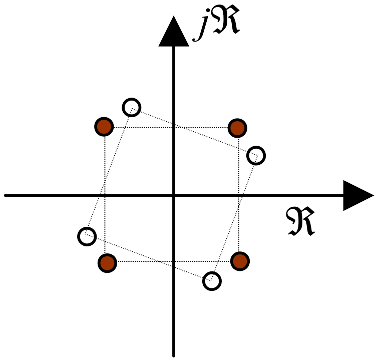


Figure 2.10: The transmission diversity of 2 for the rotated constellation.

these constellations on the context of Multiple-antennas in [Dam98] by imposing design criteria on algebraic codes [TSC98]. This idea was then generalized in [DAMB02] to propose new block ST practical architecture, where the rotated constellation are used to code information symbols.

2.5.4 The FIR Channel

Consider a single antenna time-discrete finite impulse response (FIR) channel where the channel output at time k, $\tilde{\mathbf{y}}_k$, is given by

$$\tilde{\mathbf{y}}_k = \tilde{\mathbf{h}}_k \tilde{\mathbf{s}}_k + \tilde{\mathbf{w}}_k = \sum_{l=0}^{L} \tilde{\mathbf{h}}_l \tilde{\mathbf{x}}_{k-l} + \tilde{\mathbf{w}}_k \tag{2.13}$$

where $\tilde{\mathbf{s}}_k$ is the transmitted symbol at time k, $\tilde{\mathbf{h}}_l$ is the channel impulse response, and $\tilde{\mathbf{w}}_k$ is an additive noise term. Assume that the channel impulse response, $\tilde{\mathbf{h}}_l$ is zero for $l \neq [0, L]$ and that a burst of N symbols, $\tilde{\mathbf{x}}_0, ..., \tilde{\mathbf{x}}_{N-1}$, is transmitted. Then this system may be written on matrix form as

$$\begin{bmatrix} \tilde{\mathbf{y}}_0 \\ \tilde{\mathbf{y}}_1 \\ \vdots \\ \tilde{\mathbf{y}}_{N-1} \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{h}}_0 & 0 & \dots & \dots & 0 \\ \tilde{\mathbf{h}}_1 & \tilde{\mathbf{h}}_0 & \dots & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & \dots & \tilde{\mathbf{h}}_L & \dots & \tilde{\mathbf{h}}_0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{x}}_0 \\ \tilde{\mathbf{x}}_1 \\ \vdots \\ \tilde{\mathbf{x}}_{N-1} \end{bmatrix} + \begin{bmatrix} \tilde{\mathbf{w}}_0 \\ \tilde{\mathbf{w}}_1 \\ \vdots \\ \tilde{\mathbf{w}}_{N-1} \end{bmatrix}$$

which is on the form of (4.20).

2.6 Computational Complexity

When analyzing an algorithm it is useful to establish how the computational complexity varies with parameters such as the size n, where n is the dimension of the solution space, or the SNR. Herein, focus will be on the dependence on n. An investigation into how the complexity depends on n will yield useful information about when a specific algorithm is well suited. However, a direct study of how much time an algorithm requires to solve a specific problem of size, n, will generally depend on the particular hardware on which the algorithm is implemented. For this reason it is more common to instead study how the complexity varies with n and to classify the algorithm based on this behavior.

As the concept of algorithm complexity will play a fundamental role in this work it is useful to give definitions of what is meant by statements such as polynomial and exponential complexity. To this end, consider the following definitions [NW88].

- $Definition$ 1 A function $f(n)$ is said to be in $\mathcal{O}(g(n))$ if there exist constants c and K such that $f(n) \leq cg(n)$ for all $n \geq K$.
- $Definition$ 2 A function $f(n)$ is said to be in $\Omega(g(n))$ if there exist constants c and K such that $f(n) \geq cg(n)$ for all $n \geq K$.
- $Definition$ 3: A function $f(n)$ is said to grow polynomially if there exists a constant $a \geq 0$ such that $f(n) \in \mathcal{O}(n^a)$.
- $Definition$ 4: A function $f(n)$ is said to grow exponentially if there exist constants $a > 1$ and $b > 1$ such that $f(n) \in \Omega(a^n) \cap \mathcal{O}(b^n)$.

The complexity class $\mathcal{P}$ (polynomial time) is the set of all problems for which an algorithm with complexity $\mathcal{O}(p(n))$ exists, with $p(n) \in \mathbb{Z}[n]$. The complexity class $\mathcal{NP}$ (non-deterministic polynomial time) is the set of all problems for which a given solution can be checked for correctness in polynomial time, even if finding the solution in polynomial time is only possible with a genie-aided algorithm (hence nondeterministic). A problem is called $\mathcal{NP}$-hard if an algorithm for solving it can be translated into one solving any other $\mathcal{NP}$-problem. This notation is often used to express the number of operations an algorithm takes to solve a problem as a function of the problem size. As an example, standard matrix multiplication of two $n \times n$ matrices takes $\mathcal{O}(n^3)$ operations.

2.7 Chapter Summary

In this chapter, we described the general channel model (2.6) used in this thesis. We start by discussing the characteristics of the radio channel such as attenuation, multipath, the Doppler effect and fading. diversity techniques are introduced and discussed briefly. Some application of the general model $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}$ are then shown.

The strategies to be discussed in next chapter are the so-called detection strategies, *i.e*, how the receiving side in the communication system obtains estimates of the transmitted vector. we will present an optimal and a sub-optimal detection techniques, for which some complexity and performance comparison are discussed.

Chapter 3

Detection Fundamentals

Given a channel model as described in the previous chapter, the task of the receiver is to detect the transmitted signal $\mathbf{x}$ from $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}$, *i.e.*, construct an estimate $\hat{\mathbf{x}}$, given $\mathbf{y}$ and $\mathbf{H}$. The operation is described by the general block diagram of Figure 3.1. We assume that the detector, *i.e.*, the receiver in the transmission system, has perfect knowledge of the channel matrix $\mathbf{H}$, while no channel knowledge is necessary or exploited at the transmitter side. The data symbol vector $\mathbf{x} = [\mathbf{x}_1, .., \mathbf{x}_n]^T$ to be transmitted is selected from the constellation $\mathcal{A}^n$.

In this chapter, various optimum and sub-optimum multiuser receivers, inspired from the work of Chabbouh [Cha04], are presented, as well as discussions of their performances and computational complexity.

3.1 Maximum Likelihood Detection

Consider a wireless communication model diagram shown in Figure 3.1. To communicate over this channel, we are faced with the task of detecting a set of n transmitted symbols from a set of m observed signals. Our observations are corrupted by the non-ideal communication channel, typically modelled as a linear system followed by an additive noise vector. To assist us in achieving our goal, we draw the transmitted symbols from a known finite alphabet $\mathcal{A}$ of size L. The detector's role is then to choose one of the L^n possible transmitted symbol vectors based on the available data. Our intuition correctly suggests that an optimal detector should return $\hat{\mathbf{x}}$, the symbol vector whose (posterior) probability of having

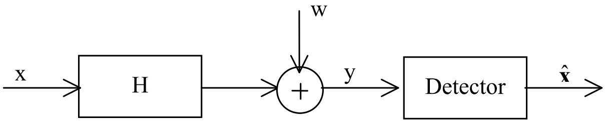


Figure 3.1: General Detection Setup: transmitted symbol vector $\mathbf{x} \in \mathcal{A}^n$, channel matrix $\mathbf{H} \in \mathbb{R}^{m\times n}$, additive noise vector $\mathbf{w} \in \mathbb{R}^m$ and detected symbol $\hat{\mathbf{x}} \in \mathbb{R}^n$.

been sent, given the observed signal vector $\mathbf{y}$, is the largest:

$$\begin{aligned} \hat{\mathbf{x}} &= \arg\max_{\mathbf{x}\in\mathcal{A}^n} p(\mathbf{x}\,\text{was send}|\mathbf{y}\,\text{is observed}) && (3.1)\\ &= \arg\max_{\mathbf{x}\in\mathcal{A}^n} \frac{p(\mathbf{y}\,\text{is observed}|\mathbf{x}\,\text{was send})p(\mathbf{x}\,\text{was send})}{p(\mathbf{y}\,\text{is observed})} && (3.2)\end{aligned}$$

equation 3.2 is known as the Maximum A posteriori Probability (MAP) detection rule. Making the standard assumption that the symbol vectors $\mathbf{x} \in \mathcal{A}^n$ are equiprobable, *i.e.*, that $p(\mathbf{x}\quad \text{was send})$ is constant, the optimal MAP detection rule can be written as:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}\in\mathcal{A}^n} p(\mathbf{y}\,\text{is observed}|\mathbf{x}\,\text{was send}) \tag{3.3}$$

A detector that always returns an optimal solution satisfying (3.3) is called a Maximum Likelihood (ML) detector. If we further assume that the additive noise $\mathbf{w}$ is white and Gaussian, then we can express the ML detection problem of Figure 3.1 as the minimization of the squared Euclidean distance metric to a target vector $\mathbf{y}$ over an n-dimensional finite discrete search set:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{A}^n} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \tag{3.4}$$

where borrowing terminology from the optimization literature we call the elements of $\mathbf{x}$ *optimization variables* and $f(\mathbf{x}) = \|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2$ the *objective function.*

In the special case of an AWGN channel, the interference channel model becomes $\mathbf{y} = \mathbf{x} + \mathbf{w}$, and so $\mathbf{y}$ is a noise-perturbed version of $\mathbf{x}$. The minimum-distance rule (3.4) simplifies to

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathcal{A}^n} \|\mathbf{y} - \mathbf{x}\|_2^2 \tag{3.5}$$

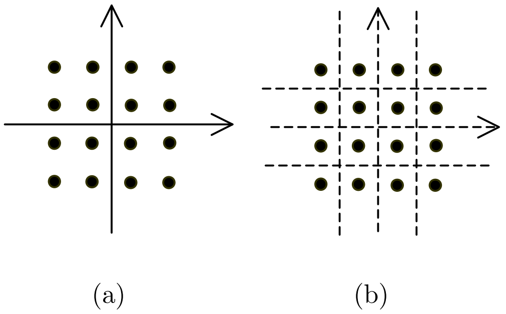


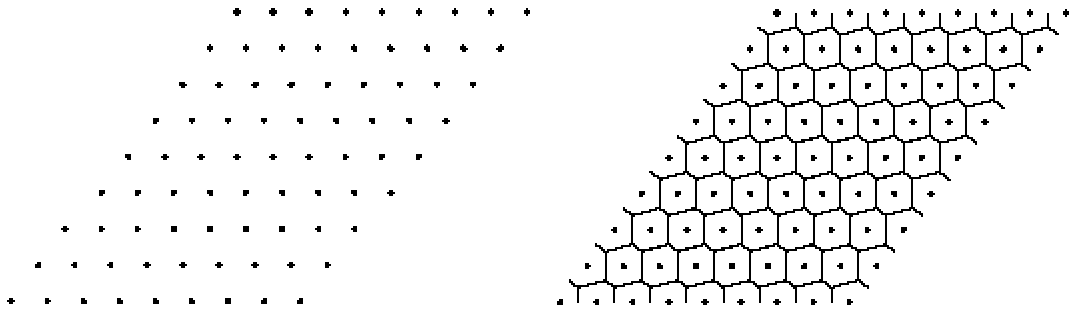
Figure 3.2: (a)Bounded lattice representing the uncoded set of vectors $\mathcal{A}^2$. (b)Corresponding decision regions for the AWGN channel.

Since each component of the uncoded vector $\mathbf{x}$ affects only the corresponding component of $\mathbf{y}$, and since the noise vector is uncorrelated, the ML detector can be decoupled into a set of symbol-by-symbol optimizations; *i.e.*,

$$\hat{\mathbf{x}}[i] = \arg \min_{\mathbf{x}[i] \in \mathcal{A}} \|\mathbf{y}[i] - \mathbf{x}[i]\|_2^2 \quad i = 1, .., n \tag{3.6}$$

which can be solved using a symbol-by-symbol minimum-distance decision device or slicer. The decision regions, corresponding to the values of $\mathbf{y}$ for which each of the possible decisions is made, are depicted in figure 3.2(b). The ability to decouple the ML detector into component wise minimizations is indicated by the fact that the boundaries of the decision regions form an orthogonal grid. The minimization for each of the n components of x requires the comparison of $|\mathcal{A}|$ differences, so complexity is linear in n.

In the general case in which linear interference is present, we have $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}$, and the ML vector detector generally cannot be decomposed into n smaller problems. We can see this by first recognizing that the action of $\mathbf{H}$ on the set of all possible uncoded $x \in \mathcal{A}^n$ vectors is to map the points of the bounded orthogonal lattice in figure 3.2(a) to the points of a bounded lattice with generators along the directions of the columns of $\mathbf{H}$, like the bounded lattice in figure 3.3(a) . The decision regions of (3.4) are now generally polytopes as shown in figure 3.3(b), and the decoupling of the problem is no longer possible. The minimization of (3.4) requires the comparison of $|\mathcal{A}|^n$ differences, so complexity is exponential in n. In fact, the least-squares integer program in (3.4) for general $\mathbf{H}$ matrices has been shown to be nondeterministic polynomial-time hard ($\mathcal{NP}$-hard) [Ver89]. The high complexity of the ML detector has invariably precluded its use in practice, so lower-complexity detectors that provide exact and approximate solutions to (3.4) are used, which we review in the next sections.



(a) (b)

Figure 3.3: (a) Bounded lattice representing all possible vectors $\mathbf{Hx}$ for an interference channel. (b)Corresponding decision regions.

3.2 Branch and Bound

Branch and Bound is a general discret search method [LD60]. Suppose we wish to minimize a function $f(\mathbf{x})$, where $\mathbf{x}$ is restricted to some feasible regions (*e.g.*, $\mathbf{x} \in \mathcal{A}^n \equiv \{+1, -1\}^n$). To apply branch and bound, one must have a means of computing a lower bound on an instance of the optimization problem and a means of dividing the feasible region of a problem to create smaller subproblems. There must also be a way to compute an upper bound (feasible solution) for at least some instances; for practical purposes, it should be possible to compute upper bounds for some set of nontrivial feasible regions.

The method starts by considering the original problem with the complete feasible region, which is called the root problem. The lower-bounding and upper-bounding procedures are applied to the root problem. If the bounds match, then an optimal solution has been found and the procedure terminates. Otherwise, the feasible region is divided into two regions (see figure 3.4), each strict subregions of the original, which together cover the whole feasible region; ideally, these subproblems partition the feasible region. These subproblems become children of the root search node.

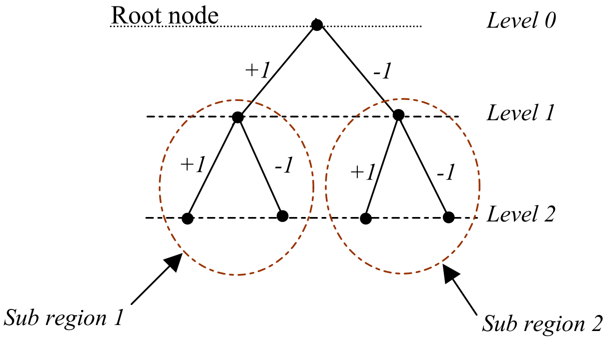


Figure 3.4: Example of the Branch and Bound method.

The algorithm is applied recursively to the subproblems, generating a tree of subproblems. If an optimal solution is found to a subproblem, it is a feasible solution to the full problem, but not necessarily globally optimal. Since it is feasible, it can be used to prune the rest of the tree: if the lower bound for a node exceeds the best known feasible solution, no globally optimal solution can exist in the subspace of the feasible region represented by the node. Therefore, the node can be removed from consideration. The search proceeds until all nodes have been solved or pruned, or until some specified threshold is met between the best solution found and the lower bounds on all unsolved subproblems. In the system model presented at the first chapter, when the components of the vector $\mathbf{x}$ are in $\{-1, 1\}$, we can use branch and bound method such that in each branching, one of the variables is fixed to 1 in one branch and -1 in the other.

The received signal at each time is : $\mathbf{y} = \mathbf{Hx} + \mathbf{w}$, where $\mathbf{H}$ is the channel matrix. The output of the filter matched to $\mathbf{H}$ can be written as

$$\mathbf{r} = \mathbf{H}^T\mathbf{y} = \mathbf{Gx} + \mathbf{b} \tag{3.7}$$

where $\mathbf{G} = \mathbf{H}^T\mathbf{H}$ is the $n \times n$ correlation matrix and $\mathbf{b}$ is the gaussian noise vector with zero mean and autocorrelation matrix $\frac{\sigma^2\mathbf{H}}{2}$. The ML detection criterion with perfect knowledge of the channel is equivalent to searching the point $\hat{\mathbf{x}} \in \{+1, -1\}^n$ such that

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \{+1,-1\}^n} \|\mathbf{r} - \mathbf{Gx}\|_2^2 \tag{3.8}$$

The optimal solution to (3.8) can be obtained by examining each of the 2^n possible $\mathbf{x}$'s. There are intelligent ways to compute such combinations. In [LPPB03], an optimal algorithm based on the branch and bound method with an iterative lower bound update was proposed. It was shown that the proposed method can significantly decrease the average computational cost. Suppose $\mathbf{G} = \mathbf{L}^T\mathbf{L}$ is the Cholesky decomposition of $\mathbf{G}$. Then (3.8) is equivalent to

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \{-1,+1\}^n} \|\mathbf{L}\mathbf{x} - (\mathbf{L}^{-1})^T \mathbf{r}\|_2^2 \tag{3.9}$$

Denote $\bar{\mathbf{r}} = (\mathbf{L}^{-1})^T \mathbf{r}$, $\mathbf{d} = \mathbf{L}\mathbf{x}$, and denote the k^{th} component of $\mathbf{d}$ and $\bar{\mathbf{r}}$ by $\mathbf{d}_k$ and $\bar{\mathbf{r}}_k$, respectively. Consequently, (3.9) becomes [LPPB03]

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \{-1,+1\}^n} \sum_{k=1}^{n} (\mathbf{d}_k - \bar{\mathbf{r}}_k)^2 \tag{3.10}$$

Since $\mathbf{L}$ is a lower triangular matrix, $\mathbf{d}_k$ depends only on $(\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k)$. When the decisions for the first k coordinates of vector $\mathbf{x}$ are fixed, the term

$$\xi_k = \sum_{i=1}^{k} (\mathbf{d}_i - \bar{\mathbf{r}}_i)^2 \tag{3.11}$$

becomes a lower bound of (3.10). Besides the use of the lower bound, the general Branch and Bound Detector (BBD) method [Ber98] has several variations in searching the nodes, including the depth-first search, breadth first search, best-first search, etc. Since the observation vector $\mathbf{r}$ is generated from a statistical model (3.7), [LPPB03] proposed an efficient BBD-based algorithm Dthat reduces the average computational cost significantly, compared with other optimal algorithms.

In the next section, we presented the sphere decoding [FP85] which can be categorized as a depth-first branch and bound detection algorithm [LPPB03].

3.3 Sphere Decoding

The sphere decoding algorithm is an optimum-ML detection technique. It promises to find the optimal solution with low computational costs under some conditions. SD was first introduced by Finke and Pohst [FP85] in the context of the closest point search in lattices but it has become very popular in digital communication literature. The Sphere decoder can easily be adapted to decode coded/uncoded communication system. For example this decoder has been used in the context of

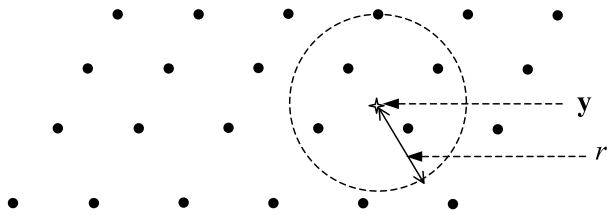


Figure 3.5: Geometric representation of the sphere decoding algorithm..

space-time block codes [Dam98, DAMB00, DSAM03]. In [Bru02] Sphere Decoder is also used in the context of multi-carrier CDMA systems.

The Sphere Decoder algorithm has been applied first to the communication context in [VB93, VB99]. It has been highlighted that the so called Sphere Decoder algorithm is well adapted to decode multidimensional modulation schemes in presence of Fading. A Generalized Sphere Decoder specially adapted to Multiple-antenna system has been proposed in [DBAM00] which makes possible the ML decoding of Multiple-antenna system with arbitrary number of transmit and receive antennas. Sphere Decoder was initially proposed for real valued systems. The generalization of SD to complex valued systems has been made in [DCB00].

The optimal ML detection which leads to solving the combinatorial optimization problem (3.4) and finding an exact solution for it, is in general $\mathcal{NP}$-hard. As entries of $\mathbf{x}$ are integer, $\mathbf{x}$ spans a rectangular n-dimensional lattice and for a real matrix $\mathbf{H}$, $\mathbf{Hx}$ spans a skewed lattice. Therefore, given the real vector $\mathbf{y}$ and the skewed lattice $\mathbf{Hx}$, the problem (3.4) would be equivalent to finding the closest lattice point to $\mathbf{y}$ in euclidean sense (figure 3.5).

The basic idea of SD is to limit search only to the lattice points $\mathbf{Hx}$ that lie in a sphere of radius r around the given vector $\mathbf{y}$ and in this way save on computations. It is clear that the closest point inside the sphere is also the closest point in the lattice. Therefore, there is no need to make an exhaustive search over all lattice points. Moreover, if the radius of sphere is properly chosen one can limit the number of operations used in order to find the desired point in sphere. The sphere decoder (or "sphere detector", as we may also call it in the context of MIMO detection) solves

$$\min_{\mathbf{x}\in\{-1,+1\}^n} (\mathbf{x}-\rho)^T\mathbf{H}^T\mathbf{H}(\mathbf{x}-\rho) \tag{3.12}$$

where ρ is the center of our search sphere. We observe that

$$\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 = (\mathbf{x}-\rho)^T\mathbf{H}^T\mathbf{H}(\mathbf{x}-\rho)+\mathbf{y}^T(\mathbf{I}-\mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T)\mathbf{y} \tag{3.13}$$

where $\rho = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{y}$ is the unconstrained maximum likelihood estimate of $\mathbf{x}$. The true (constrained) maximum likelihood estimate is therefore

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\{-1,+1\}^n}\|\mathbf{y}-\mathbf{H}\mathbf{x}\|_2^2 = \arg\min_{\mathbf{x}\in\{-1,+1\}^n}(\mathbf{x}-\rho)^T\mathbf{H}^T\mathbf{H}(\mathbf{x}-\rho) \tag{3.14}$$

The sphere decoder may thus be used to find $\hat{\mathbf{x}}$.

Solving (3.12) is generally difficult unless $\mathbf{H}$ has orthogonal columns, in which case the n-dimensional search becomes n simple 1-dimensional searches. Otherwise, an exhaustive search needs to examine 2^n different hypotheses. The sphere decoder avoids an exhaustive search by examining only those points that lie inside a sphere

$$(\mathbf{x}-\rho)^T\mathbf{H}^T\mathbf{H}(\mathbf{x}-\rho)\leq r^2 \tag{3.15}$$

with the given radius r large enough to contain the solution. The algorithm is described and refined in [FP85] and has its origins in finding the shortest vector in a lattice. Its application as a decoder for fading channels is described in [VB99], and as a maximum likelihood decoder for multiple antenna channels in [DCB00]. As we show below, sphere decoding uses the same Cholesky factorization of the channel matrix.

We assume, for the moment, that $r \geq 0$ has been chosen so that the sphere (3.15) contains the solution to (3.12) and possibly some additional points of the lattice. Let $\mathbf{U}$ be an upper triangular $n\times n$ matrix chosen such that $\mathbf{U}^T\mathbf{U} = \mathbf{H}^T\mathbf{H}$ (using, for example, Cholesky factorization). Let the entries of $\mathbf{U}$ be denoted $\mathbf{u}_{ij}$, $i \leq j = 1,...,n$, and assume, without loss of generality, that $\mathbf{u}_{ij} > 0$. Then (3.15) may be written

$$(\mathbf{x}-\rho)^T\mathbf{U}^T\mathbf{U}(\mathbf{x}-\rho) = \sum_{i=1}^{n}\mathbf{u}_{ii}^2[\mathbf{x}_i-\rho_i+\sum_{j=i+1}^{n}\frac{\mathbf{u}_{ij}}{\mathbf{u}_{ii}}(\mathbf{x}_j-\rho_j)]^2\leq r^2 \tag{3.16}$$

Each term in the sum over i is nonnegative. The sphere decoder establishes bounds on $\mathbf{x}_1,...,\mathbf{x}_n$ by examining these terms in subsets.

Starting with $i = n$, and throwing out the terms $i = 1,...,n-1$, we obtain from (3.16)

$$\mathbf{u}_{nn}^2(\mathbf{x}_n - \rho_n)^2 \leq r^2$$

or

$$\lceil \rho_n - \frac{r}{\mathbf{u}_{nn}} \rceil \leq \mathbf{x}_n \leq \lfloor \rho_n + \frac{r}{\mathbf{u}_{nn}} \rfloor \tag{3.17}$$

(The function $\lceil . \rceil$ finds the smallest integer greater than or equal to its argument, and $\lfloor . \rfloor$ finds the largest integer less than or equal to its argument; these functions are used because the constellation is assumed to be set of consecutive integers.) After computing the lower and upper bounds in (3.17), the sphere decoder chooses a candidate value for $\mathbf{x}_n$ and computes the implications of this choice on $\mathbf{x}_{n-1}$. To find the influence of the choice of $\mathbf{x}_n$ on $\mathbf{x}_{n-1}$, the sphere decoder looks at the two terms $i = n-1, n$ in (3.16), throws out the remaining terms, and obtains the inequality

$$\mathbf{u}_{n-1,n-1}^2[\mathbf{x}_{n-1} - \rho_{n-1} + \frac{\mathbf{u}_{n-1,n}}{\mathbf{u}_{nn}}(\mathbf{x}_n - \rho_n)]^2 + \mathbf{u}_{nn}^2(\mathbf{x}_n - \rho_n)^2 \leq r^2$$

which yields the upper bound

$$\mathbf{x}_{n-1} \leq \lfloor \rho_{n-1} + \frac{\sqrt{r^2 - \mathbf{u}_{nn}^2(\mathbf{x}_n - \rho_n)^2}}{\mathbf{u}_{n-1,n-1}} - \frac{\mathbf{u}_{n-1,n}}{\mathbf{u}_{nn}}(\mathbf{x}_n - \rho_n) \rfloor \tag{3.18}$$

and a corresponding lower bound. The sphere decoder now chooses a candidate for $\mathbf{x}_{n-1}$ within the range given by the upper and lower bounds, and proceeds to $\mathbf{x}_{n-2}$, and so on.

Eventually, one of two things happens: 1) the decoder reaches $\mathbf{x}_1$ and chooses a value within the computed range; 2) the decoder finds that no point in the constellation falls within the upper and lower bounds obtained for some $\mathbf{x}_i$. In the first case, the sphere decoder has a candidate solution for the entire vector $\mathbf{x}$, computes its radius (which cannot exceed r), and starts the search process over, using this new smaller radius to find any better candidates. In the second case, the decoder must have made at least one bad candidate choice for $\mathbf{x}_{i+1}, ..., \mathbf{x}_n$. The decoder revises the choice for $\mathbf{x}_{i+1}$ (which immediately preceded the attempt for $\mathbf{x}_i$) by finding another candidate value within its range, and proceeds again to try $\mathbf{x}_i$. If no more candidates are available at $\mathbf{x}_{i+1}$, the decoder backtracks to $\mathbf{x}_{i+2}$, and so on.

The performance of the algorithm is closely tied to the choice of the initial

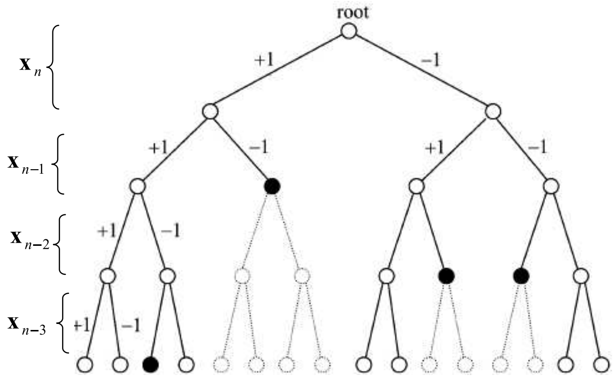


Figure 3.6: Example of the tree search in SD.

radius r. The radius should be chosen large enough so that the sphere contains the solution to (3.12). However, the larger r is chosen, the longer the search takes. If r is chosen too small, the algorithm could fail to find any point inside the sphere, requiring that r be increased. For good choices of r (we have more to say about how to choose r later), the algorithm appears to be roughly cubic in n. This is a vast improvement over an exhaustive search, which is exponential in n.

The SD adopts a tree search approach to obtain samples of $\mathbf{x}$. Figure 3.6 presents an example of the tree search adopted in [VH02]. The search starts from a root node and begins with examining possible choices of $\mathbf{x}_n$ that may satisfy (3.12). For each choice of $\mathbf{x}_n$, the possible choices of $\mathbf{x}_{n-1}$ are examined. The procedure continues for the rest of the elements of $\mathbf{x}$ similarly. Each choice of an element of $\mathbf{x}$ is indicated by a branch in the tree. Also, for convenience of demonstration, it is assumed that the elements of $\mathbf{x}$ are binary. The search collects samples of $\mathbf{x}$ by running through the branches of the tree from top to bottom and left to right. It is worth noting that many searches in the tree encounter nodes where no branches beyond them can lead to a point within the sphere, *i.e.*, satisfy (3.12). In Figure 3.6, these are indicated by bold nodes. We refer to these as terminal nodes.

3.4 Linear detector

The linear detector inverts the channel matrix and right-multiplies it by the received vector. This method separates the layered data streams, but also amplifies the Gaussian noise. The output of the multiplication is then sliced to the nearest symbol in the QAM constellation. The linear detector does not exploit the tree structure. Instead it detects each symbol independently, like n different tree searches of depth 1. To reduce complexity of ML detector, the constraints imposed on a feasible solution $x \in \{-1, +1\}^n$ can be relaxed. A simple constraint to impose is to restrict the solution vector to be contained within a closed convex set (CCS). Examples of CCSs of dimension n are $\mathbb{R}^n$ and ellipsoid of dimension n. The corresponding optimization problem is known as a CCS constrained quadratic program (CCSQP) defined as:

$$CCSQP : \arg \min_{\mathbf{x} \in \Omega^n} f(\mathbf{x}) \tag{3.19}$$

where the cost function is well defined in a convex set $\Omega^n \supset \{-1, +1\}^n$. In this work, we will concentrate on cases where the columns of the channel matrix $\mathbf{H}$ are linearly independent hence $\mathbf{H}^T\mathbf{H}$ is positive definite. In this case, the objective function $f(.)$ is strictly convex in $\mathbf{x}$ and has a well-defined unique minimizer over a convex set. Note that we require the constraint set for each relaxation to contain the feasible set of the original problem. The solution can then be mapped to the feasible set of the original problem by taking the sign of each component of the relaxed solution vector.

3.4.1 Zero Forcing detector

The fully unconstrained ML detector, $\Omega \equiv \mathbb{R}$, is denoted the decorrelating detector. Here, a valid solution vector, ρ_{zf} is found in $\mathbb{R}^n$ as each symbol estimate can take on any real value, *i.e.*, no constraints are imposed. The case is denoted an unconstrained quadratic program (UQP). This relaxation effectively removes the constraints and converts the discrete optimization problem into a continuous one. The UQP is:

$$\min_{\mathbf{x} \in \mathbb{R}^n} \mathbf{x}^T\mathbf{H}^T\mathbf{H}\mathbf{x} - 2\mathbf{y}^T\mathbf{H}\mathbf{x} \tag{3.20}$$

This relaxation effectively removes the constraints and converts the discrete optimization problem into a continuous one. Since the cost function is convex in its variable, this problem has a unique minimum

$$\rho_{zf} = \mathbf{x} + \mathbf{H}^{+}\mathbf{w} \tag{3.21}$$

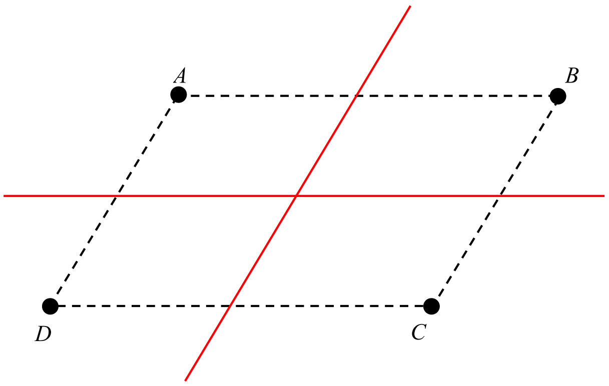


Figure 3.7: Hypotheses and decision quadrants in two dimensional case for decorrelating (ZF) detector.

Where $\mathbf{H}^+ = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ is the Moore-Penrose pseudo-inverse of the real channel matrix $\mathbf{H}$. The discrete feasible solution $\hat{\mathbf{x}}_{zf} = sign(\rho_{zf}) \in \mathcal{A}^n$ that is closest in the l_2 norm to ρ_{zf} is returned.

To illustrate the decorrelating detector we draw the decision regions in the two dimensional case. In the domain $\mathbf{H}^+\mathbf{y}$ where the noise is spherically symmetric and Gaussian the decision regions are given by perpendicular bisections of the segments between the different hypotheses denoted with A, B, C and D. In three (or more) dimensional case the decision region are cones with vertices at origin. The mapping with the decorrelating detector matrix transforms the channel output to one quadrants. The final selection is performed then with the $sign$ function.

3.4.2 Generalized MMSE detector

The constraint on each $\mathbf{x}_i \in \{-1, 1\}$ is equivalent to $\mathbf{x}_i^2 = 1$ which implies $\mathbf{x}^T\mathbf{x} = n$ at any feasible point. Although the last constraint is nonconvex, a relaxation of the form $\mathbf{x}^T\mathbf{x} \leq n$ results in a convex set. The estimate $\hat{\mathbf{x}}$ is the solution of the optimization problem

$$\min_{\mathbf{x}^T\mathbf{x} \leq n} \mathbf{x}^T\mathbf{H}^T\mathbf{H}\mathbf{x} - 2\mathbf{y}^T\mathbf{H}\mathbf{x} \tag{3.22}$$

the convex set $\|\mathbf{x}\|^2 \leq n$ can be thought of as the interior of an n-dimensional hypersphere of radius $\sqrt{n}$. The solution of the above problem, derived in [YYU99] in the context of linear modulation, is the generalized MMSE detector

$$\rho_{MMSE} = (\mathbf{H}^T\mathbf{H} + \lambda^*\mathbf{I})^{-1}\mathbf{H}\mathbf{y} \tag{3.23}$$

where λ^* is the optimum Lagrange multiplier corresponding to the global constraint (4.4). Note that (4.5) reduces to the MMSE solution [MH94] for $\lambda^* = \sigma^2$.

3.5 Interference cancellation detectors

The idea of interference cancellation has been mainly applied to the cancellation of MAI in MC-CDMA systems [Bau01]. However, the same principle could be applied to detect the transmitted signal $\mathbf{x}$ from $\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w}$. This basic approach generally requires multiple stages to recursively refine the mitigation of the interference. Detectors based on this approach generally employ one of the following two main interference cancellation methods.

3.5.1 Successive Interference Canceller

The key idea in a SIC based receiver is serial cancellation of the ISI, where the individual data streams are successively decoded and stripped away layer-by-layer. The algorithm starts by detecting the symbol (*e.g.*, using ZF or MMSE) of an arbitrarily chosen layer, assuming that the other symbols from the remaining layers are interference. Upon detection of the chosen symbol, its contribution from the received signal vector is subtracted and the procedure is repeated until all symbols are detected. In practice, error propagation will be encountered, especially in the absence of an adequate temporal coding for each layer. The error rate performance will therefore be dominated by the first stream decoded by the receiver (which is also the stream experiencing the smallest diversity order). An improved SIC processor is obtained by selecting the stream with the highest Signal to Interference plus Noise Ratio (SINR) at each decoding stage. Such receivers are known as Ordered SIC (OSIC) receivers or in the case of MIMO literature as V-BLAST detectors [WFGV98], because they have been used successfully for BLAST architectures. OSIC receivers reduce the probability of error propagation by realising a selection diversity gain at each decoding step. The OSIC algorithm requires slightly higher complexity than the SIC algorithm resulting from the need to compute and compare the SINRs of the remaining streams at each stage.

A major problem with the SIC or the OSIC methods for ISI cancellation is the delay inherent in the implementation of the canceller, since it requires one symbol delay per layer [PH93]. This problem may be alleviated to some extent by devising methods that perform interference cancellation in parallel [Mos96].

The idea of feeding back decisions to mitigate the effects of interference for future symbols was first used by Austin [Aus67] in the context of ISI channels. Duel-Hallen [Hal93] introduced the idea to CDMA systems, while Foschini [Fos96] brought the idea to multiple antenna systems via the Bell Labs Layered Space-Time (BLAST) detection algorithm.

3.5.2 Parallel Interference Canceller

In SIC receivers, the interference estimates are created and removed from the received signal before making decisions on the transmitted symbol estimates. In detectors based on PICs, the interference estimation and the cancellation process are executed simultaneously at each stage for all the layers. In [PH94], it was shown for an asynchronous Direct Sequence CDMA (DS-CDMA) that SIC receivers are superior to PIC receivers in a Rayleigh fading channel without power control. PIC based detectors, on the other hand, exhibit better performance under ideal power control. This is not surprising, since the parallel scheme treats all the users fairly and simultaneously. Therefore, if all the users' powers at the receiver are the same, they all experience the same amount of interference. When dealing with point to point MIMO architectures, it is plausible to assume that the signal transmitted from different antennas have a similar power at the receiver, or alternatively power control techniques can be easily employed, since all the symbols are transmitted by one user only.

The original work on PICs in [PH94] employs standard detection techniques at each stage, such as MF or ZF detectors, to estimate the MAI. The interference was then simultaneously removed from all the users for the next stage. Between two stages, demodulation of the users' data and hard decision were performed to regenerate the MAI. Decoding was only performed at the last stage of the cancellation process, if a channel encoder was used. As previously mentioned, this might lead to decision errors at each stage, thus decreasing the reliability of the estimated interference and hence the overall performance of the receiver. If a channel encoder is employed, at each stage, it is possible to perform hard output decoding or soft output decoding of the codewords to obtain a more reliable estimate of the MAI and hence reduce the error propagation. This results in a greater interference cancellation and better performance [MVU01]. However, if decoding is required at each stage, not only is a longer processing delay introduced in the system, but also hardware complexity is increased, as it is necessary to replicate the detector and the decoder as many times as the number of the receiver's stages. Therefore, in an iterative implementation of PIC receivers where the output of the decision device at each iteration is fed back to the PIC for the following iteration, we would require only one realization of the detection chain at the expenses of a longer processing delay.

3.6 The Semidefinite Relaxation

The semidefinite relaxation (SDR) approach to detection for the linear channel was originally introduced to the area of digital communications in [MDW+02]. The underlying philosophy of the SDR algorithm is, instead of solving the computationally complex ML detection problem, to solve a simpler problem. The value of this is that by carefully selecting the simplified problem its solution will correspond to the true ML solution with a high degree of accuracy. Although not as widely adapted by the community as the sphere decoder the SDR algorithm has been successfully applied to various detection problems in a number of publications. As in [TR01] the algorithm has been further considered in the CDMA context in [WLA03, MDWC04]. In a few cases the algorithm has also been investigated in other contexts. In for example [SLW03] the SDR algorithm is used as an inner decoder in a system employing a concatenated coding scheme consisting of an inner space-time block-code (STBC) and an outer turbo code. In this scenario the SDR algorithm is used to obtain soft information from the inner STBC which is subsequently passed to the outer turbo decoder.

Previously it has been shown that the SDR decoder, where it is applicable, has better performance than a class of commonly used suboptimal detectors [MDW+02]. This is accomplished by showing that the SDR represents a relaxation of the original ML detection problem and that the class of detectors under consideration represent further relaxations of the SDR.

Consider again the optimization problem of (3.4), *i.e.*

$$\hat{x} = \arg \min_{\mathbf{x} \in \{-1,+1\}^n} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \tag{3.24}$$

where H is a $m \times n$ real valued channel matrix. Note that the case of a complex channel with a QPSK constellation can also be written on this form by doubling the dimension of the original problem. The semidefinite relaxation algorithm attempts to approximate the solution of (3.4) by forming a convex problem which has the property that the solution thereof serves as an estimate for the solution to (3.4).

To form this convex problem note that the criterion function of (3.4) can be written as

$$\|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 = \mathbf{x}^T\mathbf{H}^T\mathbf{H}\mathbf{x} - 2\mathbf{y}^T\mathbf{H}\mathbf{x} + \mathbf{y}^T\mathbf{y}$$

Thus $\hat{\mathbf{x}}$ can equivalently be obtained through

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \{-1,+1\}^n} \mathbf{x}^T\mathbf{H}^T\mathbf{H}\mathbf{x} - 2\mathbf{y}^T\mathbf{H}\mathbf{x} \tag{3.25}$$

since $\mathbf{y}^T\mathbf{y}$ does not depend on $\mathbf{x}$. The solution to the above problem (3.25) requires a search over all possible combinations of the components of the vector $\mathbf{x}$. It is thus clear that the computational complexity increases exponentially with n.

The technique of semidefinite programming has a potential to reduce computational complexity without sacrificing performance. Let us start with the problem in (3.25) which we reformulate as [HR98]

$$\hat{\mathbf{s}} = \arg \min_{\mathbf{s} \in \{-1,+1\}^{n+1}} \mathbf{s}^T\mathbf{L}\mathbf{s} \quad , \mathbf{L} = \begin{bmatrix} \mathbf{H}^T\mathbf{H} & -\mathbf{H}^T\mathbf{y} \\ -\mathbf{y}^T\mathbf{H} & 0 \end{bmatrix} \tag{3.26}$$

For $\mathbf{s} \in \{-1, 1\}^{n+1}$ the matrix $\mathbf{s}\mathbf{s}^T$ is positive semidefinite, its diagonal entries are equal to 1, and it is a rank one matrix [HR98]. Now let $\mathbf{S} = \mathbf{s}\mathbf{s}^T$ be a matrix which satisfies these three characteristic properties. Then we can restate (3.26) as

$$\hat{\mathbf{S}} = \arg \min_{\mathbf{S}} \mathbf{L}\mathbf{S} \quad , diag(\mathbf{S}) = \mathbf{e}_1 \quad , rank(\mathbf{S}) = 1 \quad , \mathbf{S} \succeq 0 \tag{3.27}$$

where $\mathbf{e}_1$ is an all ones vector of length $n + 1$. Dropping the rank one constraint yields the basic semidefinite relaxation [HR98] of (3.27).

$$\hat{\mathbf{S}_1} = \arg \min_{\mathbf{S}} tr(\mathbf{L}\mathbf{S}) \quad , diag(\mathbf{S}) = \mathbf{e}_1 \quad , \mathbf{S} \succeq 0 \tag{3.28}$$

This is known as a semidefinite program in the matrix variable $\mathbf{S}$, because it is a linear problem in $\mathbf{S}$ with the additional semidefiniteness constraint $\mathbf{S} \succeq 0$.

To tighten the relaxation, we introduce cutting planes which have become a standard technique for solving combinatorial optimization problems through semidefinite programs [HR98]. An optimal solution to the relaxed problem in (3.28) is computed iteratively. If the solution is not feasible for the original problem in (3.27), the feasible region for (3.28) is reduced so that the current solution is no longer feasible. This is done by finding inequalities that are valid for the original problem in (3.27) but exclude the current point from the feasible region. The goal is to approximate the solution to (3.27) by using a tightest possible continuous relaxation of the feasible set of integral points. Clearly, we need to keep the number of valid linear inequalities for the feasible set, also called the cutting planes, as small as possible to limit the computational complexity.

A semidefinite program (3.28) can be solved by employing the primal-dual path-following algorithm of [HRVW96] as a basic optimization tool. We solve (3.28) using this interior-point method and then use the procedures described in [HR98] for including the cutting planes.

Since $\hat{\mathbf{S}_1} \neq \hat{\mathbf{s}}\hat{\mathbf{s}}^T$, the last thing we have to do is to approximate $\hat{\mathbf{s}}$ from $\hat{\mathbf{S}_1}$. One way is to assume that $\hat{\mathbf{S}_1}$ has rank one and let $\hat{\mathbf{s}}$ to be the sign of the last column of the $\hat{\mathbf{S}_1}$. Another way is to select $\hat{\mathbf{s}}$ to be the sign of the eigenvector corresponding to the largest eigenvalue of $\hat{\mathbf{S}_1}$.

3.7 Performances and Complexity Evaluation

In this section all optimum or sub-optimal detectors presented in the previous section are evaluated numerically by simulations. The data is generated according to a real valued version of the *i.i.d.* Rayleigh fading channel. Also, by considering such a simple scenario the result can easily be reproduced. For all algorithms, the inputs are considered to be $\mathbf{H}$ and $\mathbf{y}$ and the output is the estimate, $\hat{\mathbf{x}}$.

3.7.1 Data Model

In order to create scenarios where all previously detectors can be compared, we restrict the study to the case where the constellation is binary, *i.e.* $\mathcal{A} = \{\pm 1\}$. The channel matrix and noise are also real valued and the problem instances are generated according to

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{3.29}$$

where $\mathbf{H} \in \mathbb{R}^{m\times n}$ is referred to as the channel matrix and $\mathbf{w} \in \mathbb{R}^m$ is the additive noise. The vectors $\mathbf{y} \in \mathbb{R}^m$ and $\mathbf{x} \in \{-1,+1\}^n$ are the received signals and transmitted symbols respectively.

The transmitted symbols are modeled as *i.i.d.* random variables which are uniformly distributed over the constellation alphabet, $\mathcal{A}$. It is assumed that the constellation is centered at zero, under the *i.i.d* assumption on $\mathbf{x}$. Note that this occurs at no loss of generality since any nonzero mean of $\mathbf{x}$, and consequently $\mathbf{y}$, may be removed prior to detection without any loss in performance. Also, designing a communications system with a nonzero mean would in most scenarios require an increased transmit power and thus, the assumption is usually satisfied in practice as well.

The noise is modeled as zero mean, circularly symmetric real Gaussian, with variance σ^2 and is assumed uncorrelated between components. The Gaussian assumption is usually motivated by the notion that the noise is made up of several contributing components and is thus approximately Gaussian due to the central limit theorem. The ability to correctly detect the transmitted symbols, $\mathbf{x}$, is affected by the ratio of the signal strength and the noise power. This ratio is called the signal to noise ratio (SNR) and is herein defined as

$$\gamma \triangleq \frac{\varepsilon\{\|\mathbf{Hx}\|^2\}}{\varepsilon\{\|\Pi_{\mathbf{H}}\mathbf{w}\|^2\}} \tag{3.30}$$

where $\Pi_{\mathbf{H}} \triangleq \mathbf{H}(\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$ is the projection matrix for the projection onto the space spanned by the columns of $\mathbf{H}$. The reason for including the projection of the noise onto the columns of $\mathbf{H}$ is that the part of the noise orthogonal to $\mathbf{H}$ does not affect the ability to correctly detect $\mathbf{x}$ and is thus irrelevant to the detection problem. Note that in the special case where $m = n$ and where $\mathbf{H}$ is full rank with probability 1, the equation (3.30) can be reduced to the following more familiar expression

$$\gamma = \frac{\varepsilon\{\|\mathbf{Hx}\|^2\}}{\varepsilon\{\|\mathbf{w}\|^2\}} \tag{3.31}$$

where $\|.\|$ is the l^2-norm. Let $\mathbf{s} = \mathbf{Hx}$; then,

$$\begin{aligned}
\varepsilon\{\|\mathbf{s}\|^2\} &= \varepsilon\{\|\mathbf{s}\|^2\} \\
&= \varepsilon\{\sum_{k=1}^{m} \|\mathbf{s}_k\|^2\} \\
&= \varepsilon\{\sum_{k=1}^{m} sum_{j=1}^{n} \|\mathbf{h}_{kj}\mathbf{x}_j\|^2\} \\
&= \sum_{k=1}^{m}\sum_{j=1}^{n} \varepsilon\{\|\mathbf{h}_{kj}\|^2\}\varepsilon\{\|\mathbf{x}_j\|^2\} \\
&= E_{\mathbf{x}} \cdot n \cdot m
\end{aligned} \tag{3.32}$$

Recall that the expected value of each element of $\mathbf{x}$ is $E_{\mathbf{x}}$, and that of the elements of $\mathbf{H}$ is 1. Given that $\varepsilon\{\|\mathbf{w}\|^2\} = 2 \cdot m \cdot \sigma^2/2$, substituting (3.32) into (3.31), the average SNR can be written as

$$\gamma = \frac{E_s \cdot n}{\sigma^2} \tag{3.33}$$

The factor 2 in the average power of the noise appears because the real and imaginary parts of $\tilde{\mathbf{w}}$ each have variance $\sigma^2/2$; consequently, the variance of each component of $\mathbf{w}$ is σ^2.

3.7.2 Complexity study

In this section we compare the computational complexity of the various detection schemes that have been considered. We assume that transmission is done in *bursts*. For these bursts we assume the channel matrix $\mathbf{H}$ to be constant, and hence the matrices used in receiver processing need only be calculated once per burst. In high-speed transmission systems burst lengths of several thousands of symbols are possible.

Consider the linear solutions where the estimate of $\hat{\mathbf{x}}$ is returned as $sgn(\mathbf{My})$. This process requires the construction of $\mathbf{M}$ then the application of $\mathbf{M}$ to $\mathbf{y}$. The sign testing is ignored since its complexity is negligible. The complexity of the filter $\mathbf{M}$ varies slightly depending on the particular linear detector under study. The determination $\mathbf{M}$ has a complexity budget as shown in Table 3.1

The scenario of the ML detector is a little different. The minimization of the

Table 3.1: Complexity computation of ZF and MMSE

step	Procedure	ZF *flops*	MMSE *flops*
pre-processing	$\mathbf{H}^T\mathbf{H}$	$2m^2n$	$2m^2n$
pre-processing	$(. + \sigma^2\mathbf{I})$	0	$2n^2$
pre-processing	$(.)^{-1}$	n^3	n^3
pre-processing	$(.)^{-1}\mathbf{H}^T$	$2n^2m$	$2n^2m$
processing	$(.)^{-1}\mathbf{H}^T\mathbf{y}$	$2mn$	$2mn$

objective function $f : \mathbf{x} \mapsto \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2$ over $\mathbf{x} \in \{-1, +1\}^n$ requires the computation of 2^n metrics. The complexity budget for each metric computation is as shown in Table 3.2. Since the ML detector requires the computation of 2^n of these metrics the total complexity for the detection is $2^n m(2n + 3)$[1]. The detection process is to select the vector $\mathbf{x}$ that minimize the objective function. The complexity of linear detector is polynomial in n, $\mathcal{O}(n^3)$, whereas the complexity of the exhaustive search (ML) is exponential in n.

The main computation in using the iterative nulling and cancellation algorithm is the determination of the optimal ordering of the nulling and cancellation steps, and the computation of the corresponding nulling vectors. These steps have computational complexity of order $\mathcal{O}(n^4)$. When the number n is large the repeated use of the pseudo-inverse to calculate the nulling vectors may lead to numerical instability. Nevertheless, a square-root algorithm based on QR decomposition of the channel matrix and unitary transformations is used in [Has00] to avoid the repeated computation of the nulling vectors. Instead, the QR decomposition is computed only once. Not only is computation complexity reduced $\mathcal{O}(n^3)$, but also the numerical robustness is improved by this square-root algorithm.

For Branch And Bound Algorithm and the Sphere Decoder, the number of iterations of the processing of the search of the optimal point is random. So we run simulations to estimate the complexity in this step. Both algorithms carry out some operations that can call pre-processing step, the complexity budget for this step is shown in Table 3.3 For estimation of the complexity of the processing step, we compare the efficiencies of the Branch and Bound Detector (BBD) and

[1] This amount can be reduced by storing computation between different evaluations. For example, if $\mathbf{x}_1$ and $\mathbf{x}_2$ differs only from one coordinate, the computation of $\mathbf{H}\mathbf{x}_2$ can be reduced using partial result of the computation of $\mathbf{H}\mathbf{x}_1$. Nevertheless, the computation complexity is still $\mathcal{O}(2^n)$

Table 3.2: Complexity computation of ML

step	Procedure	ML $flops$
pre-processing	$\mathbf{Hx}$	$2mn$
processing	$\mathbf{y} - \mathbf{Hx}$	m
processing	$\|.\|^2$	$2m$

Table 3.3: Complexity computation of pre-processing step for SD and BBD

Procedure	SD $flops$	BBD $flops$
$\mathbf{G} = \mathbf{H}^T\mathbf{H}$	$\frac{n^3}{2} + \frac{n^2}{2}$	$\frac{n^3}{2} + \frac{n^2}{2}$
$\mathbf{G} = \mathbf{L}\mathbf{L}^T$	$\frac{n^3}{6}$	$\frac{n^3}{6}$
$compute\ l_{ij}$	$\frac{n^2}{2} + \frac{n}{2}$	0
$(.)^{-1}$	n^3	$\frac{n^3}{2}$
$Total$	$\frac{5n^3}{3} + \frac{n^2}{2} + \frac{n}{2}$	$\frac{7n^3}{6} + \frac{n^2}{2}$

the Sphere Decoding (SD) algorithms in terms of the number of floating-point operations (Addition / Multiplication) versus the signal to noise ratio in the worst-case. In fact, for hardware implementation any detector would be designed for the worst-case complexity rather than the average complexity.

As shown previously in this chapter, the BBD and SD methods provide the optimum detection performance. Nevertheless, their worst-case complexities grow exponentially in the number n [JO05]. The result of the comparison of the processing complexity of the SD and BB algorithms is given in figure 3.8. We can see that the Branch and Bound method have a lower complexity compared to the Sphere Decoder when an uncoded 4-QAM modulation is used .

The complexity of the Semidefinite Relaxation (SDR), called also Semidefinite programming (SDP), algorithm based on interior-point [HRVW96] methods is determined by two factors:

- The number of iterations required to achieve the stopping criterion. It has been shown [HRVW96] that given a solution accuracy ϵ, the worst-case number of iterations required to satisfy the stopping criterion is of the order of $\sqrt{2n}log(1/\epsilon)$
- The operational cost at one iteration. It has been shown that the computational complexity per iteration is $\mathcal{O}(n^3)$

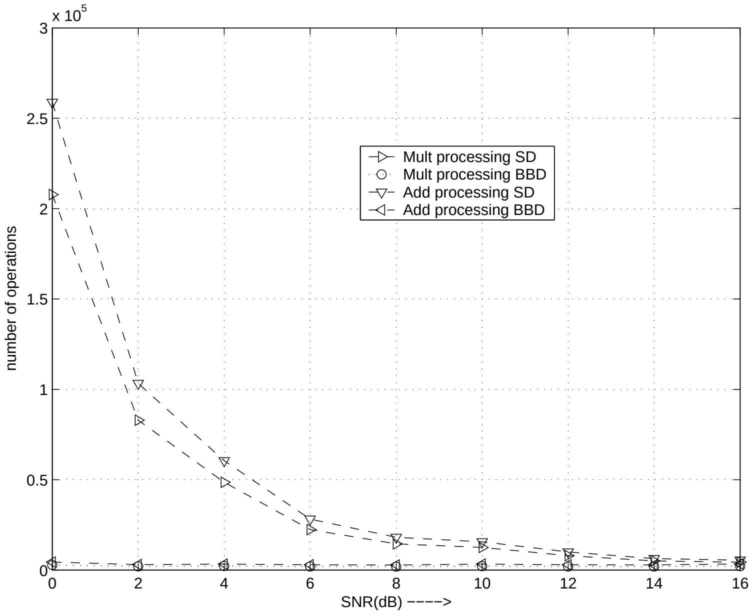


Figure 3.8: Computation complexity comparison of the processing step (SD and BBD) with $n = m = 10$ and 4-QAM modulation in the worst-case, $r = \sqrt{\alpha n}\sigma$ where α is chosen so that [HV02].

The SD relaxation detector is efficient in that its complexity is of the order of $\mathcal{O}(n^{3.5})$, where n is the dimension of the ML detection problem.

3.7.3 Performance

In Rayleigh fading, the average error probability Bit Error Rate BER decays according to $BER \sim 1/SNR^{\nu}$ at high SNR, where ν is the diversity order and reflects the system's tolerance of and robustness toward channel fading. The bit error rate performance of the linear detection scheme is shown in figure 3.9. Notes that the maximum likelihood detector (MLD), the branch and bound detector (BBD) and the sphere decoding (SD) have the same performances.

We have applied the SDP method to a system sending symbols in $\{-1, 1\}$ for different values of the lattice dimension $n = m$ $(n = 6\, and\, 10)$. The results are on figure 3.10(a). For the same system a comparison of the performance of the four algorithms is done in the case of a lattice dimension $n = 12$. The result of this comparison is on figure 3.10(b). We can see that the SD and BBD algorithms outperform the SDP and VBLAST methods. Moreover, they have exactly the same performances.

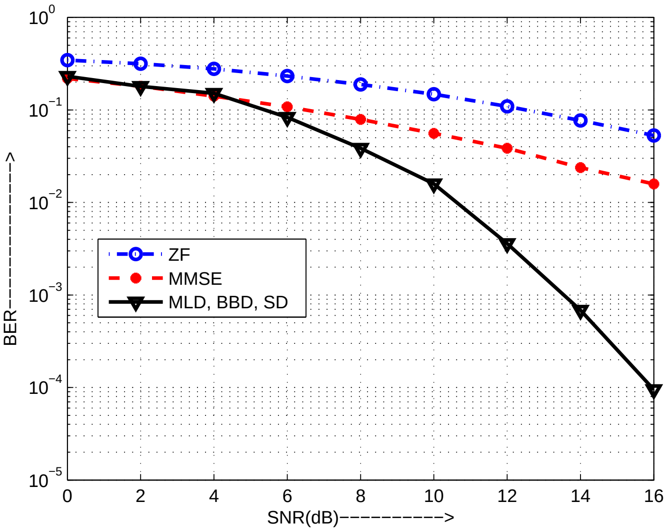
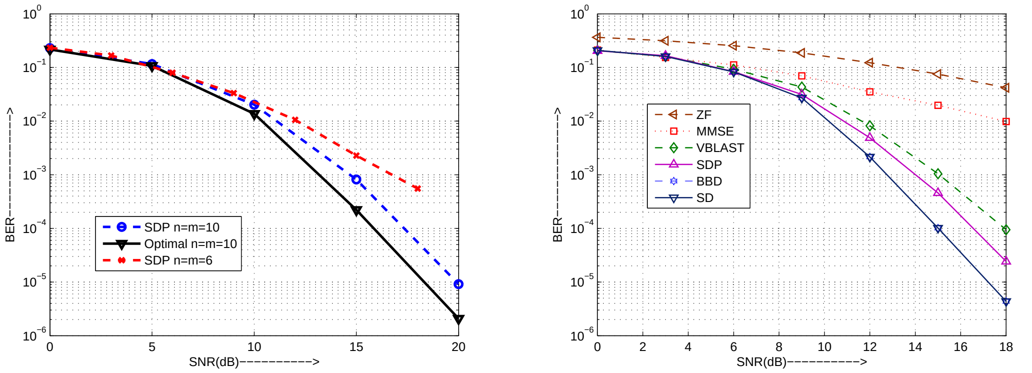


Figure 3.9: Average bit error rates achieved for linear zero-forcing (ZF) and linear minimum-mean-square error (MMSE) $N = M = 5$, $4 - QAM$ uncoded system.



(a) Performances of the SDP algorithm where $n = m = 10$

(b) Performances of the different algorithms (SDP, VBLAST, SD and BB) where $n = m = 12$

Figure 3.10: Performance comparison on Rayleigh fading channel for an uncoded systems using $4 - QAM$ modulation

3.8 Conclusion

In this chapter, most popular detectors schemes were introduced. We characterized the performance and the complexity of these receivers, such as ZF, MMSE, SDP, SD and BBD for Rayleigh flat fading channels assuming perfect estimation in the receiver. These results are all well known from the previous literature. A major problem in obtaining an efficient implementation of any detection algorithm is their inherent sequential structure. Our goal here was to establish a performance baseline so that these results could be compared with performance of the proposed detection algorithm in the next chapter.

Chapter 4

Proposed detection method

The detection fundamentals are presented in the previous chapter. The ML detection, which is an $NP - hard$ problem, can be implemented using a "smart" efficient search algorithm, *e.g.*, the Sphere Decoding (SD) [VB93, VB99] at a reasonable average computational cost. Unfortunately, the SD is not well suited for VLSI implementation. Consequently, there has been much interest in implementing suboptimal detection algorithms. The most fastest suboptimum detectors having in general bad performance are given by the linear receivers (ZF and MMSE). A class of nonlinear detectors that offer better performance with only a modest increase in complexity is based on successive cancellation [Bau01]. Recently semidefinite programming (SDP) approach has been shown to be a promising approach to combinatorial problems [MDW+02]. However, the main problem concerning the hardware implementation of the existing decoding algorithms is the lack of parallelism caused by their iterative structure.

This chapter introduces a new detection algorithm based on a geometrical approach to detect the transmitted signal $\mathbf{x} \in \xi \triangleq \{-1, +1\}^n$ from $\mathbf{y} = \mathbf{Hx} + \mathbf{w}$, given the received vector $\mathbf{y} \in \mathbb{R}^m$ and the channel matrix $\mathbf{H} \in \mathbb{R}^{m \times n}$, $m \geq n$. The proposed detection algorithm can achieve near-optimum performance while its implementation complexity is $\mathcal{O}(n^3)$, where n is the dimension of search space. The new proposed algorithm called Geometrical Intersection and Selection Detector (GISD), associates a powerful geometrical treatment to the classical optimization process, *i.e.*, $intensification$ and $diversification$. The intensification means that, starting from a given solution $\mathbf{x}^0 \in \xi$, a local search of the potential better solutions in the neighborhood of $\mathbf{x}^0$ is performed. However, the diversification is a method to select efficiently subset $\xi_{start} \subset \xi$, of starting solution, which allows us to escape from potential local minima.

The strategy to select the subset ξ_{start} is based on a geometrical approach inspired by the very original work of H Artes and D Seethaler [ASH03]. The fundamental difference is that computation in the proposed algorithm is applied to the real domain, instead of the complex domain, and this strongly reduces the complexity. A patent [NB05] based on our strategy detection, which focuses on intensification/diversification approach to reduce ML detection complexity, has been published. After a submission of an IEEE Communications letter of our study, one of reviewers mentionned that the Canonical Basis intersection and selection detector CBISD, variant of the proposed method, has been earlier published in paper [SG01].

In this chapter, the proposed geometrical approach detection method are developed. Section 4.1 introduces the singular value decomposition properties for a given matrix **A**. Also, we present the distribution of the condition number of real i.i.d random channel matrix, and discuss the effect of this number on the linear detector methods. The intensification step of the GISD decoder is given in section 4.2, and exact complexity analysis of this step is done. In section 4.3 different methods of the diversification step are considered including repeated random start, Bose-ChaudhurI-Hocquenghem (BCH) and geometrical approach methods. Section 4.4 gives the flow chart of GISD detector including all variants. Moreover, an example is presented to illustrate the efficacty and simplicity of the proposed detection method. The computational complexity and the performance of the GISD are given in sections 4.5 and 4.6. The chapter ends with some simulation results showing the impact of channel estimation errors on the performance of the proposed GISD detector.

4.1 Singular Value Decomposition

The aim of this section is to collect the basic information needed to understand the Singular Value Decomposition (SVD) as used throughout this thesis. We start giving the definition of SVD for a generic, rectangular matrix **A** and discussing some related concepts.

4.1.1 Definition

Any $m \times n$ matrix $\mathbf{A}$ can be written as the product of three matrices

$$\mathbf{A} = \mathbf{UDV}^T \tag{4.1}$$

The columns of the $m \times m$ matrix $\mathbf{U}$ are mutually orthogonal unit vectors, as are the columns of the $n \times n$ matrix $\mathbf{V}$. The $m \times n$ matrix $\mathbf{D}$ is diagonal; its diagonal elements, λ_i, called singular values, are such that $\lambda_1 \geq \lambda_2 \geq .... \geq \lambda_n \geq 0$.

Some important properties now follow.

4.1.2 Properties of the SVD

- *Property 1*: The singular values give valuable information on the singularity of a square matrix $\mathbf{A}$. The matrix $\mathbf{A}$ is nonsingular if and only if all its singular values are different from zero. Most importantly, the ratio

  $$Cond(\mathbf{A}) = \frac{\lambda_1}{\lambda_n} \tag{4.2}$$

  called *condition number*, measures the degree of singularity of $\mathbf{A}$. When $1/Cond(\mathbf{A})$ is comparable with the arithmetic precision of machine, the matrix $\mathbf{A}$ is ill-conditioned and, for all practical purposes, can be considered singular.

- *Property 2*: If $\mathbf{A}$ is a rectangular matrix, the number of nonzero λ_i, equals the rank of $\mathbf{A}$.

- *Property 3*: If $\mathbf{A}$ is a square, nonsingular matrix, its inverse can be written as

  $$\mathbf{A}^{-1} = \mathbf{VD}^{-1}\mathbf{U}^T$$

  Be $\mathbf{A}$ singular or not, the pseudoinverse of $\mathbf{A}$, $\mathbf{A}^+$, can be written as

  $$\mathbf{A}^+ = \mathbf{VD}_0^{-1}\mathbf{U}^T$$

  with $\mathbf{D}_0^{-1}$ equals to $\mathbf{D}^{-1}$ for all nonzero singular values and zero otherwise. If $\mathbf{A}$ is nonsingular, then $\mathbf{D}_0^{-1} = \mathbf{D}^{-1}$ and $\mathbf{A}^+ = \mathbf{A}^{-1}$.

- *Property 4*: The columns of $\mathbf{U}$ are the left singular vectors corresponding to the nonzero singular values of $\mathbf{A}$, and form an orthogonal basis for the range of $\mathbf{A}$. The columns of $\mathbf{V}$ are the right singular vectors corresponding to the nonzero singular values of $\mathbf{A}$, and are each orthogonal to the null space of $\mathbf{A}$.

- *Property 5*: The squares of the nonzero singular values are the nonzero eigenvalues of the $n \times n$ matrix $\mathbf{A}^T\mathbf{A}$ and $m \times m$ matrix $\mathbf{A}\mathbf{A}^T$. The columns of $\mathbf{U}$ are eigenvectors of $\mathbf{A}\mathbf{A}^T$, the columns of $\mathbf{V}$ are eigenvectors of $\mathbf{A}^T\mathbf{A}$. Moreover, $\mathbf{A}\mathbf{u}_k = \lambda_k \mathbf{v}_k$ and $\mathbf{A}^T\mathbf{v}_k = \lambda_k \mathbf{u}_k$ where $\mathbf{u}_k$ and $\mathbf{v}_k$ are the columns of $\mathbf{U}$ and $\mathbf{V}$ corresponding to λ_k.

- *Property 6*: One possible distance measure between matrices can use the $Frobenius\ norm$. The Frobenius norm of a matrix $\mathbf{A}$ is simply the sum of the squares of the entries a_{ij} of $\mathbf{A}$, or

$$\|\mathbf{A}\|_F = \sqrt{\sum_{i,j} |a_{ij}|^2} = \sum_i |\lambda_i|^2$$

4.1.3 Detection performances and ill-conditioned channel occurs

Ill-conditioned components present in the channel effectively increase the linear dependence of the input streams and makes stream separation and decoding a difficult task. For example, current schemes like space multiplexing (V-BLAST) literally break down in the presence of correlation levels close to one or high Ricean factors. The degradation in performance can be attributed to the sensitivity of the MMSE suppression algorithm to rank and conditioning of the channel matrix. When the correlation between various paths in the channel increases, the condition number of the channel matrix also increases. As the matrix becomes "more singular" (elements in the matrix become more correlated) *cond*($\mathbf{A}$) approaches infinity. The distribution of *cond*($\mathbf{A}$) for a random channel is shown in Figure 4.1. In this plot the distribution of *cond*($\mathbf{A}$) is plotted as variation in the magnitude of *cond*($\mathbf{A}$).

The impact of the condition number behavior on the average bit error rate (BER) performance of optimal and sub-optimal detection depends on the probability with which Ill-conditioned ($\lambda_1 >>> \lambda_n$) "*bad*" channels occur. The figure 4.2 shows the cumulative distribution function (cdf) of *cond*(A). It is seen that

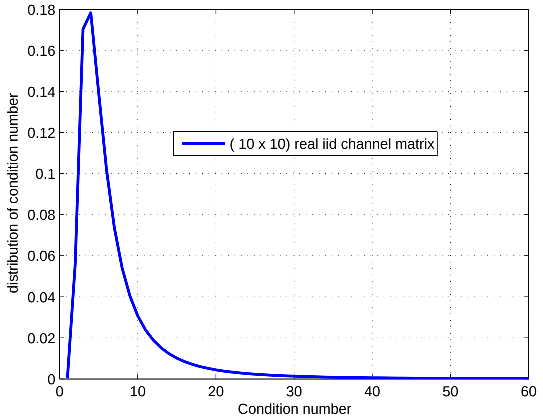


Figure 4.1: Distribution of the condition number for a (10 × 10) real iid random channel.

the probability that $cond(A)$ exceeds a value of 10 and 20 is 16% and 5%, respectively. However, the probability that $cond(A)$ is less than or equal to 8 is 77%.

The figure 4.2 suggests that bad channels occur frequently enough to cause a significant degradation of the average performance of suboptimal detection schemes. In fact, the noise enhancement of MMSE or ZF algorithms increases when $cond(\mathbf{A})$ is large, and this degrades the performance of BLAST and dramatically increases computational complexity of the sphere decoding algorithm. Experiments suggest that the performance of suboptimal detection schemes strongly depends on the channel's condition number. In figure 4.3, we show the bit error rate (BER) of various detection schemes versus the condition number of the two experimental channel realizations (channel $C1$: when $cond(\mathbf{A}) < 8$ and channel $C2$: when $cond(\mathbf{A}) > 10$). In this simulation, we used a 10×10 channel with independent and identically distributed Gaussian channel matrix entries, 4-QAM modulation. It can be seen that there is a significant performance gap between linear detection in the two cases of channel $C1$ and channel $C2$. While the performance of ML detection is fairly robust to bad channel realizations, it is note worthy that the computational complexity of the Sphere Decoding algorithm for ML detection significantly increase for these channels. Thus, there is a strong demand for computationally efficient suboptimal detectors that are able to achieve near-ML performance.

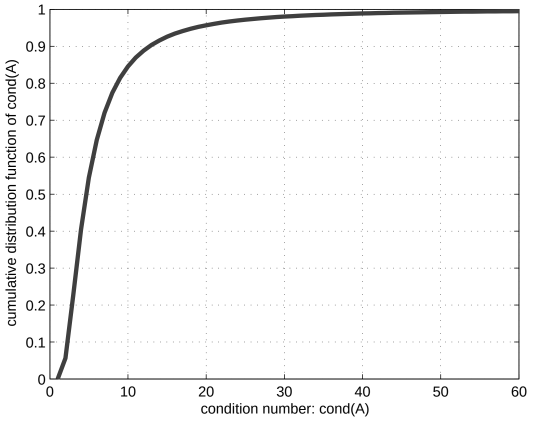
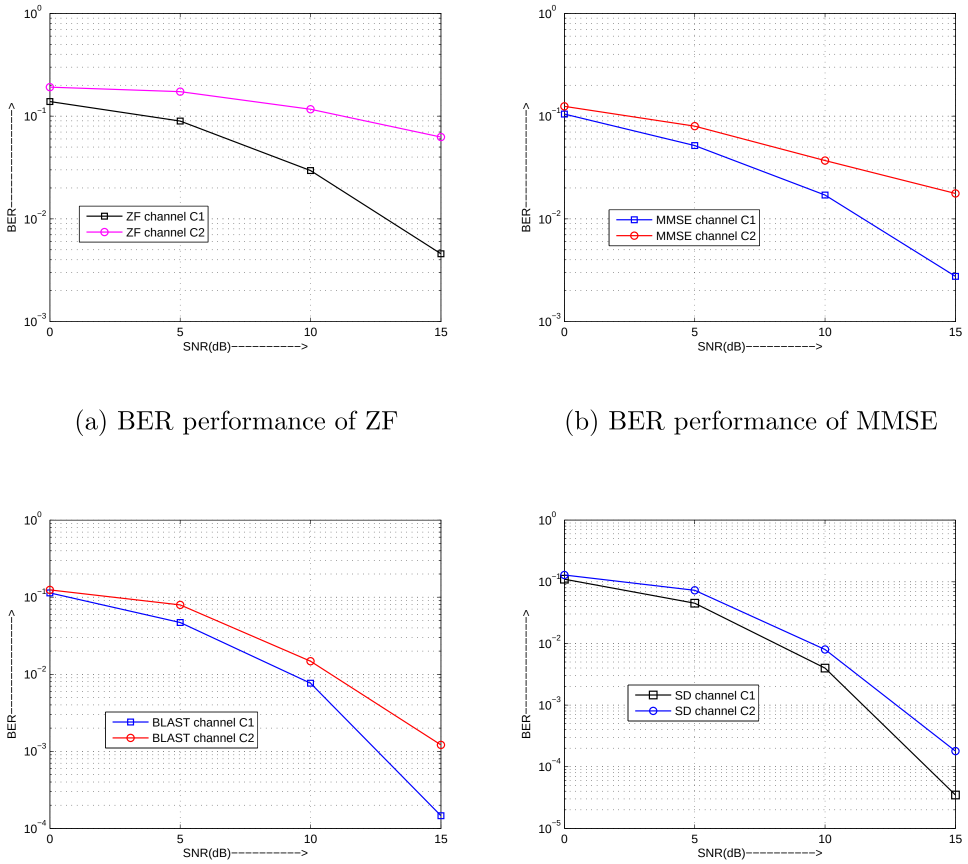


Figure 4.2: Cumulative distribution function of the condition number for (10×10) real iid random channel.



(a) BER performance of ZF

(b) BER performance of MMSE

(c) BER performance of BLAST

(d) BER performance of SD

Figure 4.3: Performances of detectors in presence of channel $C1$ or channel $C2$.

4.1.4 Geometrical interpretation of ill-conditioned channel

The starting point for developing the proposed detection methods in this thesis is a geometrical analysis of the decision regions of zero forcing detection methods in the case of ill-conditionned channels. In fact, after ZF detection, we obtain

$$\rho_{zf} = \mathbf{H}^{+}\mathbf{y} = \mathbf{x} + \mathbf{H}^{+}\mathbf{w} \tag{4.3}$$

Where $\mathbf{H}^{+} = (\mathbf{H}^{T}\mathbf{H})^{-1}\mathbf{H}^{T}$ is the Moore-Penrose pseudo-inverse of the real channel matrix $\mathbf{H}$. This is the undistorted data $\mathbf{x}$ corrupted by the noise $\ddot{\mathbf{w}} = \mathbf{H}^{+}\mathbf{w}$ that is correlated with covariance matrix

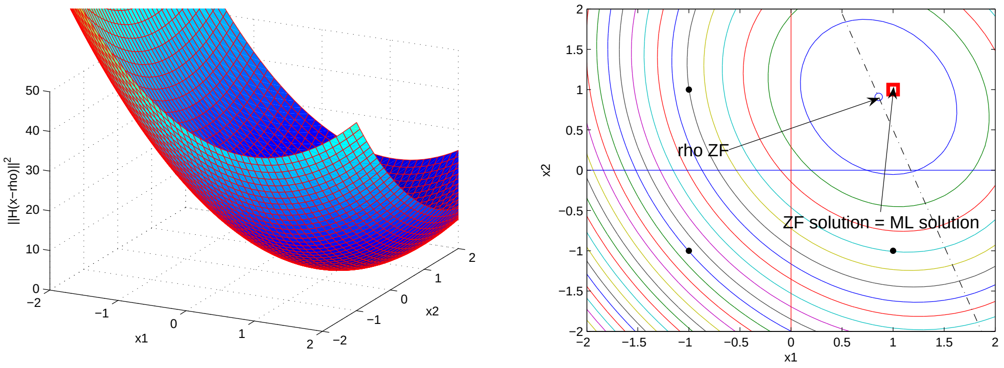
$$\begin{aligned} \mathbf{R}_{\ddot{\mathbf{w}}} &= \sigma^{2}(\mathbf{H}^{T}\mathbf{H})^{-1} \\ &= \sigma^{2}\mathbf{V}\mathbf{D}^{-2}\mathbf{V}^{T} \end{aligned} \tag{4.4}$$

Hence, the contour surfaces of the probability density function of $\ddot{\mathbf{w}}$ are hyperellipsoids whose m^{th} principal axis is given by the m^{th} eigenvector $\mathbf{v}_m$ of $\mathbf{R}_{\ddot{\mathbf{w}}}$. Thus, ZF detection results in a distortion of the noise pdf relative to the spherical pdf geometry of $\mathbf{w}$.

The ML detection performs the minimization of the objective function $f(\mathbf{x})$ over a nonconvex set ξ. The optimal ML solution $\hat{\mathbf{x}}$ is given as

$$\begin{aligned} \hat{\mathbf{x}} &= \arg\min_{\mathbf{x}\in\xi} f(\mathbf{x}) \\ &= \arg\min_{\mathbf{x}\in\xi} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \\ &= \arg\min_{\mathbf{x}\in\xi} \|\mathbf{H}(\mathbf{x} - \rho_{zf})\|_2^2 \\ &= \arg\min_{\mathbf{x}\in\xi} (\mathbf{x} - \rho_{zf})^{T}\mathbf{H}^{T}\mathbf{H}(\mathbf{x} - \rho_{zf}) \\ &= \arg\min_{\mathbf{x}\in\xi} (\mathbf{x} - \rho_{zf})^{T}\mathbf{V}\mathbf{D}^{2}\mathbf{V}^{T}(\mathbf{x} - \rho_{zf}) \end{aligned} \tag{4.5}$$

The cost function $f(\mathbf{x})$ is a quadratic function and assumes the shape of a hyperparaboloid as illustrated in figure 4.4 and figure 4.5 for a 2-dimensional case. The sections of the surface $f(\mathbf{x}) = const$, are hyperellipsoids (ellipses in the 2-dimensional case). The orientation and the shape of these ellipsoids depend on the eigenvalues of the matrix $\mathbf{H}^{T}\mathbf{H}$. It is easy to show that the axes of the hyperellipsoids are aligned with the eigenvectors of $\mathbf{H}^{T}\mathbf{H}$ and that their lengths are inversely proportional to the square roots of the corresponding eigenvalues. In the 2-dimensional case, if the two eigenvalues are very different the ellipses are thin and long, while, if the eigenvalues are equal the ellipses degenerate into circles. The figures 4.4 and 4.5 show the surface $f(\mathbf{x}) = const$ after ZF detection



(a) Graph of $f(x) = \|\mathbf{H}(\mathbf{x} - \rho_{zf})\|_2^2$. The minimum point of this surface is ρ_{zf}.

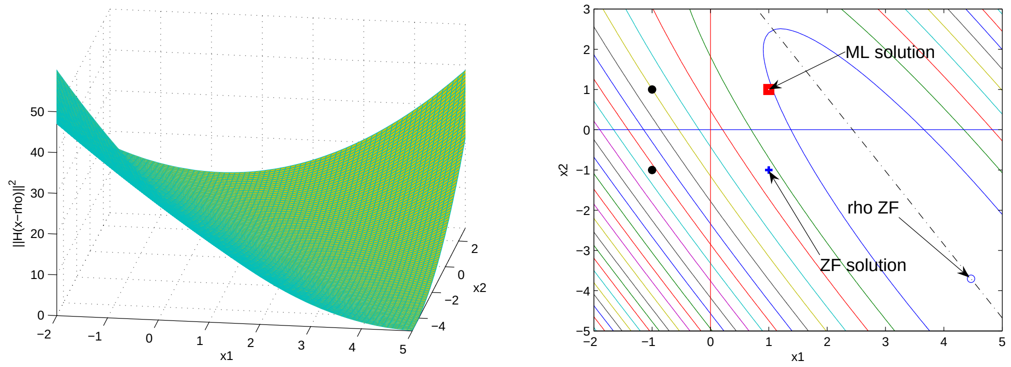
(b) Contour of $f(x)$. Each ellipsoidal curve has constant $f(x)$.

Figure 4.4: Sections of the surface $f(\mathbf{x}) = const$ and ZF regions in the ZF-detection domain for good conditioned real (2;2) channel and BPSK modulation, SNR=0 dB, and $cond(\mathbf{H}) = 1,3428$.

for a good ($cond(\mathbf{H}) = 1,3428$) and a bad ($cond(\mathbf{H}) = 5,6284$) realization of a real-valued $(2,2)$ channel and the ZF decision regions (the four quadrants). For the good channel, the ZF and ML solution are similar. For the bad channel, they are very different. Experiments indicate that for a bad channel, the largest ZF-domain noise component whose direction is given by the principal axis $\mathbf{v}_2$ tends to dominate all the other noise components. Hence, this dominant noise component causes the main part of the bad channel effects that are responsible for the poor performance of linear detection.

4.2 The intensification step

The intensification or the greedy search method can be considered as a local search. This section describes an overview of basic local search. Essentially, a local search consists in moving from one feasible solution to another in its neighborhood. A particular optimization problem can be specified by identifying



(a) Graph of $f(x) = \|\mathbf{H}(\mathbf{x} - \rho_{zf})\|_2^2$. The minimum point of this surface is ρ_{zf}.

(b) Contour of $f(x)$. Each ellipsoidal curve has constant $f(x)$.

Figure 4.5: Sections of the surface $f(\mathbf{x}) = const$ and ZF regions in the ZF-detection domain for ill-conditioned real (2;2) channel and BPSK modulation, SNR=0 dB, and $cond(\mathbf{H}) = 5,6284$.

a set of solutions ξ with a cost function $f(\mathbf{x})$ that assigns a numerical value to each solution $\mathbf{x}$. When applied to a minimization problem, an optimal solution is a solution (or a set of solutions) with the minimum possible cost in a feasible solution space of the problem.

Local search is a generally applicable approach that can be used to find approximate solutions to difficult optimization problems. A local search strategy starts from an arbitrary solution $\mathbf{x}^1 \in \xi$. At each step k, the best solution $\mathbf{x}^{k+1}$ is chosen in the neighborhood $\mathcal{N}_Q(\mathbf{x}^k)$ of the current solution $\mathbf{x}^k$. The neighborhood of $\mathbf{x}^k$ is a subset of ξ and can be defined in the following way. The Q-neighborhood of a point $\mathbf{x}^0 \in \xi$ is the set $\mathcal{N}_Q(\mathbf{x}^0) = \{\mathbf{x} : d_H(\mathbf{x}, \mathbf{x}^0) \leq Q\}$, where $d_H(.,.)$ is the standard Hamming distance. For all $\mathbf{x} \in \mathcal{N}_Q(\mathbf{x}^0)$, $\mathbf{x}^0$ and $\mathbf{x}$ differ by at most Q components. The total number of vectors in $\mathcal{N}_Q(\mathbf{x}^0)$ is $|\mathcal{N}_Q(\mathbf{x}^0)| = \sum_{i=1}^{Q} \binom{n}{i}$. As illustrated in Figure 4.6, for $Q = 1$ this defines the points linked to $\mathbf{x}^0$ by an edge of the n-cube, while for $Q = 2$ it defines the points laying on the same face of the n-cube as $\mathbf{x}^0$. A Q-order greedy search methods finds a solution to

$$f(\mathbf{x}^k) \leq f(\mathbf{x}) \quad \forall \mathbf{x} \in \mathcal{N}_Q(\mathbf{x}^k)$$

To allow for a fast evaluation of moves in the neighborhood, we set $Q = 1$ so that only a low number of possible moves has to be inspected. In addition,

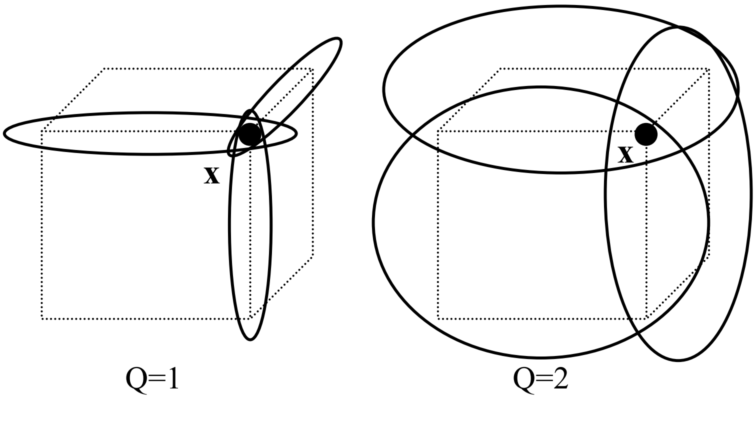


Figure 4.6: Three dimensional visualization of a $Q = 1$ and $Q = 2$ neighborhood. In each case, all neighbors of the point $\mathbf{x}$ are encircled.

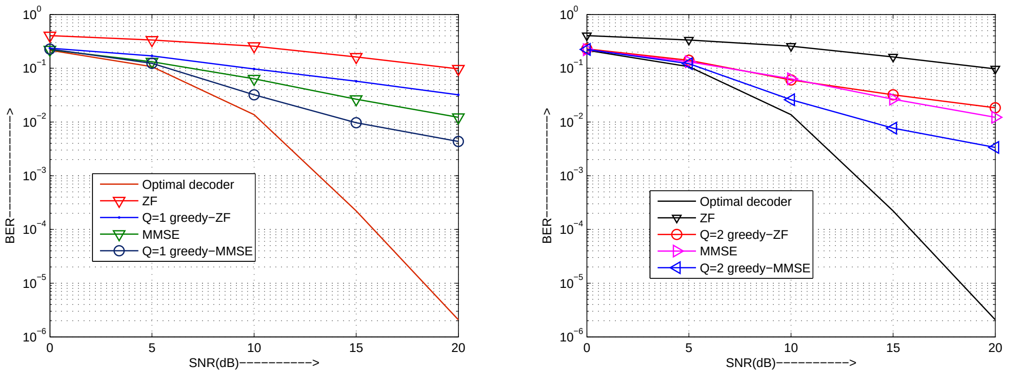
the figure 4.7 show a comparison of the BER performances between the 1^{st}-order ($Q = 1$) and 2^{sd}-order ($Q = 1$) greedy search methods starting from ZF or MMSE detection solution when $n = m = 10$. Also, we can show the that the choice of $Q = 1$ is sufficient for the greedy search to obtain good performance. Hence, a move corresponds to changing the value of a single variable (*i.e.* setting $\mathbf{x}(p) = -\mathbf{x}(p)$ for some p). In this study we develop the $Q = 1$ greedy search methods, and refer to this as the 1^{st}-order greedy detector. The cardinality of $\mathcal{N}_{Q=1}(\mathbf{x}^k)$ is equal to n. For example, assume $n = 4$. Then, the neighborhood of $(+1, +1, +1, +1)$ is $(-1, +1, +1, +1)$, $(+1, -1, +1, +1)$, $(+1, +1, -1, +1)$, and $(+1, +1, +1, -1)$.

The 1^{st}-order ($Q = 1$) greedy detector starts from a given point $\mathbf{x}^1$ and performs iteratively. At each iteration k, we select the best neighbor $\mathbf{x}^{k+1}$ defined as:

$$\mathbf{x}^{k+1} = \min_{\mathbf{x} \in \mathcal{N}_Q(\mathbf{x}^k)} f(\mathbf{x}) \tag{4.6}$$

If $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x})$ for all $\mathbf{x} \in \mathcal{N}_Q(\mathbf{x}^k$ and $f(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k)$ then $\mathbf{x}^{k+1}$ becomes the new starting point of the iteration $k + 1$, otherwise, the algorithm stops and return $\mathbf{x}^k$. By convention, $\mathbf{x}^{k,p} \in \mathcal{N}_{Q=1}($ $\mathbf{x}^k$ will differ only from $\mathbf{x}^k$ by the p^{th} coordinate, $\mathbf{x}^{k,p}(p) = -\mathbf{x}^k(p)$).

The exhaustive computation in (4.6), which includes n computations of the objective function $f(\mathbf{x}^{k,p})$ where $p = 1, .., n$, can be simplified. In fact, at each step k, the computation of $f(\mathbf{x}^{k,p})$, where $\mathbf{x}^{k,p}$ is the p^{th} neighbor of $\mathbf{x}^k$, can be



(a) BER of 1^{sd}-order greedy search methods

(b) BER of 2^{sd}-order greedy search methods

Figure 4.7: Performances comparison of 1^{st}-order and 2^{sd}-order greedy search methods in case of uncoded BPSK modulation and $\mathbf{H} \in \mathbb{R}^{10\times10}$.

replaced by $\delta(\mathbf{x}^{k,p}, \mathbf{x}^k) = \frac{1}{4}[f(\mathbf{x}^{k,p}) - f(\mathbf{x}^k)]$:

$$\begin{aligned} \delta(\mathbf{x}^{k,p}, \mathbf{x}^k) &= \frac{1}{4}[\|\mathbf{y} - \mathbf{H}\mathbf{x}^{k,p}\|_2^2 - \|\mathbf{y} - \mathbf{H}\mathbf{x}^k\|_2^2] \\ &= \mathbf{G}(p,p) + \eta[(\mathbf{H}\mathbf{x}^k)^T\mathbf{H}(:,p) - \mathbf{y}^T\mathbf{H}(:,p)] \end{aligned} \quad (4.7)$$

where $\mathbf{G} = \mathbf{H}^T\mathbf{H}$ is the Gram matrix of the channel matrix $\mathbf{H}$, $\eta = -sign(\mathbf{x}^k(p))$, $\mathbf{H}(:,p)$ and $\mathbf{G}(:,p)$ represent respectively the p^{th} column of the channel matrix $\mathbf{H}$ and the Gram matrix $\mathbf{G}$.

4.2.1 Intensification step study

The greedy search method partitionned the set of all feasible solutions to K subsets defined as:

$$\xi = \cup_{i=1}^{K}\Pi_i \quad where \quad \forall(\mathbf{u},\mathbf{v}) \in \Pi_i \quad greedy(\mathbf{u}) = greedy(\mathbf{v}) \quad (4.8)$$

where $greedy(u)$ is result of the greedy search starting from the feasible point $\mathbf{u}$. By convention Π_1 will be the subset that leads to the ML solution (global optimum). Figure 4.8 shows the cumulative distribution function of the number of subsets K. In the simulation, we used $m \times n$ channel matrix $\mathbf{H}$ where $n = m$, for simplification, and the SNR equal to 10 dB. It is seen that the probability

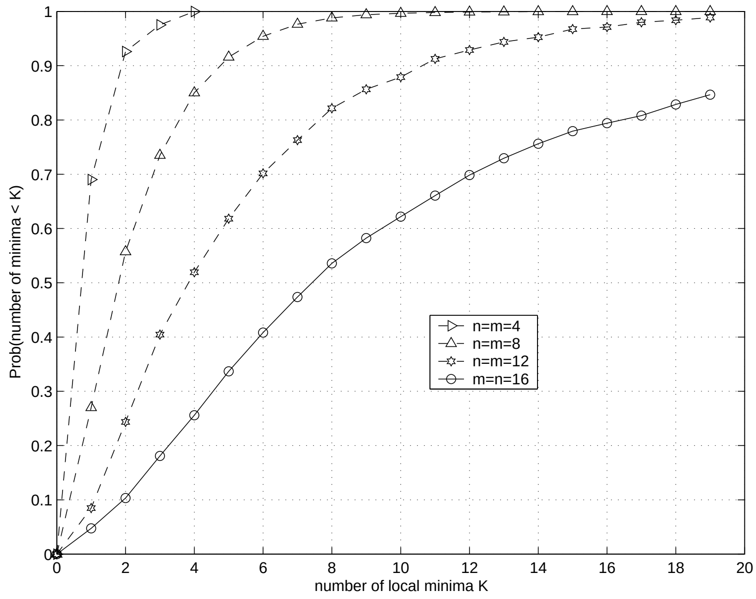


Figure 4.8: The cumulative distribution function of local minima (number of subset K), when SNR=10 dB.

that K exceeds a value of 6, 12, and 18 is 60%, 30%, and 17% respectively in the case of $n = m = 16$. This suggests that the number of local minima occur frequently enough to cause a significant degradation of the average performance of the greedy search method.

To study the bit error rate (BER) performance of the greedy search methods, The idea is to start the greedy search algorithm from the suboptimal solution given by the zero forcing $\hat{\mathbf{x}}_{zf}$ or the MMSE detector $\hat{\mathbf{x}}_{mmse}$. The performances of the $greedy - ZF$ and $greedy - MMSE$ detection schemes are illustrated in figure 4.7(a). As expected, they are clearly better, in term of bit error rate, than the linear detection methods. However, the performance of the greedy search algorithm depends on the starting point of the search. The goal of a good detector is to search the global minimum of the objective function $f(.)$.

4.2.2 computation complexity of the intensification step

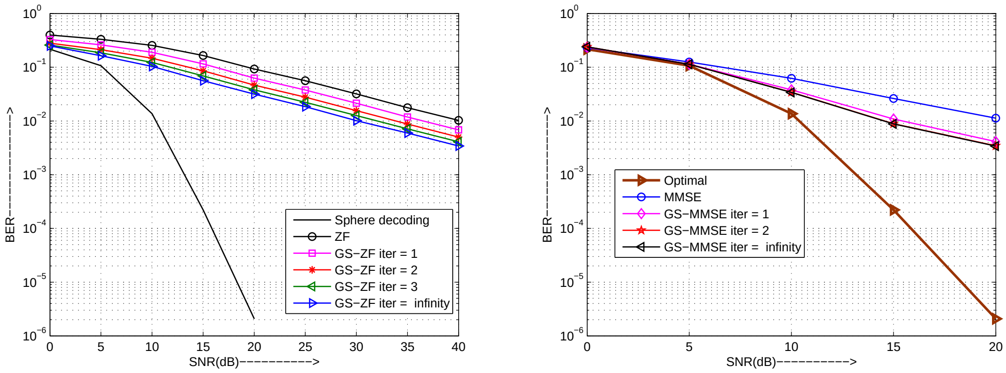
We now compare the computational complexity of the 1^{st}-order and the 2^{sd}-order greedy search methods. For a given starting point the intensification step can be subdivided into two parts. The first one, called pre-processing, is the calculation of the Gram matrix $\mathbf{G} = \mathbf{H}^T\mathbf{H}$. The dominant complexity of this step is $\mathcal{O}(n^3)$.

Table 4.1: Complexity computation of processing step of the 1^{st}-order and the 2^{sd}-order greedy search methods when θ is the iteration number

Operation	1^{st}-order	2^{st}-order
add/sub	$n^2(3\theta+1)+2n\theta$	$n^2(5\theta+1)+4n\theta$
$Mult$	$n^2(2\theta+1)+n\theta$	$n^2(3\theta+1)+2n\theta$

The second one, called processing, has to be performed for each received data vector $\mathbf{y}$. The average number of additions and multiplications of the processing steps is expressed in table 4.1.

The figure 4.9 shows the performances of greedy search method using various numbers of iterations, θ, on different starting point. The results indicate that, for MMSE or ZF starting point, this method has approximately achieved convergence after $\theta = 2$ iterations, in the case of MMSE starting point, since the BER performance has stabilized.



(a) GS starting from ZF solution

(b) GS starting from MMSE solution

Figure 4.9: The effect of iteration number θ in BER performance when $m = n = 10$ and $\xi \triangleq \{-1, +1\}^n$

4.3 The diversification step

Before the search for a locally optimal solution can begin, it has to be decided how to obtain an initial feasible solution. It is sometimes practical to execute local search from several different starting points and to choose the best result.

Next, a ”good” neighborhood has to be chosen for the problem at hand and a method for searching it. The choice is normally guided by intuition because very little theory is available as a guide. A clear trade-off can be seen between small and large neighborhoods. A larger neighborhood would seem to hold promise of providing better local optima but will take longer to search. Design of effective local search algorithms has been and remains very much an art. The analysis of the performance of a standard local search algorithm is concerned with the following:

- Time complexity, *i.e.* the time required by the algorithm to reach the final answer.
- Size of the neighborhood to be searched.
- The number of iterations required to reach a locally optimal solution

The intensification step based on greedy search method requires a well suited diversification, *don't put all you eggs in one basket*, to overcome local minima. Let us assume that the subset $\Pi_1 \subset \xi$ contains all starting points leading to the ML solution (*i.e.* $\mathbf{x} \in \Pi_1 \Leftrightarrow \mathbf{x}_{ml} = greedy(\mathbf{x})$). Our criterion are thus: Firstly, the diversification step should give a small cardinality subset ξ_{start} that verifies $\xi_{start} \cap \Pi_1 \neq \emptyset$ with a very high probability. Secondly, it should have a small computational complexity.

The main part of our research concerns the examination of different ways of diversification. In the following, we suggest some possible methods.

4.3.1 Repeated random start local search

Local search techniques involve iteratively improving upon a solution point by searching in its neighborhood for better solutions. If better solutions are not found, the process terminates; the current point is taken as a locally optimal solution. Since local search performs poorly when there are multiple local optima, a modification of this technique has been suggested in which local search is repeated several times using randomly selected starting points. This process is computationally expensive; after each iteration, search starts from a point very far away from the optimum and no information obtained from previous iterations is reused. Random start is commonly associated with local search as a means of overcoming local optima. More generally, it is a common technique for taking a statistically unreliable process, one which fails more than it succeeds, and through repeated trials producing a new reliable process.

4.3.1.1 Definition

In graph theoretic terms, we can represent each feasible point by a node and each neighbor move by an arc joining the two nodes. Successive neighbor moves will then correspond to paths in the graph. An "only-down" path will refer to a path along which the objective function value is non-increasing. More precisely $\mathbf{x}^1, \mathbf{x}^2, ..., \mathbf{x}^p$ is an "only-down" path if $f(\mathbf{x}^1) \geq f(\mathbf{x}^2) \geq .... \geq f(\mathbf{x}^p)$

Repeated random start local search is an adaptation of the intensification step which repeatedly starts from a random feasible point and follows an "only-down" path toward a local minimal. The algorithm specifies the number of such repeated starts and chooses the best of the local minimal thus generated. For visualization the full Repeated Random Start (RRS) local search algorithm is described compactly in Algorithm 1.

Algorithm 1: RRS local search algorithm

```
Data: $p$: The a priori total number of random starts points, $\mathbf{G} = \mathbf{H}^T\mathbf{H}$
      is the Gram matrix of the channel matrix $\mathbf{H}$ and $\mathbf{y}$ is the received
      vector.
Result: $\mathbf{x}_{RRS} \in \xi$ RRS solution.
begin
    for $(j = 1; j \leq p; j++)$ do
        $\mathbf{x}_{start}(:, j) = sign(rand(n, 1) - 0.5)$
        $[List(1 : n, j), List(n + 1, j)] = greedysearch(\mathbf{y}, \mathbf{G}, \mathbf{H}, \mathbf{x}_{start}(:, j))$
    $List = (sortrows(List^T, n + 1))^T$
    $\mathbf{x}_{RRS} \longleftarrow List(1 : n, 1)$
end
```

In figure 4.10, we have plotted the performance of the RRS-GS using different p values in a Rayleigh fading channel when $\mathbf{H} \in \mathbb{R}^{10\times10}$. We observe that in this case, the best RRS-GS performances is given in the case of $p = 4n$. A simple explanation can be provided by the fact that the probability $P(\xi_{start} \cap \Pi_1)$ increases with the parameter p.

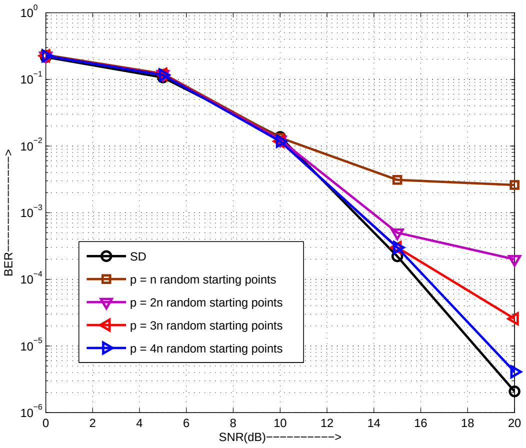


Figure 4.10: Average BER of RRS-GS based detectors with different values of parameter p for $n = m = 10$ system using uncoded BPSK.

4.3.2 BCH diversification

The basic idea behind the diversification is to generate a "smart" starting subset ξ_{start}. It is obvious that a starting point $\mathbf{x}^k$ that has a small hamming distance to another starting point already in the subset ξ_{start}, will not contribute much to the diversification. Therefore, a starting point is not added to the starting subset if its distance to the ξ_{start} is below a certain threshold δ. We call δ the *diversity parameter*. A starting point $\mathbf{x}^k$ can be added to the subset ξ_{start} if the following holds:

$$\min_{\mathbf{x}^i \in \xi_{start}} d_H(\mathbf{x}^k, \mathbf{x}^i) \geq \delta \tag{4.9}$$

Using the diversity parameter δ, the diversity of the starting subset can be controlled as higher values for δ will increase the diversity of the staring subset while lower values will decrease it. A high value of δ will allow only starting point that have a large distance to all starting points in the subset ξ_{start} and will lead -perhaps after a few iterations of intensification step - to a low number of solution points. A low value of δ will allow increasing of computation complexity of the decoder.

In literature, the Bose-Chaudhuri-Hocquenghem (BCH) codes [BRC60, Hoc59] are a class of cyclic codes that append $n - k$ parity bits to a message of k bits so that each code word is n bits long. The code parameters (n, k, d_{min}) are of the

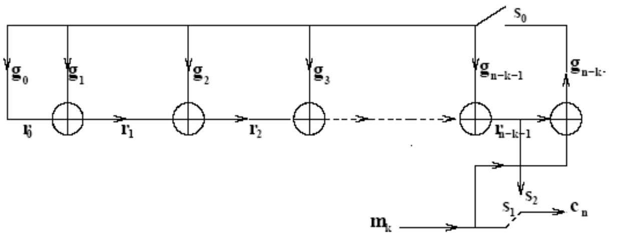


Figure 4.11: Shift Register Encoding using BCH Codes.

form $n = 2^m - 1$, $n - k \leq mt$, for positive integers m and t, and the minimum Hamming distance is $d_{min} \leq 2t + 1$. The codes are specified by their generator polynomials in $GF(2)$ which has the general form $G(D) = g_0 + g_1 D + ... + g_{n-k} D^{n-k}$. The parity bits appended to the message in the systematic generation of the codeword corresponding to the message polynomial $M(D)$ are the coefficients of the remainder of $\frac{D^k M(D)}{G(D)}$. This encoding process is usually implemented by a shift register. The general setup is shown in figure 4.11 in which switches s_0 and s_1 are closed and s_2 open for the first k cycles while the message m_k of length k is input. For the next $n - k$ cycles switch s_2 is closed and switches s_0 and s_1 are open.

The figure 4.12 shows the BER performance comparison between different detection algorithms based on repeated random start or extended BCH code in diversification step. It can be easily observed that the extended BCH code allows good diversification to the intensification step. Moreover the generation of BCH code is more adapted to hardware implementation.

4.3.3 Geometrical diversification

A geometrical approach leads us to an efficient method to select ξ_{start}. In what follows, the singular value decomposition (SVD) of channel matrix $\mathbf{H} = \mathbf{U}\Sigma\mathbf{V}^T$ is used, where the diagonal matrix Σ contains the singular values $\{\lambda_k\}_{k=1}^n$ and the unitary matrices $\mathbf{U}$ and $\mathbf{V}$ contain, respectively, the left $\{\mathbf{u}_k\}_{k=1}^m$ and right $\{\mathbf{v}_k\}_{k=1}^n$ singular vectors of the matrix $\mathbf{H}$ as columns. We assume that the λ_k are indexed in increasing order, *i.e.* $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$.

Let $\rho_{zf} = \mathbf{H}^+\mathbf{y}$ be the solution given by the ZF detector, *i.e.* $f(\rho_{zf}) = 0$. For all points $\mathbf{x} \in \xi$, $\mathbf{x} - \rho_{zf}$ is a vector of $\mathbb{R}^n$ and can be expressed in the base $\mathbf{V}$ as $\mathbf{x} - \rho_{zf} = \sum_{k=1}^n \alpha_k \mathbf{v}_k$ where $\{\alpha_k\}_{k=1}^n$ are real coefficients. We can express the

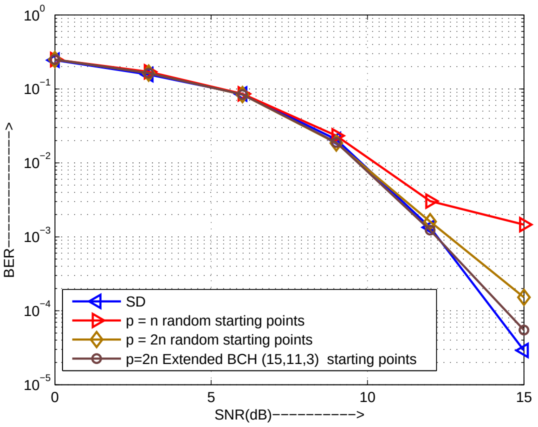


Figure 4.12: Performance Comparison between RRS-GS based detectors and GS using extended BCH code for $n = m = 16$ system.

value of the objective function at the feasible point $\mathbf{x}$ as:

$$\begin{aligned} f(\mathbf{x}) &= \|\mathbf{y}-\mathbf{Hx}\|_2^2 \\ &= \|\mathbf{H}(\mathbf{x}-\rho_{zf})\|_2^2 \\ &= \|\sum_{k=1}^{n}\alpha_k\lambda_k\mathbf{v}_k\|_2^2 + c \\ &= \sum_{k=1}^{n}\alpha_k^2\lambda_k^2 + c \end{aligned} \tag{4.10}$$

The constant c is independent of the vector symbol $\mathbf{x}$ and can hence be ignored in the metric computation. In the following, for simplicity of description, we set $c = 0$.

Let us define $\triangle_k = \{z \in \mathbb{R}^n / z = \rho_{zf} + \gamma\mathbf{v}_k,\ \gamma \in \mathbb{R}\}$ as a line in $\mathbb{R}^n$ originating from the point ρ_{zf} and along a direction $\mathbf{v}_k$. Since $\lambda_1 \leq \lambda_2 \leq ... \leq \lambda_n$, we can note that objective function $f(.)$ increases much slower along the first D lines $\triangle_1, ..., \triangle_D$ than along the last $n - D$ lines. The idea of the diversification step is to choose the feasible points in the "vicinity" of the lines $\triangle_1, ..., \triangle_D$ in order to create a starting subset ξ_{start} which contains *a priori* good starting point. A first level of diversity is then obtained by the use of the D independent lines. Then, for each line $\triangle_k$, $k = 1, ..., D$ a second level of diversity is given by different algorithm variants as following.

4.3.3.1 Hypercube Intersection and Selection method

Let us define $\xi \triangleq \{-1,+1\}^n$. Geometrically, this set comprises the vertices of the unit hypercube of dimension n. The basic idea of this method is to find the intersection points between $\triangle_k$ and the faces of the unit hypercube. Given a line $\triangle_k$, there are at most $2n$ feasible points $\mathcal{I}_k \subset \{-1,+1\}^n$ representing the projection on $\{-1,+1\}^n$ of all intersection points between $\triangle_k$ and the $2n$ hyperplanes defined as

$$\mathcal{PH} = \{z \in \mathbb{R}^n / z(i) = s,\ s = -1, 1 \quad and \quad i = 1...n\} \tag{4.11}$$

Let us study the intersection between a given line $\triangle_k$ and the hyperplane $\mathcal{P}(i,s) = \{z \in \mathbb{R}^n / z(i) = s\}$. The problem is to obtain $\gamma_k^{s,i}$ expressed in the form:

$$\rho_{zf}(i) + \gamma_k^{s,i}\mathbf{v}_k(i) = s \tag{4.12}$$

This intersection includes two cases:

- If $\mathbf{v}_k(i) \neq 0$: $\gamma_k^{s,i} = \frac{s-\rho_{zf}(i)}{\mathbf{v}_k(i)}$ then the generated point is $\beta_k^{s,i} = \gamma_k^{s,i}\mathbf{v}_k + \rho_{zf}$. The returned point is $\bar{\beta}_k^{s,i}$ where the p^{th} coordinate is equal to $sign((s - \rho_{zf}(i))v_k(p) + v_k(i)\rho_{zf}(p)) \times sgn(v_k(i))$, $p = 1, .., n$.
- If $\mathbf{v}_k(i) = 0$: the returned point $\bar{\beta}_k^{s,i} = \hat{\mathbf{x}}_{zf}$.

Thus $\mathcal{I}_k = \{\bar{\beta}_k^{s,i}\}_{i=1..n}^{s=-1,1}$ contains at most $2n$ distinct feasible points. The starting subset for the intensification is defined as:

$$\xi_{start} = \cup_{k=1}^{D} \xi_k \tag{4.13}$$

where $\xi_k \subset \mathcal{I}_k$ contains the C best candidate points of $\mathcal{I}_k$, *i.e.* the C distinct feasible points of $\mathcal{I}_k$ which minimize the objective function $f(.)$. In figure 4.13 we show all intersections points $\{\beta_k^{s,i}\}_{i=1..n}^{s=-1,1}$ and their corresponding candidate points $\{\bar{\beta}_k^{s,i}\}_{i=1..n}^{s=-1,1}$ in case $n = 2$.

4.3.3.2 Canonical Basis Intersection and Selection method

The method of slowest descent is a general method for solving discrete optimization problems developed in [SG01]. Here we will only describe its application to diversification step. The idea of the method of slowest descent is only to consider the discrete solutions $\mathbf{x} \in \{+1,-1\}^n$ that are closest to the D lines in $\mathbb{R}^n$

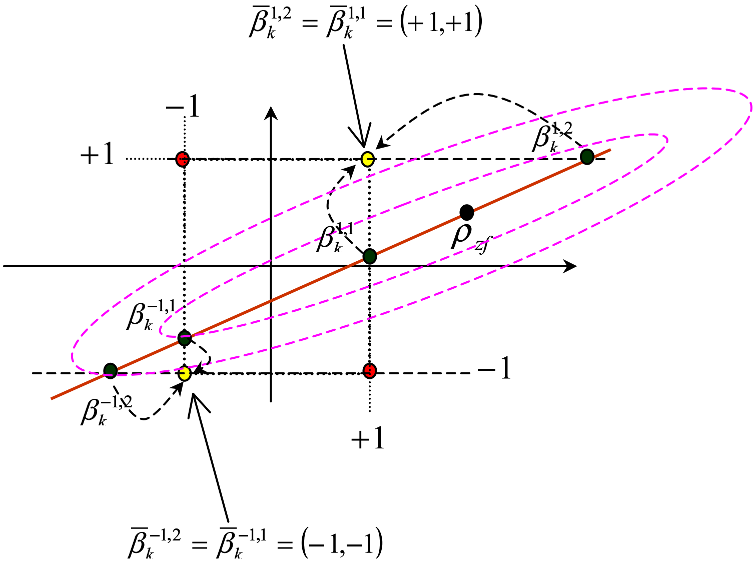

$\bar{\beta}_k^{1,2} = \bar{\beta}_k^{1,1} = (+1,+1)$
-1
$+1$
$\beta_k^{1,2}$
$\beta_k^{1,1}$
ρ_{zf}
$\beta_k^{-1,1}$
-1
$\beta_k^{-1,2}$
$+1$
$\bar{\beta}_k^{-1,2} = \bar{\beta}_k^{-1,1} = (-1,-1)$

Figure 4.13: Hypercube Intersection and Selection method : One to one mapping from $\{\beta_k^{s,i}\}_{i=1..n}^{s=-1,1}$ to $\{\bar{\beta}_k^{s,i}\}_{i=1..n}^{s=-1,1}$ for $n = 2$.

defined by ρ_{zf} and the eigenvectors belonging to the D smallest eigenvectors of $\mathbf{G} = \mathbf{H}^T\mathbf{H}$. The points can be found as follows. Let $\mathbf{v}_k$ be the k^{th} smallest eigenvectors of Gram matrix G. The set of intersection points corresponding to a line defined by ρ_{zf} and $\mathbf{v}_k$ can be expressed as

$$\{\beta_k^i = \rho_{zf} + \gamma_k^i \mathbf{v}_k,\ \gamma_k^i = \frac{\rho_{zf}(i)}{\mathbf{v}_k(i)}\} \tag{4.14}$$

where $\rho_{zf}(i)$ and $\mathbf{v}_k(i)$ denote the i^{th} elements of the respective vectors ρ_{zf} and $\mathbf{v}_k$. Each intersection point has only its i^{th} component equal to zero, *i.e.*, $\beta_k^i(i) = 0$. For simplicity, we do not consider lines that simultaneously intersect more than one coordinate hyper-plane since this event occurs with probability zero.

Any point on the line except for an intersection point has an unique closest candidate point in $\{+1, -1\}^n$. The figure 4.14 shows that an intersection point is of equal distance from its two neighboring candidate points and two neighboring intersection points share a unique closest candidate point. By carefully selecting one of the two candidate points closest to each intersection point to avoid choosing the same point twice, one can specify n distinct candidate points in $\{+1, -1\}^n$ that are closest to the line defined by ρ_{zf} and $\mathbf{v}_k$. To that end, consider the

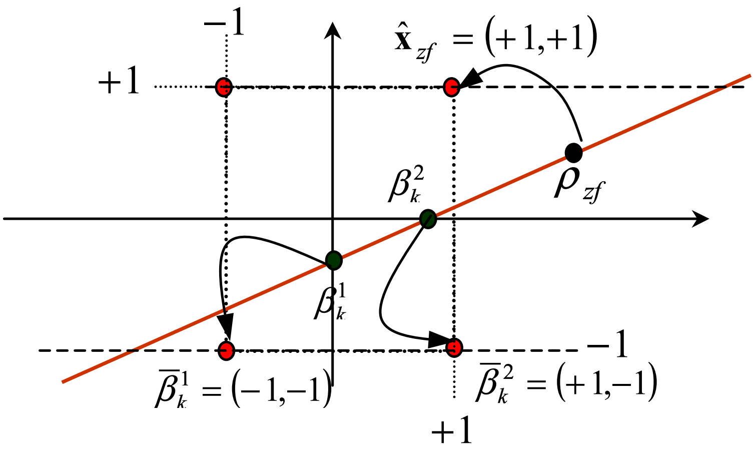

-1
$\hat{\mathbf{x}}_{zf} = (+1,+1)$
$+1$
β_k^2
ρ_{zf}
β_k^1
-1
$\bar{\beta}_k^1 = (-1,-1)$
$\bar{\beta}_k^2 = (+1,-1)$
$+1$

Figure 4.14: Canonical basis intersection method : One to one mapping from $\{\rho_{zf}, \beta_k^1, .., \beta_k^n\}$ to $\{\hat{\mathbf{x}}_{zf}, \bar{\beta}_k^1, .., \bar{\beta}_k^n\}$ for $n = 2$. Each intersection point β_k^i is of equal distance from its two neighboring candidate points. $\bar{\beta}_k^i$ is chosen to be one of these two candidate points that is on the opposite side of the i^{th} coordinate hyper-plane with respect to $\hat{\mathbf{x}}_{zf}$.

following set $\mathcal{I}_k$:

$$\{\bar{\beta}_k^i \in \xi,\ \bar{\beta}_k^i(p) = sign(\beta_k^i(p))\ p \neq i\ and\ \bar{\beta}_k^i(i) = -sign(\hat{\mathbf{x}}_{zf}(i))\} \tag{4.15}$$

It is seen that (4.15) assigns to each intersection point β_k^i a closest candidate point $\bar{\beta}_k^i$ that is on the opposite side of the i^{th} coordinate hyper-plane from $\hat{\mathbf{x}}_{zf}$. Then , the starting subset for the intensification is defined as:

$$\xi_{start} = \cup_{k=1}^{D} \xi_k \tag{4.16}$$

where $\xi_k \subset \mathcal{I}_k$ contains the C best candidate points of $\mathcal{I}_k$, *i.e.* the C distinct feasible points of $\mathcal{I}_k$ which minimize the objective function $f(.)$.

4.3.3.3 Plane Intersection and Selection method

The Plane Intersection and Selection method is based on the study of the intersection between the line defined by ρ_{zf} and the k^{th} slowest eigenvector $\mathbf{v}_k$ and all planes orthogonal to the vector $\mathbf{n}_k = sign(\mathbf{v}_k)$, *i.e.* $\mathbf{n}_k$ is the normal vector to all planes. Let l be the number of non-zero coordinates of the normal vector. There are exactly $l+1$ distinct planes, which contain all candidate points belong

ξ, defined as:

$$\mathcal{PH} = \{\mathbf{z} \in \mathbb{R}^n \;/\; \mathbf{z}^T\mathbf{n}_k = t,\ t = \{-l, -l+2, ..., l-2, l\}\} \tag{4.17}$$

In n-dimensional, a line $\triangle_k$ is either parallel to a plane $\mathcal{P}_t$ or intersects it in a single point. Let $\triangle_k$ be given by the parametric equation: $\rho_{zf} + \gamma_k^i\mathbf{v}_k$, and the plane $\mathcal{P}_t$ be given by the parametric equation: $\mathbf{z}^T\mathbf{n}_k = t$ when it's normal vector $\mathbf{n}_k$. We first check if $\triangle_k$ is parallel to $\mathcal{P}_t$ by testing if $\mathbf{n}_k^T\mathbf{v}_k = 0$ which means that the line direction vector $\mathbf{v}_k$ is perpendicular to the plane normal $\mathbf{n}_k$. If this is true, then $\triangle_k$ and $\mathcal{P}_t$ are parallel and either never intersect or else $\triangle_k$ lies totally in the plane $\mathcal{P}_t$. Disjointness or coincidence can be determined by testing whether any specific point of $\triangle_k$, say ρ, is contained in $\mathcal{P}_t$, that is whether it satisfies the implicit line equation: $\mathbf{n}_k^T(\rho - \mathbf{x}_{k,t}) = 0$, where $\mathbf{x}_{k,t} \in \xi$ belongs to the plane $\mathcal{P}_t$.

Let us study the intersection between a given line $\triangle_k$ and the plane $\mathcal{P}_t$. The problem is to obtain γ_k^t expressed in the form:

$$\gamma_k^t = \frac{t - \mathbf{n}_k^T\rho_{zf}}{\mathbf{n}_k^T\mathbf{v}_k}, \quad where\, t = \{-l, -l+2, ..., l-2, l\} \tag{4.18}$$

then the generated point is $\beta_k^t = \gamma_k^t\mathbf{v}_k + \rho_{zf}$. The returned point is $\bar{\beta}_k^t = sign(\beta_k^t)$.

Thus $\mathcal{I}_k = \{\bar{\beta}_k^t\}_{t=-l,-l+2,...,l-2,l}$ contains at most $l+1$ distinct feasible points. The starting subset for the intensification is defined as:

$$\xi_{start} = \cup_{k=1}^{D}\xi_k \tag{4.19}$$

where $\xi_k \subset \mathcal{I}_k$ contains the C best candidate points of $\mathcal{I}_k$, *i.e.* the C distinct feasible points of $\mathcal{I}_k$ which minimize the objective function $f(.)$. In figure 4.15, we show all intersections points $\{\beta_k^t\}_{t=-l,-l+2,...,l-2,l}$ and their corresponding candidate points $\{\bar{\beta}_k^t\}_{t=-l,-l+2,...,l-2,l}$ in case $n = 2$.

4.4 GISD detector flow chart

The different steps and variants of geometrical intersection and selection detector, presented in the previous section, are more clearly shown in Figure 4.16. The GISD algorithm depends from two parameters D (the number of studied slowest eigenvectors directions) and C (the number of the best candidate points at each direction).

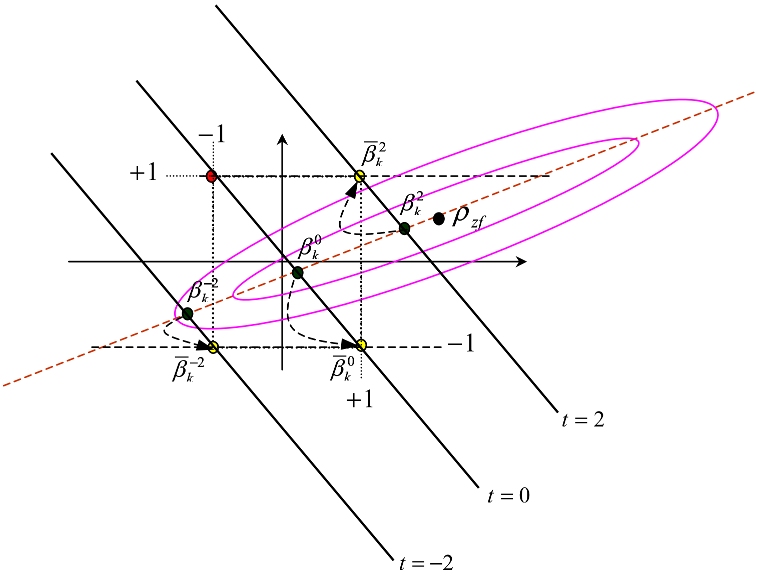


Figure 4.15: Plane Intersection and Selection method: One to one mapping from $\{\beta_k^t\}_{t=-l,-l+2,\ldots,l-2,l}$ to $\{\bar{\beta}_k^t\}_{t=-l,-l+2,\ldots,l-2,l}$ for $n = 2$.

The following example illustrates the procedure: Let us give an 4-dimensional example to illustrate the algorithms HISD, CBISD and PISD. The parameters used by the GISD algorithm are $n = 4$, $D = 1$ and $C = 3$:

$$\begin{aligned}
\mathbf{H} &= \begin{pmatrix} -1.01 & -1.43 & 0.30 & -0.09 \\ -0.12 & -1.02 & 0.72 & -0.55 \\ -0.21 & -0.92 & 0.97 & -0.32 \\ 0.68 & 1.17 & -0.34 & -0.72 \end{pmatrix} \\
\mathbf{x} &= \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix}^T \\
\mathbf{w} &= \begin{bmatrix} 0.18 & 0.50 & -0.67 & 0.53 \end{bmatrix}^T \\
\mathbf{y} &= \begin{bmatrix} -2.05 & -0.47 & -1.15 & 1.32 \end{bmatrix}^T
\end{aligned}$$

The coordinates of $\mathbf{y}$ with respect to the lattice are $\rho_{zf} = [2.89, -0.90, -1.39, 0.08]^T$. The first slowest eigenvector direction $\mathbf{v}_1$ can be found using the singular value decomposition of the Gram matrix $\mathbf{G} = \mathbf{H}^T\mathbf{H}$.

- **Detection using the HISD**: The algorithm determines the line $\triangle_1$ defined by $\{z \in \mathbb{R}^n / z = \rho_{zf} + \gamma \mathbf{v}_k,\ \gamma \in \mathbb{R}\}$ and generates a set $\mathcal{I}_1$ of all intersection points between this line and $\mathcal{PH}$. The generated subset is

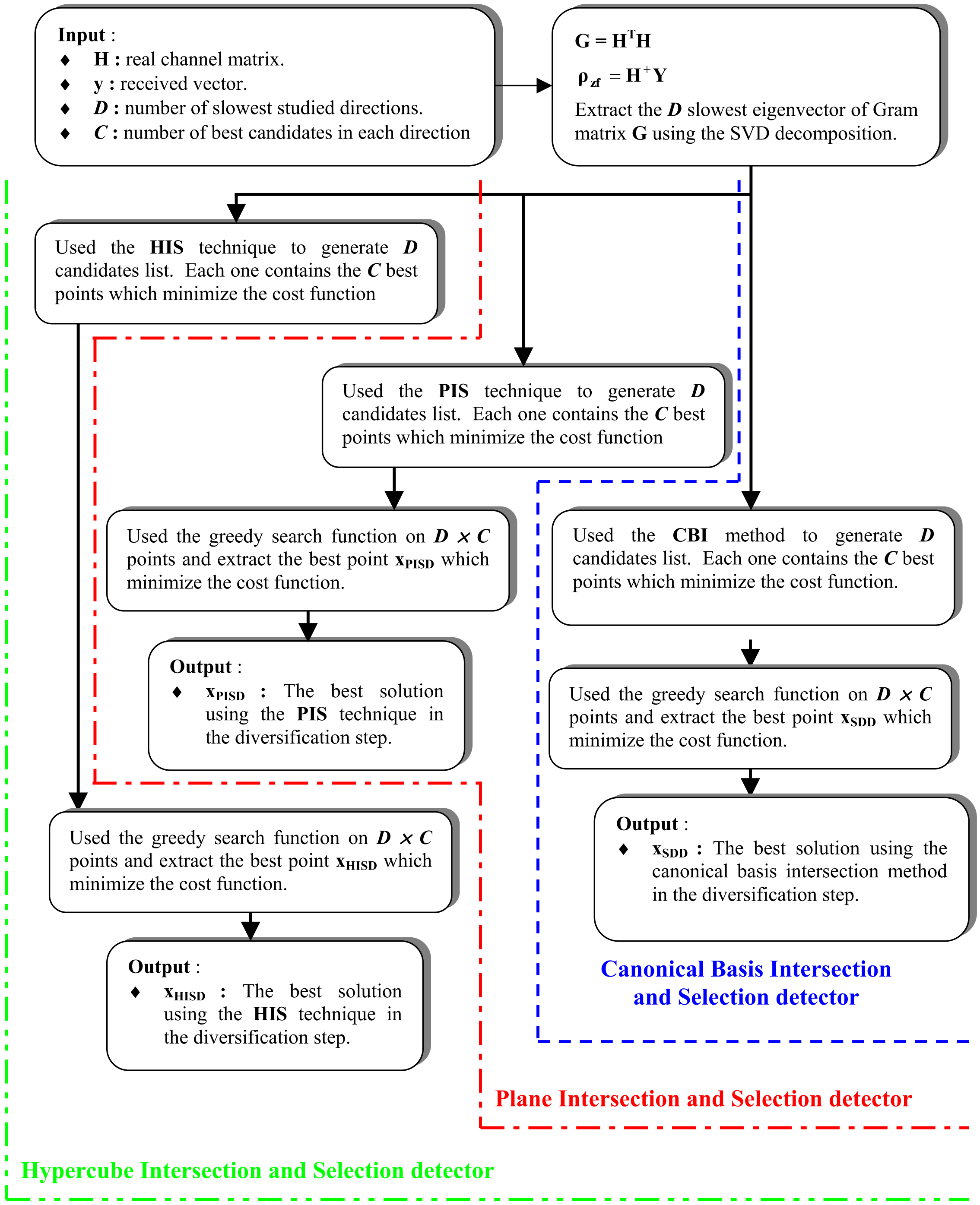

Input :
♦ **H :** real channel matrix.
♦ **y :** received vector.
♦ ***D*** **:** number of slowest studied directions.
♦ ***C*** **:** number of best candidates in each direction
$\mathbf{G} = \mathbf{H}^T\mathbf{H}$
$\boldsymbol{\rho}_{zf} = \mathbf{H}^+\mathbf{Y}$
Extract the ***D*** slowest eigenvector of Gram matrix **G** using the SVD decomposition.
Used the **HIS** technique to generate ***D*** candidates list. Each one contains the ***C*** best points which minimize the cost function
Used the **PIS** technique to generate ***D*** candidates list. Each one contains the ***C*** best points which minimize the cost function
Used the greedy search function on $D \times C$ points and extract the best point $\mathbf{x}_{\mathbf{PISD}}$ which minimize the cost function.
Output :
♦ $\mathbf{x}_{\mathbf{PISD}}$ **:** The best solution using the **PIS** technique in the diversification step.
Used the **CBI** method to generate ***D*** candidates list. Each one contains the ***C*** best points which minimize the cost function.
Used the greedy search function on $D \times C$ points and extract the best point $\mathbf{x}_{\mathbf{SDD}}$ which minimize the cost function.
Output :
♦ $\mathbf{x}_{\mathbf{SDD}}$ **:** The best solution using the canonical basis intersection method in the diversification step.
Used the greedy search function on $D \times C$ points and extract the best point $\mathbf{x}_{\mathbf{HISD}}$ which minimize the cost function.
Output :
♦ $\mathbf{x}_{\mathbf{HISD}}$ **:** The best solution using the **HIS** technique in the diversification step.
Canonical Basis Intersection and Selection detector
Plane Intersection and Selection detector
Hypercube Intersection and Selection detector

Figure 4.16: Flowchart of the proposed algorithm.

Table 4.2: This table give all point in $sign(\mathcal{I}_1)$ when we used the HISD detector.

Point	$\mathbf{x}^1$	$\mathbf{x}^2$	$\mathbf{x}^3$	$\mathbf{x}^4$
coordinate 1	1	1	-1	1
coordinate 2	1	1	1	-1
coordinate 3	1	-1	1	-1
coordinate 4	-1	1	-1	1
$\Vert\mathbf{y}-\mathbf{H}\mathbf{x}^i\Vert_2^2$	2.90	6.01	7.98	9.46

Table 4.3: Application of greedy search in the case of HISD algorithm.

Point	$\mathbf{x}^1$	$\mathbf{x}^2$	$\mathbf{x}^3$
$\Vert\mathbf{y}-\mathbf{H}\cdot GS(\mathbf{x}^i)\Vert_2^2$	1.01	1.01	2.47

$\mathcal{I}_1 = \{\bar{\beta}_1^{s,i}\}_{i=1..4}^{s=-1,1}$. In general, the subset $sign(\mathcal{I}_1) \subset \{-1,1\}^4$ contained at most 8 distinct feasible points after redundancy suppress. In our example, the subset $sign(\mathcal{I}_1)$ contains only four feasible points. For each point $\mathbf{x}^i \in sign(\mathcal{I}_1)$, $i = \{1,2,3,4\}$, the HISD algorithm calculates the quantity $\Vert\mathbf{y}-\mathbf{H}\mathbf{x}^i\Vert_2^2$ and sorts them in ascending order, see table 4.2. Finally, we used the greedy search method to search the best neighbor solution to the Maximum likelihood problem starting from just the first three feasible points $\mathbf{x}^1$, $\mathbf{x}^2$, and $\mathbf{x}^3$ (see table 4.3). The Hypercube Intersection and Selection detector generated solution is $\mathbf{x}_{HISD} = GS(\mathbf{x}^1) = [1,1,1,1]^T$ which has the minimum euclidean distance.

- **Detection using the PISD**: The algorithm determines the line $\triangle_1$ defined by $\{z \in \mathbb{R}^n/z = \rho_{zf} + \gamma\mathbf{v}_k,\ \gamma \in \mathbb{R}\}$ and generates a set $\mathcal{I}_1$ of all intersection points between this line and $\mathcal{P}_t$. The generated subset is $\mathcal{I}_1 = \{\bar{\beta}_1^t\}_{t=-4,-2,0,2,4}$. In our example, the subset $sign(\mathcal{I}_1) \subset \{-1,1\}^4$ contained 5 distinct feasible points after redundancy suppress. For each point $\mathbf{x}^i \in sign(\mathcal{I}_1)$, $i = \{1,2,..,5\}$, the PISD algorithm calculates the quantity $\Vert\mathbf{y}-\mathbf{H}\mathbf{x}^i\Vert_2^2$ and sorts them in ascending order, see table 4.4. Finally, we used the greedy search method to search the best neighbor solution to the Maximum likelihood problem starting from just the first three feasible points $\mathbf{x}^1$, $\mathbf{x}^2$, and $\mathbf{x}^3$ (see table 4.5). The Plane Intersection and Selection detector generated solution is $\mathbf{x}_{PISD} = GS(\mathbf{x}^1) = [1,1,1,1]^T$ which has the minimum euclidean distance.

Table 4.4: This table give all point in $sign(\mathcal{I}_1)$ when we used the PISD detector.

Point	$\mathbf{x}^1$	$\mathbf{x}^2$	$\mathbf{x}^3$	$\mathbf{x}^4$	$\mathbf{x}^5$
coordinate 1	1	1	1	-1	1
coordinate 2	1	1	1	1	-1
coordinate 3	1	1	-1	1	-1
coordinate 4	1	-1	1	-1	1
$\Vert\mathbf{y}-\mathbf{H}\mathbf{x}^i\Vert_2^2$	1.01	2.90	6.01	7.98	9.46

Table 4.5: Application of greedy search in the case of PISD algorithm.

Point	$\mathbf{x}^1$	$\mathbf{x}^2$	$\mathbf{x}^3$
$\Vert\mathbf{y}-\mathbf{H}\cdot GS(\mathbf{x}^i)\Vert_2^2$	1.01	1.01	1.01

- **Detection using the CBISD**: In our example, this algorithm gives the same results as the Plane Intersection and Selection detector.

4.5 Computation complexity of the GISD

This section evaluates the computational complexity of the proposed detection algorithm. We first compute the number of multiplications and additions/subtractions required for each geometrical technique approach in the diversification step. Assume for simplicity that $m = n$. In the worst case scenario, the complexity budget for diversification is shown in Table 4.6

GISD's computation complexity depends on the parameters D and C. The new algorithm can be subdivided in two parts. The first one is the calculation of the Gram matrix $\mathbf{G} = \mathbf{H}^T\mathbf{H}$, the Moore-Penrose pseudo-inverse of channel matrix defined as $\mathbf{H}^+ = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T$, and uses the brute-force singular value decomposition for the D smallest eigenvectors estimation. Thus, the dominant complexity of the preparatory steps is $\mathcal{O}(n^3)$ per data block. The complexity

Table 4.6: Complexity computation of HIS, PIS and CBIS techniques for diversification step.

Op	HIS	PIS	CBIS
Add/sub	$2n^3+\frac{9}{2}n^2-\frac{5}{2}n$	n^3+4n^2	n^3+2n^2
Mult	$2n^3+3n^2$	n^3+5n^2	n^3+3n^2

Table 4.7: Complexity of processing part of Geometrical Intersection and Selection Detector.

Detector	Add / Sub	Mult
HISD	$2Dn^3 + [\frac{9}{2} + C(3\theta+1)]Dn^2 + (2\theta - \frac{5}{2})n$	$2Dn^3 + [3 + C(2\theta+1)]Dn^2 + \theta n$
CBISD	$Dn^3 + [2 + C(2\theta+1)]Dn^2 + 2\theta n$	$Dn^3 + [3 + C(2\theta+1)]Dn^2 + \theta n$
PISD	$Dn^3 + [4 + C(3\theta+1)]Dn^2 + 2\theta n$	$Dn^3 + [5 + C(2\theta+1)]Dn^2 + \theta n$

of the first part can be shared by several consecutive L received vectors if the channel variations are slow. The second part of the new algorithm has to be performed for each received data vector. The worst case number of additions and multiplications of the second part is expressed in table 4.7. Note that, the greedy search method is iterated in average $\theta = 2$ times. In the worst case scenario, the total number of visited feasible points by HISD, PISD, and CBISD algorithm are $(DCn\theta + 2Dn)$, $(DCn\theta + Dn)$, and $(DCn\theta + D(n+1))$ respectively.

The average complexity of well-known SD method has been claimed to be polynomial time over certain ranges of rate, SNR and dimension, while the worst case complexity is still exponential. However, recently, Jalden derive an exponential lower bound on the average complexity of SD [JO04]. The performance of the proposed GISD is close to the ML performance, while the order of the worst case complexity is lower as compared to SD (polynomial vs. exponential). Also the main problem with the hardware implementation of existing decoding algorithms (SD,SDP and VBLAST) is the lack of parallelism caused by its heuristic structure. The proposed sub-optimal algorithm GISD can overcome this problem. For example, we can use the greedy search function in parallel way over a C independent data starting points. Moreover, each slowest eigenvectors direction can be studied separately.

4.6 Simulations and Discussions

In this section, we carry out some computer simulations to demonstrate the proposed detection approaches. In addition, we will discuss the characteristics of different variants of the GISD from the simulations results.

4.6.1 Simulations

All experiments described here are for a $2M \times 2N$ real channel matrix system. We provide simulation results demonstrating the performance of GISD algorithms.

4.6.1.1 ***Experiment 1*: MIMO**

In this experiment, we will compare the performance of HISD, PISD, CBISD, SD (sphere decoding) and other known detection algorithms. The first three algorithms are implemented according to GISD flowchart in figure 4.16, respectively. We fix the transmit antennas to $N = 5$ and change receive antennas M from 5, up to 6. The constellation $4 - QAM$ is used. According to the simulation model defined in chapter 2, the received signal in a given symbol interval can be written as

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{w}} \tag{4.20}$$

where the entries of the channel matrix $\tilde{\mathbf{H}} \in \mathbb{C}^{M \times N}$ and $\tilde{\mathbf{w}} \in \mathbb{C}^{M \times 1}$ are independent and identically distributed zero mean complex Gaussian random variables with unit and σ^2 variance, respectively. The equivalent real model contains $2M$ equations and $2N$ integer unknowns that assume the values from $\{\pm 1\}$. The channel is assumed to be quasi-static. The matrix $\tilde{\mathbf{H}}$ maintains constant during every interval of $L = 100$ symbols, and then changes randomly. In the figure 4.17, It can be seen the good performance of the new proposed geometrical approach, when $D = 2$ and $C = 4$ comparing to the SDP detector. In fact, the required SNR for a BER of 10^{-4} is 3.8 dB lower than that of SDP detector.

Figure 4.18 shows the bit error rate performance of different variants of the Geometrical Intersection and Selection detector with $4-QAM$ uncoded modulation using $N = 5$ transmit antennas and $M = 6$ receive antennas over a quasi-static Rayleigh fading channel. In this scenario, we notice that for $D = 2$ and $C = 4$ The proposed detector can achieve a quasi-ML performance. All GISD variants outperform the MMSE and the SDP detector.

4.6.1.2 ***Experiment 2*: MC-CDMA**

In this experiment, we will present the performance of HISD, PISD, CBISD, SD (sphere decoding) and other known detection algorithms for a downlink MC-CDMA system. The channel coefficients are modified for each transmitted symbol. All users have the same power. We assume the power control being perfect,

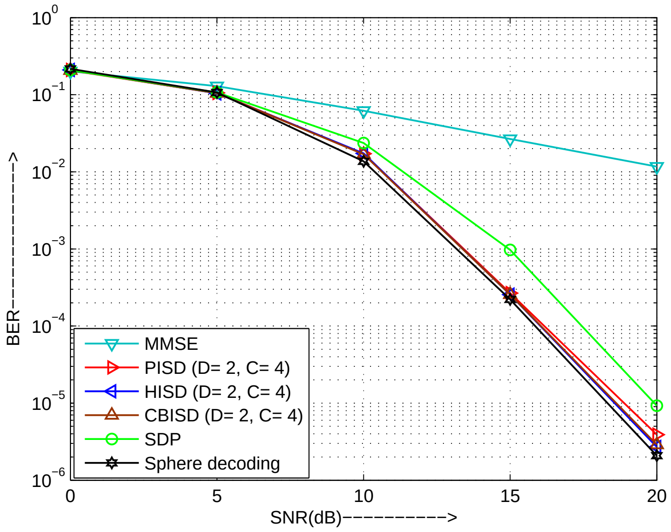


Figure 4.17: BER versus SNR for $N = 5$ and $M = 5$ MIMO system, comparison of GISD variants, SDP and MMSE detectors.

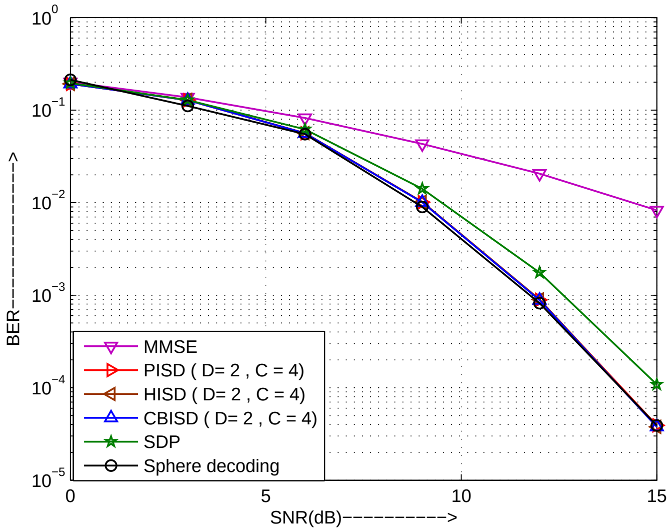


Figure 4.18: BER versus SNR for $N = 5$ and $M = 6$ MIMO system, comparison of GISD variants, SDP and MMSE detectors.

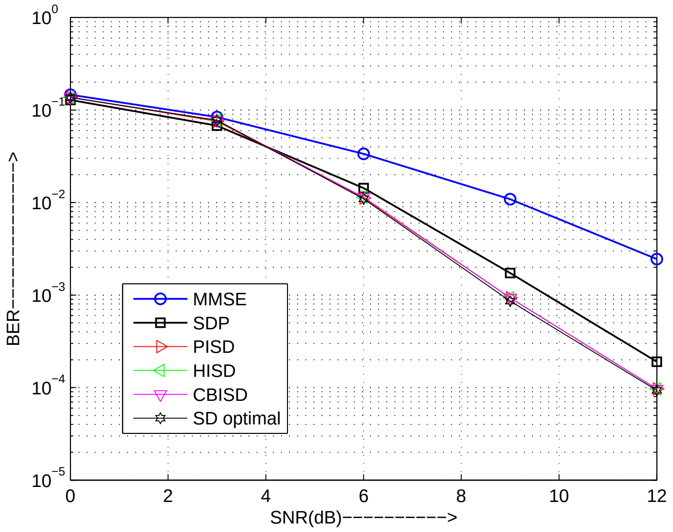


Figure 4.19: Comparison between SD detection, GISD variants ($D = C = 4$) and others sub-optimum detector in a Rayleigh fading channel ($N_u = 16$ users)

i.e., at each time i, the received symbol power is equal to the transmitted symbol power. Each user symbol is spread over $L_c = N_p = 16$ sub-carriers with a real Walsh-Hadamard sequence. Figure 4.19 shows the performance of GISD variants in a Rayleigh fading channel for a fully loaded downlink MC-CDMA system with $N_u = 16$ users and employing an uncoded 4-QAM modulation. It can be seen the excellent performance of the proposed detector using $D = 4$ and $C = 4$. In fact, the required SNR for a BER of 10^{-3} is 0.8 dB lower than that of SDP detector.

4.7 Channel Estimation Errors

In previous study, we always assumed that we have perfect channel knowledge at the receiver, which allows us to compare the performance of different decoders. However, the channel information is typically not perfect. A channel estimator extracts from the received signal approximate channel coefficients during the transmission. One method to accomplish this is to transmit pilot tones prior to the transmission, by turning off all transmitter antennas except the i^{th} antenna at some time instance and sending a pilot signal using i^{th} antenna. The fading coefficients $\tilde{h}_{ij}$ are then estimated. Another way to estimate the channel fading coefficients is to embed the pilot bits inside the signal or send an orthogonal sequence.

4.7.1 Error Model

The impact from the channel estimation errors will degrade the performance of the system. To study the impact of the channel estimation errors on the Geometrical Hypercube Intersection and Selection detectors algorithm, we introduce the error model at the receiver [ZO03]

$$\tilde{\mathbf{H}} = \gamma\tilde{\mathbf{H}}_0 + \sqrt{(1-\gamma^2)}\tilde{\mathbf{E}} \tag{4.21}$$

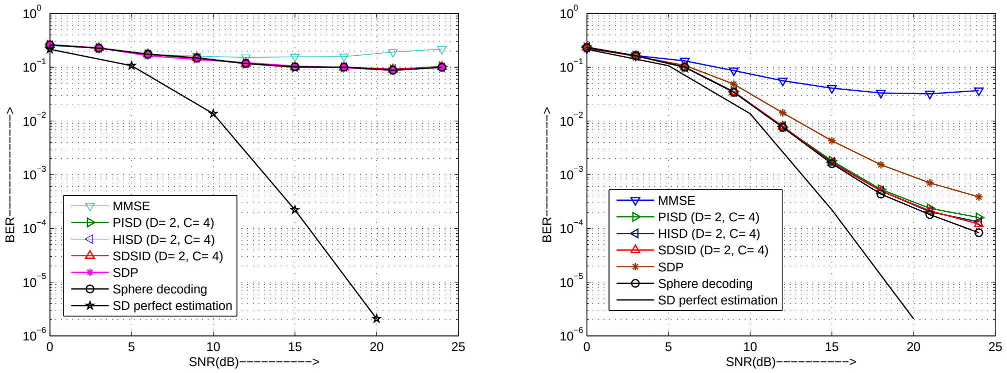
where $\tilde{\mathbf{H}}_0$ represent the true channel matrix and $\tilde{\mathbf{E}}$ denotes the channel estimation error. The elements of $\tilde{\mathbf{E}}$ are assumed to be zero mean, unit variance and complex Gaussian. Here,$\gamma \in [0, 1]$ is a measurement of how accurate the channel estimation is. The value $\gamma = 1$ indicates no estimation decreases.

4.7.2 Simulation Results

As shown in Figure 4.20(a) and 4.20(b), the channel estimation errors with different γ have given the sphere decoding algorithm with adaptive radius almost the same bit error probability as the maximum-likelihood decoder. In this simulation, we compared the performance of the sphere decoding algorithm against GISD and other decoders including SDP and the MMSE, where codewords are modulated using 4-QAM modulator, the number of the transmitter antennas $N = 5$ and the receiver antennas $M = 5$. Both the sphere decoders and the GISD decoder outperform the SDP decoder as well as MMSE decoder. Although we change the value of γ, it is interesting to note that the performance of the proposed GISD detector is usually better than SDP detector.

4.8 Conclusion

A method for quasi-maximum likelihood decoding based on intensification and diversification is introduced. The proposed GISD provides a wealth of trade-off between the complexity and the performance. The GISD allows a near-ML performance with constant polynomial-time, *i.e.* $\mathcal{O}(n^3)$, computational complexity (unlike the SD that has exponential-time complexity for low SNR). Simulation



(a) Bit error rate with channel estimation error, $\gamma = 0.9$

(b) Bit error rate with channel estimation error, $\gamma = 0.99$

Figure 4.20: BER versus SNR. Comparison of different detector methods (SD, PISD, HISD, SDSID, MMSE and SDP) with the same channel estimation error for a 5×5 uncoded MIMO system, iid channel matrix assumption.

results have shown that the GISD detector introduced only a small performance degradation compared to ML detector. The effect of channel estimation error on proposed detector performances was evaluated.

The inherent parallelism of GISD detector allows high throughput, low complexity and low latency decoder. The new approach method can be efficiently employed in the case of CDMA, MC-CDMA, MIMO systems.

Chapter 5

Extended GISD detector

In the previous chapter we have developed a robust detection technique, called GISD, based on a geometrical approach to resolve the maximum likelihood detection problem. By searching only over two or three smallest singular directions, this method, provides a good compromise between computational complexity and BER performances comparing to the sphere decoder.

The most computation complexity of the GISD detector is concentrated in the pre-processing part. In fact, this step needs calculation of the unconstrained ML solution given by $\rho_{zf} = \mathbf{H}^{+}\mathbf{y}$ and extraction of the D smallest singular vectors of the channel matrix $\mathbf{H}$ using singular value decomposition. Nevertheless, recently work in [SG01] gives a simple method, based on the fixed-step gradient descent, to extimate jointly ρ_{zf} and the first smallest singular vector.

In section 5.1, we develop a new method to estimate the uncontrained ML solution and the D smallest singular vector of channel matrix. This technique is based on the fixed-step gradient descent, deflation method and Rayleigh quotient iteration. In section 5.2, motivated by the success of the GISD in demodulating BPSK signaling, we investigate the application of the geometrical approach detection for 16 quadrature amplitude modulation (16-QAM). Also, we explain why the proposed method gives a poor performance. Section 5.3 proposes a new soft quasi-ML detector that maximizes the log-likelihood function by deploying the GISD. A list geometrical intersection and selection detector is presented to yield soft-decision output by storing a list of symbol sequence candidates. However, the computation complexity of the new soft-output detector is marginally increased compared to the original GISD algorithm.

5.1 Pre-processing complexity reduced

Motivated by Spasojevic work in [SG01], this section gives an extended method to reduce the complexity of the GISD pre-processing step which consists of:

- A simple zero-forcing (ZF) detection

  $$\rho_{zf} = \mathbf{H}^{+}\mathbf{y}$$

  where $\mathbf{H}$ is an $m \times n$ real channel matrix, $\mathbf{y}$ is the received signal of length m, $\mathbf{H}^{+} = (\mathbf{H}^{T}\mathbf{H})^{-1}\mathbf{H}^{T}$ denotes the Moore-Penrose pseudo-inverse of the matrix $\mathbf{H}$. The complexity of finding the ZF estimation is essentially determined by the complexity of finding the $\mathbf{H}^{+}$ matrix. The simplest and direct way of calculating the pseudo-inverse is by means of QR factorization. The complexity of ZF detector is of cubic order, $\mathcal{O}(n^3)$ when $m = n$.

- Extract the D smallest eigenvectors of Gram matrix $\mathbf{G} = (\mathbf{H}^{T}\mathbf{H})$ which corresponded to the D smallest right singular vectors of channel matrix $\mathbf{H}$. A direct method using the Singular Value Decomposition (SVD) can be used. However, the brute-force computation complexity of this method is $\mathcal{O}(n^3)$ and the required space to store the data structure is $\mathcal{O}(mn + 2n^2)$.

The complexity of GISD pre-processing step can be reduced using more other efficient and simple methods. Furthermore, the eigenvectors of $\mathbf{G}$ are not functions of the received signal. Both the linear operator required for estimation of ρ_{zf} and the eigenvectors of $\mathbf{G}$ can be precomputed. Some classical methods that can be used for largest eigenvector estimation are the Power's method, Rayleigh's method, the inverse iterations, the Arnoldi method, and the Lanczos method (see Appendix B for more information on these methods). However, these methods are not able to directly calculate the unconstrained estimation ρ_{zf} and the D smallest eigenvectors of matrix $\mathbf{G}$ needed by the GISD detector.

5.1.1 Fixed-step gradient descent method

In a recent work, [SG01] investigates a simple iterative method based on the fixed-step gradient descent algorithm to joint by estimate the smallest singular vector of channel matrix $\mathbf{H}$ and the unconstrained minimizer ρ_{zf} of the cost function

$f(\mathbf{x}) = \|\mathbf{y} - \mathbf{Hx}\|_2^2$. In the following, we briefly give the overview of the theory of such method.

The minimization of the cost function $f()$ over $\mathbb{R}^n$ is given by an optimal solution

$$\begin{aligned} \rho_{zf} &= \arg\min_{\mathbf{a}\in\mathbb{R}^n} f(\mathbf{a}) \\ &= \arg\min_{\mathbf{a}\in\mathbb{R}^n} \|\mathbf{y} - \mathbf{Ha}\|_2^2 \\ &= \arg\min_{\mathbf{a}\in\mathbb{R}^n} \mathbf{a}^T\mathbf{H}^T\mathbf{Ha} - 2\mathbf{y}^T\mathbf{Ha} + \mathbf{y}^T\mathbf{y} \end{aligned} \tag{5.1}$$

The iteration steps of the fixed-step gradient descent algorithm

$$\mathbf{a}^{k+1} = \mathbf{a}^k + \mu\mathbf{g}(\mathbf{a}^k) \tag{5.2}$$

where $\mathbf{g}(\mathbf{a}) = -\frac{\partial}{\partial\mathbf{a}}f(\mathbf{a})$ denotes the gradient of $f(a)$, $\mathbf{a} \in \mathbb{R}^n$, and $\mu > 0$ is a design constant that needs to be set to a sufficiently small value to assure convergence. Note that, at the stationary point of the gradient algorithm ρ_{zf}; $\mathbf{g}(\rho_{zf}) = 0$ and, therefore, ρ_{zf} is also a stationary point of the cost function.

The fixed step gradient descent method's iteration mapping function $\mathbf{M}^{\mathbf{g}}(\mathbf{a})$ is

$$\mathbf{M}^{\mathbf{g}}(\mathbf{a}) = \mathbf{a} + \mu\mathbf{g}(\mathbf{a}) \tag{5.3}$$

then $\mathbf{M}^{\mathbf{g}}(\mathbf{a})$ can be approximated with the first two terms of its Taylor expansion in the neighborhood of ρ_{zf} defined as $\mathcal{U}(\rho_{zf}) \subset \mathbb{R}^n$. That is, for any $\mathbf{a}^k \in \mathcal{U}(\rho_{zf})$; the following approximation holds:

$$\mathbf{M}^{\mathbf{g}}(\mathbf{a}^k) \approx \rho_{zf} + \mathbf{J}^{\mathbf{g}}(\rho_{zf})(\mathbf{a}^k - \rho_{zf})$$

where $\mathbf{J}^{\mathbf{g}}(\mathbf{a}) \triangleq \frac{\partial}{\partial\mathbf{a}}\mathbf{M}^{\mathbf{g}}(\mathbf{a})$ is the Jacobian matrix of the mapping $\mathbf{M}^{\mathbf{g}}(\mathbf{a})$. Or, equivalently,

$$(\mathbf{a}^{k+1} - \rho_{zf}) \approx \mathbf{J}^{\mathbf{g}}(\rho_{zf})(\mathbf{a}^k - \rho_{zf})$$

Let defined $\delta\mathbf{a}^k = \mathbf{a}^k - \rho_{zf}$. It is now clear that $\delta\mathbf{a}^{k+1} = \mathbf{J}^{\mathbf{g}}(\rho_{zf})\delta\mathbf{a}^k$ defines the iterations of the power method (see Appendix B) for computation of the largest eigenvector of the Jacobian matrix $\mathbf{J}^{\mathbf{g}}(\rho_{zf})$. Thus, for a sufficiently large k.

$$\mathbf{a}^k = \rho_{zf} + c_p\lambda_{max}[\mathbf{J}^{\mathbf{g}}(\rho_{zf})]$$

follows from the properties of the power method for some constant c_p. Here $\lambda_{max}[\mathbf{J^g}(\rho_{zf})]$ denotes the maximum eigenvector of $\mathbf{J^g}(\rho_{zf})$. That is, $\mathbf{a}^k$ is a point on the line defined by ρ_{zf} and the largest eigenvector of $\mathbf{J^g}(\rho_{zf})$. The Jacobian of the fixed step gradient descent iteration mapping can be expressed in terms of the Hessian of the cost function as

$$\mathbf{J^g}(\mathbf{a}) = \mathbf{I} - \mu\mathbf{K}(\mathbf{a})$$

where $\mathbf{K} = \frac{\partial^2}{\partial\mathbf{a}\partial\mathbf{a}^T}f(a)$ denotes the Hessian matrix of the cost function $f(a)$. It is clear that $\mathbf{K}(\mathbf{a})$ and $\mathbf{J^g}(\mathbf{a})$ have equal eigenvectors and that their eigenvalues have the following relationship:

$$\lambda_i[\mathbf{J^g}(\mathbf{a})] = 1 - \mu\lambda_i[\mathbf{K}(\mathbf{a})]$$

where $\lambda_i[\mathbf{K}(\mathbf{a})]$ denotes that i^{th} eigenvalue of matrix $\mathbf{K}$. For convergence it is required that

$$|\lambda_i[\mathbf{J^g}(\mathbf{a})]| = |1 - \mu\lambda_i[\mathbf{K}(\mathbf{a})]| < 1$$

The necessary and sufficient condition for the convergence of the fixed-step gradient descent algorithm is that the step-size parameter μ satisfy the double inequality [Hay02]

$$0 < \mu < \frac{2}{|\lambda_{max}[\mathbf{J^g}(\mathbf{a})]|}$$

A practical requirement for convergence is $\mu \leq 2/tr[\mathbf{K}(\mathbf{a})]$ where $tr[\mathbf{K}]$ denotes the trace of matrix $\mathbf{K}$. Furthermore, one can see that the largest eigenvector of $\mathbf{J^g}(\mathbf{a})$ corresponds to the smallest eigenvector of $\mathbf{K}(\mathbf{a})$. Finally, the minimum eigenvector can be obtained based on two successive iterates of the fixed step gradient descent method as $\mathbf{v}_1 \approx \mathbf{a}^{k+1} - \mathbf{a}^k$; for a sufficiently large k and $\rho_{zf} = \mathbf{a}^{k+1}$.

5.1.2 An extended of fixed-step gradient descent method

The previous section shows a method which determines ρ_{zf} and the first smallest eigenvector $\mathbf{v}_2$ of matrix $\mathbf{K} = \frac{\partial^2}{\partial\mathbf{a}\partial\mathbf{a}^T}f(a) = \mathbf{G}$ of the cost function $f(a)$. This section describe an extended method to determine the $D-1$ smallest eigenvector $\{\mathbf{v}_2, \mathbf{v}_3, ..., \mathbf{v}_D\}$ of matrix $\mathbf{G}$. The basic idea of the proposed method is the used of deflation and Rayleigh's methods on the Jacobian matrix $\mathbf{J} = \mathbf{I} - \mu\mathbf{G}$ of the mapping $\mathbf{M}^{\mathbf{g}}(\mathbf{a})$. Thus, we found the $(D-1)$ largest eigenvectors of the matrix $\mathbf{J}$ corresponds to the $(D-1)$ smallest eigenvectors of the Hessian matrix $\mathbf{K} = \mathbf{G}$.

5.1.2.1 Deflation method

A well-known technique in eigenvalue methods is the so-called Wielandt deflation [Saa92]. Suppose that we have computed the eigenvalue α_1 of largest modulus and its corresponding eigenvector $\mathbf{z}_1$ of a given matrix $\mathbf{A}$ by some algorithm such as, in the simplest case, the power method. A common problem is to compute the next dominant eigenvalue α_2 of $\mathbf{A}$ and its corresponding eigenvector $\mathbf{z}_2$. An old artifice for achieving this is to use a deflation procedure: a rank one modification of the original matrix is performed so as to displace the eigenvalue α_1 to the origin, while keeping all other eigenvalues unchanged. Thus, the eigenvalue α_2 becomes the dominant eigenvalue of the modified matrix and, therefore, the power method can subsequently be applied to this matrix to compute the next dominant pair $(\alpha_2, \mathbf{z}_2)$. The deflated matrix is of the form

$$\mathbf{B}_1 = \mathbf{A} - \alpha_1 \mathbf{z}_1 \mathbf{z}_1^T$$

thus, $\mathbf{B}_1$ has the same eigenvectors as $\mathbf{A}$, and the same eigenvalues as $\mathbf{A}$ except that the largest one has been replaced by 0. Thus we can use the power method with Rayleighs coefficient to find the next largest eigenvector of $\mathbf{A}$ and so on. In our case, we applicate the deflation technique on the matrix $\mathbf{J} = \mathbf{I} - \mu\mathbf{G}$.

5.1.2.2 Rayleigh method

The Rayleigh quotient iteration (RQI) method presented in Algorithm 2 is the simplest iterative method to finds the eigenvalue of matrix $\mathbf{A}$ which has the largest absolute value and a corresponding eigenvector. In general, Rayleigh quotient iteration will need fewer iterations to find an eigenvalue/eigenvector pair than the power method.

5.1.2.3 Method summary/complexity

In summary, the extended fixed-step gradient descent method requires the following:

1. Estimate ρ_{zf} and the first smallest eigenvector $\mathbf{v}_1$ of matrix $\mathbf{G} = \mathbf{H}^T\mathbf{H}$

Algorithm 2: The RQI method

```
Data: $\mathbf{z}_0$ arbitrary, a matrix $\mathbf{A}$, and $\varepsilon$ : tolerance.
Result: $(\alpha_{max}, \mathbf{z}_{max})$ largest eigenpair of matrix $\mathbf{A}$.
begin
    $\mathbf{z} = \mathbf{z}_0/\|\mathbf{z}_0\|_2$
    $flag = 1$
    while $flag == 1$ do
        $\mathbf{y} = \mathbf{A}\mathbf{z}$
        $\alpha = \mathbf{z}^T\mathbf{y}$
        $\mathbf{y} = \mathbf{y}/\|\mathbf{y}\|_2$
        $err = \mathbf{y} - \mathbf{z}$
        if $\|err\|_2 < \varepsilon$ then
            $flag = 0$
        $\mathbf{z} = \mathbf{y}$
    $\alpha_{max} = \alpha$
    $\mathbf{z}_{max} = \mathbf{z}$
end
```

using the fixed-step gradient descent algorithm appendix C.3. Let's $\mathbf{J}_{k=1} = \mathbf{I} - \mu\mathbf{G}$.

2. Generate $\mathbf{J}_{k+1} = \mathbf{J}_k - \alpha_k \mathbf{v}_k \mathbf{v}_k^T$ using the deflation procedure on the matrix $\mathbf{J}_k$ and its largest eigenpair $(\mathbf{v}_k, \alpha_k)$.

3. Use the rayleigh method to finds the largest eigenvector of matrix $\mathbf{J}_{k+1}$ corresponds to the $(k+1)^{th}$ smallest eigenvector of the matrix $\mathbf{G}$.

4. Repeted step 2 and 3 until we find the $D-1$ smallest eigenvectors of the matrix $\mathbf{G}$.

Regarding the complexity of the proposed method, the following considerations hold. In every step, we must multiply the $n \times 1$ vector $\mathbf{z}$ to the $n \times n$ matrix, which is possible with complexity $\mathcal{O}(n^2)$.

5.1.3 Performances of the extended method

In this section, we compare the performance of HISD detector (using channel inversion matrix and SVD decomposition to determinate the vector ρ_{zf} and the D smallest eigenvectors of the channel matrix $\mathbf{H}$) and an modified HISD detector called EHISD (the extended fixed-step gradient descent method is used to

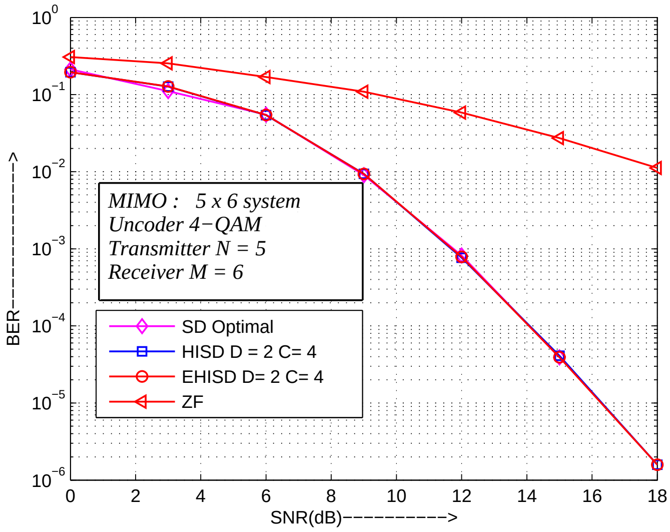


Figure 5.1: BER versus SNR, comparison between sphere decoder ,EHISD ($D = 2$ and $C = 4$), and HISD ($D = 2$ and $C = 4$)

estimate jointly ρ_{zf} and the D smallest singular vectors of the channel matrix $\mathbf{H}$) under $N = 5$ transmit antennas and $M = 6$ receive antennas MIMO system. Perfect channel estimation is assumed. The Rayleigh flat fading channel is considered. The channel coefficients are modeled with an independent zero mean complex Gaussian random variable with variance 0.5 per dimension. Figure 5.1 shows that the EHISD has the same BER performance as the HISD detector. However, its computation complexity of pre-processing step is smaller than the HISD.

5.2 16-QAM GISD detection

Consider the standard linear channel model

$$\tilde{\mathbf{y}} = \tilde{\mathbf{H}}\tilde{\mathbf{x}} + \tilde{\mathbf{w}} \tag{5.4}$$

where $\tilde{\mathbf{y}}$ is the received signal of length M, $\tilde{\mathbf{H}}$ is an $M \times N$ channel matrix, $\tilde{\mathbf{x}}$ is the length N vector of transmitted symbols,and $\tilde{\mathbf{w}}$ is a length M complex normal zero-mean noise vector with covariance $\sigma^2\tilde{\mathbf{I}}$. The symbols of $\tilde{\mathbf{x}}$ belong to some known complex constellation (16-QAM), *i.e.*, the real part and the imaginary part of $\tilde{\mathbf{x}}_i$ for $i = 1, 2, .., N$ belong to the set $\{\pm 1, \pm 3\}$. The real-valued equivalent of the model (5.4) is given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{5.5}$$

where $\mathbf{H}$ is an $m \times n$ real valued channel matrix with $m = 2M$ and $n = 2N$. Assuming that $\mathbf{H}$ is known at the receiver, the optimal detector minimizing the average error probability of detection is given by the Maximum Likelihood (ML) detector which solves the following combinatorial optimization problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x} \in \{\pm 1, \pm 3\}^n} \|\mathbf{y} - \mathbf{H}\mathbf{x}\|_2^2 \tag{5.6}$$

The problem (5.6) is a combinatorial problem and can be solved in a brute-force fashion by searching over all of the 4^{2N} possibilities. Clearly, as N increases, this option becomes impractical. Thus, we propose an approximate solution to the problem via geometrical approach using the GISD detector.

5.2.1 Extended GISD

To further improve the search of the sub-optimal solution in the case of 16-QAM modulation, The GISD detector based on an intensification stategy (concentrate the search in a localized region) and a diversification stategy (direct the search to unexplored regions) can be used as following:

- $diversification$: As shown in chapter 3, the idea of the diversification step is to choose the feasible points in the "vicinity" of the lines $\triangle_1, ..., \triangle_D$ in order to create a starting subset ξ_{start} which contains *a priori* good and promising starting point. A first level of diversity is then obtained by the use of the D independent lines. Then, for each line $\triangle_k$, $k = 1, ..., D$ a second level is obtained by the intersection between $\triangle_k$ and the faces of the hypercube having ξ as vertices in the case of hypercube intersection and selection detector variant. Given a line $\triangle_k$, there are at most $4n$ feasible points $\mathcal{I}_k \subset \{\pm 1, \pm 3\}^n$ representing the projection on search space of all intersection points between $\triangle_k$ and $\mathcal{PH} = \{z \in \mathbb{R}^n / z(i) = s\}_{s=-3,-1,1,3}^{i=1..,n}$. In figure 5.2, we show all intersections points between $\mathcal{PH}$ and the k^{th} line directed by the k^{th} smallest singular vector of the channel matrix $\mathbf{H}$.

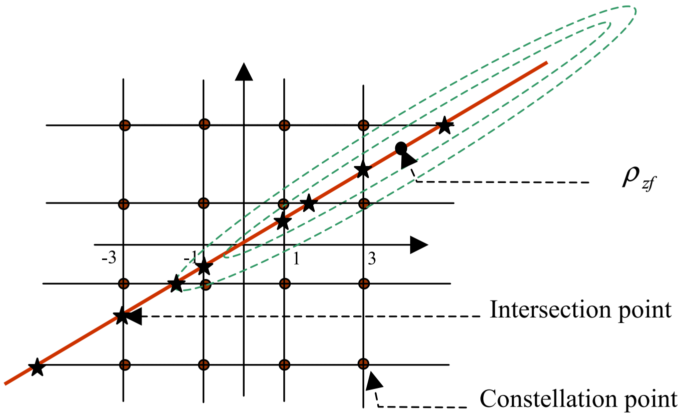


Figure 5.2: Diversification using hypercube intersection and selection variant for 16-QAM modulation and n=2

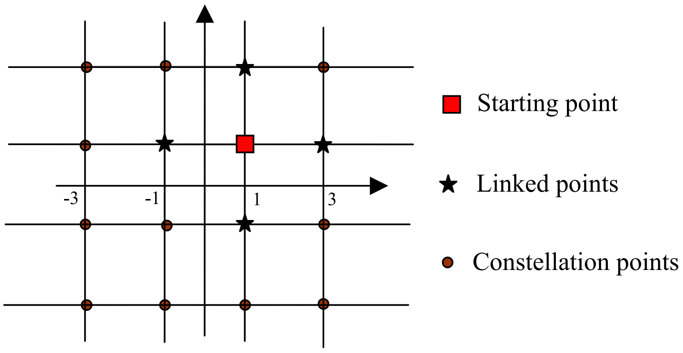
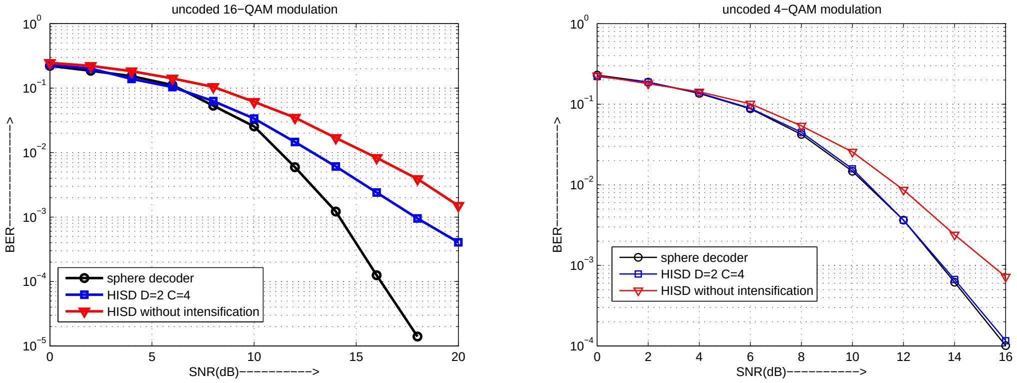


Figure 5.3: Two dimensional visualization of a $Q = 1$ neighborhood in case of 16-QAM modulation

- *intensification*: Intensification refers to focusing on the search on promising start solutions. Hence, we choose to apply a simple greedy search procedure as described in chapter 3. In figure 5.3, we illustrate the points linked to starting point $\mathbf{x}^0$ where $Q = 1$ for 2-dimensional case.

5.2.2 Simulation results

The bit error rate of the modulation 16-QAM is presented in figure 5.4(a) for five receive antennas and five transmit antennas uncoded MIMO system. We compare BER performances of the proposed HISD detector where $D = 2$ and $C = 4$, the sphere decoder (optimal detector), and the HISD without the intensification phase. We note that the performance of the HISD is dramatically bad as compared to that of the sphere decoding. In fact, the required SNR for a BER of 10^{-3} is 3.8 dB higher than that of SD detector. One of the reason of this bad performance can be explained by the high number of local minima makes any approach with greedy search methods doomed to failure. The second reason is that diversification step don't give a very good promising start points (see figure 5.4(b)).



(a) Performance of the SD, HISD and HISD without intensification where uncoded 16-QAM modulation is used

(b) Performance of the SD, HISD and HISD without intensification where uncoded 4-QAM modulation is used

Figure 5.4: Performance comparison for $M \times N$ uncoded MIMO systems with $N = M = 5$, $C = 4$ and $D = 2$.

5.3 Soft-output detection

In wireless communications, quite often we wish to estimate the $n \times 1$ information bearing symbol vector $\mathbf{s}$ from the $m \times 1$ data vector $\mathbf{y}$ in the block coding model

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{w} \tag{5.7}$$

where $\mathbf{s}$ has entries belonging to a finite alphabet $\tilde{\mathcal{A}}$, $\mathbf{H}$ is a known $m \times n$ real or complex matrix, and $\mathbf{w}$ is a $m \times 1$ Gaussian noise vector. This problem is encountered in many applications including single-antenna block transmissions, space-time (ST) multi-antenna transmissions, or, in multi-user detection of CDMA, MC-CDMA transmissions. When $\mathbf{s}$ is drawn from QPSK or rectangular QAM constellations, the block coding model (5.7) can be easily transformed to a real model where $\mathbf{s}$, $\mathbf{y}$, $\mathbf{H}$ and $\mathbf{w}$ all belong to the real field. Since in this paper we only focus on QPSK and rectangular QAM signalling, we assume that (5.7) is a real model.

When $\mathbf{s}$ is uncoded and $\mathbf{w}$ is white Gaussian, the optimal solution of (5.7) in the sense of minimizing the bit error rate (BER) is offered by the maximum likelihood (ML) decoder. However, when $\mathbf{s}$ is coded with some kind of error control code (ECC), soft iterative detection is capable of approaching the ultimate performance limit dictated by the capacity of the channel $\mathbf{H}$ using maximum a posteriori (MAP) decoding. Such an approach has been followed in [HB03], where a soft, so called List Sphere Decoding (LSD) algorithm, has been derived to compute the extrinsic information based on a list of candidates obtained inside a preset sphere.

To generate soft information in the case of GISD algorithm, a certain number of candidates points including GISD solution are required. Thus, GISD can be modified to be a List Geometrical Intersection and Selection Decoder, which finds a list of most likely points, and the soft information can then be generated based on these points [HB03]. Since generating such information increases the computational complexity for selecting a specific number of candidate points N_c, we can use efficient architecture of the GISD (pipelining and parallelism) for high throughput. Suppose that each entry $\mathbf{s}(m)$ of the symbol vector $\mathbf{s}$ in (5.7) is obtained by mapping a $Q \times 1$ binary vector $\mathbf{x}_{<m>}$ with ± 1 entries, and let $\mathbf{x} = [(\mathbf{x}^T_{<1>}, \mathbf{x}^T_{<2>}, \ldots.., \mathbf{x}^T_{<n>}]^T$. The MAP decoder for obtaining $\mathbf{x}$ from $\mathbf{y}$ minimizes the bit error rate (BER) by evaluating the log-likelihood ratio (LLR) [HOP96] of the a posteriori probability of each bit $\mathbf{x}(k)$

$$L_D(\mathbf{x}(k)|\mathbf{y}) = \ln \frac{P[\mathbf{x}(k) = +1|\mathbf{y}]}{P[\mathbf{x}(k) = -1|\mathbf{y}]} \tag{5.8}$$

Assume $\{\mathbf{x}(k)\}$ are independent due to the random interleaver, Equation (5.8) can be further expressed as:

$$L_D(\mathbf{x}(k)|\mathbf{y}) = L_A(\mathbf{x}(k)) + \ln \frac{\sum_{\mathbf{x}\in\mathbb{X}_{k,+1}} P[\mathbf{y}|\mathbf{x}] \cdot \exp\{\frac{1}{2}\mathbf{x}^T_{[k]} L_{A,[k]}\}}{\sum_{\mathbf{x}\in\mathbb{X}_{k,-1}} P[\mathbf{y}|\mathbf{x}] \cdot \exp\{\frac{1}{2}\mathbf{x}^T_{[k]} L_{A,[k]}\}} \tag{5.9}$$

where $\mathbb{X}_{k,+1} := \{\mathbf{x}|\ \mathbf{x}(k) = +1\}$, $\mathbb{X}_{k,-1} := \{\mathbf{x}|\ \mathbf{x}(k) = -1\}$, $L_A(\mathbf{x}(k)) = \frac{P[\mathbf{x}(k)=1]}{P[\mathbf{x}(k)=-1]}$ denoting the *a priori* information of $\mathbf{x}(k)$, $\mathbf{x}_{[k]}$ is the sub-vector of $\mathbf{x}$

obtained by omitting its k^{th} element $\mathbf{x}(k)$, and likewise $L_{A,[k]}$ is obtained from L_A by omitting its k^{th} element $L_A(\mathbf{x}(k))$.

Since $\mathbf{w}$ is white Gaussian, using the max-log approximation [RVH95], we can approximate the *extrinsic* information of $\mathbf{x}(k)$ as [HB03]

$$\begin{aligned} L_E(\mathbf{x}(k)|\mathbf{y}) \;\approx\; & \frac{1}{2}\max_{\mathbf{x}\in\mathbb{X}_{k,+1}}\{\frac{-1}{\sigma^2}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|_2^2+\mathbf{x}_{[k]}^T L_{A,[k]}\} \\ & -\frac{1}{2}\max_{\mathbf{x}\in\mathbb{X}_{k,-1}}\{\frac{-1}{\sigma^2}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|_2^2+\mathbf{x}_{[k]}^T L_{A,[k]}\} \end{aligned} \quad (5.10)$$

Unfortunately, even with simplification, computing $L_E(\mathbf{x}(k)|\mathbf{y})$ is exponential in the length of the bit vector $\mathbf{x}$ or the number of symbols in the constellation $\tilde{\mathcal{A}}$: To find the maximizing hypotheses in (5.10) for each $\mathbf{x}(k)$, there are 2^{nQ-1} hypotheses to search over in each of the two terms. For even a moderate block size n, or bits per symbol Q, this complexity may be overwhelming.

We are interested in computing (5.10). Finding the maximum likelihood estimate $\hat{\mathbf{s}}$ does not necessarily help, because, although it is the estimate that makes $f(\mathbf{s}) = \|\mathbf{y}-\mathbf{H}\mathbf{s}\|_2^2$ smallest, it is not necessarily the estimate that maximizes the two terms in (5.10).

However, The GISD structure generate a list $\mathfrak{L}$ of the N_c points $\mathbf{s}$ that make $f(\mathbf{s})$ smallest. In addition, this list contains the maximizer of (5.10) with high probability. Hence, $\mathfrak{L}$ contains the *GISD* estimate and N_c-1 neighbors for which $f(\mathbf{s})$ is smallest. The "soft" information about any given bit $\mathbf{x}(k)$ is essentially contained in $\mathfrak{L}$ because if there are many entries with $\mathbf{x}(k) = 1$ then it can be concluded that the likely value for $\mathbf{x}(k)$ is indeed one, whereas if there are few entries in $\mathfrak{L}$ with $\mathbf{x}(k) = 1$, then the likely value is minus one.

Equation (5.10) is approximated using $\mathfrak{L}$ as

$$\begin{aligned} L_E(\mathbf{x}(k)|\mathbf{y}) \;\approx\; & \frac{1}{2}\max_{\mathbf{x}\in\mathfrak{L}\cap\mathbb{X}_{k,+1}}\{\frac{-1}{\sigma^2}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|_2^2+\mathbf{x}_{[k]}^T L_{A,[k]}\} \\ & -\frac{1}{2}\max_{\mathbf{x}\in\mathfrak{L}\cap\mathbb{X}_{k,-1}}\{\frac{-1}{\sigma^2}\|\mathbf{y}-\mathbf{H}\mathbf{s}\|_2^2+\mathbf{x}_{[k]}^T L_{A,[k]}\} \end{aligned} \quad (5.11)$$

where $\mathbf{s} = map(\mathbf{x})$. For different LGISD variants, the number of candidate points N_c is alway equal to $DCn\theta$ where D denote the number of studied slowest direction, C define the number of candidate point at each slowest direction, and θ is the iteration number of greedy search method.

The LGISD is a generalization of the GISD. Rather than finding only the best argument, if finds the best N_c ones. It stores each of these arguments $\mathbf{x}_p$ and their corresponding value v_l in a list $\mathfrak{L} = \{\mathbf{x}_p, v_l\}$ for $l = 1...N_c$. The LGISD is implemented by modifying the GISD algorithm. Each time a possible argument is found, the LGISD checks whether it is better than any of the arguments in the list, and if so it exchanges them. This search requires an order of N_c comparisons, and is executed for every vector checked in the intensification step.

5.3.1 Simulation results

In this section, we present simulations using a parallel concatenated (turbo) ECC with rate $R = 1/2$, as in [BGT93]. Each constituent convolutional code has memory 2, feedback polynomial $G_r(D) = 1+D+D^2$, and feedforward polynomial $G(D) = 1 + D^2$.The interleaver size of the turbo code is 512 information bits. We choose the number of inner iterations for the turbo decoding module to be 10. As in [HB03], we generate independent Rayleigh flat fading channels between transmit/receive antennas and assume perfect channel estimation at the receiver end.

Consider a BLAST system with N transmit and M receive antennas as shown in figure 5.5. Figures 5.6 and 5.7, respectively, show the diagrams of the BLAST transmitter and receiver. The information bits $\mathbf{b}$ are first encoded by an ECC module to yield $\mathbf{c}$, and then go through a random interleaver. Interleaved bits $\tilde{\mathbf{c}}$ are mapped to 4-QAM symbols. 4-QAM symbol vector $\mathbf{x}$ is transmitted using BLAST scheme. At the receiver end, the soft detector first generates the bit metrics and then rearranges them.

In the 8×8 MIMO system case, figure 5.8(a) compares the performance of proposed soft output LGISD versus the List sphere decoding (LSD) with candidate lists of maximal length $N_{cand} = 1024$ as show in [HB03] and the shifted spherical list APP detector [BGBF03]. It is seen that for the 8×8 scenario the shifted spherical list APP detector has a slightly better performance than the LGISD decoder with parameters ($D = 3$ and $C = 4$).

In figure 5.8(b), the BER performance between three soft output detectors is compared: List sphere decoding, List geometrical intersection and selection detector, and soft output semidefinite programming. The result presented is for $N = 16$ and $M = 16$ MIMO system. The BER difference is not even noticeable

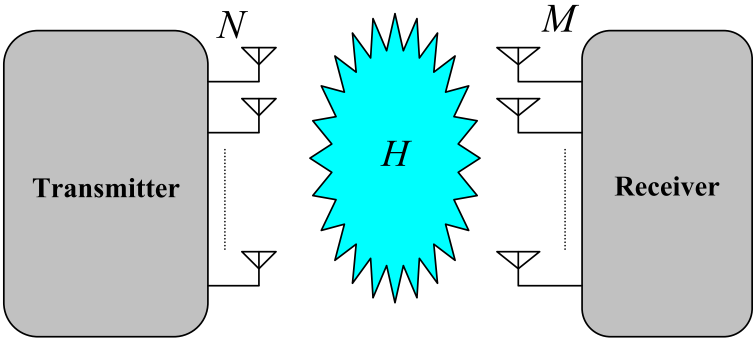


Figure 5.5: Diagram of a MIMO system.

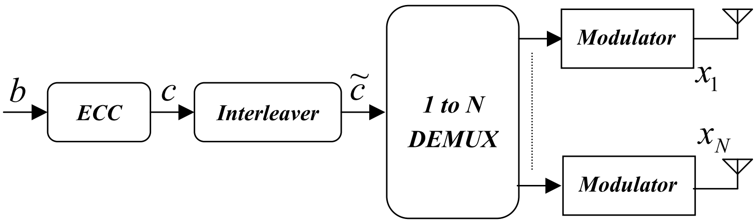


Figure 5.6: Diagram of a BLAST transmitter employing error control code (ECC).

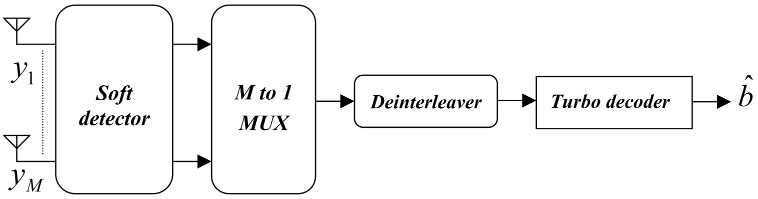
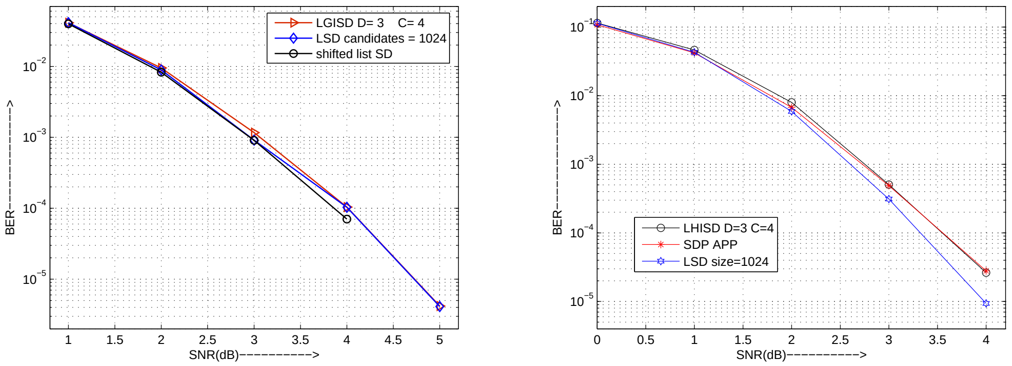


Figure 5.7: Diagram of a BLAST receiver employing turbo decoder.

between the LGISD with parameters $D = 3$ and $C = 4$ and the soft output SDP. At the bit error rate of 10^{-4}, LGISD performance is only less than 0.2 dB from the LSD with candidate lists length $N_{cand} = 1024$.



(a) Performance of LGISD, LSD and shifted list sphere decoder, $m = n = 8$

(b) Performance of LGISD, LSD and soft output SDP detector, $m = n = 16$

Figure 5.8: BER curves as a function of SNR for $m \times n$ channel transmitting 4-QAM with $R = 1/2$ memory 2 turbo code.

5.4 Conclusion

In this chapter, we extended the application of the fixed-step gradient descent method for joint estimation of D slowest descent directions and the unconstrained minimizer ρ_{zf}. The developed method overcome the high computation of the SVD decomposition and the matrix inversion suggested in the first version of the GISD detector. The technique of intensification and diversification has been applicated to 16-QAM modulation, the result (figure 5.4(a)) have demontrated that the diversification step don't give a very good promising start points. This unresolved problem will be one of the topics for future work. The GISD has been extended to generate a soft output decision for forward error correcting codes.

Chapter 6

VLSI implementation of GISD detection

The GISD have potential to be of great use in future wireless communications systems due to its ability to greatly reduce the size of the exponential search space that needs to be processed. However, for this to be useful, it must also be practical for implementation in very large scale integrated (VLSI) circuits. The parallelism of the algorithm is explored based on the data dependency analysis and an efficient hardware architectures are developed with two levels of parallelisms:

1. The parallel execution of the D geometrical intersection and selection modules, where D denotes the number of studied directions.
2. The parallel execution of the C greedy search modules, where C denotes the number of best starting solutions at each studied directions.

In this chapter, An implementation on a FPGAs/DSPs multiprocessor motherboard of the GISD detection technique is discussed. Section 6.1 is mainly focussed on a comparative performance/complexity study where other norms are used in both intensification and diversification steps. The impacts of GISD parameters on performances and computation complexity are looked in section 6.2. The rapid prototyping platform, used for hardware implementation, was presented in section 6.3. Section 6.4 investigates various quantization schemes of the GISD. Moreover, a finite word length analysis for an uncoded 5×5 MIMO systems is given. The GISD detection block diagram corresponding to case where the parameter $D = 1$ is presented in section 6.5. Moreover, a globally-asynchronous locally-synchronous technique, to interconnect different GISD unit's, are presented in section 6.6. The chapter ends with a basic VLSI implementation of the GISD detector where $D = 1$ and $C = 4$.

6.1 Simplified norm algorithm

The simplified norm algorithm was first introduced to sphere decoding in [AMM+05] and can be used to reduce complexity of the GISD algorithm on both the circuit and the algorithmic levels, at the cost of a minor performance degradation. The main idea is to approximate the $l2 - norm$ by a $lp - norm$ respectively according to

$$\begin{aligned} f(\mathbf{x}) &= \|\mathbf{y} - \mathbf{Hx}\|_2^2 \\ &\approx \|\mathbf{y} - \mathbf{Hx}\|_p^p,\ p \neq 2 \end{aligned} \tag{6.1}$$

The three properties that all norms satisfy are:

1. $\|\mathbf{x}\|_p = \sum_{i=1}^n (|\mathbf{x}(i)|^p)^{1/p} \geq 0$, and $\|\mathbf{x}\|_p = 0$ only if $\mathbf{x} = \mathbf{0}$.
2. $\|\alpha\mathbf{x}\|_p = |\alpha|\|\mathbf{x}\|_p$, where α is a scalar.
3. $\|\mathbf{x} + \mathbf{y}\|_p \leq \|\mathbf{x}\|_p + \|\mathbf{y}\|_p$ (triangle inequality).

Suppose we have two norms, Say the p-norm and the q-norm where $p \leq q$, then the following inequality is satisfied

$$1 \leq \frac{\|\mathbf{x}\|_p}{\|\mathbf{x}\|_q} \leq n^{\frac{q-p}{pq}}$$

Since the ratio between any two norms length are bounded below and above by constants, any two norms may be considered equivalent. It can be observed from the inequality that as $q \longrightarrow \infty$, $\frac{q-p}{pq}$ approaches $\frac{1}{p}$ and $\|\mathbf{x}\|_p$ approaches $\|\mathbf{x}\|_\infty$. The most commonly used approximate norms are the $l1$- and $l\infty$-norms which are defined as

- $l1$-norm: The $l1$-norm is the vector norm defined to be the sum of the absolute vector components

$$\|\mathbf{x}\|_1 = \sum_{i=1}^n (|\mathbf{x}(i)|)$$

where $\mathbf{x}$ is a vector of order n. This norm is used most often when looking for the most robust answer since the $l1$-norm is not affected greatly by outliers. An outlier is, given a set of data, an extreme measurement that stands out from the rest of the data and may be an incorrectly recorded observation.

- $l\infty$-norm: The $l\infty$-norm is the vector norm defined to be the maximum value of the absolute component values of the vector.

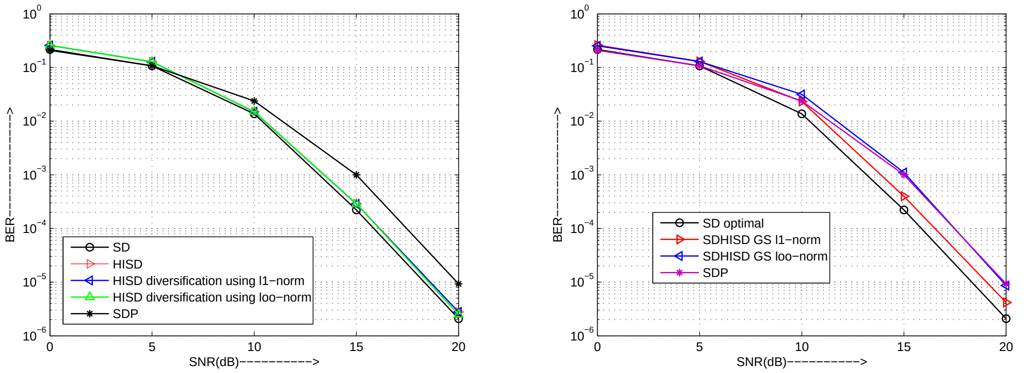
$$\|\mathbf{x}\|_\infty = max(|\mathbf{x}(1)|, |\mathbf{x}(2)|, ..., |\mathbf{x}(n)|)$$

where $\mathbf{x}$ is a vector of order n. This norm is used most often when gross discrepancies are to be avoided with the data.

In order to fully assess the impact of the above described norm approximations on throughput, we shall study the influence of the reduced complexity norms on the different GISD steps.

- $Intensification$: Approximating the $l2$-norm by the $l1$-norm or the $l\infty$-norm on the intensification step results in a modified GISD algorithm that no longer implements an ML detector. The impact on BER of using the $l1$-norm or the $l\infty$-norm instead of the $l2$-norm is quantified in figure 6.1(b) for $N = 5$ transmit antennas and $M = 5$ receive antennas MIMO systems with uncoded 4-QAM modulation. It can be seen that the GISD using the $l1$-norm still have a good BER performance comparing to the SDP detector. In fact, at the bit error rate of 10^{-5}, it is only less than 1 dB from the Sphere detector; whereas the robust SDP is about $1,6$ dB. In table 6.1, we compare the amount of arithmetic operations needed to determine the best neighbors point to a given start feasible point using different norms. Note that θ is the number of iteration of greedy search method typically equal to 2.

- $Diversification$: To select best starting candidate points in the geometrical diversification, we can approximate the $l2$-norm by the $l1$-norm or the $l\infty$-norm. Figure 6.1(a) shows the case of uncoded $M \times N$ MIMO system ($M = N = 5$). As we seen, the HISD variant still shows a better result than SDP not only in BER but also in the complexity that it enjoys. The number of additions/subtractions and multiplications of HISD using different norm in diversification step is expressed in table 6.2.



(a) Using $l1$-norm or the $l\infty$-norm in diversification step

(b) Using $l1$-norm or the $l\infty$-norm in intensification step

Figure 6.1: Performance comparison for 5 × 5 MIMO systems with uncoded 4-QAM modulation, $C = 4$ and $D = 2$.

Table 6.1: Complexity of greedy search function using different norms when $n = 2N$, and θ is the iteration number.

Greedy search	Add / Sub	Mult
$l2$-norm Appendix C.1.1	$\theta n^3 + 2\theta n^2$	$\theta n^3 + \theta n^2$
$l1$-norm Appendix C.1.2	$\theta n^3 + 2\theta n^2$	θn^3
$l\infty$-norm Appendix C.1.3	$\theta n^3 + \theta n^2$	θn^3

Table 6.2: Complexity of diversification step using different norms when $n = 2N$.

HISD Evaluation	Add / Sub	Mult
$l2$-norm Appendix C.2.1	$2n^3 + 4n^2$	$2n^3 + 2n^2$
$l1$-norm Appendix C.2.2	$2n^3 + 4n^2$	$2n^3$
$l\infty$-norm Appendix C.2.3	$2n^3 + 2n^2$	$2n^3$

Table 6.3: Relative Cost per Operation.

Addition	1
Multiplication	10
Division	40
Square Root	50

6.2 GISD Parameters Impact

The BER performance and computational complexity of the GISD detector depends heavily on it's two parameters:

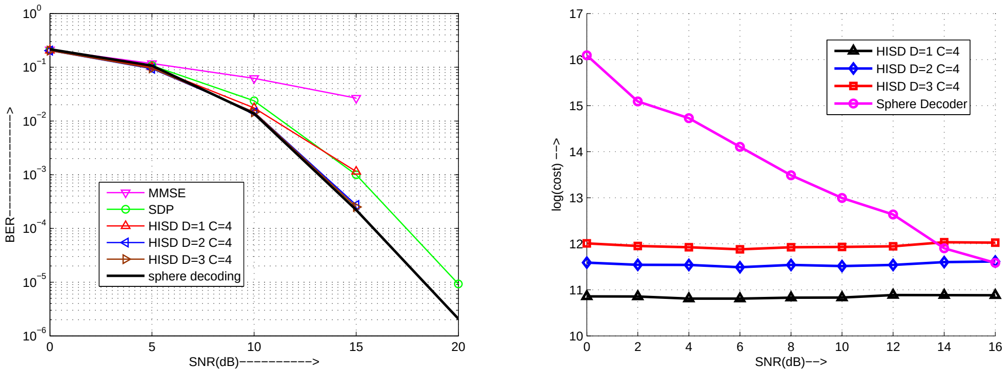
- D: number of smallest right singular vectors of the channel matrix $\mathbf{H}$.
- C: number of best candidates on each smallest direction.

In the following, we looked to parameters impact on the bit error rate (BER) and the computation complexity. For the estimation of the impact of the parameters on the HISD complexity. We will study the complexity of the SD and HISD in their searching process, we assign a relative cost for each operation (addition, multiplication, division, square-root). Table 6.3 show the relative cost assigned to each type of operation. Note that the cost of an addition is set to 1 and all other operation costs are with respect to this baseline value [Net03].

1. *BER/complexity versus D*

   Simulation results comparing the different BER performances when we fix the number of candidate points C to 4 and increase the number of slowest eigenvector directions $D \in \{1, 2, 3\}$ are given in figure 6.2(a). This simulation demonstrates that the HISD with only two search direction $D = 2$ offers a significant performance gain over the SDP detector. Searching one more direction $D = 3$ do not results in some additional performance improvement. Further increase in the number of search directions only results in a diminishing improvement in performance. The result of the comparison of the complexity of the HISD and sphere decoder algorithms is on figure 6.2(b). We can see that the computation complexity of HISD is almost constant compared to the Sphere Decoder.

2. *BER/complexity versus C*



(a) BER performance of HISD detector for a fixed candidates number $C = 4$ and various smallest right singular vectors $D \in \{1, 2, 3\}$

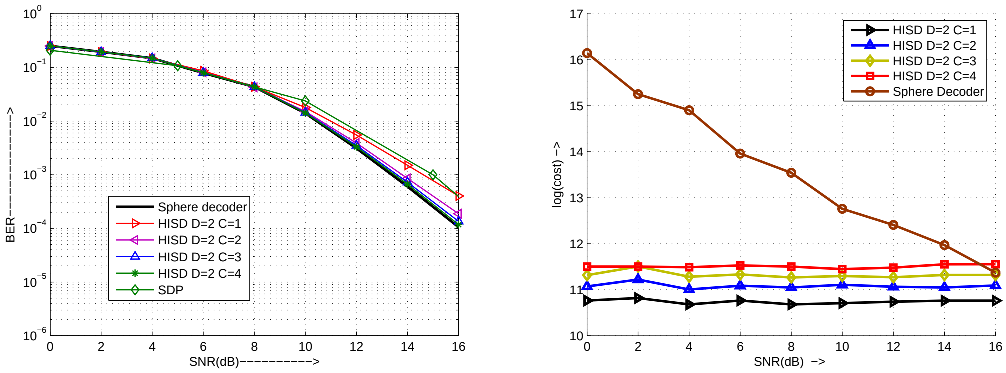
(b) Comparison of the processing complexity of the sphere decoder $radius = \sqrt{\alpha n \sigma}$ [HV02] and HISD algorithms when we increase the parameter D

Figure 6.2: BER/Complexity of HISD detector for 5×5 MIMO systems with uncoded 4-QAM modulation, $C = 4$ and $D \in \{1, 2, 3\}$.

Here, we consider HISD performance for uncoded 4-QAM modulation. As in the case of the previous analysis, we assess the impact on BER performance of various value of parameter C. We fix the number of slowest eigenvector directions D to 2, the number of transmitting antennas N to 5, the number of receiving antennas M to 5, and increase the number of candidate points $C \in \{1, 2, 3, 4\}$. Figure 6.3(a) compares the BER of the sphere decoding with that of the HISD variants as a function of the parameter C. As C increases, the HISD performs close to the SD. The result of the comparison of the complexity of the SD and BB algorithms is on figure 6.3(b)). We can see that the GISD method have a lower complexity compared to the Sphere Decoder.

6.3 Architecture description

Our rapid prototyping platform architecture, named PALMYRE [Bom04], is based on a peripheral component interconnect (PCI) Sundance Multiprocessor motherboard where one DSP-based module and one FPGA module are plugged. As illustrated in Figure 6.4, two different communication formats can be used: a 8-bit bidirectional format, denoted by slow port, allowing 20 Mbps transfer rate,



(a) BER performance of HISD detector for a fixed the smallest right singular vectors number $D = 2$ and increase the number of candidates point $C \in \{1, 2, 3, 4\}$

(b) Comparison of the processing complexity of the sphere decoder $radius = \sqrt{\alpha n \sigma}$ [HV02] and HISD algorithms when we increase the parameter C

Figure 6.3: BER/Complexity of HISD detector for $M \times N$ MIMO systems with uncoded 4-QAM modulation, $N = M = 5$, $n = 2N$, $D = 2$ and $C \in \{1, 2, 3, 4\}$.

and a 16-bit bidirectional format, denoted by quick port, allowing 200 Mbps throughput.

The SW module uses the TMS320C6701 DSP from Texas Instrument. This component is based on a very long instruction word (VLIW) architecture making it possible to compute 8 operations per cycle at a 167MHz frequency. The FPGA is a XC2V2000 Virtex II with 2 Mega system gates. Memory blocks are also available in the FPGA. Dedicated components are used on the SW module to make possible data exchanges between the DSP peripherals and the communication ports. The FPGA is configured using a bitstream sent by a DSP.

6.3.1 Key requirements

- *Scalability*: Communication systems offer a large variety of new signal processing algorithms, in particular new coding and detection strategies. The complexity of these algorithms can become very large, therefore an extend able platform is required. A modular approach, where additional processing power can be added (DSPs and/or FPGAs) is desirable.

- *Flexibility*: The signal processing load is best spread among different processing units. The most common in communications are FPGAs, DSPs.

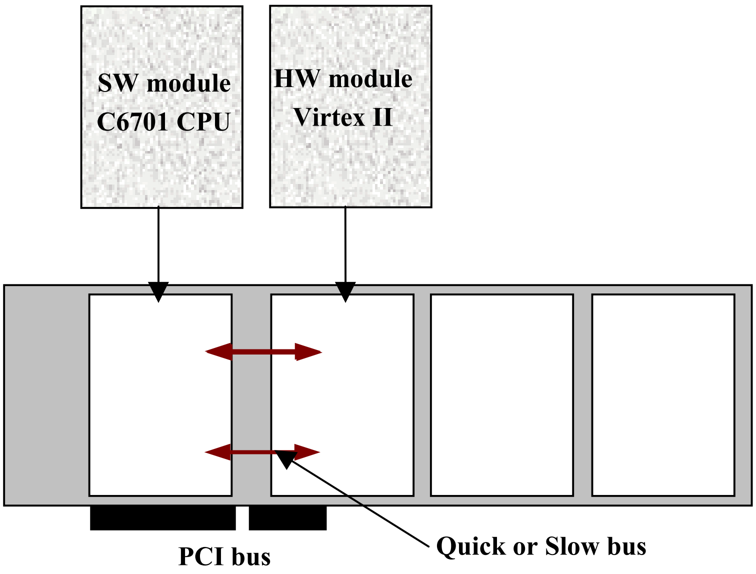


Figure 6.4: Sundance architecture description.

> High rate data-path dominated operations, *e.g.* transmit and receive filters, are best located in FPGAs, whereas more control oriented operations, *e.g.* signal detection, are best implemented in DSPs. Due to the possibility to resort to the C programming language for programming DSPs, *e.g.* using TIs Code Composer Studio [Ins]. DSPs are pre-destinate for algorithms under research. The possibility to partition the processing load in a flexible way onto DSPs and FPGAs is a key requirement.

In order to account for these constraints a modular based testbed structure is preferred. The testbed is located in host DSP, containing the modules for transmission, pre-processing step of HISD and reception. The block diagram of the transmitter and receiver is indicated in figure 6.5.

6.4 Finite word length analysis

While the HISD is developed using floating-point arithmetic, its implementation using Very-Large-Scale-Implementation (VLSI) requires fixed-point arithmetic for the sake of hardware cost and speed. The reduction of the bit width almost linearly reduces the design size, hardware complexity and power consumption. However, the stability of the algorithm and the performance may suffer from excessive finite word length effects, due to the overflow and quantization noise,

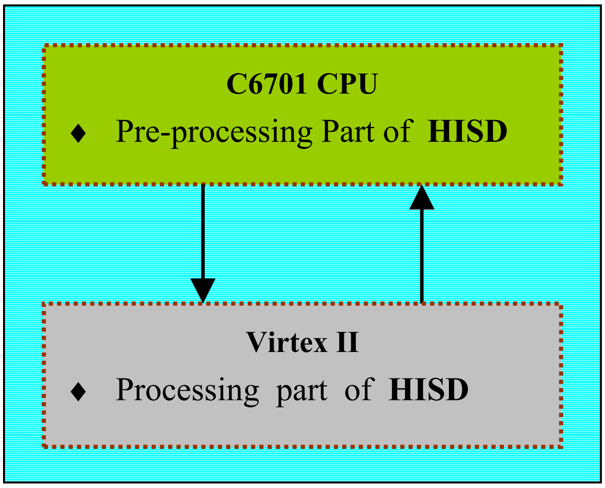


Figure 6.5: Block diagram of the HISD testbed.

unless all signals are scaled properly and sufficient word length is assigned. So it is important to find a reduced word-length with negligible performance degradation.

In this section, the finite word length effects is analyzed on the performance of the proposed VLSI architecture for the HISD algorithm. The possible trade off between hardware complexity and detecting performance is discussed. It should be noted that the quantization analysis is specific to the proposed VLSI architecture, and the quantization results are dependent on the system model stated earlier. Let $q(w, t)$ denote a quantization scheme in which totally w bits are used, of which t bits are used for the integer part of the value. With this quantization scheme, a value has t bits of dynamic range and $w-t$ bits of precision. The quantization schemes of various variables of the HISD have been analyzed in Matlab and systemC as following:

1. Write floating-point model of GISD in Matlab (*floating-point*).

2. Convert the Matlab model to C++ model (*floating-point*).

3. Analyze the dynamic range of different variables (*floating-point*).

4. Use systemC fixed-point data types (*fixed-point*).

5. Generate a dynamic link library to be used by Matlab (*fixed-point*).

6. Simulate the Fixed point model and look to BER performance (*fixed-point*).

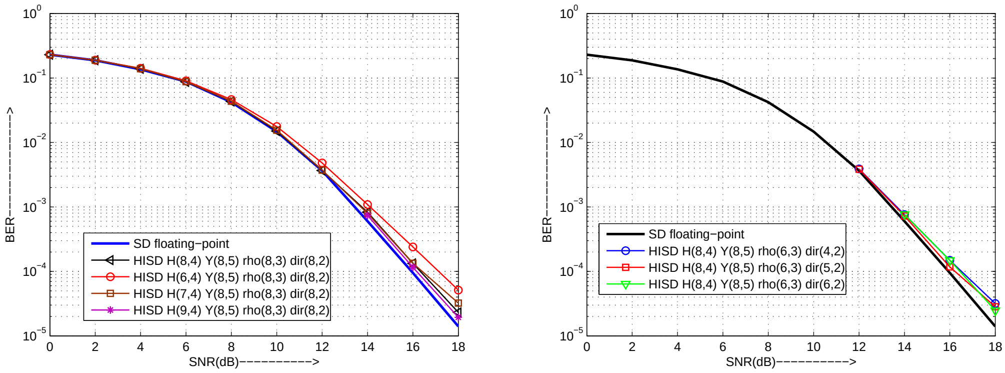
The least possible finite word length required for each variable is firstly determined in turn by assuming that other variables are in infinite precision, then the word length of each variable is further refined with all of variables considered in finite precision. The simulation is based on the case of uncoded multiple antenna system with $N = 5$ transmit antennas and $M = 5$ receive antennas to simplify the analysis, but it is straightforward to generalize the results to the case of any number of antennas.

- Quantization of the channel matrix $\mathbf{H}$: it is crucial to the behavior of the geometrical intersection and selection detector, as it determines the computational precision of the objective function values in different feasible points. A small word length may result in poor performance, though a large word length may cost more hardware. The maximal absolute value of $\mathbf{H}$ is observed to be smaller than 7. Thus, 4 bits of dynamic range are needed for the quantization of the channel matrix. Various fractional precisions for $\mathbf{H}$ from 2 bits to 5 bits have been examined in this work (figure 6.6(a)). It turns out to be no significant when the number of fractional bits is lager than 4. Thus the $q(8, 4)$ scheme is the optimal choice.

- Quantization of the k^{th} smallest eigenvector $\mathbf{v}_k$: the set of vectors $\{\mathbf{v}_i\}_{i=1}^{n}$ is an orthonormal basis of $\mathbb{R}^n$. The maximal absolute value of different coordinates of each vector is less or equal to one. Thus, 2 bits of dynamic range are needed for the quantization of the smallest eigenvector. Various fractional precisions for $\mathbf{v}_k$ from 2 bits to 4 bits have been examined in the figure 6.6(b). By our simulation, $q(6, 2)$ scheme is sufficient for the quantization of the vector $\mathbf{v}_k$.

- Quantization of the received vector $\mathbf{y}$: the maximal absolute value of different coordinates of the received vector is less or equal to 15. Thus, 5 bits of dynamic range are needed for the quantization. To determine the number of bits required to represent fractional precision, we fix t to 5 and compare the simulated detector performance using varying w bits until the performance loss becomes unacceptable. The results are plotted versus BER in Figure 6.7(a). To meet a requirement of a BER floor below 10^{-4}, we select 3 bits to use for fractional precision.

- Quantization of ρ_{zf}: Various quantization schemes for the unconstrained solution ρ_{zf}: $q(6, 2)$, $q(6, 3)$ and $q(6, 4)$, have been investigated for GISD

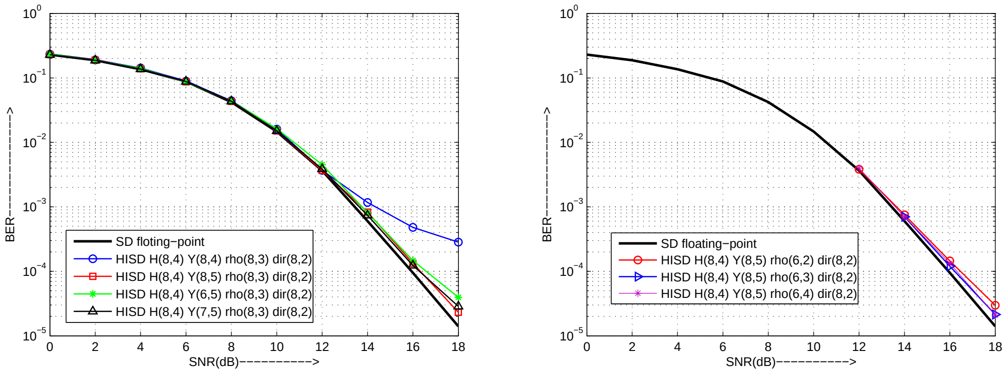


(a) Quantization of $\mathbf{H}$

(b) Quantization of $\mathbf{v}_k$

Figure 6.6: Performance comparison of various quantization schemes for $\mathbf{H}$ and $\mathbf{v}_k$: 5×5 MIMO system with uncoded 4-QAM modulation, $C = 4$ and $D = 2$.

detector where $D = 2$ and $C = 4$. Infinite precision and finite word length simulation results are shown in figure 6.7(b). It can be seen that the quantization scheme $q(6, 3)$ perform well compared to the sphere decoding infinite precision scheme.



(a) Quantization of $\mathbf{y}$

(b) Quantization of ρ_{zf}

Figure 6.7: Performance comparison of various quantization schemes for $\mathbf{y}$ and ρ_{zf}: 5×5 MIMO system with uncoded 4-QAM modulation, $C = 4$ and $D = 2$.

The quantization scheme for the geometrical intersection and selection detector is summarized in table 6.4.

Table 6.4: Quantization schemes summary.

Variable	Quantization Scheme
$\mathbf{H}$	$q(8,4)$
$\mathbf{y}$	$q(8,5)$
$\mathbf{v}_k$	$q(6,2)$
ρ_{zf}	$q(6,3)$

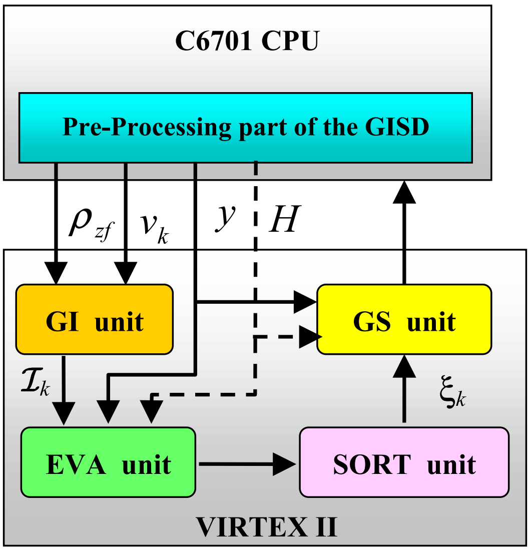


Figure 6.8: Block diagram of the GISD detector.

6.5 GISD Architecture

Among various of sub-optimal detectors, the GISD is attractive to hardware implementations due to lower complexity and numerical stability. In this section, we propose a VLSI architecture for the GISD detector and the correspond implementation results. The block diagram that illustrates the detector architecture is shown in Figure 6.8.

The geometrical intersection (GI) module calculate all intersections points and projected them on set $\{-1,1\}^n$ where $n = 2N$. We assumed that the vectors $\mathbf{v}_k$ and ρ_{zf} have been pre-calculated using a pre-processor, *e.g.*, DSP in our implementation. The evaluation (EVA) module calculate the value of the objective function $f() = \|\mathbf{y} - \mathbf{Hx}\|_p^2$ for all the feasible points generated by the GI module. Then, the SORT module sorts different values of $f()$ in ascending order and select the C best feasible points having the minimum values. The GS module performs a greedy search on C starting feasible points, and finally outputs the best detected symbols. To interconnect the different modules, we use the globally asynchronous locally synchronous technique described in next section.

6.6 Globally Asynchronous Locally Synchronous

The inter-module communication is based on Globally Asynchronous - Locally Synchronous (GALS) principe [Cha84, Bom04]. Usually, a GALS circuit is defined as a set of locally synchronous modules communicating with each other via asynchronous wrappers. A wrapper generates the clock signal for its module and realizes the communication between modules. The idea of the GALS approach is to combine the advantages of synchronous and asynchronous design methodologies while avoiding their disadvantages.

Globally-asynchronous locally-synchronous (GALS) operation employs a self-timed communication scheme on a coarse grained block level and combines the following features:

- All major modules are designed in accordance to proven synchronous clocking disciplines.
- Data exchange between any two modules strictly follows a full handshake protocol.
- Each module is allowed to run from its own local clock.
- Any asynchronous circuitry necessary for coordinating the clock-driven with the self-timed operation is combined to ”self-timed wrappers” arranged around each clock domain.

Figure 6.9 depicts a block level schematic of a GALS module with its self-timed wrapper surrounding the locally synchronous module. The wrapper contains an arbitrary number of GALS ports and a local clock generator.

6.7 Implementation Results

The proposed VLSI architecture of the HISD is modelled in very high speed integrated circuit hardware description language (VHDL) and synthesized in Xilinx

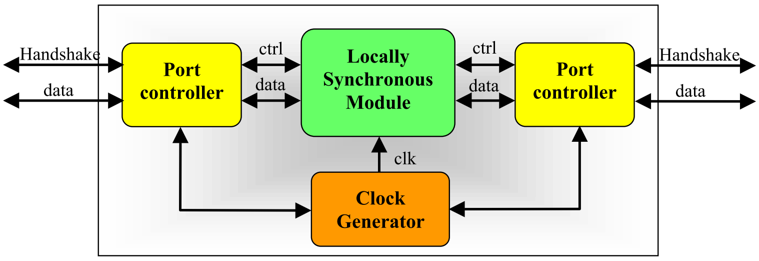


Figure 6.9: General GALS module.

Table 6.5: Virtex2 synthesis results.

	GI unit	EVA unit	SORT unit	GS unit
Maximum Frequency (Mhz)	119.182	54.262	142.798	51.395
Critical Path (ns)	8.391	18.429	7.003	19.457
Required cycles	31	260	44	163
Number of Slices/10752	321	473	282	156
Throughput (Mbps)	39.66	2.09	32.45	3.15

ISE Software. It has been implemented using a Xilinx Virtex2 XC2V2000-6 with package ff896. The sample data generated in MATLAB, *i.e.*, $\mathbf{H}$, $\mathbf{y}$, $\mathbf{v}_k$ and ρ_{zf} are inputed as test vectors into the Modelsim to verified the VHDL model of the HISD. It is observed that the output samples $\mathbf{x}_{hisd}$ of the VHDL model corresponds to that from MATLAB. For the simulated system of 5×5 antennas, the implementation results for FPGA of different HISD's unit are summarized in table 6.5.

6.8 Conclusion

The hardware implementation of the geometrical intersection and selection detector algorithm has been discussed. Firstly, a modified evaluation criterion is proposed based on the l_1-norm and/or l_∞-norm instead of the squared l_2-norm, which reduce complexity on the circuit level at only a small SNR penalty. Secondly, various quantization schemes of the GISD are investigated and the optimal choice considering the tradeoff between the hardware complexity and the performance is discussed. The numerical simulation results show that the quantization schemes that have been developed are effective in approximating the infinite precision schemes. Thirdly, the GISD hardware implementation over the PALMYRE system platform has been discussed. Finally, an actual VLSI implementation for a 5 × 5 uncoded MIMO system using 4-QAM modulation was presented. The GISD decoder, where $D = 1$ and $C = 4$, achieves throughput of 2Mbps at all SNR.

Chapter 7

Conclusions

It is well known that the optimal maximum likelihood detection problem is $\mathcal{NP}$-hard. Most of the research in detection techniques has focused on developing new or improved existing suboptimal detection schemes that are more feasible to implement. In this work, we have presented a new suboptimal detection method capable to give a good approximation of the optimal solution and reduce the huge complexity of the ML detection problem.

After outlining the characteristics and the baseband model of the linear wireless channel, a few motivating examples of systems which have previously been studied in the literature and which may be modeled as linear wireless channels are given. An overview of the most common optimal and suboptimal detection strategies have been presented in chapter 3. Typically, there is a trade-off between performances and computation complexity of the given detectors, *e.g*, The linear detectors which are also the fastest have in general worse performance than the semidefinite relaxation and interference cancellation detectors which are computationally more complex caused by their iterative structure. The next section summarizes the contributions of this thesis.

7.1 Contributions

The main results of the thesis are the following:

- The design of a new suboptimal method for the maximum likelihood detection problem. This method is based on two complementary "*real time*" operational research techniques called *intensification* and *diversification.*

- The intensification step is a simple local search schema, called *greedy search method.* Moreover, this step has a good convergence property (two or three iterations). For a given starting point the computational complexity of the intensification method is $\mathcal{O}(n^2)$.

- The diversification step is motivated by the search of all feasible points in the "vicinity" of the D lines originating from the uncontrained ML solution $\rho_{zf} = \mathbf{H}^{+}\mathbf{y}$ and along the D right smallest singular vectors of the matrix $\mathbf{H}$. Depending on the considered geometrical approach, this step contains of three distinct variants: Hypercub Intersection and Selection (HIS), Plane Intersection and Selection (PIS) and Canonical Basis Intersection and Selection (CBIS).

- The proposed method is able to achieve near-ML performance with constant polynomial-time, *i.e.* $\mathcal{O}(n^3)$, computational complexity (unlike the SD that has exponential-time complexity for low SNR).

- We proposed a new solft output detection method, called List-GISD to generate a soft input decision for forward error correcting codes.

- The most computation complexity of the GISD detector is concentrated in the pre-processing part. In fact, This step needs calculation of the unconstrained ML solution given by $\rho_{zf} = \mathbf{H}^{+}\mathbf{y}$ and extraction of the D smallest singular vectors of the channel matrix $\mathbf{H}$ using singular value decomposition. To reduce the complexity of the pre-processing step of the GISD, we propose an new method based on the basic work in [SG01].

- A basic VLSI implementation of the GISD detection technique is presented where $D = 1$ and $C = 4$. However, the acheived throughput is 2 Mbps. This poor hardware performance can be very easly increased if we explorate the parallel structure of proposed schemas.

7.2 Topics for Future Work

While some questions have found their answers in this work there are still many issues not resolved. A few thoughts about some of these are given below.

- To improve the receiver performance, the proposed GISD detector and Turbo decoding technique will be combined. The soft output of the Turbo decoder is fed back to improve the detection. This improvement then benefits the decoding in return.

- The GISD is suitable for pipeline VLSI implementation which allows fast data processing. Moreover, the parallel structure will be explored in a future work.

- Develop a systolic VLSI architecture to GISD pre-processing step. The main purpose of this work is to give a complet hardware implementation of the GISD detector.

Appendix A

Properties of the Real-Valued Model

In this appendix we collect some properties of the mappings

$$\tilde{\mathbf{H}} \mapsto \mathbf{H} = \left[\begin{array}{cc} \Re(\tilde{\mathbf{H}}) & -\Im\tilde{\mathbf{H}} \\ \Im(\tilde{\mathbf{H}}) & \Re(\tilde{\mathbf{H}}) \end{array} \right] \tag{A.1}$$

$$\mathbb{C}^{M\times N} \rightarrow \mathbb{R}^{2M\times 2N} = \mathbb{R}^{m\times n} \tag{A.2}$$

and

$$\tilde{\mathbf{x}} \mapsto \mathbf{x} = \left[\begin{array}{c} \Re(\tilde{\mathbf{x}}) \\ \Im(\tilde{\mathbf{x}}) \end{array} \right] \tag{A.3}$$

$$\mathbb{C}^{N} \rightarrow \mathbb{R}^{2N} = \mathbb{R}^{n} \tag{A.4}$$

from complex-valued matrices and vectors to real-valued matrices and vectors, also [Tel95]:

$$\tilde{\mathbf{H}}^{H} \Leftrightarrow \mathbf{H}^{T} \tag{A.5}$$

$$\|\tilde{\mathbf{x}}\|^2 = \sum_{k=1}^{N} |x[k]|^2 = \sum_{k=1}^{N} (\Re(x[k])^2 + \Im(x[k])^2) = \|\mathbf{x}\|^2 \tag{A.6}$$

From this follows

$$\begin{aligned} trace(\tilde{\mathbf{x}}\tilde{\mathbf{x}}^{H}) &= trace(\mathbf{x}\mathbf{x}^{T}) && \text{(A.7)} \\ \tilde{\mathbf{H}}\tilde{\mathbf{x}} = \tilde{\mathbf{y}} &= \mathbf{H}\mathbf{x} = \mathbf{y} && \text{(A.8)} \\ trace(\mathbf{H}) &= 2\Re(trace(\tilde{\mathbf{H}})) && \text{(A.9)} \end{aligned}$$

If $\Im(\tilde{\mathbf{x}})$ and $\Re(\tilde{\mathbf{x}})$ are independent vectors of independent random variables with variance $\sigma^2 = \frac{\tilde{\sigma}^2}{2}$ each,

$$\varepsilon\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^\dagger\} = \tilde{\sigma}^2\mathbf{I} \Leftrightarrow \varepsilon\{\mathbf{x}\mathbf{x}^T\} = \frac{\tilde{\sigma}^2}{2}\mathbf{I} = \frac{\sigma^2}{2}\mathbf{I} \tag{A.10}$$

Appendix B

Extraction of smallest singular vectors

In preprocessing step, we are given an $n \times n$ matrix $\mathbf{G} = \mathbf{H}^T\mathbf{H}$ and we want to compute the last D right singular vectors of channel matrix $\mathbf{H}$ which correspond to the last D eigenvectors of the matrix $\mathbf{G}$. Singular vectors are usually computed via the Singular Value Decomposition of $\mathbf{H}$. There are many algorithms that either exactly compute the SVD of a matrix in $\mathcal{O}(mn^2 + nm^2)$ time. The GISD detector needs just the D smallest eigenvectors of the matrix $\mathbf{G}$. thus, it is of interest to find an approximation $\mathbf{B}$ of a specified rank D to the matrix $\mathbf{G}$. In the following, we give a brief high-level presentation of different faster methods to extract the D smallest eigenvector of channel matrix.

B.1 The power method

Probably the oldest algorithm for approximating eigenvalues and corresponding eigenvectors of a matrix is the power method. This method is an important tool in its own right when conditions are appropriate. It is very simple and only requires matrix-vector products along with two vectors of storage.

At step five of the power method ($i = imax(\mathbf{w})$), i is the index of the element of $\mathbf{w}$ with largest absolute value. It is easily seen that the contents of $\mathbf{z}$ after

Algorithm 3: The power method

```
Data: $\mathbf{z}_0$ arbitrary, and the Gram matrix $\mathbf{G}$.
begin
    $\mathbf{z} = \mathbf{z}_0/\|\mathbf{z}_0\|_\infty$
    for $p = 1, 2, 3, ...$ do
        $\mathbf{w} = \mathbf{G}\mathbf{z}$
        $\alpha = \frac{\mathbf{w}^T\mathbf{z}}{\mathbf{z}^T\mathbf{z}}$
        $i = imax(\mathbf{w})$
        $\mathbf{z} = \mathbf{z}/(\mathbf{e}_i^T\mathbf{w})$
end
```

k-steps of this iteration will be the vector

$$\begin{aligned}\mathbf{z}_k &= (\frac{1}{\mathbf{e}_i^T\mathbf{G}^k\mathbf{z}_0})\mathbf{G}^k\mathbf{z}_0 \\ &= (\frac{\beta_k}{\mathbf{e}_i^T\mathbf{G}^k\mathbf{z}_0})(\frac{1}{\beta_k}\mathbf{G}^k\mathbf{z}_0)\end{aligned}$$

for any nonzero scalar β_k and where $\mathbf{e}_i$ is the i^{th} column of the $i \times i$ identity matrix. In particular, this iteration may be analyzed as if the vectors had been scaled by $\beta_k = \lambda_1^k$ at each step, with λ_1^k an eigenvalue of $\mathbf{G}$ with largest magnitude. If $\mathbf{G}$ is diagonalizable with eigenpairs $\{(\mathbf{v}_j, \lambda_j),\ 1 \leq j \leq n\}$ and $\mathbf{z}_0$ has the expansion $\mathbf{z}_0 = \sum_{j=1}^n \gamma_j \mathbf{v}_j$ in this basis then

$$(\frac{1}{\lambda_1^k})\mathbf{G}^k\mathbf{z}_0 = \sum_{j=1}^n \gamma_j \frac{\lambda_j^k}{\lambda_1^k}\mathbf{v}_j$$

If λ_1 is a largest and simple eigenvalue then

$$\frac{\lambda_j^k}{\lambda_1^k} \rightarrow 0, \quad 2 \leq j \leq n$$

It follows that $\mathbf{z}_k \rightarrow \mathbf{v}_1/(\mathbf{e}_i^T\mathbf{v}_1)$, where $i = imax(\mathbf{w})$, at a linear rate with a convergence factor of $|\frac{\lambda_2}{\lambda_1}|$.

Furthermore, the power method can be extended to search also for other eigenvalues; for example, the smallest one and the second largest one. First, if $\mathbf{G}$ is nonsingular, we can apply the power method to $\mathbf{G}^{-1}$ to find the smallest eigenvalue because $(1/\lambda_n)$ is the largest eigenvalue of $\mathbf{G}^{-1}$. Second, if we need

Table B.1: Krylov subspace methods

	$\mathbf{Av} = \lambda\mathbf{v}$	$\mathbf{Av} = \mathbf{b}$
$\mathbf{A} = \mathbf{A}^T$	Lanczos	CG
$\mathbf{A} \neq \mathbf{A}^T$	Arnoldi	GMRES

more eigenvalues and λ_1 is already known, we can use a reduction method to construct a matrix $\mathbf{B}$ that has the same eigenvalues and eigenvectors as $\mathbf{G}$ except for λ_1, which is replaced by zero eigenvalue. For the power method, the rate of convergence is dictated by the ratio of absolute values of the second largest and the largest eigenvalue of the matrix $\mathbf{G}$. Depending on the eigenvalue ratio, this method can converge very slowly.

B.2 Iterative methods and Krylov space

These methods are dominant in computing large matrices because direct methods are either impossible or too slow hence infeasible in practice. First, there is no direct method for eigenvalue problems when dimension of the matrix is greater than 4. Any eigenvalue solvers must be iterative. On the other hand, direct methods for solving linear systems like Gaussian elimination require $\mathcal{O}(n^3)$ operations, which is too time-consuming. Iterative methods are approximated methods, which only require $\mathcal{O}(n^2)$ operations. They can compute solutions much faster with errors which can be tolerant. In practice, this is often good enough.

Krylov methods are one important types of iterative methods. These methods often attempt to generate better approximations from Krylov subspace. Given a matrix $\mathbf{A}$ and a vector $\mathbf{b}$, the associated Krylov sequence is the set of vectors: $\mathbf{b}, \mathbf{Ab}, \mathbf{A}^2\mathbf{b}, \mathbf{A}^3\mathbf{b}, ...$ The corresponding Krylov subspaces are the spaces spanned by successively larger groups of these vectors in the Krylov sequence. Krylov methods can be summarized in table B.1:

Although these four methods work in different situations, they share some similar structures. They all tend to construct the orthogonal basis of Krylov subspace by multiplying matrix A with a vector. The all have one step which compute the multiplication of a matrix and a vector. See algorithm 4 and algorithm 5. The computation of $\mathbf{Av}$ is the most time-consuming part of each iteration. It costs $\mathcal{O}(n^2)$ time. Suppose the number of iteration is k, the running time for Krylov methods totally is $\mathcal{O}(kn^2)$.

B.2.1 Lanczos method

For a given real symmetric matrix $\mathbf{A}$ of size $n \times n$, the Lanczos method (algorithm 4) starts from a nonzero vector $\mathbf{b} \in \mathbb{R}^n$ and generates two sequences of numbers (α_k) and (β_k) as follows. Put $\beta_0 = 0$, $\mathbf{z}_0 = null$, $\mathbf{z}_1 = \mathbf{b}/\|\mathbf{b}\|_2$, and for $k = 1, 2, ...,$

$$\alpha_k =< \mathbf{z}_k, \mathbf{A}\mathbf{z}_k >, \quad \beta_k \mathbf{z}_{k+1} = \mathbf{A}\mathbf{z}_k - \alpha_k \mathbf{z}_k - \beta_{k-1}\mathbf{z}_{k-1}$$

where β_k is taken such that $\|\mathbf{z}_{k+1}\|_2 = 1$. The vectors $\{\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_\ell\}$ are an orthonormal basis of the ℓ^{th} Krylov subspace spanned by $\mathbf{b}, \mathbf{Ab}, ..., \mathbf{A}^{\ell-1}\mathbf{b}$. The coefficients α_k and β_k are collected in the tridiagonal matrices:

$$\mathbf{T}_\ell = \begin{pmatrix} \alpha_1 & \beta_1 & & & \\ \beta_1 & \alpha_2 & \beta_2 & & \\ & \beta_2 & \alpha_3 & \ddots & \\ & & \ddots & \ddots & \beta_{\ell-1} \\ & & & \beta_{\ell-1} & \alpha_\ell \end{pmatrix}$$

For $\ell < n$ The eigenvalues of $\mathbf{T}_\ell$ are called *Ritz* values, and they are easier to compute because of the tridiagonal nature of $\mathbf{T}_\ell$ and because ℓ is smaller than n. Some of the Ritz values turn out to be accurate approximations of some of the eigenvalues of $\mathbf{A}$, also when ℓ is much smaller than n. The Lanczos method is discussed in [Lan50, ABFH00].

Algorithm 4: Lanczos Algorithm

```
Data: $\mathbf{b}$ arbitrary, $\mathbf{z}_1 = \mathbf{b}/\|\mathbf{b}\|_2$, $\beta_0 =$ and $\mathbf{z}_0 = null$.
begin
    for $p = 1, 2, 3....$ do
        $\mathbf{q} = \mathbf{A}\mathbf{z}_p$
        $\alpha_p = \mathbf{z}_p^T\mathbf{q}$
        $\mathbf{q} = \mathbf{q} - \beta_{p-1}\mathbf{z}_{p-1} - \alpha_p\mathbf{z}_p$
        $\beta_p = \|\mathbf{q}\|_2$
        $\mathbf{z}_{p+1} = \mathbf{q}/\beta_p$
        compute eigenvalues and eigenvectors of $\mathbf{T}_p$
end
```

B.2.2 Arnoldi method

Suppose $\mathbf{Z}$ is an $n \times n$ unitary matrix that reduces $\mathbf{A}$ to upper Hessenberg form; *i.e.*, $\mathbf{Z}^T\mathbf{G}\mathbf{Z} = \mathbf{T}$ for some upper Hessenberg matrix, $\mathbf{T}$. For any index $1 \leq \ell \leq n$, let $\mathbf{T}_\ell$ denote the ℓ^{th} principal submatrix of $\mathbf{T}$:

$$\mathbf{T}_\ell = \begin{pmatrix} t_{11} & t_{12} & \cdots & t_{1\ell} \\ \beta_2 & t_{22} & \cdots & t_{2\ell} \\ & \ddots & \ddots & \vdots \\ & & \beta_\ell & t_{\ell\ell} \end{pmatrix}$$

The Arnoldi method [Arn51, Saa80] builds up the matrices $\mathbf{T}$ and $\mathbf{Z}$ one column at a time starting with the unit vector $\mathbf{z}_1 \in \mathbb{R}^n$, although the process is typically stopped well before completion, with $\ell \ll n$. The algorithm only accesses $\mathbf{G}$ through matrix-vector products, making this approach attractive when $\mathbf{G}$ is large and sparse.

Different choices for $\mathbf{z}_1$ produce distinct outcomes for $\mathbf{T}_\ell$. The defining recurrence may be derived from the fundamental relation

$$\mathbf{A}\mathbf{Z}_\ell = \mathbf{Z}_\ell\mathbf{T}_\ell + \beta_{\ell+1}\mathbf{z}_{\ell+1}\mathbf{e}_\ell^T$$

where $\mathbf{e}_\ell$ is the ℓ^{th} column of the $\ell \times \ell$ identity matrix. the ℓ^{th} column of $\mathbf{T}_\ell$ is determined so as to force $\mathbf{z}_{\ell+1}$ to be orthogonal to the columns of $\mathbf{Z}_\ell$, and $\beta_{\ell+1}$ then is determined so that $\|\mathbf{z}_{\ell+1}\| = 1$. Provided $\mathbf{T}_\ell$ is unreduced, the columns of $\mathbf{Z}_\ell$ constitute an orthonormal basis for the order-ℓ Krylov subspace $\kappa_\ell(\mathbf{A}, \mathbf{z}_1) = span\{\mathbf{z}_1, \mathbf{A}\mathbf{z}_1, \mathbf{A}^2\mathbf{z}_1, ..., \mathbf{A}^{\ell-1}\mathbf{z}_1\}$. Since $\mathbf{Z}_\ell^T\mathbf{Z}\mathbf{Z}_\ell = \mathbf{T}_\ell$, the matrix $\mathbf{T}_\ell$ is a RitzGalerkin approximation of $\mathbf{A}$ on this subspace, as described by Saad [Saa80]. The eigenvalues of $\mathbf{T}_\ell$ are called Ritz values and will, in many circumstances, be reasonable approximations to some of the eigenvalues of $\mathbf{A}$. An eigenvector of $\mathbf{T}_\ell$ associated with a given Ritz value θ can be used to construct an eigenvector approximation for $\mathbf{A}$. Indeed, if $\mathbf{T}_\ell\mathbf{y} = \theta\mathbf{y}$, then the Ritz vector $\mathbf{x} = \mathbf{Z}_\ell\mathbf{y}$ yields the residual

$$\begin{aligned} \|\mathbf{A}\mathbf{x} - \theta\mathbf{u}\|_2 &= \|\mathbf{A}\mathbf{Z}_\ell\mathbf{y} - \theta\mathbf{Z}_\ell\mathbf{y}\|_2 \\ &= \|(\mathbf{A}\mathbf{Z}_\ell - \mathbf{Z}_\ell\mathbf{T}_\ell)\mathbf{y}\|_2 \\ &= |\beta_{\ell+1}||\mathbf{e}_\ell^T\mathbf{y}| \end{aligned}$$

where $|\beta_{\ell+1}| \ll 1$, the columns of $\mathbf{Z}_\ell$ nearly span an invariant subspace of $\mathbf{A}$. It easily follows that θ is a Ritz value and $\mathbf{x}$ a correspond Ritz vector. The central idea behind the Arnoldi factorization is to construct eigenpairs of the large matrix $\mathbf{A}$ from the eigenpairs of the small matrix $\mathbf{T}$. The explicit steps needed to form a p-Step Arnoldi factorization are shown in algorithm 5.

Algorithm 5: Arnoldi Algorithm

```
Data: $\mathbf{b}$ arbitrary, $\mathbf{z}_1 = \mathbf{b}/\|\mathbf{b}\|_2$.
begin
    for $j = 1, .., \ell - 1$ do
        $\mathbf{q} = \mathbf{A}\mathbf{z}_j$
        for $i = 1..j$ do
            $t_{ij} = \mathbf{z}_i^T \mathbf{q}$
            $\mathbf{q} = \mathbf{q} - t_{ij}\mathbf{z}_i$
        $t_{j+1j} = \|\mathbf{q}\|_2$
        $\mathbf{z}_{j+1} = \mathbf{q}/t_{j+1j}$
end
```

Appendix C

Algorithms

C.1 Intensification step

In this section, we give the MATLAB source code for the 1^{st}-order greedy detector for different norms

C.1.1 Greedy Search function using l_2-norm

```
function [final, dist]=first_order_greedy_l2(H,y,start,dist);
% Input arguments:
% 1. H : rayleigh channel matrix
% 2. y : received vector
% 3. start : starting vector belongs to {-1,+1}^n
% 4. dist  : ||A*start-y||^2
% output arguments:
% 1. final : best neighbor
% 2. dist  : ||A*final-y||^2
[m,n] = size(H); flag = 1;
while (flag == 1)
    flag = 0;
    for k = 1: n
        x = start; x(k) = -1 * x(k); Dk = sum((H * x - y).^2);
        if Dk < dist
            dist = Dk; index = k; flag = 1;
        end;
    end;
```

```
      if (flag == 1)
          start(index) = -1 * start(index);
    end;
end;
final = start;
%% end of file
```

The previous algorithm can be simplified as following:

```
function [final,dist]=Enh_first_order_greedy_l2(H,G,y,start,dist);
% Input arguments:
% 1. H : rayleigh channel matrix
% 2. G : Gram matrix G= transpose(H)*H
% 2. y : received vector
% 3. start : starting vector belongs to {-1,+1}^n
% 4. dist : ||A*start -y||^2
% output
% 1. final : best neighbor
% 2. dist : ||A*final -y||^2
[m,n] = size(H); z = H*start; flag = 1;
while (flag == 1)
      flag = 0;
      for k = 1: n
          x = start; x(k) = -1 * x(k); nu=sign(start(k));
          delta = nu*(y-z)'*H(:,k)+G(k,k);
          if delta > 0
              dist = dist-4*delta; index = k; flag = 1;
          end;
      end;
      if (flag == 1)
          start(index) = -1 * start(index); z= H*start;
    end;
end;
final = start;
%% end of file
```

C.1.2 Greedy Search function l_1-norm

```
function [final,dist]=first_order_greedy_l1(H,y,start);
% Input arguments:
```

```
% 1. H : rayleigh channel matrix
% 2. y : received vector
% 3. start : starting vector belongs to {-1,+1}^n
% output arguments:
% 1. final : best neighbor
% 2. dist  : sum(abs(A*final-y))
[m,n] = size(H); flag = 1; dist = sum(abs(H*start-y));
while (flag == 1)
    flag = 0;
    for k = 1: n
        x = start; x(k) = -1 * x(k); Dk = sum(abs(H*x-y));
        if Dk < dist
            dist = Dk; index = k; flag = 1;
        end;
    end;
    if (flag == 1)
       start(index) = -1 * start(index);
   end;
end;
final = start;
%% end of file
```

C.1.3 Greedy Search function using l_∞-norm

```
function [final,dist]=first_order_greedy_linfinity(H,y,start);
% Input arguments:
% 1. H : rayleigh channel matrix
% 2. y : received vector
% 3. start : starting vector belongs to {-1,+1}^n
% output arguments:
% 1. final : best neighbor
% 2. dist  : max(abs(A*final-y))
[m,n] = size(H); flag = 1; dist = max(abs(H*start-y));
while (flag == 1)
    flag = 0;
    for k = 1: n
        x = start; x(k) = -1 * x(k); Dk = max(abs(H*x-y));
        if Dk < dist
```

```
            dist = Dk; index = k; flag = 1;
        end;
    end;
    if (flag == 1)
       start(index) = -1 * start(index);
   end;
end;
final = start;
%% end of file
```

C.2 Diversification step

In this section, we give the MATLAB source code for the hypercube intesection and selection (HIS) function for different norms

C.2.1 HIS function using l_2-norm

```
function [list_can]=his_l2(H,y,rho,dir,nbr_can);
% Input arguments:
% 1. H : rayleigh channel matrix
% 2. y : received vector
% 3. rho : unconstrained solution
% 4. dir : studied direction
% 5. nbr_can : number of candidates
% output arguments:
% 1. list_can : C best candidates point and their associated l2 norm
[m,n] = size(H);
% listed and project (on {-1,+1}^n) all intersection points
% between the line {z in R^n / z = rho + aplha* dir ,alpha in R}
% and all unit cube faces.
list = hypercube_intersection(rho,dir);
% Evaluation
for k = 1: 2*n
    list(n+1,k) = sum((H*list(1:n,k)-y).^2);
end;
% Selection of the best nbr_can candidates
 list      = (sortrows(list',dim_in+1))';
 list      = supp_red(list);
 list_can = sel_cand(list,nbr_can);
%% end of file
```

C.2.2 HIS function using l_1-norm

```
function [list_can]=his_l1(H,y,rho,dir,nbr_can);
% Input arguments:
% 1. H : rayleigh channel matrix
% 2. y : received vector
% 3. rho : unconstrained solution
% 4. dir : studied direction
% 5. nbr_can : number of candidates
% output arguments:
% 1. list_can : C best candidates point and their associated l1 norm
[m,n] = size(H);
% listed and project (on {-1,+1}^n) all intersection points
% between the line {z in R^n / z = rho + aplha* dir ,alpha in R}
% and all unit cube faces.
list = hypercube_intersection(rho,dir);
% Evaluation
for k = 1: 2*n
    list(n+1,k) = sum(abs(H*list(1:n,k)-y));
end;
% Selection of the best nbr_can candidates
 list     = (sortrows(list',dim_in+1))';
 list     = supp_red(list);
 list_can = sel_cand(list,nbr_can);
%% end of file
```

C.2.3 HIS function using l_∞-norm

```
function [list_can]=his_linfinity(H,y,rho,dir,nbr_can);
% Input arguments:
% 1. H : rayleigh channel matrix
% 2. y : received vector
% 3. rho : unconstrained solution
% 4. dir : studied direction
% 5. nbr_can : number of candidates
% output arguments:
```

```
% 1. list_can : C best candidates point and their associated l infinity no
[m,n] = size(H);
% listed and project (on {-1,+1}^n) all intersection points
% between the line {z in R^n / z = rho + aplha* dir ,alpha in R}
% and all unit cube faces.
list = hypercube_intersection(rho,dir);
% Evaluation
for k = 1: 2*n
    list(n+1,k) = max(abs(H*list(1:n,k)-y));
end;
% Selection of the best nbr_can candidates
 list      = (sortrows(list',dim_in+1))';
 list      = supp_red(list);
 list_can = sel_cand(list,nbr_can);
%% end of file
```

C.3 fixed step gradient descent algorithm

```
function [rho,dir1] = fixed_step_gradient_descent(H,y,start,tol)
% Input arguments:
% 1. H : rayleigh channel matrix
% 2. y : received vector
% 3. start : random start vector
% 4. tol : represents a given termination condition
% output arguments:
% 1. rho    : unconstrained solution
% 2. dir1   : first smallest singularvector of H
G=H'*H;
mu= 2/trace(G);
flag =1;
while(flag==1)
    xi = start - mu * (start'*G-2*Y'*H)';
    diff = xi-start;
    if (norm(diff)< toler)
        flag =0;
    end;
    start = xi;
end;
```

```
rho = start;
dir1 = diff/norm(diff);
```

Appendix D

Branch and Bound detector

The BBD algorithm maintains a node stack called $OPEN$, and a scalar called $UPPER$, which is equal to the minimum feasible cost found so far, *i.e.*, the "current-best" solution. Define k to be the level of a node (virtual root node has level 0). Label the branch with $\mathbf{d}_k(\mathbf{x}_1, \mathbf{x}_2, .., \mathbf{x}_k)$, which connects the two nodes $(\mathbf{x}_1, ..., \mathbf{x}_k)$ and $(\mathbf{x}_1, ..., \mathbf{x}_{k+1})$. The node $(\mathbf{x}_1, .., \mathbf{x}_k)$ is labeled with the lower bound ξ_k. Also, define $\mathbf{z}_k = \sum_{i=1}^{k} \mathbf{x}_i \mathbf{L}_i - \bar{\mathbf{r}}$, where $\mathbf{L}_i$ denotes the i^{th} column of $\mathbf{L}$. Denote $[\mathbf{z}_k]_j$ as the j^{th} component of vector $\mathbf{z}_k$ and $\mathbf{l}_{ij}$ as the $(i,j)^{th}$ element of $\mathbf{L}$. The branch and bound algorithm proceeds as follows [Ber98]:

1. Pre-compute $\bar{\mathbf{r}} = (\mathbf{L}^{-1})^T \mathbf{r}$;

2. Initialize $k = 0$, $\mathbf{z}_k = \bar{\mathbf{r}}$, $\xi_k = 0$, $UPPER = +\infty$ and $OPEN = NULL$;

3. Set $k = k + 1$. Choose the node in level k such that $\mathbf{x}_k = -sign([\mathbf{z}_{k-1}]_k)$. if $k < n$, append the node with $\mathbf{x}_k = sign([\mathbf{z}_{k-1}]_k)$ to the end of the $OPEN$ list.

4. Compute $\mathbf{z}_k = \mathbf{z}_{k-1} + \mathbf{x}_k \mathbf{L}_i$.

5. Compute $\xi_k = \xi_{k-1} + [\mathbf{z}_k]_k^2$.

6. If $\xi_k \geq UPPER$ and the $OPEN$ list is not empty, drop this node. Pick the node from the end of the $OPEN$ list, set k equal to the level of this node and go to step 4.

7. If $\xi_k < UPPER$, $k = n$ and the $OPEN$ list is not empty, update the "Current-Best" solution and $UPPER = \xi_k$. Pick the node from the end of the $OPEN$ list, set k equal to the level of this node and go to step 4.

8. If $\xi_k < UPPER$ and $k \neq n$, go to step 3.
9. If $\xi_k < UPPER$, $k = n$ and the $OPEN$ list is empty update the "Current-Best" solution and $UPPER = \xi_k$.
10. For all other cases, stop and report the "current-best" solution.

Bibliography

[ABFH00] J. I. Aliaga, D. L. Boley, R. W. Freund, and V. Hernández. A lanczos-type method for multiple starting vectors. *Math. Computation*, 69:1577–1601, May 2000.

[Ala98] S. M. Alamouti. A simple transmit diversity technique for wireless communications. *IEEE Journal on Select Areas in Communications*, 6:1451–1458, October 1998.

[AMM$^+$05] B. Andreas, B. Moritz, W. Markus, Z. Martin, F. Wolfgang, and B. Helmut. Vlsi implementation of mimo detection using the sphere decoding algorithm. *IEEE JOURNAL OF SOLID-STATE CIRCUITS*, 40:1566–1577, july 2005.

[Arn51] W. E. Arnoldi. The principle of minimized iterations in the solution of the matrix eigenvalue problem. *Quart. Appl. Math*, 9:17–29, 1951.

[ASH03] H. Artes, D. Seethaler, and F. Hlawatsch. Efficient detection algorithms for mimo channels: A geometrical approach to approximate ml detection. *IEEE Trans. Signal Processing, Special Issue on MIMO Wireless Communications*, 51(11):2808–2820, Nov 2003.

[Aus67] M. Austin. Decision feedback equalization for digital communication over dispersive channels. Technical report, MIT Research Laboratory of Electronics, August 1967.

[Bau01] J. Y. Baudais. *Etude des modulations a porteuses multiples et a spectre etale : analyse et optimisation.* PhD thesis, INSA Rennes, Mai 2001.

[BB92] K. Boulle and J. C. Belfiore. Modulation scheme designed for the rayleigh fading channel. *CISS*, March 1992.

[Ber98] D. Bertsekas. *Network Optimization, continuous and discrete models*, chapter 10, pages 483–492. Athena Scientific, Belmont, Massachusetts, 1998.

[BGBF03] J. Boutros, N. Gresset, L. Brunel, and M. Fossorier. Soft-input soft-output lattice sphere decoder for linear channels. San Francisco, Dec 2003.

[BGT93] C. Berrou, A. Glavieux, and P. Thitimajshima. Near shannon limit error-correcting coding and decoding: turbo-codes. pages 1064–1070, May 1993.

[Bom04] P. Bomel. *Plate-forme de prototypage rapide fondée sur la synthèse de haut niveau pour applications de radiocommunications.* PhD thesis, UBS, Lorient-France, December 2004.

[BRC60] Bose, Ray, and Chaudhuri. On a class of error-correcting binary group codes. *Information and Control*, 3:68–79, 1960.

[Bru02] L. Brunel. Optimum and sub-optimum multiuser detection based on sphere decoding for multi-carrier code division multiple access systems. *ICC*, 3:1526–1530, 2002.

[CBJ93] A. Chouly, A. Brajal, and S. Jourdan. Orthogonal multicarrier techniques applied to direct sequence spread spectrum cdma systems. *IEEE Globecom*, pages 1723–1728, November 1993.

[Cha84] D. M. Chapiro. *Globally-Asynchronous Locally-Synchronous Systems.* PhD thesis, Stanford University, October 1984.

[Cha04] S. Chabbouh. Algorithmes de détection et problèmes d'optimisation, juin 2004.

[Dam98] M. O. Damen. *Joint coding/decoding in a multiple access system, Application to mobile communications.* PhD thesis, ENST, Paris, 1998.

[DAMB00] M. O. Damen, K. Abed-Meraim, and J. C. Belfiore. Sphere decoding of space-time codes. *ISIT*, June 2000.

[DAMB02] M. O. Damen, K. Abed-Meraim, and J. C. Belfiore. Diagonal algebraic space-time block codes. *IEEE Trans. on Information Theory*, 48(3):628–636, March 2002.

[DBAM00] M. O. Damen, J. C. Belfiore, and K. Abed-Meraim. Generalized sphere decoder for asymmetrical space-time communication architecture. *Electronics Letters*, 36:166–167, 2000.

[DCB00] M. O. Damen, A. Chkeif, and J-C. Belfiore. Lattice codes decoder for space-time codes. *IEEE Communications Letters*, 4:161–163, 2000.

[DJ98] E. Dinan and B. Jabbari. Spreading codes for direct sequence cdma and wideband cdma cellular network. *IEEE Communications Magazine*, 36(9):48–54, 1998.

[DSAM03] M. O. Damen, A. Safavi, and K. Abed-Meraim. On cdma with space-time codes over multipath fading channels. *IEEE Trans. on Wireless Communications,*, 2:11–19, january 2003.

[FG98] G. Foschini and M. Gans. On the limits of wireless communications in a fading environment when using multiple antennas. *Wireless Personal Communications*, 6(3):311–335, Mars 1998.

[Fos96] G. J. Foschini. Layered space-time architecture for wireless communication in a fading environment when using multi-element antenna. *Bell labs Tech. Journal*, 1(2):41–59, 1996.

[FP85] U. Fincke and M. Pohst. Improved methods for calculating vectors of short length in a lattice, including a complexity analysis. *Mathematics of Computation*, 44:463–471, April 1985.

[Hal93] A. D. Hallen. Decorrelating decision-feedback multiuser detector for synchronous cdma channel. *IEEE Trans. Communications*, 41:285–290, Feb 1993.

[Has00] B. Hassibi. A fast square-root implementation for blast. *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, 2:1255–1259, 2000.

[Hay02] S. Haykin. *Adaptative Filter Theory*, chapter Method of steepest descent, pages 208–211. Tom Robbins, fourth edition edition, 2002.

[HB03] B. M. Hochwald and S. T. Brink. Achieving near-capacity on a multiple-antenna channel. *IEEE Transactions on Communications*, 51(3):389–399, March 2003.

[HH02] B. Hassibi and B. M. Hochwald. High-rate codes that are linear in space and time. *IEEE Trans. Inform. Theory*, 48(7):1804–1824, july 2002.

[Hoc59] A. Hocquenghem. Codes corecteurs d'erreurs. *Chiffres*, 2:147–156, 1959.

[HOP96] J. Hagenauer, E. Offer, and L. Papke. Iterative decoding of binary and block convolutional codes. *IEEE Trans. Info. Theory*, 42:429–445, March 1996.

[HR98] C. Helmberg and F. Rendl. Solving quadratic (0,l)-problems by semidefinite programs and cutting planes. *Mathematical Programming*, (82):219–315, 1998.

[HRVW96] C. Helmberg, F. Rendl, R. J. Vanderbei, and H. Wolkowicz. An interior point method for semidefinite programming. *SIAM Journal Optim*, (6):342–361, 1996.

[HV02] B. Hassibi and H. Vikalo. On the expected complexity of integer least-squares problems. *IEEE ICASSP*, page Orlando, May 2002.

[Ins] T. Instruments. http://www.ti.com.

[JO04] J. Jalden and B. Ottersten. An exponential lower bound on the expected complexity of sphere decoding. *IEEE ICASSP*, 4:393–396, May 2004.

[JO05] J. Jalden and B. Ottersten. On the complexity of sphere decoding in digital communications. *IEEE Transactions on Signal Processing*, 53(4):1474–1484, Apr 2005.

[Lan50] C. Lanczos. An iteration method for the solution of the eigenvalue problem of linear differential and integral operators. *Research Nat. Bur. Standards*, 45:255–282, 1950.

[LD60] A. H. Land and A. G. Doig. An automatic method for solving discrete programming problems. *Econometrica*, 28:497–520, 1960.

[LPPB03] J. Luo, K. Pattipati, P.Willett, and L. Brunel. Branch-and-bound-based fast optimal algorithm for multiuser detection in synchronous cdma. *in Proc. IEEE Int. Conf. Communications*, vol. 5:3336–3340, May 2003.

[MCH01] D. Mottier, D. Castelain, and J. F. Helard. Optimum and sub-optimum linear mmse multi-user detection formulti-carrier cdma transmission systems. *Vehicular Technology Conference*, 2:868–872, October 2001.

[MDW$^+$02] W. K. Ma, T. N. Davidson, K. M. Wong, Z. Q. Lou, and P. C. Ching. Quasi-maximum-likelihood multiuser detection using semi-definite relaxation with application to synchronous cdma. *IEEE Transactions on Signal Processing*, 50:912–922, 2002.

[MDWC04] W. K. Ma, T. N. Davidson, K. M. Wong, and P. C. Ching. A block alternating likelihood maximization approach to multiuser detection. *IEEE Transactions on Signal Processing*, 52:2600–2611, 2004.

[MH94] U. Madhow and M. L. Honig. Mmse interference suppression for direct-sequence spread spectrum cdma. *IEEE Transactions on Communications*, 42:3178–3188, Dec 1994.

[Mos96] S. Moshavi. Multi-user detection for ds-cdma communications. *IEEE Communications Magazine*, pages 124–136, Oct 1996.

[MVU01] S. Marinkovic, B. Vucetic, and A. Ushirokawa. Space-time iterative and multistage receiver structures for cdma mobile communications systems. *IEEE Journal. Sel. Areas Commun*, 19(8):1594–1604, August 2001.

[NB05] A. Nafkha and E. Boutillon. *Procédé de décodage de codes matriciels, module de décodage associé, et applications mettant en uvre un tel procédé.* Brevet français n 05 02664, March 2005.

[Net03] N. Networks. R1-031085: Complexity comparison of ofdm hs-dsch receivers and advanced receivers for hsdpa and associated text proposal. *3GPP TSG-RAN-1 , Seoul, Korea.*, Oct 2003.

[NW88] G.L. Nemhauser and L.A. Wolsey. *Integer and Combinatorial Optimization.* John Wiley, New York, 1988.

[PH93] P. Patel and J. Holtzman. Analysis of a ds-cdma successive interference cancellation scheme using correlations. *IEEE Int. Conference on Communications*, 1:76–80, 1993.

[PH94] P. Patel and J. M. Holtzman. Performance comparison of a ds/cdma system using successive interference cancellation (ic) scheme and a parallel ic scheme under fading. *ICC*, pages 510–514, May 1994.

[Pro00] Proakis. *Digital Communications*, chapter Characterization of communication signals and systems, pages 148–221. McGraw-Hill, New york, fourth edition, 200.

[PSM82] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein. Theory of spread-spectrum communications–a tutorial. *IEEE Transactions on Communications*, 30:855–884, May 1982.

[Rap96] T. S. Rappaport. *Wireless Communications - Principles and Practice*, chapter Mobile Radio Propagation, pages 177–255. Prentice Hall Communications Engineering and Emerging Technologies Series. Prentice Hall PTR, New Jersy,, December 1996.

[RVH95] P. Robertson, E. Villebrun, and P. Hoeher. A comparison of optimal and suboptimal map decoding algorithms operating in the log domain. pages 1009–1013, june 1995.

[Saa80] Y. Saad. Variations on arnoldis method for computing eigenelements of large unsymmetric matrices. *Linear Algebra Application*, 34:269–295, 1980.

[Saa92] Y. Saad. Numerical methods for large eigenvalue problems. *Algorithms and Architectures for Advanced Scientific Computing*, Halsted Press, 1992.

[SG01] P. Spasojevic and C. N. Georghiades. The slowest descent method and its application to sequence estimation. *IEEE transactions on communications*, 49(9):1592–1604, Sep 2001.

[Sha48] C. E. Shannon. A mathematical theory of communication. *Tech. Journal*, 27:379–423, 1948.

[SLW03] B. Steingrimsson, Z. Q. Lou, and K. W. Wong. Soft quasi-maximum-likelihood detection for multiple-antenna wireless channels. *IEEE Transactions on Signal Processing*, 51, Nov 2003.

[Tel95] E. Telatar. Capacity of multi-antenna gaussian channels. *European Transactions on Telecommunications*, 10:585–595, October 1995.

[TJC99a] V. Tarokh, H. Jafarkhani, and A. R. Calderbank. Space-time block codes from orthogonal designs. *IEEE Trans. Inf. Theory*, 45:1456–1467, July 1999.

[TJC99b] V. Tarokh, H. Jafarkhani, and A. R. Calderbank. Space-time block coding for wireless communications: performance results. *Selected Areas in Communications*, 17(03):451–460, March 1999.

[TR01] P. H. Tan and L. K. Rasmussen. The application of semidefinite programming for detection in cdma. *IEEE Journal on Select Areas in Communications*, 19:1442–1449, 2001.

[TSC98] V. Tarokh, N. Seshadri, and A. R. Calderbank. Space-time codes for high data rates wireless communications: Performance criterion and code construction. *IEEE Trans. Inf. Theory*, 44(3):744–765, Mars 1998.

[VB93] E. Viterbo and E. Biglieri. A universal lattice code decoder. *GRETSI*, Sept 1993.

[VB99] E. Viterbo and J. Boutros. A universal lattice code decoder for fading channel. *IEEE Trans. Information Theory*, 45:1639–1642, July 1999.

[Ver89] S. Verdu. Computational complexity of optimal multiuser detection. *Algorithmica*, 4:303–312, 1989.

[Ver98] S. Verdú. *Multiuser Detection.* Cambridge University Press, 1998.

[VH02] H. Vikalo and B. Hassibi. Modified fincke-pohst algorithm for low complexity iterative decoding over multiple antenna channels. *IEEE International Symposium on Information Theory*, page 390, 2002.

[WFGV98] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela. Vblast: an architecture for realizing very high data rates over the richscattering wireless channel. *ISSSE98*, pages 295–300, 1998.

[Win87] J. H. Winters. On the capacity of radio communication systems with diversity in a rayleigh fading environment. *IEEE journal on selected areas in communications*, 5(5):871–878, june 1987.

[WLA03] X. M. Wang, W. S. Lu, and A. Antoniou. A near-optimal multiuser detector for ds-cdma systems using semidefinite programming relaxation. *IEEE Transactions on Signal Processing*, 51:2446–2450, 2003.

[YLF93] N. Yee, J.P. Linnartz, and G. Fettweis. Multi-carrier cdma in indoor wireless radio networks. pages 109–113, Yokohama, september 1993. IEEE International Symposium on Personal, Indoor and Mobile Radio Communications,.

[YYU99] A. Yener, R. D. Yates, and S. Ulukus. A nonlinear programming approach to cdma multiuser detection. *Asilomar Conference on Signals, Systems and Computers*, Oct 1999.

[ZO03] X. Zhang and B. Ottersten. Performance analysis of v-blast structure with channel estimation errors. *IEEE Workshop on Signal Processing Advances in Wireless Communications*, june 2003.