



# **Complexity and Performance Evaluation of Detection Schemes for Spatial Multiplexing MIMO Systems**

**Auda M. Elshokry**

A Thesis Submitted to Faculty of Engineering, Islamic University Gaza in partial  
fulfillment of the requirements for the degree of

Master of Science

in

Electrical Engineering

Dr. Ammar Abu Hudrouss, Chair

January 2010

Gaza, Palestine

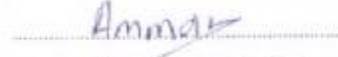


## نتيجة الحكم على أطروحة ماجستير

بناءً على موافقة عمادة الدراسات العليا بالجامعة الإسلامية بغزة على تشكيل لجنة الحكم على أطروحة الباحث/ عودة محمد عودة الشكري لنيل درجة الماجستير في كلية الهندسة قسم الهندسة الكهربائية/أنظمة التحكم وموضوعها:

### Complexity and Performance Evaluation of Detection Schemes for Spatial Multiplexing MIMO Systems

وبعد المناقشة العلنية التي تمت اليوم الأربعاء 11 صفر 1431هـ، الموافق 2010/01/27م الساعة الثانية عشرة والنصف ظهراً، اجتمعت لجنة الحكم على الأطروحة والمكونة من:

	مشرفاً ورئيساً	د. عمار أبو هدروس
	مناقشاً داخلياً	د. فادي النحل
	مناقشاً خارجياً	د. أنور موسى

وبعد المداولة أوصت اللجنة بمنح الباحث درجة الماجستير في كلية الهندسة / قسم الهندسة الكهربائية/أنظمة التحكم.

واللجنة إذ تمنحه هذه الدرجة فإنها توصيه بتقوى الله ولزوم طاعته وأن يسخر علمه في خدمة دينه ووطنه.

والله ولي التوفيق،،،

عميد الدراسات العليا

  
د. زياد إبراهيم مقداد

# **Complexity and Performance Evaluation of Detection Schemes for Spatial Multiplexing MIMO Systems**

by

Auda M. Elshokry

Committee Chairman: Dr. Ammar Abu Hudrouss

## **Abstract**

Multiple Input Multiple Output (MIMO) multiplexing is a promising technology that could greatly increase the channel capacity without additional spectral resources. The challenge is to design low complexity and high performance algorithms that capable of accurately detecting the transmitted signals.

In this study, the general model of MIMO communication system was introduced in addition to several MIMO Spatial Multiplexing (SM) detection techniques. The BER performance and computational complexity of the optimal and sub-optimal MIMO detection schemes have been analyzed and compared to each other. For ease of understanding and fair comparison, the MIMO detection techniques are categorized into three main categories; viz., linear schemes, successive interference cancelation, and tree-search techniques. Different aspects have been considered and discussed in this evaluation such as; signal to noise ratio, channel matrix conditionality, number of transmit and receive antennas, and other performance limiting factors. The complexity evaluations and performance comparisons and graphs have been generated using an optimized simulator. This simulator has been developed using MATLAB® platform, hence, it can be considered as a reference implementation for any further research on the field of MIMO SM detection.

# تقييم الأداء ودرجة التعقيد لتقنيات الإكتشاف في أنظمة الاتصالات اللاسلكية متعددة المداخل والمخارج

للباحث: عودة محمد الشكري  
مشرف البحث: د. عمار أبو هديروس

## المخلص

يعتبر نظام الاتصالات متعدد المداخل والمخارج من التقنيات الواعدة في مجال الاتصالات اللاسلكية. و تتمثل أهمية مثل هذه الأنظمة في قدرتها الفائقة على زيادة سعة قناة الاتصال وزيادة معدل الإرسال دون الحاجة إلى موارد إضافية. وتتضمن تلك الأنظمة تحدياً رئيسياً يتمثل في تصميم خوارزميات قادرة على كشف الإشارات المرسله بأداء عالي و درجة تعقيد مقبولة.

في هذه الدراسة، تم عرض النموذج التقليدي لنظام الاتصالات اللاسلكي متعدد المداخل والمخارج بالإضافة إلى وصف عدة خوارزميات استكشاف للإشارات المرسله. وتم مقارنة أداء وتحليل درجة التعقيد الحسابي لتلك الخورزميات. و لسهولة الفهم والمقارنة الموضوعية تم تصنيف خوارزميات الاستكشاف لأنظمة الاتصالات اللاسلكية متعددة المداخل والمخارج إلى ثلاث فئات رئيسية ، وهي: خوارزميات خطية ، خوارزميات إلغاء التداخل المتعاقبة ، و خورزميات البحث الشجري. أثناء دراسة وتقييم تلك الخورزميات تم اعتبار العديد من المتغيرات والعوامل المقيدة و المؤثرة في النظام مثل ؛ نسبة الإشارة إلى نسبة الضوضاء، حالة قناة الإتصال، عدد هوائيات الإرسال والإستقبال. تم استخدام برنامج "MATLAB" لإنتاج الرسوم البيانية التي توضح وتقارن الأداء ودرجة التعقيد لجميع خوارزميات الاستكشاف. وقد تم تطوير نظام شامل لخورزميات الاستكشاف يمكن اعتباره كمرجع لإجراء مزيد من البحوث والدراسات في مجال تقنيات الاستكشاف في أنظمة الاتصالات اللاسلكية متعددة المداخل والمخارج.

*To all whom I love*

## **Acknowledgements**

I would like to express thanks to my advisor Dr. Ammar Abu Hudrouss for his invaluable advice and comments from deciding the thesis topic to revising the work. His guidance and consistent encouragement allowed me complete this work. I am deeply grateful to Dr. Anwar Mousa and Dr. Fadi Alnahal for accepting to be on my thesis committee and for providing many insightful discussion and comments. Thanks to Dr. Manar Mohaisen for the helpful discussions and notes throughout my research. I am extremely impressed with his advising and generosity. Finally, I thank my parents and my wife for their support, patience and prayers through my life.

## Contents

<b>Abstract</b> .....	iii
<b>Contents</b> .....	ivvii
<b>List of Figures</b> .....	x
<b>List of Tables</b> .....	xi

## CHAPTER 1

### INTRODUCTION

1.1 Background .....	1
1.2 Motivation .....	2
1.3 Objectives .....	3
1.4 Thesis Organization .....	4
1.5 Terminology .....	4
1.5.1 Abbreviations .....	4
1.5.2 List of Terms .....	6

## CHAPTER 2

### OVERVIEW OF MIMO SYSTEMS

2.1 Introduction .....	10
2.2 MIMO Diversity Techniques .....	11
2.2.1 Transmit Diversity .....	11
2.2.2 Receive Diversity .....	12
2.3 MIMO Spatial Multiplexing Techniques .....	14
2.4 Advantages of MIMO Systems .....	15
2.5 MIMO System Model .....	16
2.6 Spatial Multiplexing and Detection Problem .....	18
2.7 Summary .....	20

## CHAPTER 3

### DETECTION TECHNIQUES FOR MIMO SPATIAL MULTIPLEXING SYSTEMS

3.1 Linear Detection Techniques .....	21
3.1.1 Zero-Forcing .....	22

3.1.2 Minimum Mean Square Error .....	23
3.2 V-BLAST Detection .....	25
3.2.1 Zero-Forcing VBLAST (ZF-VBLAST) .....	26
3.2.2 Minimum Mean Square Error VBLAST (MMSE-VBLAST) .....	29
3.3 QR Decomposition Based Detection .....	31
3.3.1 Zero-Forcing QR Decomposition (ZF-QRD).....	31
3.3.2 Minimum Mean Square Error QR Decomposition (MMSE-QRD) .....	34
3.4 Tree-Search Detection Techniques .....	36
3.4.1 Sphere Decoding .....	37
3.4.2 QRD-M Detection .....	50
3.4.3 SD and QRD-M Performance Comparison .....	58
3.5 Summary .....	60

## **CHAPTER 4**

### **COMPLEXITY ANALYSIS OF MIMO SM DETECTION TECHNIQUES**

4.1 Introduction .....	61
4.2 Complexity of Arithmetic Operations .....	62
4.3 Complexity analysis of linear detections .....	66
4.3.1 Complexity of Zero-Forcing .....	66
4.3.2 Complexity of MMSE .....	68
4.4 Complexity analysis of SIC (VBLAST) detection.....	69
4.4.1 Complexity of ZF-VBLAST .....	69
4.4.2 Complexity of MMSE-VBLAST .....	73
4.5 Complexity of QR Decomposition Based Detection.....	75
4.5.1 Complexity of Zero-Forcing QR Decomposition (ZF-QRD) Detection.....	75
4.5.2 Complexity of Zero-Forcing Sorted QR Decomposition (ZF-SQRD).....	79
4.5.3 MMSE Sorted QR Decomposition (MMSE-SQRD) .....	81
4.5.4 MMSE QR Decomposition (MMSE-SQRD) .....	83
4.6 Complexity comparison of linear, SIC and QRD detection techniques .....	83
4.7 Complexity analysis of tree search algorithms.....	84
4.7.1 Complexity of sphere decoding detection .....	85
4.7.2 Complexity of QRD-M algorithm detection.....	88
4.7.3 Complexity Comparison of QRD-M and SD .....	88

4.8 Summary .....90

**CHAPTER 5**

**CONCLUSION AND FUTURE WORK**

5.1 Conclusion .....91  
5.2 Future work .....93  
References .....95

## List of Figures

Figure 2.1 Transmit diversity .....	12
Figure 2.2: Receive diversity: selection combining.....	13
Figure 2.3: MIMO receive diversity: maximal ratio combining .....	14
Figure 2.4: MIMO Spatial Multiplexing system .....	15
Figure 2.5: SM system model including both transmitter and receiver main functional blocks .....	17
Figure 3.1: MIMO SM with linear receiver. ....	21
Figure 3.2: BER of linear detection algorithms.....	24
Figure 3.3: BER of VBLAST detection schemes .....	30
Figure 3.4: BER of QRD detection schemes.....	35
Figure 3.5: Geometrical representation of the idea behind SD algorithm .....	37
Figure 3.6: Fincke-Pohst strategy .....	38
Figure 3.7: Schnorr-Euchner strategy .....	39
Figure 3.8: Sequence of testing the hypothesis .....	47
Figure 3.9: Example illustrating SD search tree. Nodes visited by the algorithm are shown in black .....	49
Figure 3.10: Flowchart of QRD-M detection algorithm .....	53
Figure 3.11: the final tree showing the detection levels and the estimate $\hat{x}$ .....	58
Figure 3.12: BER of SD and QRD-M performance for several values of $M$ .....	59
Figure 4.1 Number of floating point operations for linear, VBLAST and QRD detection techniques of a MIMO system with $N_t = N_r$ antennas .....	84

## List of Tables

Table 3.1: Pseudocode for the ZF-VBLAST detection algorithm

Table 3.2: Pseudocode for the MMSE-VBLAST detection algorithm

Table 3.3: Pseudocode of Sorted ZF-QRD detection algorithm

Table 4.1: Computational complexity of arithmetic operations

Table 4.2: ZF QR decomposition procedures

Table 4.3: Steps of detection stage

Table 4.4: ZF-SQRD algorithm

Table 4.5: MMSE-SQRD algorithm

Table 4.6: Complexity comparison for a QRSK MIMO system with  $N_t = N_r = 4$

Table 4.7: Complexity comparison for 16 QAM MIMO system with  $N_t = N_r = 4$

# CHAPTER 1

## INTRODUCTION

---

### 1.1 Background

In the recent few years, Multiple Input Multiple Output (MIMO) systems have drawn a significant attention in the area of wireless technologies. The earliest ideas in this field go back to the work by A.R. Kaye and D.A. George (1970) and W. van Etten (1975, 1976). In 1984 and 1986, Jack Winters and Jack Salz at Bell Laboratories published several papers on beamforming related applications. A. Paulraj and T. Kailath proposed the concept of Spatial Multiplexing (SM) using MIMO in 1993. Their US Patent No. 5,345,599 issued in 1994 on spatial multiplexing emphasized applications to wireless broadcast. In 1996, Greg Raleigh and Gerard J. Foschini refined new approaches to MIMO technology, which considers configurations where multiple transmit antennas are co-located at one transmitter to improve the link throughput effectively. Bell Labs was the first to demonstrate a laboratory prototype of spatial multiplexing in 1998, where spatial multiplexing is a principal technology proposed to improve the performance to increase the capacity of MIMO communication systems.

In the industry, Iospan Wireless Inc. developed the first commercial system in 2001 that used MIMO-OFDMA technology. Iospan technology supported both diversity coding and spatial multiplexing. In 2005, Airgo Networks had developed a pre-11n version based on their patents on MIMO. Following that in 2006, several

companies (Broadcom, Intel,...) have fielded a MIMO-OFDM solution based on a pre-standard for IEEE 802.11n WiFi standard. Also in 2006, several companies (Beceem Communications, Samsung, Runcom Technologies, etc.) developed MIMO-OFDMA based solutions for IEEE 802.16e WIMAX broadband mobile standard. All upcoming 4G systems will also employ MIMO technology. Several research groups have demonstrated over 1 Gbit/s prototypes.

MIMO communications systems can exploit spatial multiplexing (SM) approach to increase the channel capacity and improve spectral efficiency as well. Therefore, the MIMO SM-based system is one of currently promising techniques that can achieve high-speed wireless communications networks. In MIMO SM-based systems, independent data streams are transmitted from sufficiently-separated antennas. This results in a linear increase in the channel capacity proportional to the minimum number of receive and transmit antennas. However, MIMO SM-based system requires powerful signal processing procedures at the receiver to efficiently recover the signal transmitted from the multiple antennas, and hence to explore the advantages of MIMO systems. Therefore, the potential advantages of MIMO system can be guaranteed and the wireless system will work in the best possible way.

Some special detection techniques have been proposed in the literature in order to exploit the high spectral capacity offered by MIMO systems. These techniques are grouped into three main categories: linear, nonlinear, and tree-search.

## **1.2 Motivation**

Day by day, wireless communication systems require significantly higher spectral efficiency (i.e., higher transmission rate measured in bit/second/Hz) and improved quality of service. The intuitive solution to increase the system capacity is

to assign additional bandwidth where the capacity can be increased linearly. The spectral resources assigned to wireless communications are not only expensive, but also limited. Thus, in many cases it is infeasible to use more spectral resources.

MIMO Spatial Multiplexing (SM) seems to be the ultimate solution to increase the system capacity without requiring the need to additional spectral resources. In SM, multiple signals are transmitted instantaneously via enough spaced-antennas. At the receiver side, the main challenge resides in designing signal processing techniques, i.e., detection techniques, capable of separating those transmitted signals with acceptable complexity and achieved performance.

Motivated by the importance of the detection techniques as an important factor in determining both the feasibility and performance of the MIMO-SM systems, this study only considers the receiver structure for the MIMO-SM techniques. The study includes a detailed performance analysis of detection algorithms. Also, deep understanding of the affecting factors on the SM performance are covered including, the number of transmit and receive antennas, constellation size.

### **1.3 Objectives**

As the MIMO detection is a challenging topic for researchers and communication system designers, huge research efforts were done in the recent years giving the birth to a variety of detection schemes that differ in strategy adapted, computational complexity, and performance. This thesis mainly achieves the following objectives;

- Get a more fundamental understanding of MIMO technology
- Introduce a good and useful MIMO spatial multiplexing model

- Evaluate several MIMO-SM detection techniques by comparing BER performance simulations and analyzing the computational complexities
- Evaluate and find an efficient MIMO-SM detection techniques in terms of performance and complexity that is recommended to hardware implementation

## **1.4 Thesis Organization**

In chapter 2, the key concepts behind MIMO communication theory, particularly spatial multiplexing and MIMO system model, have been reviewed. In addition, the system descriptions and assumptions used throughout this thesis have been presented. Chapter 3 was the core of this thesis; it firstly presented the theory behind most MIMO detection schemes described in literature. Secondly, an independent code for each detection algorithm has been written. This chapter also included BER performance comparisons among detection algorithms of the aforementioned three categories. Chapter 4 presented a complexity analysis and comparison for all detection schemes that have been described and discussed in chapter 3. Chapters 5 concluded the most important attained results and suggested different research topics for future work.

## **1.5 Terminology**

### **1.5.1 Abbreviations**

MIMO: Multiple-Input Multiple-Output

SISO: Single-Input Single-Output

SNR: Signal to Noise Ratio

SC: Selection Combining

MRC: Maximal Ratio Combining

EGC: Equal Gain Combining

$N_t$  : Number of transmit antennas

$N_r$  : Number of receive antennas

$x$  : Transmitted symbol vector

$\Omega$  : Constellation set

QAM: Quadrature Amplitude Modulation

$H$  : Channel matrix

CN (0,1): Complex Gaussian distribution with zero mean and unity variance

$r$  : Received symbol vector

$n$  : Noise vector

iid: independent and identically distributed

SM: Spatial Multiplexing

OFDM: Orthogonal Frequency Division Multiplexing

OFDMA: Orthogonal Frequency Division Multiple Access

MLD: Maximum Likelihood Detector

SD: Sphere Decoding

QRD-M: QR Decomposition with M-algorithm

ZF: Zero-Forcing

MMSE: Minimum Mean Square Error

$(\cdot)^\dagger$  : Moore-Penrose pseudo-inverse

$G$  : Filtering (weighting) matrix

$\sigma^2$  : Noise variance

QPSK: Quadrature phase-shift keying

$E_b$  : Average bit energy

$N_o$  : Noise power

BER: Bit Error Rate

VBLAST: Vertical Bell Laboratories Layered Space Time

$Q(\cdot)$  : Demodulation function

QRD: QR Decomposition

$Q$  : A unitary matrix

$R$  : An upper triangular

$Q^H$  : Hermitian transpose of  $Q$

$y$  : Modified received signal vector

$p$  : Permutation vector

SD: Sphere Decoding

$d$  : Sphere radius

FP: Fincke-Pohst searching strategy

SE: Schnorr-Euchner searching strategy

QRD-M: QR-Decomposition with M-algorithms

$M$  : Number of survival candidates at each detection level

## **1.5.2 List of Terms**

### **1.5.2.1 3GPP LTE**

LTE stands for Long Term Evolution and it is the name given to a project within the 3GPP to improve the universal mobile telecommunications standard (UMTS). The LTE project resulted in Release 8 of the UMTS standard, including extensions and

modifications of the UMTS system. LTE Advanced is a more recent project that addresses the 4G technology requirements, such as increased data rates and reduced latency.

#### **1.5.2.2 Array Gain**

In MIMO communication systems, array gain means a power gain of transmitted signals that is achieved by using multiple-antennas at transmitter and/or receiver.

#### **1.5.2.3 Diversity Gain**

In wireless communications, diversity gain is the increase in signal-to-interference ratio due to some diversity scheme, or how much the transmission power can be reduced when a diversity scheme is introduced, without a performance loss.

#### **1.5.2.4 QAM**

Quadrature amplitude modulation is a modulation scheme. It conveys two message signals/ streams, by changing the amplitudes of two carrier waves, using the amplitude-shift keying (ASK) digital modulation scheme or amplitude modulation (AM) analog modulation scheme. These two waves, usually sinusoids, are out of phase with each other by  $90^\circ$  and are thus called quadrature carriers.

#### **1.5.2.5 OFDM**

OFDM is a frequency-division multiplexing (FDM) scheme utilized as a digital multi-carrier modulation method. A large number of closely-spaced orthogonal sub-carriers are used to carry data. The data is divided into several parallel data streams or channels, one for each sub-carrier. Each sub-carrier is modulated with a conventional modulation scheme (such as QAM or PSK) at a low symbol rate, maintaining total

data rates similar to conventional single-carrier modulation schemes in the same bandwidth

### **1.5.2.6 QR Decomposition**

The QR decomposition (also called a QR factorization) of a matrix is a decomposition of the matrix into an orthogonal and a right triangular matrix. QR decomposition is often used to solve the linear least squares problem

### **1.5.2.7 Unitary matrix**

A unitary matrix is an  $N \times N$  complex matrix  $U$  satisfying the condition  $U^*U = UU^* = I_N$ , where  $I_N$  is the identity matrix and  $U^*$  is the conjugate transpose of  $U$ . Noting that  $U$  must have an inverse which equal conjugate transpose conjugate transpose  $U^*$ , i.e.  $U^{-1} = U^*$ .

### **1.5.2.8 Upper triangular matrix**

The upper triangular matrix (right triangular matrix) is a special kind of square matrix where the entries below the main diagonal are zero

### **1.5.2.9 Depth-first search**

Depth-first search is an algorithm for searching a tree, tree structure, or graph. One starts at the root (selecting some node as the root in the graph case) and explores as far as possible along each branch before backtracking.

#### **1.5.2.10 Breadth-first search**

Breadth-first search is a graph search algorithm that begins at the root node and explores all the neighboring nodes. Then for each of those nearest nodes, it explores their unexplored neighbor nodes, and so on, until it finds the goal.

#### **1.5.2.11 Flops**

In computing, FLOPS (or flop/s) is an acronym meaning Floating Point Operations Per Second. The FLOPS is a measure of a computer's performance, especially in fields of scientific calculations that make heavy use of floating point calculations.

## CHAPTER 2

### OVERVIEW OF MIMO SYSTEMS

---

#### 2.1 Introduction

During the last decade, the intensive work of researchers on Multiple-Input Multiple-Output (MIMO) techniques has demonstrated their key role in increasing the channel reliability and improving the spectral efficiency in wireless communication systems without the need to additional spectral resources [1].

To meet the exaggerated demands on high transmission rate in Single-Input Single-Output (SISO) wireless communication systems, the capacity can be increased by allocating additional bandwidth which is not always possible because spectral resources are not only expensive but also scarce [2].

Recent developments have shown that using spatial multiplexing MIMO systems can increase the capacity in wireless communication substantially without requiring extra-bandwidth [1], [3]. In MIMO systems, multiple antenna elements are deployed at the transmitter and/or the receiver, as shown in Figure 2.1. Thus, instead of sending only one signal at every time instant, i.e. time slot,  $N_t$  signals are transmitted at the same time instant using the same frequency band. As a result, the capacity of the overall system is linearly proportional to  $N_t$ , which is a considerable increase in the capacity and the spectral efficiency is improved as well. In general, MIMO techniques are grouped into two main categories, namely; MIMO diversity techniques and MIMO spatial multiplexing techniques. MIMO diversity techniques

can provide higher signal-to-noise ratio (SNR), which improve transmission reliability. On the other hand, MIMO spatial multiplexing techniques can provide a linear increment in the channel capacity without requiring additional spectral resources. In this study, we restrict our work on MIMO spatial multiplexing systems and their related detection schemes due to the strong need for such systems in the future communication systems (i.e. 4G technologies).

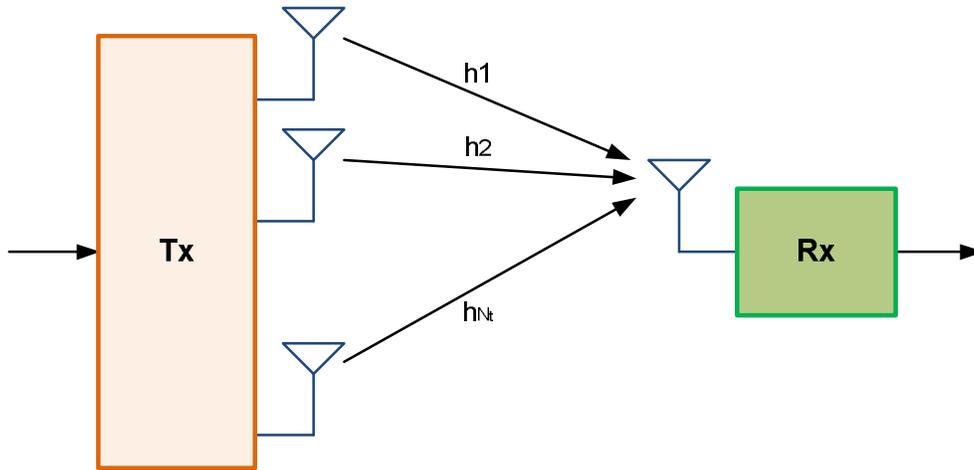
## **2.2 MIMO Diversity Techniques**

The key idea in MIMO diversity techniques is that the same data stream is transmitted from multiple antennas or received at more than one antenna. MIMO diversity schemes are impressively effective in increasing the diversity gain where consequently performance is improved [4]. Diversity can be implemented at the transmit end (transmit diversity), at the receive end (receive diversity) or at both ends of the wireless link. Generally, MIMO diversity techniques can provide higher SNR and improve transmission reliability as a result.

### **2.2.1 Transmit Diversity**

Transmit diversity improves the signal quality and achieves a higher SNR ratio at the receiver side; it involves transmitting data stream through multiple antennas and receiving by single antenna or more. Transmit diversity can effectively mitigate multipath fading effects as multiple antennas afford a receiver several observations of the same data stream. Each antenna will experience a different interference environment and if one antenna experienced a deep fade, then it is likely that another has a sufficient signal. Thus, transmit diversity can help improve the reliability of the data reception and data decoding as well. The most popular examples of these transmit diversity techniques include Alamouti code [5] and orthogonal codes

proposed by Taroukh et al. [6]. Figure 2.1 depicts the whole system for an exemplary  $N_t$  transmit antenna system.



**Figure 2.1 Transmit diversity**

### 2.2.2 Receive Diversity

Receive diversity are widely used in wireless communication systems; it can be achieved by receiving redundant copies of the same signal. The idea behind receive diversity is that each antenna at the receive end can observe an independent copy of the same signal. Therefore the probability that all signals are in deep fade simultaneously is significantly reduced. This type of diversity hasn't particular settings or requirements on the transmit end, but requires a receiver that could simultaneously process all received signals and combines them by a proper combining method [4]. There are several classical methods for combining the different diversity branches at the receiver [7], [8], the most important of which and most widely used are Selection Combining (SC), Maximal Ratio Combining (MRC) and Equal Gain Combining (EGC).

## I. Selection Combining

Selection combining shown in Figure 2.2 is the simplest form of receive diversity combining methods. Fundamentally it estimates the instantaneous SNR for each of the received signals and selects the particular receiver output with the strongest SNR among  $N_t$  diversity branches; where  $N_t$  is the number of receive antennas in the system.

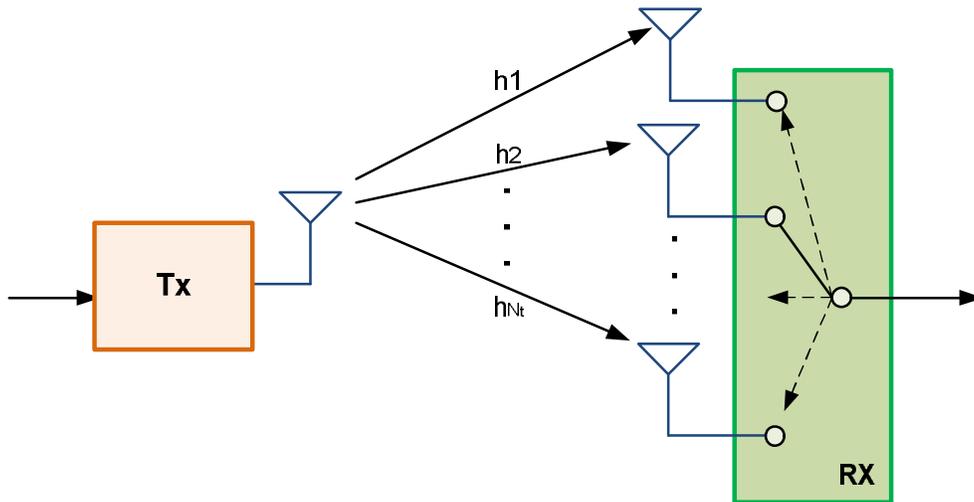


Figure 2.2: Receive diversity: selection combining

## II. Maximal Ratio Combining

The selection combining technique ignores information from all diversity branches except the particular branch that has the highest SNR. This drawback is mitigated by using Maximal Ratio Combining, in which the information from all branches is combined in order to maximize the output SNR [9]. MRC works by weighting each branch with the complex conjugate of their particular channel coefficients and then do summation to produce the received signal as shown in Figure 2.3.

### III. Equal Gain Combining

Equal Gain Combining (EGC) is similar to Maximal Ratio Combining without weighting the signals before summation [10]. In EGC co-phasing is needed to avoid signal cancellation. The average SNR improvement of EGC is typically about 1 dB worse than with MRC, but still simpler to implement than MRC.

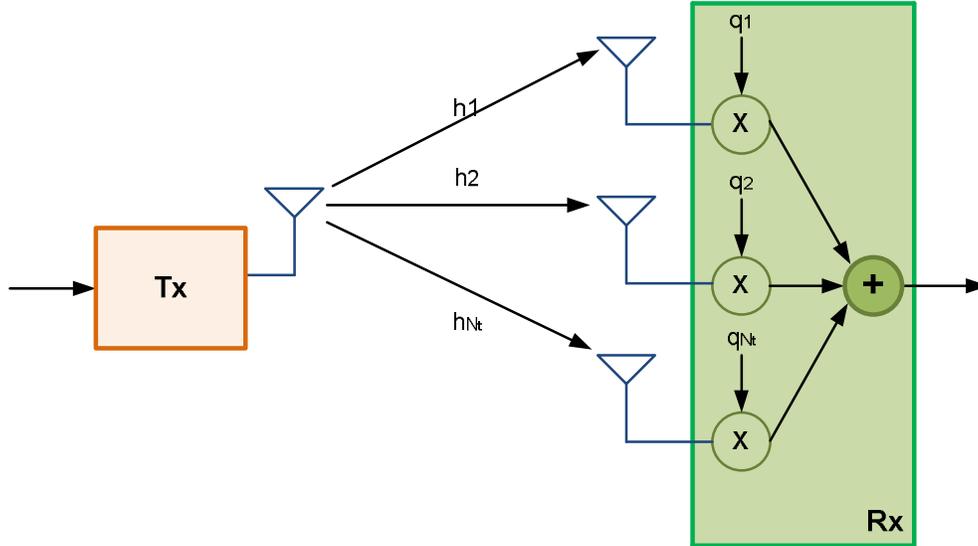
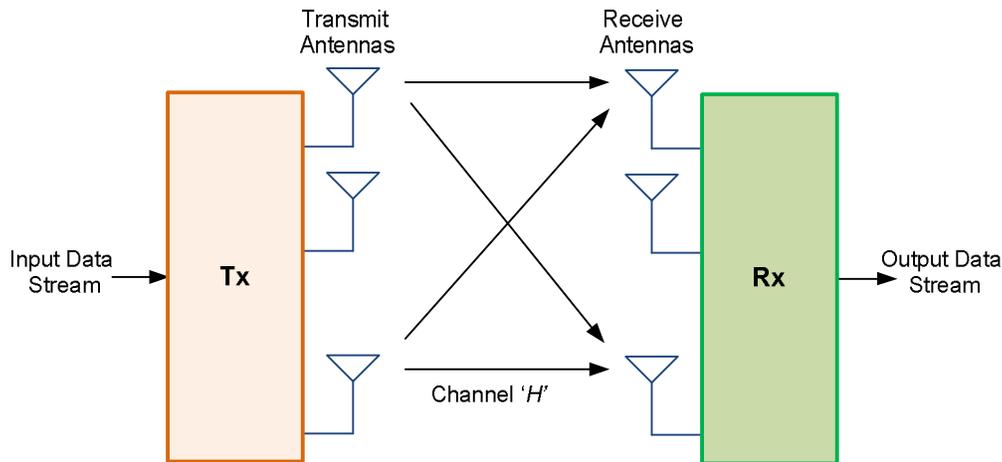


Figure 2.3: MIMO receive diversity: maximal ratio combining

### 2.3 MIMO Spatial Multiplexing Techniques

Spatial Multiplexing (SM) has been utilized in MIMO systems to provide higher transmission rate without allocating additional bandwidth or increasing the transmit power [11]. The VBLAST [3] was the practical implementation approach of spatial multiplexing technique. Spatial multiplexing involves deploying multiple antennas at both transmitter and receiver ends as shown in Figure 2.4. Input data streams can be divided into different independent sub streams and then transmitted simultaneously via sufficiently-separated antennas ( $L/2$  or more, to obtain highly uncorrelated and independent signal). It has been shown in [2] that utilizing spatial

multiplexing schemes under certain conditions and assumptions, can linearly increase capacity with relation to the minimum of the number of transmit antennas and the number of receive antennas.



**Figure 2.4: MIMO Spatial Multiplexing system**

## 2.4 Advantages of MIMO Systems

The increased demand on higher transmission rate for the cutting edge wireless applications makes MIMO technology very important for the future wireless communication systems. MIMO systems can provide different advantages over single input single output conventional system [12]. These advantages can be summarized in the followings:

- MIMO systems exploit multiple antennas diversity at transmitter/receiver. This gives the possibility to increase system reliability by one of the transmit diversity techniques in which the same signal is transmitted through multiple antennas. Different copies of the signal can be observed at the receiver and the probability that at least one of the copies is not experiencing a deep fade

increases. Thus the receiver can successfully recover the signal with decreased bit/symbol error rate and overall system performance is improved as well.

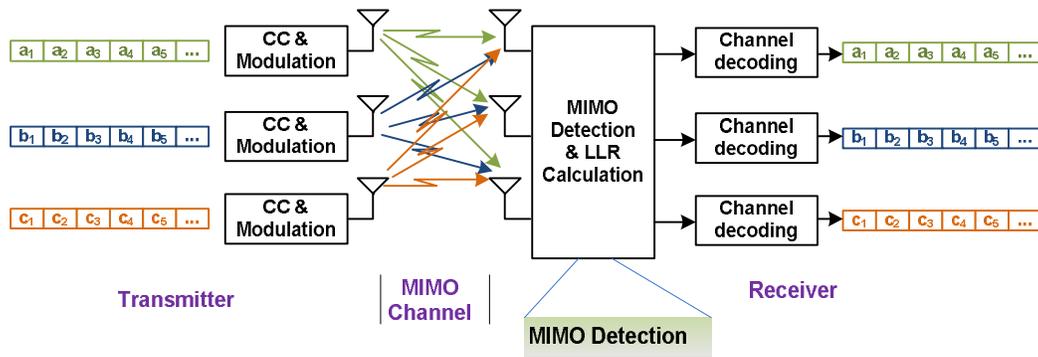
- MIMO systems can proportionally increase the achievable data-rates by spatial multiplexing, i.e. transmitting multiple independent data streams within the allocated bandwidth. Thereafter, the receiver can separate/recover the data streams under certain channel conditions such as a rich scattering surrounding wireless channel.
- MIMO systems can produce different gains such as array gain, diversity gain and multiplexing gain. Despite the fact that these gains compete each other, they may combined to increase the coverage area and to reduce the required transmit power [4]. Assume that there are  $N_r$  receive antennas and only one transmit antenna, then the average SNR is approximately  $N_r$ , then it can be found that the coverage area is increased by a multiplicative factor  $N_r\gamma$ , where  $\gamma$  is the average SNR per branch. This can be used to increase the coverage area for a fixed transmitted power, or it can be used to reduce the transmitted power requirement for a given coverage area.

The most benefit behind using MIMO technology is that all above advantages are achieved without requiring any additional bandwidth for the wireless system. Moreover, the offered benefits can help meet the challenges posed by both the impairments in the wireless channel as well as resource constraints and limitations. Therefore, MIMO technology constitutes a breakthrough in a wireless communication system design.

## 2.5 MIMO System Model

In this thesis, we consider a conventional MIMO SM system with  $N_t$  transmit antennas and  $N_r$  receive antennas where  $N_t \leq N_r$  as shown in Figure 2.5. Independent

data streams  $a$ ,  $b$ , and  $c$ , are encoded and modulated before being transmitted. Herein, consider a transmitted vector  $\mathbf{x} = [x_1, x_2, \dots, x_{N_t}]^T$  whose elements are drawn independently from a complex constellation set  $\Omega$ , e.g. Quadrature Amplitude Modulation (QAM) constellation. The vector is then transmitted via a MIMO channel characterized by the channel matrix  $\mathbf{H}$  whose element  $h_{i,j} : CN(0,1)$ <sup>1</sup> is the complex channel coefficient between the  $j^{\text{th}}$  transmit and  $i^{\text{th}}$  receive antennas. The received vector  $\mathbf{r} = [r_1, r_2, \dots, r_{N_r}]^T$  can then be given as following,



**Figure 2.5: SM system model including both transmitter and receiver main functional blocks**

$$\mathbf{r} = \mathbf{H}\mathbf{x} + \mathbf{n} , \quad (2.1)$$

where the elements of the vector  $\mathbf{n} = [n_1, n_2, \dots, n_{N_r}]^T$  are drawn from independent and identically distributed (i.i.d.) circular symmetric Gaussian random variables. The system model of (2.1) is then given in the matrix form as following.

$$\begin{bmatrix} r_1(k) \\ \vdots \\ r_{N_r}(k) \end{bmatrix} = \begin{bmatrix} h_{11} & \cdots & h_{1N_t} \\ \vdots & \ddots & \vdots \\ h_{N_r1} & \cdots & h_{N_rN_t} \end{bmatrix} \begin{bmatrix} x_1(k) \\ \vdots \\ x_{N_t}(k) \end{bmatrix} + \begin{bmatrix} n_1(k) \\ \vdots \\ n_{N_r}(k) \end{bmatrix}.$$

<sup>1</sup> Normal distribution with a zero mean and unity variance

## 2.6 Spatial Multiplexing and Detection Problem

Spatial Multiplexing (SM) seems to be the ultimate solution to increase the system capacity without the need to additional spectral resources. The basic idea behind SM [4] is that a data stream is demultiplexed into  $Nt$  independent substreams as shown in Figure 2.5, and each substream is then mapped into constellation symbols and fed to its respective antenna. The symbols are taken from a QAM constellation. The encoding process is simply a bit to symbol mapping for each substream, and all substreams are mapped independently. The total transmit power is equally divided among the  $Nt$  transmit antennas. At the receiver side, the main challenge resides in designing powerful signal processing techniques, i.e., detection techniques, capable of separating those transmitted signals with acceptable complexity and achieved performance. Given perfect channel knowledge at the receiver, a variety of techniques including linear, successive, tree search and maximum likelihood decoding can be used to remove the effect of the channel and recover the transmitted substreams, see for example [13-15]. Different research activities have been carried out to show that the spatial multiplexing concept has the potential to significantly increase spectral efficiency [11], [16]. Further research has been carried out on creating and evaluating enhancements to the spatial multiplexing concepts, such as combining with other modulation schemes like OFDM (Orthogonal Frequency Division Multiplexing) [17]. In general, this technique assumes channel knowledge at the receiver and the performance can be further improved when the knowledge of the channel response is available at the transmitter. However, SM does not work well in low SNR environments as it is more difficult for the receiver to recognize the multiple uncorrelated paths of the signals [18], [19].

The main challenge in the practical realization of MIMO wireless systems lies in the efficient implementation of the detector [20] which needs to separate the spatially multiplexed data streams. So far, several algorithms offering various trade-offs between performance and computational complexity have been developed [21]. Linear detection (low complexity, low performance) constitutes one extreme of the complexity/ performance region, while Maximum Likelihood Detector (MLD) detection algorithm has an opposite extreme (high complexity, optimum performance).

Maximum Likelihood Detector (MLD) is considered as the optimum detector for the system of (2.1) that could effectively recover the transmitted signal at the receiver based on the following minimum distance criterion,

$$\tilde{x} = \mathbf{arg}_{x_k \in \{x_1, x_2, \dots, x_N\}} \min \|r - Hx_k\|^2, \quad (2.2)$$

where  $x$  is the estimated symbol vector. Using the above criterion, MLD compares the received signal with all possible transmitted signal vector which is modified by channel matrix  $H$  and estimates transmit symbol vector  $x$ . Although MLD achieves the best performance and diversity order, it requires a brute-force search which has an exponential complexity in the number of transmit antennas and constellation set size. For example, if the modulation scheme is 64-QAM and 4 transmit antenna, a total of  $64^4 = 16777216$  comparisons per symbol are required to be performed for each transmitted symbol. Thus, for high problem size, i.e. high modulation order and high transmit antenna ( $Nt$ ), MLD becomes infeasible.

The computational complexity of a MIMO detection algorithm depends on the symbol constellation size and the number of spatially multiplexed data streams, but

often on the instantaneous MIMO channel realization and the signal-to-noise ratio [22]. On the other hand, the overall decoding effort is typically constrained by system bandwidth, latency requirements, and limitations on power consumption. In order to solve the detection problem in MIMO systems, research has been focused on sub-optimal detection techniques which are powerful in terms of error performance and are practical for implementation purposes as well that are efficient in terms of both performance and computational complexity. Two such techniques are Sphere Decoding (SD) and QR Decomposition with M-algorithm (QRD-M) which utilize restrict tree search mechanisms. These algorithms and more linear and non-linear detection techniques will be described and discussed in details in Chapter 3 .

## **2.7 Summary**

This Chapter presented an overview of Multiple Input Multiple Output systems, spatial multiplexing and detection problem. It has been shown that there was an intensive research on developing an efficient detection algorithm for MIMO spatial multiplexing systems to meet the exaggerated demands on high transmission rate for the cutting edge wireless communication systems. In MIMO technology, system performance is improved using spatial diversity techniques. But with spatial multiplexing the channel capacity is linearly increased as independent data streams are transmitted from the multiple transmit antennas and received by multiple antennas at the receiver.

The main challenge in MIMO SM system is the design of detection code with acceptable complexity and achieved performance. The conventional MIMO SM system model has been also described. This model will be utilized in the design of all SM detection schemes in chapter 3.

## CHAPTER 3

### DETECTION TECHNIQUES

#### FOR MIMO SPATIAL MULTIPLEXING SYSTEMS

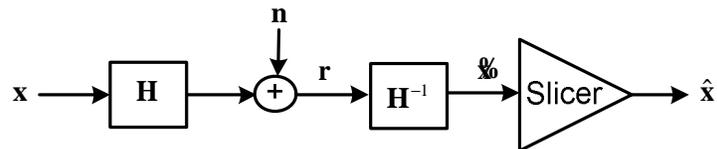
---

Several MIMO detection techniques were proposed in the literature. In this study, a variety of these techniques will be evaluated using different predetermined performance and complexity criteria.

MIMO detection techniques are categorized into three main categories; linear schemes, successive interference cancelation, and tree-search techniques. These techniques are explained in details in the following sections.

### 3.1 Linear Detection Techniques

The idea behind linear detection techniques is to linearly filter received signals using filter matrices, as depicted in Figure 3.1. This category includes Zero-Forcing (ZF) and Minimum Mean Square Error (MMSE) techniques. Although linear detection schemes are easy to implement, they lead to high degradation in the achieved diversity order and error performance due to the linear filtering.



**Figure 3.1: MIMO SM with linear receiver.**

### 3.1.1 Zero-Forcing

Zero-Forcing (ZF) technique is the simplest MIMO detection technique, which was proposed in [3]. Where filtering matrix is constructed using the ZF performance-based criterion. The drawback of ZF scheme is the susceptible noise enhancement and loss of diversity order due to linear filtering [21], [22]. ZF can be implemented by using the inverse of the channel matrix  $H$  to produce the estimate of transmitted vector  $\mathcal{X}$ .

$$\begin{aligned}\mathcal{X} &= H^\dagger r \\ &= H^\dagger (Hx) \\ &= x\end{aligned}\tag{3.1}$$

where  $(\cdot)^\dagger$  denotes the pseudo-inverse. But when the noise term is considered, the post-processing signal is given by:

$$\begin{aligned}\mathcal{X} &= H^\dagger R \\ &= H^\dagger (Hx + n) \\ &= x + H^\dagger n\end{aligned}\tag{3.2}$$

with the addition of the noise vector, ZF estimate, i.e.  $\mathcal{X}$ , consists of the decoded vector  $x$  plus a combination of the inverted channel matrix and the unknown noise vector. Because the pseudo-inverse of the channel matrix may have high power when the channel matrix is ill-conditioned, the noise variance is consequently increased and the performance is degraded.

To alleviate for the noise enhancement introduced by the ZF detector, the MMSE detector was proposed, where the noise variance is considered in the construction of the filtering matrix  $G$ .

### 3.1.2 Minimum Mean Square Error

Minimum Mean Square Error (MMSE) approach alleviates the noise enhancement problem by taking into consideration the noise power when constructing the filtering matrix using the MMSE performance-base criterion. The vector estimates produced by an MMSE filtering matrix becomes

$$\tilde{x} = \left[ \left[ (H^H H + (\sigma^2 I))^{-1} \right] H^H \right] r, \quad (3.3)$$

where  $\sigma^2$  is the noise variance. The added term ( $1/SNR = \sigma^2$ , in case of unit transmit power) offers a trade-off between the residual interference and the noise enhancement. Namely, as the SNR grows large, the MMSE detector converges to the ZF detector, but at low SNR it prevents the worst Eigen values from being inverted.

At low SNR, MMSE becomes Matched Filter [23]:

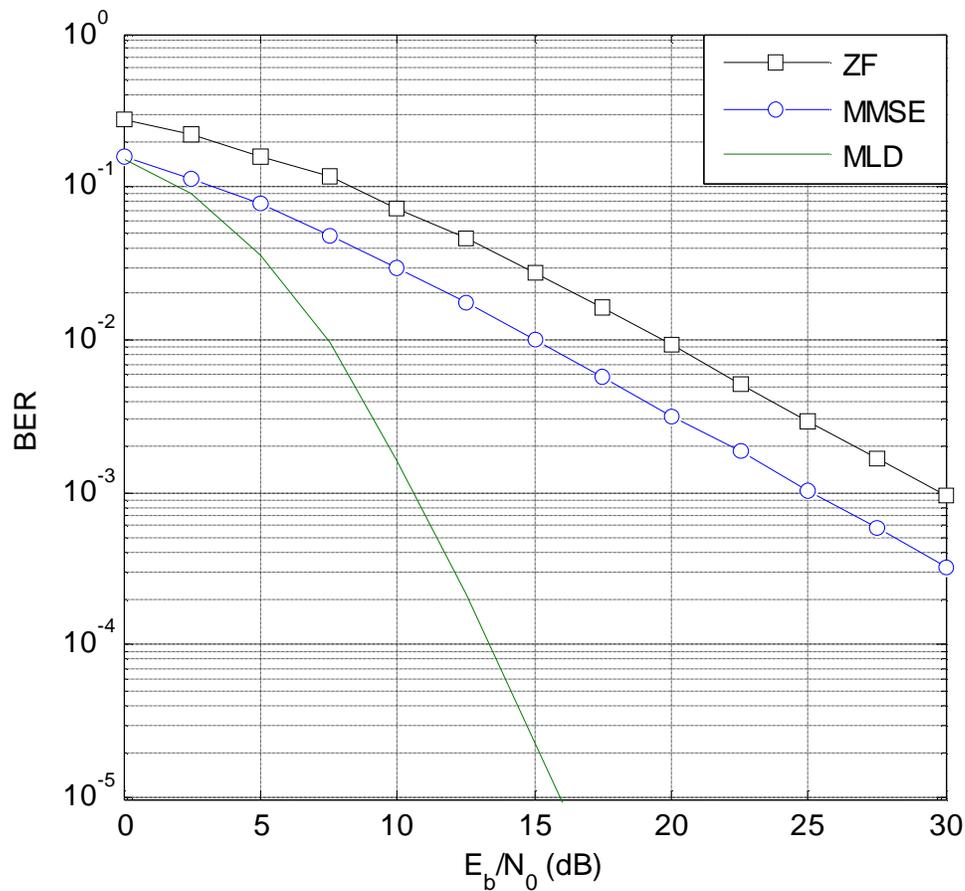
$$\left[ (H^H H + (\sigma^2 I))^{-1} \right] H^H \approx \sigma^2 H^H \quad (3.4)$$

At high SNR, MMSE becomes ZF:

$$\left[ (H^H H + (\sigma^2 I))^{-1} \right] H^H \approx (H^H H)^{-1} H^H \quad (3.5)$$

Figure 3.2 shows performance estimation of the linear detectors, the simulations are done for a  $(N_t, N_r) = (4, 4)$  system with QPSK modulation. The  $E_b/N_o$ , ranges between 0 dB and 30 dB in step of 5 dB. The Bit Error Rate (BER) is calculated by performing 1,500,000 trials at each  $E_b/N_o$  point.

A new realization of  $H$  was chosen in each trial and for each  $E_b/N_o$  value. We observe that there is no convergence of the MMSE and ZF performance curves for high SNR in the simulation results. In this example MMSE curve performs better than ZF by about 5 dB at an error rate of  $10^{-3}$ . Both the ZF and MMSE detectors show a diversity order of more than  $N_r - N_t + 1$ , but less than  $N_r$  [24]. The ML detector is the optimum one and shows the full diversity order of  $N_r$ . Generally, the linear detection schemes are favourable in terms of computational complexity, but their BER performance is severely degraded due to the noise enhancement in the ZF detector case, and when the channel matrix is ill-conditioned.



**Figure 3.2: BER of linear detection algorithms**

### 3.2 V-BLAST Detection

Although linear detection techniques are easy to implement, they lead to high degradation in the achieved diversity order due to the linear filtering. Another approach that takes advantage of the diversity potential of the additional receive antennas, uses nonlinear techniques such as Successive Interference Cancellation (SIC) (for instance, V-BLAST decoder).

The Vertical Bell Laboratories Layered Space Time (V-BLAST) scheme was originally proposed by Foschini [1] and has been discussed in details in literature. The main idea behind the V-BLAST architecture (i.e., transmitter) is to demultiplex the data stream into several sub-streams and transmit them simultaneously. At the receiver side, each antenna observes all the transmitted signals, which are mixed due to the environment surrounding the wireless propagation channel. V-BLAST detection algorithm detects the signals one after another in an iterative way. The construction of the filtering matrix can still be based on any of the aforementioned linear criteria, i.e. ZF or MMSE.

The V-BLAST algorithm utilizes the already detected symbol  $x_i$ , obtained by the ZF or MMSE filtering matrix, to generate a modified received vector with  $x_i$  cancelled out. Thus the modified received vector becomes with fewer interferers and better performance due to a higher level of diversity. The algorithm continues until all  $Nt$  symbols being detected.

If we rewrite the system in (2.1) into a matrix form with  $N_t = N_r = 4$ ,

$$\begin{bmatrix} r_1 \\ r_2 \\ r_3 \\ r_4 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \\ h_{41} & h_{42} & h_{43} & h_{44} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} \quad (3.6)$$

Then, using ZF or MMSE criterion, the estimate of  $x_i$  can be calculated. Assuming that this symbol is correct, it is weighted with its corresponding channel coefficient and then subtracted from the received vector  $r$ . The new modified vector  $y$  becomes :

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} h_{12} & h_{13} & h_{14} \\ h_{22} & h_{23} & h_{24} \\ h_{32} & h_{33} & h_{34} \\ h_{42} & h_{43} & h_{44} \end{bmatrix} \begin{bmatrix} x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} \quad (3.7)$$

Iteratively, the nulling matrix is computed. The newly detected symbol  $x_i$  is subtracted of the already modified received vector  $y$  to produce the following equations :

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \end{bmatrix}_{\text{mod}} = \begin{bmatrix} h_{13} & h_{14} \\ h_{23} & h_{24} \\ h_{33} & h_{34} \\ h_{43} & h_{44} \end{bmatrix} \begin{bmatrix} x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ n_3 \\ n_4 \end{bmatrix} \quad (3.8)$$

Definitely the diversity level is getting better at each stage of detection and the performance is improved because the equations become more than unknowns. This method of successive interference cancellation is continued until all  $N_t$  symbols are detected.

### 3.2.1 Zero-Forcing VBLAST (ZF-VBLAST)

The Zero-Forcing V-BLAST algorithm (ZF-VBLAST) is based on detecting the components of  $x$  one by one. For the first decision, the pseudo-inverse, i.e.,  $G$  equals  $H^\dagger$ , of the matrix  $H$  is obtained. Assume that the noise components are *i.i.d.* and that the noise is independent of  $x$ . Then, the row of  $G$ , with the least Euclidean norm, corresponds to the required component of  $x$ . That is,

$$k_1 = \arg \min_j \left( \|\underline{g}_j\|^2 \right), \quad (3.9)$$

$$\underline{\mathcal{H}}_{k_1} = \underline{g}_{k_1} r^{(1)}, \quad (3.10)$$

and,

$$\hat{x}_{k_1} = Q(\underline{\mathcal{H}}_{k_1}), \quad (3.11)$$

where  $\underline{g}_j$  is the  $j^{\text{th}}$  row of the filtering matrix  $G$ ,  $Q(\cdot)$  is the demodulation function, and the superscript is the iteration index. At the first iteration,  $r^{(1)} = r$  and  $G^{(1)} = H^\dagger$ . At the end of the first iteration, the interference due to the  $k_1^{\text{th}}$  component of  $x$  is cancelled out as follows:

$$r^{(2)} = r^{(1)} - \hat{x}_{k_1} h_{k_1},$$

$$H^{(2)} = H^{(1)-k_1} = [\mathbf{L}, h_{k_1-1}, h_{k_1+1}, \mathbf{L}]$$

Doing so until detecting the last element of  $x$ . When the sorting step in Table 3.1 (line 6) is discarded, the code is called Unsorted ZF-VBLAST or ZF-VBLAST.

Obviously, incorrect symbol detection in the early stages will create errors in the following stages; i.e. error propagation. This is a severe problem with cancellation based detection techniques particularly when the number of transmit and receive antennas are the same. The first detected symbol's performance is quite poor as it has no diversity. To reduce the effect of error propagation and to optimize the performance of VBLAST technique, it has been shown in [14] that the order of detection can increase the performance considerably. By detecting the symbols with

largest channel coefficient magnitude first, the effect of the noise vector producing an incorrect symbol can be reduced, and reducing error propagation as result.

In order to achieve best performance, it is optimal to start detecting the components of  $x$  that suffer the least noise amplification i.e the layer with the largest SNR. Then sorting step (line 6) in the code shown in Table 3.1 will be activated. This algorithm is called sorted Zero-Forcing VBLAST (SZF-VB).

**Table 3.1: Pseudocode for the ZF-VBLAST detection algorithm**

(1)	input $H, r, U$
(2)	$H^{(1)} = H$
(3)	$r^{(1)} = r$
(4)	for $i = 1, \dots, N_T$
(5)	$G_i = H_i^\dagger$
(6)	$k_i = \arg \min \ g_i^{(j)}\ ^2$
(7)	$W_i = g_i^{(k_i)}$
(8)	Exchange columns $k_i$ and $i$ in $U$
(9)	$\mathcal{X}_i = W_i \cdot r$
(10)	$\hat{x}_i = Q[\mathcal{X}_i]$
(11)	$r_{i+1} = r_i - h_i \cdot \hat{x}_i$
(12)	$H_{i+1} = H_i^{\bar{k}_i}$
(13)	end
(14)	Output $U$

It was shown in section 3.1 that ZF is the simple linear receiver with low computational complexity and suffers from noise enhancement. But it can works well at high SNR. However, in Zero- Forcing we can choose any row of  $G_i$  to null the signal from the  $i^{th}$  transmit antenna, while in ZF-VBLAST it was shown that it is best to start with the signal that has the greatest signal to noise ratio (SNR) in which is

known by ordering, which results in a better performance as seen above. The ZF solution in general is an easier solution but not optimum as it enhances the noise. Instead we have used the MMSE method, which gives us better performance.

### 3.2.2 Minimum Mean Square Error VBLAST (MMSE-VBLAST)

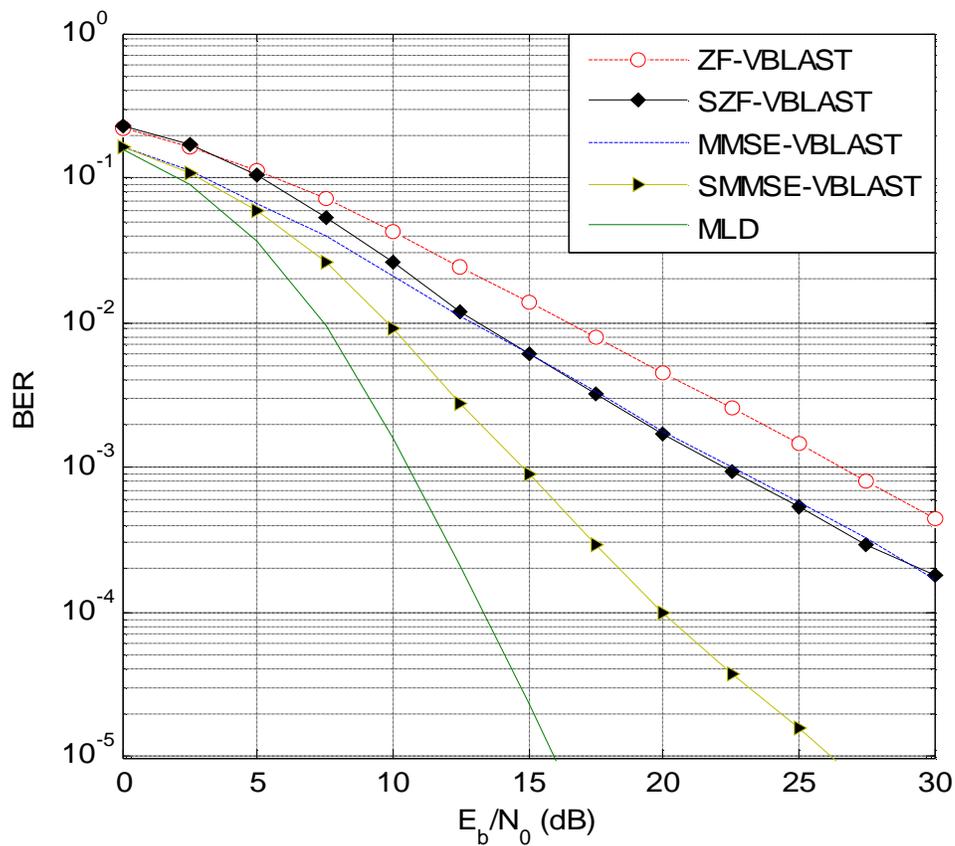
In section 3.1, it was shown that MMSE algorithm suppresses both the interference and noise components, whereas the ZF algorithm removes only the interference components. This implies that the mean square error between the transmitted symbols and the estimate of the receiver is minimized. Therefore, MMSE is superior to ZF in the presence of noise. The MMSE filtering strategy can be used with VBLAST, where the resulting detector is referred to as the MMSE-VBLAST detector.

**Table 3.2: Pseudocode for the MMSE-VBLAST detection algorithm**

(1)	input $H, r, S, U$
(2)	$H^{(1)} = \left[ \left( H^H H + (S^2 I) \right)^{-1} \right] H^H$
(3)	$r^{(1)} = r$
(4)	for $i = 1, \dots, N_T$
(5)	$G_i = H_i^\dagger$
(6)	$k_i = \arg \min \ g_i^{(j)}\ ^2$
(7)	$W_i = g_i^{(k_i)}$
(8)	Exchange columns $k_i$ and $i$ in $U$
(9)	$\mathcal{X}_i = W_i \cdot r$
(10)	$\hat{x}_i = Q[\mathcal{X}_i]$
(11)	$r_{i+1} = r_i - h_i \cdot \hat{x}_i$
(12)	$H_{i+1} = H_i^{\bar{k}_i}$
(13)	end
(14)	Output $U$

Also, we refer to the MMSE-VBLAST as the “Unsorted MMSE-VB” when the sorting stage is skipped. In this case, the components of  $x$  are detected in an ascending order.

Table 3.2 shows the pseudocode for the Sorted MMSE-VBLAST detection algorithm. The MMSE-VB detection algorithm can be obtained by the MMSE criterion in constructing the filtering matrix as shown in step (2) of the code.



**Figure 3.3: BER of VBLAST detection schemes**

The main drawback of the VBLAST detection algorithms lies in the computational complexity, because multiple calculations of the pseudo-inverse of the channel matrix are required [25].

Figure 3.3 shows the performance of various VBLAST detection schemes that utilizing both ZF and MMSE criteria with and without using optimal ordering. Comparing the simulation results of ZF-VBLAST and MMSE-VBLAST separately, the sorted detection schemes achieve an improved performance in comparison to the unsorted ones. At a target BER of  $10^{-3}$  the difference between ZF-VBLAST curves is about 4 dB and the difference between MMSE-VBLAST curves is about 7 dB. This demonstrates the impact of employing signal ordering. Note that the performance advantage of the MMSE is quite considerable in all cases. The sorted MMSE-VBLAST lags the MLD curve by about 6.7 dB at a target BER of  $10^{-4}$ .

### 3.3 QR Decomposition Based Detection

In section 3.2, it was shown that VBLAST detection algorithms imply the calculation of the pseudo-inverse of MIMO channel at each detection step. This involves expensive computational requirements and makes VBLAST algorithms enduring computational bottleneck. This computational bottleneck can be avoided using QR Decomposition based algorithm such as ZF- QRD and MMSE-QRD. In [25], [26], it was shown that QR Decomposition-based algorithms requires only a fraction of the computational efforts required by the V-BLAST detection algorithm.

#### 3.3.1 Zero-Forcing QR Decomposition (ZF-QRD)

It was shown that VBLAST algorithm can be restated in terms of QR decomposition of the channel matrix  $H$  [26-29].

$$H = QR , \tag{3.12}$$

where the  $N_t \times N_r$  matrix  $Q$  has orthogonal columns with unit norm (unitary matrix) and the  $N_t \times N_r$  matrix  $R$  is upper triangular.

Then, the received vector  $r$  in (2.1) is multiplied from the left by the Hermitian transpose of  $Q$ , [30].

$$\begin{aligned}
 r &= H \cdot x + n \\
 r &= QR \cdot x + n \\
 Q^H r &= Q^H QR \cdot x + Q^H n \\
 y &= R \cdot x + v
 \end{aligned} \tag{3.13}$$

a  $N_t \times 1$  modified received signal vector  $y$  can be written in an explicit matrix form as follows:

$$\begin{pmatrix} y_1 \\ y_2 \\ \mathbf{M} \\ y_{N_r} \end{pmatrix} = \begin{pmatrix} r_{11} & r_{12} & \mathbf{K} & r_{1N_t} \\ 0 & r_{22} & \mathbf{L} & r_{2N_t} \\ \mathbf{M} & \mathbf{M} & \mathbf{M} & \mathbf{M} \\ 0 & 0 & \mathbf{K} & r_{N_r N_t} \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ \mathbf{M} \\ x_{N_r} \end{pmatrix} + \begin{pmatrix} v_1 \\ v_2 \\ \mathbf{M} \\ v_{N_r} \end{pmatrix} \tag{3.14}$$

or

$$y_k = R_{k,k} x_k + \sum_{i=k+1}^{n_r} R_{k,j} \hat{x}_i \tag{3.15}$$

then

$$\hat{x}_k = Q \left( \frac{y_k - \sum_{i=k+1}^{n_r} R_{k,j} \hat{x}_i}{R_{k,k}} \right) \tag{3.16}$$

Note that due to the upper triangle structure of the matrix  $R$ , the last element  $x_{N_t}$ , is interference-free and can be used to estimate  $x_{N_t}$ ; hence, it is detected at first

$$\hat{x}_{N_t} = Q \begin{bmatrix} y_{N_t} \\ r_{N_t N_t} \end{bmatrix}$$

by applying the quantization operation  $Q[\cdot]$ .

Detecting  $(k = N_t - 1, \mathbf{K}, 1)$  is carried out in an equivalent way, noting that already-detected components of  $x$  are cancelled out from the received vector. These procedures are repeated up to the first component  $x_1$ .

As mentioned in section 3.2, the detection sequence is critical due to the risk of error propagation. Following the same idea as VBLAST, the symbols can be detected in order of decreasing SNR. Because the last symbol is detected first in this method, one would like the last symbol to be the best one. This requires rearranging the columns of  $H$  in increasing order of 2-norm so that the last symbol corresponding to the last column gets detected first and so on. The optimal ordering can be determined just by permuting the columns of  $x$  according to the elements of  $p$  (where  $p$  is the permutation vector).

Table 3.3 gives a pseudocode of the Sorted ZF-QRD detection algorithm. Again, we meant by 'sorted' that the signal ordering approach is employed in the code. The unsorted ZF-QRD can be obtained by discarding the sorting steps in the code (line (7) and line (8)). The algorithm consists of a decomposition part (line (1) to (16)) and a detection part (line (17) to (21)). In the decomposition part, the ordering is done in line (7) and (8) and provides the permutation vector  $p$ , the orthogonal matrix  $Q$  and the upper triangular matrix  $R$ . In the detection part, the received signal vector is sorted according to the permutation  $p$ , and the modified received signal vector  $y$  is calculated (line (17)). The following lines (17) to (21) represent the iterative detection process.

**Table 3.3: Pseudocode of Sorted ZF-QRD detection algorithm**

(1)	input $H, p = 1, 2, \dots, N_t$
(2)	$R = 0, Q = H$
(3)	for $i = 1, \dots, N_T$
(4)	$\text{norms}_i = \ q_i\ ^2$
(5)	end
(6)	for $i = 1, \dots, N_T$
(7)	$k_i = \arg \min_{j=i, \dots, N_T} (\text{norms}_j)$
(8)	Exchange $i$ -th $k$ -th columns in $R, Q, p$ and norms
(9)	$R_{i,i} = \sqrt{\text{norms}_i}$
(10)	$q_i = q_i / R_{i,i}$
(11)	for $k = i + 1, \mathbf{K}, N_t$
(12)	$R_{i,k} = q_i^H \cdot q_k$
(13)	$q_k = q_k - R_{i,k} \cdot q_i$
(14)	$\text{norms}_k = \text{norms}_k -  R_{i,k} ^2$
(15)	end
(16)	end
(17)	$y = Q^H r$
(18)	for $k = N_t, \mathbf{K}, 1$
(19)	$\hat{d}_k = \sum_{i=k+1}^{N_t} R_{k,i} \hat{x}_i$
(20)	$\hat{x}_k = Q \left( \frac{y_k - \hat{d}_k}{R_{k,k}} \right)$
(21)	end
(22)	Permutate $\hat{x}$ according to $p$

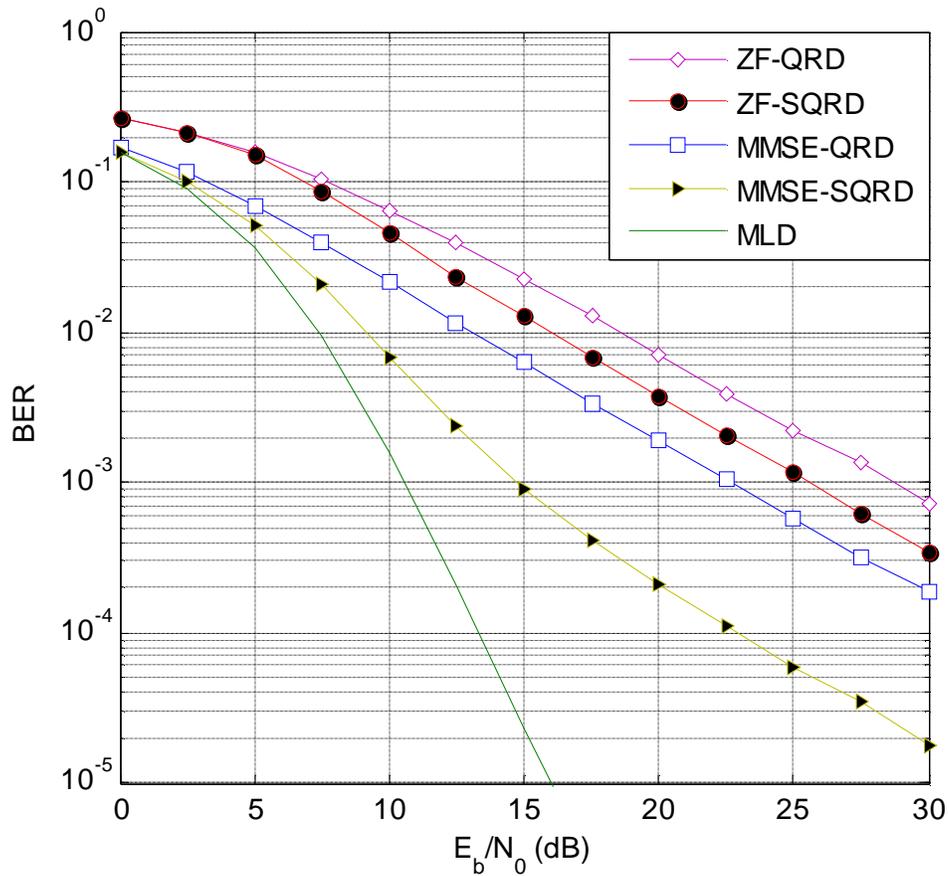
### 3.3.2 Minimum Mean Square Error QR Decomposition (MMSE-QRD)

In order to extend the QR-based detection with respect to MMSE criterion, the channel matrix  $H$  should be extended to

$$H^0 = \begin{bmatrix} H \\ S_n I \end{bmatrix} \quad (3.17)$$

and decomposed into  $Q$  and  $R$  matrices such that  $\tilde{H}^{\%} = QRP$ , with  $P$  as the column permutation matrix.

Table 3.3 gives the pseudocode of the ZF-SQRD detection algorithm. The sorted MMSE-QRD detection scheme (labelled MMSE-SQRD) is obtained by simply replacing  $H$  by  $\tilde{H}^{\%}$ . The unsorted MMSE-QRD detection scheme (labelled MMSE-QRD) is obtained by simply skipping the sorting steps in the QRD code (line (7) and line (8)).



**Figure 3.4: BER of QRD detection schemes**

Figure 3.4 shows the BER performance of the QRD-based detection schemes in addition to that of the optimum detector (MLD). It can be seen that the MMSE

algorithms perform better than ZF in all cases. At target BER of  $10^{-3}$ , MMSE-QRD leads both ZF-QRD and ZF-SQRD by about 5.5 dB and 2.5 dB respectively.

In MMSE cases, by sorting the columns of the QR decomposition the performance increases to 7.5 dB. Thus the best performance is achieved by the MMSE-SQRD detection scheme, where it lags the optimum performance by about 9 dB at target BER of  $10^{-4}$ .

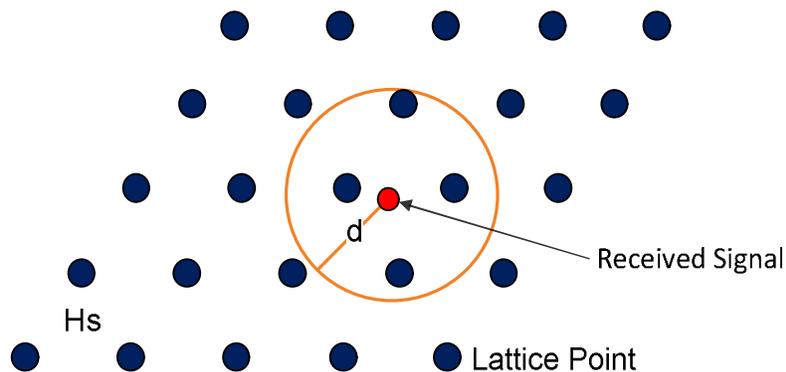
In general, both VBLAST and QRD detection algorithms without sorting have a diversity order of  $(N_r - N_t + 1)$ [9]. That is, the diversity orders of VBLAST and QRD without sorting equals one for equal number of transmit and receive antennas whatever is the number of receive antennas. This is because signals are detected independently, where the ZF or MMSE solution of each component of  $x$  is demodulated and considered as error-free in the following detection levels. Since one of the main reasons for the inaccuracy of the linear detection, VBLAST and QRD algorithms is the ill-conditionality of the channel matrix, we introduce in the following section the tree search detection.

### **3.4 Tree-Search Detection Techniques**

Several tree-search detection algorithms have been proposed in the literature that achieve quasi-ML performance while requiring lower computational complexity. In these techniques, the search problem of (2.2) is presented as a tree where nodes represent the symbols' candidates. In the following, we introduce two tree-search algorithms and discuss their advantages and drawbacks.

### 3.4.1 Sphere Decoding

Sphere Decoding (SD) approach was inspired from the mathematical problem of computing the shortest nonzero vector in a lattice [31]. SD algorithm was originally described in [32] and refined in [33] to substantially reduce the computational complexity of signal detection in MIMO communication systems. The principle of SD is to search for the closet constellation point to the received signal within a sphere with predetermined radius ' $d$ ' [34], where each transmit candidate is represented by a lattice point in a lattice field  $\{Hs\}$ . Figure 3.5 depicts a geometrical representation of the idea behind SD algorithm, the search can be restricted to be in a circle around the received signal just small enough to enclose at least one lattice point or ML solution [35], thus search time can be significantly reduced by eliminating the search of those lattice points lie outside the circle. According to the analysis available in [36], SD can transform the ML detection problem into a tree search and pruning process and achieve quasi-ML performance with polynomial average computational complexity for large range of signal to noise ratios.



**Figure 3.5: Geometrical representation of the idea behind SD algorithm**

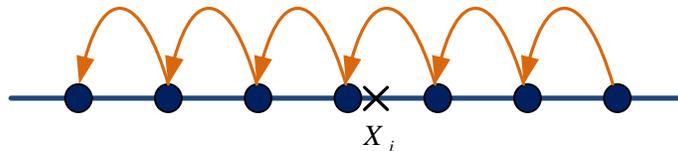
### 3.4.1.1 SD Search Strategies

The sphere decoding can be considered as a depth-first search approach with tree pruning process [37]. In SD algorithm, the most important issue is the strategy based on which signals “hypotheses” are tested per level. The SD algorithm for SM MIMO systems has two types of searching strategies, the Fincke-Pohst (FP) and the Schnorr-Euchner (SE) [38].

#### I. Fincke-Pohst Strategy

The Fincke-Pohst (FP) strategy [33] is considered in literature to be the original sphere decoding algorithm [39]. This strategy was first used in digital communications theory by Viterbo and Biglieri [31]. In [40], it was applied to find the closest point for a single antenna fading channels. This method considers all hypotheses in natural order, and the search is starting with the first one as shown in Figure 3.6. If a point is found, the radius is updated (reduced) and so forth.

An important and critical aspect of the FP strategy is that a search radius must be initialized appropriately. However, if the sphere radius  $d$  is too large, many lattice points will have to be computed and a large number of points may also be cancelled out. If it is too small, no points will be found and the algorithm must then be restarted with a larger searching radius. Both of these factors negatively influence the overall

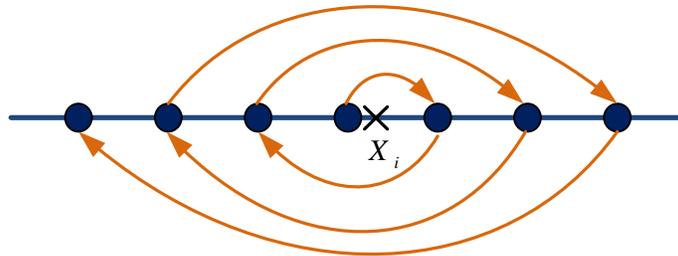


**Figure 3.6: Fincke-Pohst strategy**

computation time, and thus it is well-known that one of the main drawbacks of the FP strategy is the sensitivity of its performance to the choice of initial search radius  $d$ . A recommend choice is the distance to the Babai point [35], which is the first returned point in the search set. Then, it could be assured that at least one lattice point will be found inside the sphere.

## II. Schnorr-Euchner Strategy

Schnorr-Euchner (SE) strategy that was proposed in [41], added a small but significant refinement to the FP approach. The FP strategy searches the admissible nodes without any ordering, whereas in SE strategy, the admissible nodes of each level are spanned in a zigzag order starting with the closest middle point as depicted in Figure 3.7. SE strategy considers symbol close to the ZF solution and If a point is found, then the radius is updated (reduced) and so on. It was also concluded in [35] and [42] that the SE enumeration is more efficient than FP and has lower computational complexity by reordering the constellation searching at each level.



**Figure 3.7: Schnorr-Euchner strategy**

### 3.4.1.2 Sphere Decoding Algorithm

As stated in the introduction, the exhaustive search through the whole lattice  $\{H_s\}$  has an exponential computational complexity. This complexity is unrealizable

and thus defines a bottleneck in the practical implementation of the MIMO SM systems [43][44]. The SD algorithms can solve the ML detection problem in (3.18) by searching over a restricted subset  $\Omega$  that at least contain the ML solution.

$$\hat{x}_k = \arg \min_{x_k \in \Omega} \|r - Hx_k\|^2 \quad (3.18)$$

To describe the conventional sphere decoding algorithm, consider the QR-decomposition of the channel matrix, i.e.  $H = QR$  where  $R$  is upper triangular matrix and  $Q$  has orthogonal columns of unit norm. In the basis given by the columns of  $Q$  the system model in (2.1) can equivalently be written as

$$y = R \cdot x + v \quad , \quad (3.19)$$

where  $y = Q^H x$  and  $v = Q^H n$ . Further, the ML detection problem can equivalently be written as

$$\hat{x}_k = \arg \min_{x_k \in \Omega} \|y - Rx_k\|^2 \quad (3.20)$$

The main difference between (3.20) and (3.18) is that  $R$ , by construction, is upper triangular. The sphere decoder solves (3.20) by searching over all vectors,  $x_k \in \Omega$ , satisfying a spherical constraint on the form

$$\|y - Rx_k\|^2 \leq d^2 \quad (3.21)$$

In what follows,  $d$  will be referred to as the search radius for the obvious reason. It is straightforward to see that if  $d$  is sufficiently large, at least one vector,  $x_k \in \Omega$  satisfies (3.21) and the SD algorithm will obtain the ML estimate. Naturally, it is not practically feasible to verify (3.21) for every  $x_k \in \Omega$  as this would require an

exhaustive computational complexity equal to the original brute-force ML approach. Instead, SD algorithm finds all  $x_k \in \Omega$  satisfying (3.21) through a constrained tree search [45].

So far, note that

$$\|y - Rx\|^2 = \sum_{i=1}^m \left| x_i - \sum_{j=i}^m r_{ij} x_j \right|^2 \leq d^2$$

where  $r_{ij}$ ,  $y_i$  and  $x_i$  are the  $(i, j)^{th}$  entry of  $R$ ,  $i^{th}$  entry of  $y$  and the  $j^{th}$  entry of  $x$  respectively. Thus, a sufficient condition for (3.21) to be satisfied is given by (3.22),

$$\sum_{i=m-k+1}^m \left| x_i - \sum_{j=i}^m r_{ij} x_j \right|^2 \leq d^2 \quad (3.22)$$

for  $k = 1, \mathbf{K}, m$ . Due to the triangular structure of  $R$ , (3.22) is only a constraint on  $x_{m-k+1}, \mathbf{K}, x_m$ . In particular, if  $k = 1$ , (3.22) is equivalent to

$$\left| r_{mm} x_m - y_m \right|^2 \leq d^2 \Leftrightarrow \left| x_m - c_m \right| \leq r_{mm}^{-1} d \quad (3.23)$$

where  $c_m @ r_{mm}^{-1} d$  is the unconstrained least squares estimate of  $x_m$ . In other words the admissible, in the sense that (3.22) is not violated, values of  $x_m$  belong to a sphere of radius  $r_{mm}^{-1} d$ , centered at  $c_m$ .

Similarly, for  $k = 2$ , (3.22) is equivalent to (3.24)

$$\left| x_{m-k+1} - c_{m-k+1} \right| \leq r_{m-k+1, m-k+1}^{-1} \sqrt{d^2 - \sum_{i=m-k+2}^m \left| x_i - \sum_{j=i}^m r_{ij} x_j \right|^2} \quad (3.24)$$

$$c_{m-k+1} @ x_{m-k+1} - \sum_{j=m-k+2}^m r_{m-k+1,j} x_j \quad (3.25)$$

Where (3.25) is the unconstrained least squares estimate of  $x_{m-k+1}$  given  $x_{m-k+2}, \mathbf{K}, x_m$ . An implicit assumption is naturally that the term appearing in the square root on the right-hand side of (3.24) is positive. In the case that it is not positive, there are no  $x_{m-k+1}$  that satisfy (3.22).

The above bounds suggest an iterative (or recursive) approach for solving (3.20), and thus also the original ML detection problem in (2.2), by enumerating the admissible values of  $x_m$  which satisfied (3.23). Each admissible value of  $x_m$  yields a set of admissible values for  $x_{m-1}$  through (3.24). Similarly, given  $x_{m-1}$  and  $x_m$ , a new set of admissible values for  $x_{m-2}$  is obtained and so on down to the top of the tree.

The SD algorithm can be illustrated as a tree search procedure using (3.22) as pruning criteria to reduce the search [46],[47]. In the tree search analogy, a sequence of symbol decisions,  $\{x_{m-k+1}, \mathbf{K}, x_m\}$  corresponds to a node of the search tree at the  $k^{th}$  level, counting from the root of the tree which by default is at the  $0^{th}$  level. Any full sequence  $\{x_1, \mathbf{K}, x_m\}$  is referred to as a leaf node and corresponds in an obvious way with a symbol estimate,  $x$ , satisfying (3.21). The algorithm searches in a structured manner all nodes that satisfy (3.22) and when completing the entire set of symbol estimates,  $x$ , satisfying (3.21) will have been generated. The symbol estimate,  $x$ , yielding the smallest criterion value according to (3.20) is equal to the ML estimate. Note also that there is no possibility that the ML estimate does not belong to the set of leaf nodes visited by the algorithm (assuming there are some leaf nodes

visited). This follows since the bound in (3.21) assures that nodes not visited have a larger criterion than the optimal, ML estimate.

A general algorithmic description of the SD is given in [35] and this algorithm is reproduced in Appendix 01, Table 1. Also included in this table are modifications to restrict the search space to  $[0, \dots, Q_{max}]$ . The input to this algorithm is the vector to decode,  $r$ , and the generator matrix  $H$  of the lattice  $Hs$ .

Before illustrating the idea of SD in numerical example, it is worth to remind some important points regarding SD algorithm procedures. The SD algorithm starts the search process from the root of the tree, and then searches down along branches until the total weight of a node exceeds the square of the sphere radius  $d^2$ . At this point, the corresponding branch is pruned, and any path passing through that node is declared as improbable for a candidate solution. Then the algorithm backtracks and proceeds down a different branch. Whenever a valid lattice point at the bottom level of the tree is found within the sphere,  $d^2$  is set to the newly found point weight, thus reducing the search space for finding other candidate solutions. In the end, the path from the root to the leaf that is inside the sphere with the lowest weight is chosen to be the estimated solution  $\hat{x}$ . If no candidate solutions can be found, the tree will be searched again with a greater initial radius.

### **Example I**

Let the channel matrix  $H$ , given by

$$H = \begin{bmatrix} 0.2 & 0.83 & -1.27 \\ -0.72 & 0.91 & 0.02 \\ 0.76 & 0.4 & 0.68 \end{bmatrix}$$

suppose the elements of the transmitted vector,  $x$ , are withdrawn from  $\{1, -1, 3, -3\}$  constellation

$$x = \begin{bmatrix} -1 \\ 1 \\ 3 \end{bmatrix}$$

And, let the noise vector  $n$

$$n = \begin{bmatrix} 0.006 \\ 0.05 \\ 0.01 \end{bmatrix}$$

Then, the received signal  $r$  is given by

$$\begin{aligned} r &= Hx + n \\ &= \begin{bmatrix} -3.17 \\ 1.74 \\ 1.69 \end{bmatrix} \end{aligned}$$

The first step is the QR-decomposition of the channel matrix  $H$

$$H = QR = \begin{bmatrix} 0.19 & 0.67 & -0.72 \\ -0.68 & 0.62 & 0.4 \\ 0.71 & 0.91 & 0.57 \end{bmatrix} \begin{bmatrix} 1.066 & -0.17 & 0.23 \\ 0 & 1.28 & -0.56 \\ 0 & 0 & 1.31 \end{bmatrix}$$

Now, multiply received signal  $r$  by  $Q^H$ , thus the system can be re-written as

$$y = R \cdot x + v$$

where  $y = Q^H r$  and  $v = Q^H n$ . Then,

$$y = \begin{bmatrix} -0.57 \\ -0.37 \\ 3.94 \end{bmatrix}$$

Now the system becomes

$$\begin{bmatrix} -0.57 \\ -0.37 \\ 3.94 \end{bmatrix} = \begin{bmatrix} 1.066 & -0.17 & 0.23 \\ 0 & 1.28 & -0.56 \\ 0 & 0 & 1.31 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + v$$

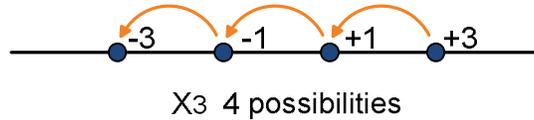
To this end, forget the noise vector, because it is included in the modified vector  $y$ .

In any tree search algorithm, including sphere decoding, the goal is to minimize

$$\arg \min_x \|y - Rx\|^2$$

In SD algorithm, the most important issue is the strategy based on which signals “hypotheses” are tested per level. For example; for  $x_3$  we have 4 possibilities, so by which  $x_3$  we should start? i.e. which search strategy, we should use? FP or SE. We usually use SE strategy because it leads to much lower complexity [42]. In SD algorithm, estimates of  $x$  is found sequentially and the search is continued until no more estimate  $\hat{x}$  has lower accumulative metric of the so far found estimate.

Now, start computing the estimate of  $x_3$  assuming the initial value for the sphere radius  $d^2 = \infty$ . The ZF solution of  $x_3$  is computed. This value is considered as the first hypothesis to test for  $x_3$ .



Due to the triangular form of the matrix  $R$ ,  $x_3$  could be calculate directly as follows:

$$\hat{x}_3 = \frac{y(3)}{R(3,3)} = \frac{3.94}{1.31} = \boxed{3.0076}$$

We choose the closet constellation point to  $\hat{x}_3 \rightarrow +3$ , then the estimate  $\hat{x}_3 = 3$ .

Now, we are going to calculate the metric  $E_3 = ?$

$$E_3 = |y(3) - R(3,3) \cdot \hat{x}_3|^2 = 1e^{-4}$$

is  $E_3 \leq d^2$ ? If yes, move to the next level.

$$\hat{x}_2 = \frac{y(2) - R(2,3)\hat{x}_3}{R(2,2)} = \frac{-0.37 - (-0.56)3}{1.28} = \boxed{1.0234}$$

Then, the estimate of  $\hat{x}_2 = 1$ , and the metric

$$E_2 = |y(2) - R(2,2) \cdot \hat{x}_2 - R(2,3) \cdot \hat{x}_3|^2 = 0.0014$$

$$\Delta_2 = E_2 + E_3 = 0.0014 + 1e^{-4} = 0.0015$$

is  $\Delta_2 \leq d^2$ ? If yes, go to the next level.

To this end,  $\hat{x}_2 = 1$  and  $\hat{x}_3 = 3$ , then  $\hat{x}_1$  can be found as follows:

$$\hat{x}_1 = \frac{y(1) - R(1,2)\hat{x}_2 - R(1,3)\hat{x}_3}{R(1,1)} = \boxed{-1.0225}$$

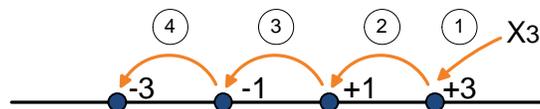
The closet constellation point to  $\mathcal{X}_3$  is "-1", then the estimate of  $\hat{x}_1 = -1$ , and the accumulative metric

$$\begin{aligned} \Delta_1 &= E_1 + \Delta_2 = E_1 + E_2 + E_3 \\ &= 0.0015 + |y(1) - R(1,1)\hat{x}_1 - R(1,2)\hat{x}_2 - R(1,3)\hat{x}_3|^2 \\ &= 0.0015 + 8.82e^{-4} \\ &= 0.0024 \end{aligned}$$

Is  $\Delta_1 \leq d^2$ ? yes

$\Rightarrow d^2 = \Delta_1$ , and restart from the beginning. According to the sequence of testing the hypothesis shown in Figure 3.8, the next hypothesis for  $\hat{x}_3$  is +1

$$\Rightarrow \boxed{\hat{x}_3 = 1}$$



**Figure 3.8: Sequence of testing the hypothesis**

Find  $E_1 = ?$

$$E_1 = |y(3) - R(3,3) \cdot \hat{x}_3|^2 = |3.94 - 1.31(1)|^2 = \boxed{6.92}$$

is  $E_1 \leq d^2$  {now  $d^2 = 0.0024$  }

No  $\Rightarrow$  stop this hypotheses and move to next hypotheses according to Figure 3.8.

$$\Rightarrow \boxed{\hat{x}_3 = -1}$$

$$E_1 = |3.94 - 1.31(-1)|^2 = \boxed{27.56}$$

is  $E_1 \mathbf{p} d^2$

No  $\hat{\mathbf{e}}$  stop and move to next hypotheses of  $\hat{x}_3$ .

$$\Rightarrow \boxed{\hat{x}_3 = -3}$$

$$E_1 = |3.94 - 1.31(-3)|^2 = \boxed{61.93}$$

is  $E_1 \mathbf{p} d^2$

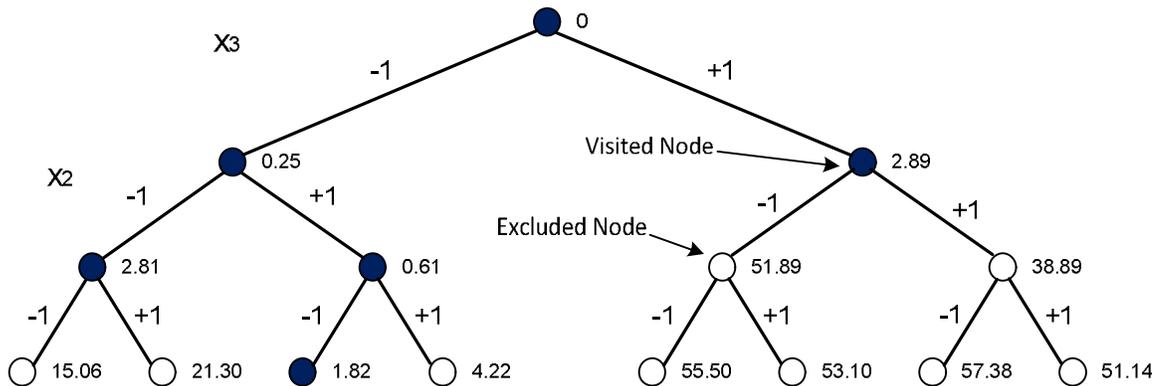
No  $\hat{\mathbf{e}}$  stop and the already found estimate is the best. Now detection algorithm is terminated, and the estimate  $\hat{x} = [-1 \ 1 \ 3]^T$

### Example II

Another example of the search tree generated by the SD algorithm is given in Figure 3.9 for the case when  $m = 3$ ,  $\Omega = \{+1, -1\}$ , and  $d^2 = 3$ , where  $y$  and  $R$  in (3.20) are given by

$$y = \begin{bmatrix} 0 \\ 3.8 \\ -1.1 \end{bmatrix}, \text{ and } R = \begin{bmatrix} 0.4 & -1.2 & -2.7 \\ 0 & 0.5 & -2.7 \\ 0 & 0 & 0.6 \end{bmatrix}$$

Each candidate symbol,  $x \in \Omega$ , is indicated by a leaf node in the tree. The metric of each node, given by the left hand side of (3.22), is indicated by the number to the right of each node. Each node with a metric less than  $d^2$  is included in the search and indicated in black. On the other hand the white nodes are not visited by the SD algorithm. The ML estimate,  $x_{ML} = [-1 \ +1 \ -1]$ , has an objective value of 1.82 in (3.20) which is also the smallest node value.



**Figure 3.9: Example illustrating SD search tree. Nodes visited by the algorithm are shown in black**

By moving down the tree until the sphere constraint in (3.22) is violated or until the bottom of the tree is reached, then moving back up to adjust previous decisions and proceed down other branches, the SD algorithm will eventually have visited all nodes satisfying (3.22). As the number of nodes visited in Figure 3.9 is less than the total number of nodes in the full tree, it follows that the SD algorithm is less complex than the brute force, ML, search over all nodes.

As indicated by Example II the complexity of the algorithm is proportional to the number of nodes that are visited by the tree search. The total number of nodes visited is however typically much larger than the number of leaf nodes visited meaning that it is the recurring verification of (3.22) which accounts for the larger part of the complexity of the algorithm, not the final number of estimates in the search sphere given by (3.21). However, the total number of nodes visited is usually much smaller than the set of all symbol vectors,  $\Omega^m$ , which implies that the SD algorithm is of substantially lower complexity than the brute force search.

The complexity analysis in Chapter 4 will indicate that by requiring that the sphere decoder visits at least one leaf node (which is essential if the sphere decoder should find an estimate,  $\hat{x}$ ) it will also follow that the search tree is left unpruned up to some fraction  $\epsilon \in (0,1]$  of the total search depth. This will however unfortunately imply that the number of nodes visited by the algorithm also grows exponentially fast, even though with smaller exponent than the full search over  $\Omega^m$ .

### 3.4.2 QRD-M Detection

The second scheme in the tree search category is the QRD-M, which was proposed to achieve quasi-ML performance while requiring fixed computational effort. QRD-M algorithm was originally discussed in [48] and was first used in signal detection in MIMO system in [49]. QRD-M algorithm can reduce the tree search complexity by selecting only  $M$  candidates at each layer instead of testing all the hypotheses of the transmitted symbol [50]. These  $M$  candidates are the smallest accumulated metric values.

#### QRD-M Algorithm

QRD-M Algorithm can be considered as a breadth-first search that has only one searching strategy. The concept of QRD-M is based on the classical  $M$ -Algorithm [51] that retains only a fixed number of symbol candidates,  $M$ , at each detection layer [49]. Basically, the idea of QRD-M Algorithm is similar to SQRD approaches for MIMO detection (section 3.3). However, instead of selecting only the closet constellation point in each layer, a total of  $M$  metrics is considered in evaluation.

The algorithm starts by applying the QR decomposition to (2.1) with  $H = QR$ , where  $Q$  is an orthogonal matrix satisfying  $QQ^H = Q^H Q = I$  and  $R$  is an upper triangular matrix with  $R_{i,j}$ ,  $j \geq i$  denoting its non-zero elements and triangular matrix  $R_{i,j}$  being positive real number, and assuming  $N_r \geq N_t$ , we have

$$r = Hx + n \quad (3.26)$$

hence

$$y = R \cdot x + v \quad (3.27)$$

where  $v = Q^H n$ .

Denoting the ZF solution as

$$\hat{x} = H^\dagger r = R^\dagger y \quad (3.28)$$

The ML detection problem can be reformulated as:

$$\begin{aligned} \hat{x}_{ML} &= \arg \min_{x \in \Omega^{N_t}} \|y - Rx\|^2 \\ &= \arg \min_{x \in \Omega^{N_t}} \left\{ \sum_{j=1}^{N_t} \left| y_j - \sum_{i=j}^{N_t} R_{j,i} x_i \right|^2 + \sum_{k=N_t+1}^{N_r} |y_k|^2 \right\} \\ &= \arg \min_{x \in \Omega^{N_t}} \left\{ \|R(\hat{x} - x)\|^2 + \|(I - RR^\dagger)\|^2 \right\} \\ &= \arg \min_{x \in \Omega^{N_t}} \left\{ \sum_{j=1}^{N_t} \left| R_{j,j}(\hat{x}_j - x_j) + \sum_{i=j+1}^{N_t} R_{j,i}(\hat{x}_j - x_j) \right|^2 + \sum_{k=N_t+1}^{N_r} |y_k|^2 \right\} \quad (3.29) \end{aligned}$$

It can be seen that the second term  $\sum_{k=N_t+1}^{N_r} |y_k|^2$  is independent of the transmitted vector

$x$ , therefore it is ignored hereafter. To minimize the metric in (3.29), QRD-M

algorithm keeps only  $M$  candidates at each detection level with the smallest accumulated metric. For example, at the first detection layer, the root node is extended to all the possible  $\Omega$  candidates of  $x_{N_i}$ , the accumulative metrics of the resulting branches are calculated and the best  $M$  candidates are retained for the next detection layer. At the second detection layer, the retained  $M$  candidates at the previous layer are extended to all possible candidates. The resulting  $(M\Omega)$  branches are sorted based on their accumulative metrics where the  $M$  branches with the smallest accumulative metrics are retained for the next detection layer.

This strategy is repeated down to the last detection layer, i.e.,  $i = 1$ . At the last step the  $x$  with the smallest overall metric is chosen as the ML decision. Note that QRD- $M$  algorithm is sub-optimal in nature and only when  $M = \Omega^{N_i}$ , it becomes exhaustive ML search. When  $M = 1$ , it is essentially zero-forcing nulling and interference cancellation. Therefore ordering is important in QRD-M, and VBLAST type of ordering gives the best performance among the various ordering schemes, as discussed in [52].

For the ease of understanding, a flowchart for the QRD-M algorithm is drawn, Figure 3.10. And the QRD-M algorithm can be summarized in the following six main steps:

Step 1: Perform QRD on  $H$

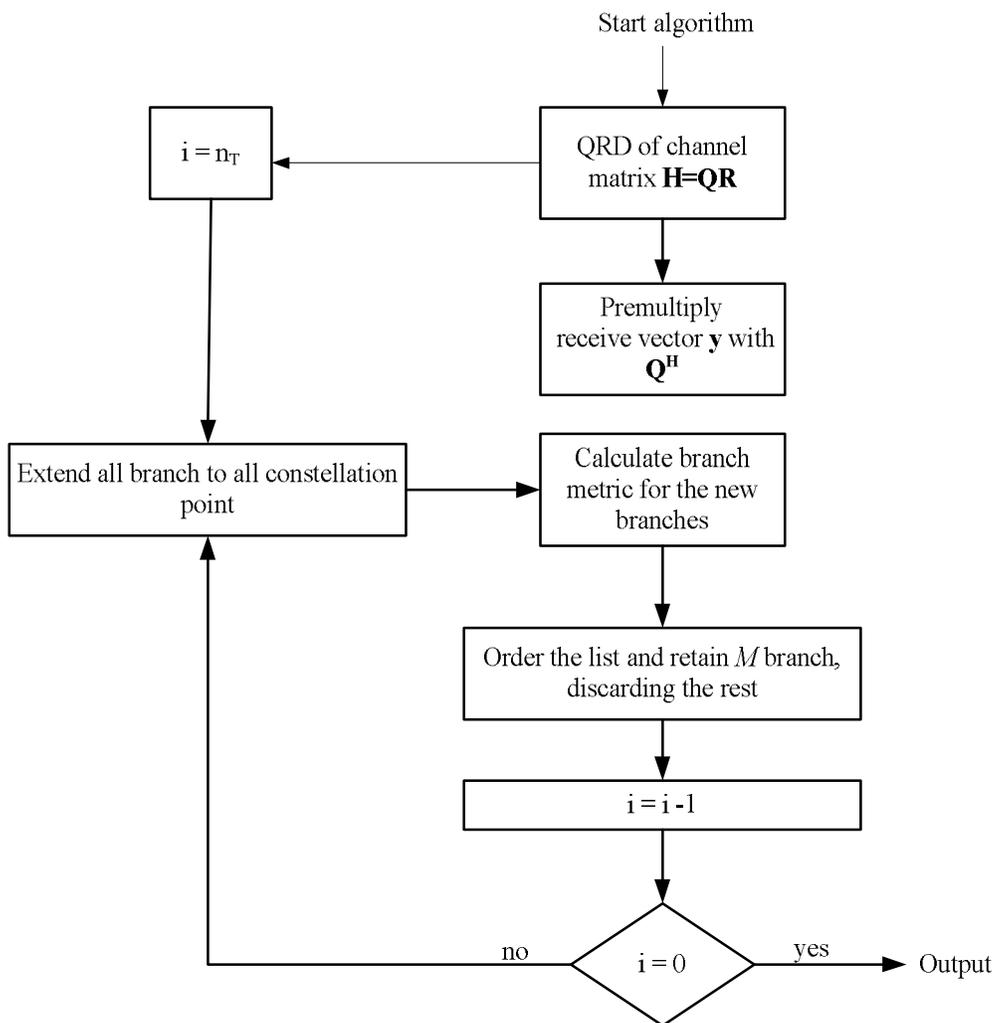
Step 2: Pre-multiply  $y$  with  $Q^H$

Step 3: Extend all branches to  $M\Omega$  nodes

Step 4: Calculate the branch metrics using (3.29)

Step 5: Order the branches according to their metrics, retaining only  $M$  branches and discarding the rest

Step 6: Move to next layer and go to step 3



**Figure 3.10: Flowchart of QRD-M detection algorithm**

### Example III

Using the same values and parameters used in the example given for SD algorithm, so

$$\begin{bmatrix} -0.57 \\ -0.37 \\ 3.94 \end{bmatrix} = \begin{bmatrix} 1.066 & -0.17 & 0.23 \\ 0 & 1.28 & -0.56 \\ 0 & 0 & 1.31 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \text{noise}$$

We could start from  $x_3$  and go up.

Let  $M = 4$

Then, we start by  $x_3$  because it is interference-free as all the possibilities of  $x_3$  are four  $\{-3, -1, 1, 3\}$ .

We calculate the metric at this level.

$$x_3 = -3$$

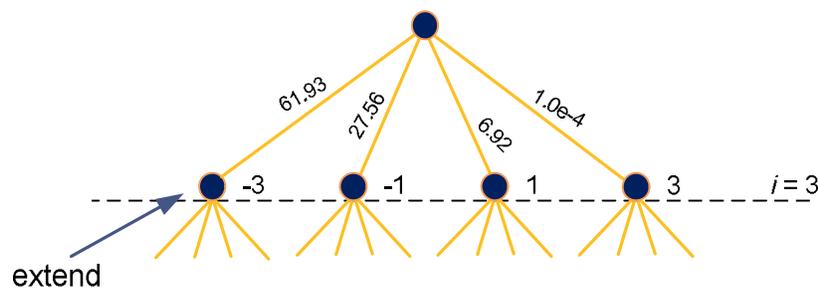
$$\boxed{x_3 = -3} \rightarrow E_{3,1} = |3.94 - 1.31(-3)|^2 = 61.93$$

$$\boxed{x_3 = -1} \rightarrow E_{3,2} = |3.94 - 1.31(-1)|^2 = 27.56$$

$$\boxed{x_3 = +1} \rightarrow E_{3,3} = |3.94 - 1.31(1)|^2 = 6.92$$

$$\boxed{x_3 = +3} \rightarrow E_{3,4} = |3.94 - 1.31(3)|^2 = 1.0e^{-4} \quad \{\text{so small}\}$$

Let us see the part of the tree, we have



Now each point (node) is extended to all possible branches. We will have 16 new nodes, and only 4 candidates should be selected among them. Let us make the calculations

Let  $x_3 = -3$ , then we should calculate the solution

$$\begin{bmatrix} x_2 \\ -3 \end{bmatrix},$$

where  $x \in \{-3, -1, 1, 3\}$

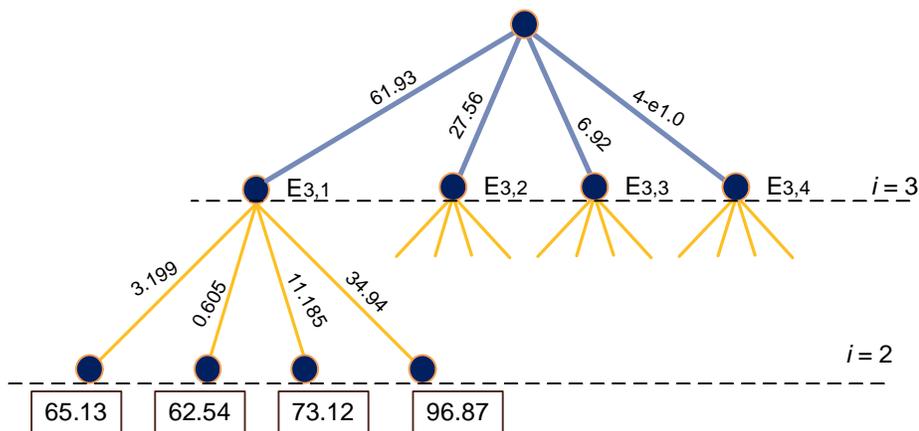
$$\begin{aligned} \boxed{x_2 = -3} &\rightarrow E_{2,1} = |y(2) - R(2,2) \times x_2 - R(2,3) \times x_3|^2 \\ &= |-0.37 - (1.28) \times (-3) - (-0.56) \times (-3)|^2 = 3.1998 \end{aligned}$$

$$\begin{aligned} \boxed{x_2 = -1} &\rightarrow E_{2,2} = |y(2) - R(2,2) \times x_2 - R(2,3) \times x_3|^2 \\ &= |-0.37 - (1.28) \times (-1) - (-0.56) \times (-3)|^2 = 0.605 \end{aligned}$$

$$\boxed{x_2 = 1} \rightarrow E_{2,3} = |-0.37 - (1.28) \times (1) - (-0.56) \times (-3)|^2 = 11.185$$

$$\boxed{x_2 = 3} \rightarrow E_{2,4} = |-0.37 - (1.28) \times (3) - (-0.56) \times (-3)|^2 = 34.94$$

Now we got the following tree:



The accumulative metric of these branches are shown in the above tree. For example

$$E_{3,1} + E_{2,1} = 61.93 + 3.1998 = 65.1298$$

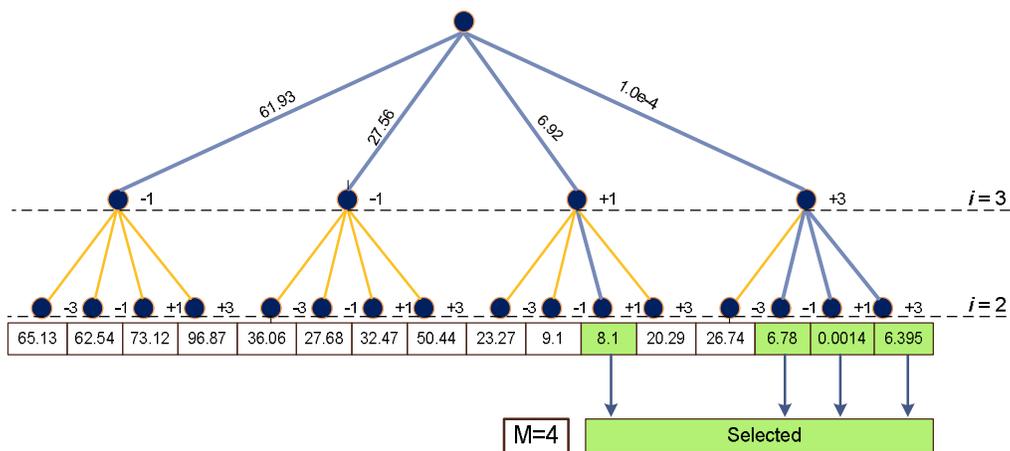
Now calculate for the next nodes at  $i = 2$  level

$$\boxed{x_3 = -1}, \left. \begin{array}{l} x_2 = -3 \quad E_{2,5} = 8.5 \\ x_2 = -1 \quad E_{2,6} = 0.1222 \\ x_2 = +1 \quad E_{2,7} = 4.915 \\ x_2 = +3 \quad E_{2,8} = 22.883 \end{array} \right\} + 27.56$$

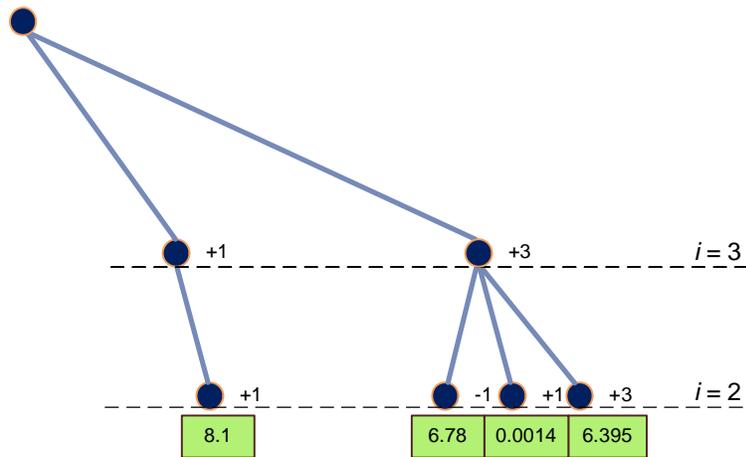
$$\boxed{x_3 = +1}, \left. \begin{array}{l} x_2 = -3 \quad E_{2,9} = 16.35 \\ x_2 = -1 \quad E_{2,10} = 2.18 \\ x_2 = +1 \quad E_{2,11} = 1.187 \\ x_2 = +3 \quad E_{2,12} = 13.368 \end{array} \right\} + 6.92$$

$$\boxed{x_3 = +3}, \left. \begin{array}{l} x_2 = -3 \quad E_{2,13} = 26.74 \\ x_2 = -1 \quad E_{2,14} = 6.780 \\ x_2 = +1 \quad E_{2,15} = 0.0014 \\ x_2 = +3 \quad E_{2,16} = 6.3948 \end{array} \right\} + 1e^{-4}$$

After extending all possible hypotheses, the new tree becomes as shown in below.



Thus our tree now is:



and the estimates of both  $\begin{bmatrix} x_2 \\ x_3 \end{bmatrix}$  are the following:

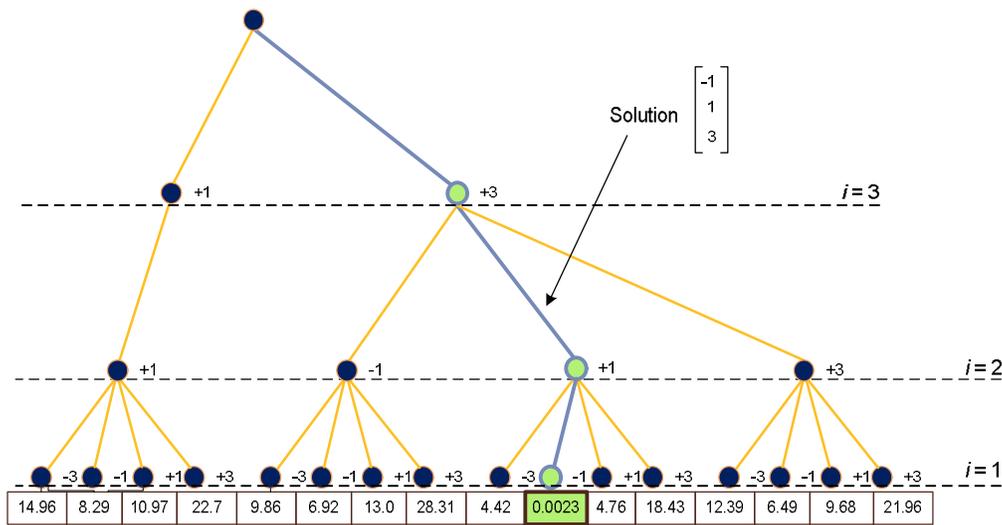
$$\begin{bmatrix} 1 \\ 1 \end{bmatrix}, \begin{bmatrix} -1 \\ 3 \end{bmatrix}, \begin{bmatrix} 1 \\ 3 \end{bmatrix}, \text{ and } \begin{bmatrix} 3 \\ 3 \end{bmatrix}$$

Now, we could start a new detection layer by extending the four selected candidates into all possible hypotheses and calculating their metrics.

$E_{1,1} = 6.59$	$E_{1,5} = 3.0776$	$E_{1,9} = 4.42$	$E_{1,13} = 6.0$
$E_{1,2} = 0.1905$	$E_{1,6} = 0.14$	$E_{1,10} = 8.82e^{-4}$	$E_{1,14} = 0.1$
$E_{1,3} = 2.8734$	$E_{1,7} = 6.30$	$E_{1,11} = 4.6712$	$E_{1,15} = 3.29$
<del><math>E_{1,4} = 14.64</math></del>	<del><math>E_{1,8} = 21.53</math></del>	<del><math>E_{1,12} = 18.429</math></del>	<del><math>E_{1,16} = 15.57</math></del>
<del>+8.1</del>	<del>+6.78</del>	<del>+0.0014</del>	<del>+6.395</del>
[ Metrics of thier Father node ]			

Figure 3.11 shows the final tree, and the lowest values corresponds the solution. The detected values are the same as the transmitted, i.e the solution is

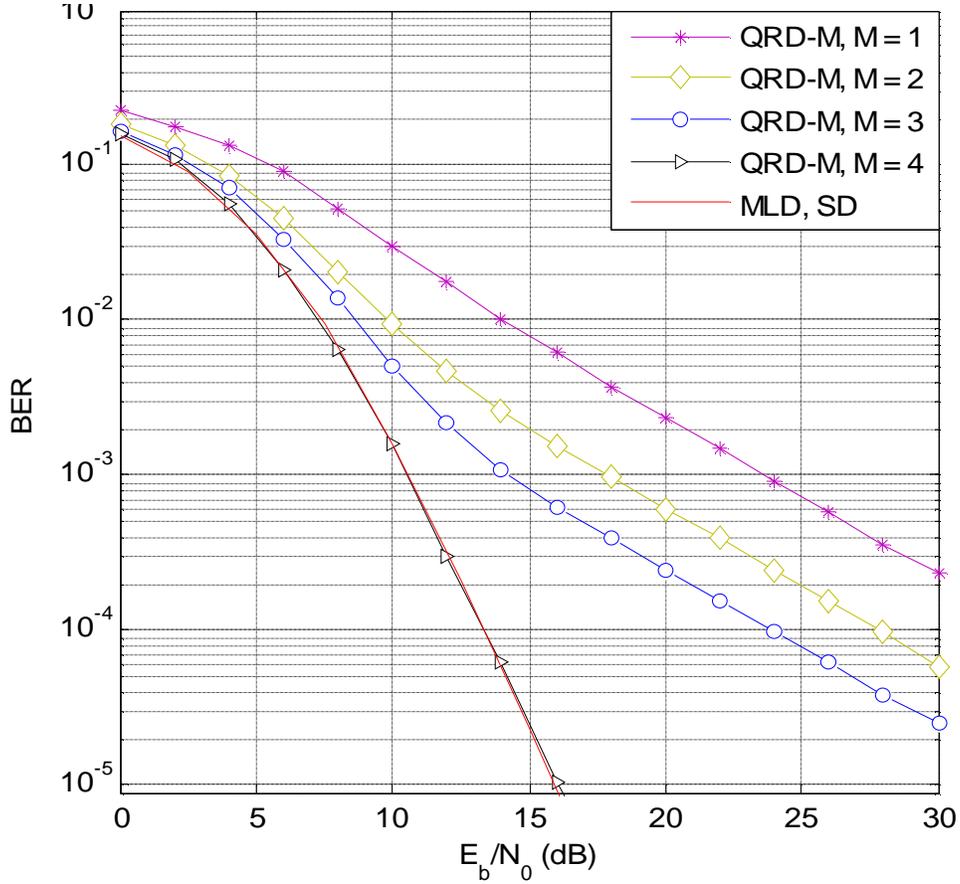
$$\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} -1 \\ 1 \\ 3 \end{bmatrix}$$



**Figure 3.11: the final tree showing the detection levels and the estimate  $\mathbf{x}$**

### 3.4.3 SD and QRD-M Performance Comparison

Figure 3.12 shows the BER performance of SD and QRD-M algorithms in 4 x 4 MIMO SM system. SD algorithm coincides with the optimum performance and the QRD-M algorithm achieves the ML performance for  $M = |\Omega|$  which equals 4 in the case of 4-QAM. Although SD achieves a quasi-ML performance, it has the following drawbacks: (i) the complexity of SD is random and depends on the conditionality of the channel matrix and the noise variance.



**Figure 3.12: BER of SD and QRD-M performance for several values of  $M$**

The worst-case complexity of SD is therefore exponential which is infeasible in computational power limited communication systems [35]. And (ii) SD has a sequential nature because it requires the update of the search radius at every time a new lattice point with smaller accumulative metric is found. This limits the possibility of parallel processing and hence reduces the detection throughput, i.e., increases the detection latency. In [53] Barbero et al. have proposed a fixed complexity sphere decoder to overcome the aforementioned drawbacks of the SD.

The QRD-M algorithm has also two drawbacks: (i) it employs a systematic tree-search without considering the noise power or the channel conditionality and (ii) for high  $N_r$  and high order modulations; detection throughput is reduced due to the

increase in the size of the search tree. Several algorithms were proposed in the literature to overcome these drawbacks by introducing a method that adaptively selects the number of retained candidates per detection layer [54].

As mentioned above, the tree-search based detection techniques provide quasi-ML performance, and the QRD-M algorithm in particular is the most amenable to hardware implementation. It should be noted that the detection complexity of SD and QRD-M was significantly higher than that of linear and SIC detection algorithms.

### **3.5 Summary**

In this chapter a variety of the MIMO SM detection schemes have been described discussed and compared in terms of performance and computational complexity. Different performance simulations have been generated for each detection categories to investigate and evaluate their BER. It has been shown that the linear detection techniques have poor performance due to the huge amplification in noise power in ZF-case. The ordering strategy involved VBLAST has important benefits but the performance improvement is limited due to error propagation. This error propagation has been alleviated by QRD algorithms. The tree-search based detection techniques; i.e., SD and QRD-M with the two promising approaches. SD has achieved MLD performance. In case of QRD-M, while the number of survival candidates increases the performance converges to that of MLD.

## CHAPTER 4

### COMPLEXITY ANALYSIS OF MIMO SM DETECTION TECHNIQUES

---

#### 4.1 Introduction

MIMO techniques will only become part of future generation wireless systems, if they are feasible in real world systems. A complexity analysis and comparison will be carried out for the most promising MIMO algorithms. This allows to estimate the potential cost of such systems and to identify possible bottlenecks for the hardware implementation. Even though there is no real consensus in the digital communications community on how exactly to interpret the concept of complexity, it is generally defined as the number of floating point operations (additions, multiplications etc.) which are required to compute the estimate of the transmitted vector  $x$  or the running time of the algorithm when implemented on some specific platform. This running time, may be defined in literature as time latency or CPU time in some references. In this thesis, the complexity of the detectors will only be computed in terms of flops, which is another measure proportional to the running time. There is also typically a tradeoff between the complexity of a detector and its performance in terms of error probability. The optimal, ML, detector which provides the minimum probability of error is often prohibitively complex while the computationally simplest detectors will have a poor performance in terms of error probability. An investigation of the complexity performance tradeoff of many detectors proposed in the literature is given in [55] in the context of CDMA. A similar

comparison of the sphere decoding algorithm and some extensions of the lattice basis reduction aided detectors in the context of MIMO can be found in [56].

Simple, closed form, expressions for the complexity are rare and it is often convenient to instead characterize the rate at which the complexity grows with some variable  $w$ . If the complexity of an algorithm is in  $O(p(w))$  for some polynomial  $p(w)$  the algorithm is said to be of polynomial complexity.

The implementation of MIMO-systems requires new hardware architectures since highly parallel algorithms need to be executed. To identify suitable hardware structures the algorithms have to be analyzed and cut down to their basic components. The goal is to provide guidelines for fast prototyping. This detailed analysis of the MIMO-signal processing algorithms includes recommendations of implementation issues and estimation of the complexity.

In this chapter, the complexity is analyzed and compared for the MIMO SM detection algorithms introduced in Chapter 3: linear algorithms( ZF, MMSE), SIC algorithms (VBLAST), tree-search algorithms (SD,QRD-M), and MLD.

## 4.2 Complexity of Arithmetic Operations

Before determining the complexity of the MIMO SM algorithms, a number of general rules will be introduced, namely, the complexity of a matrix multiplication, the conversion from complex complexity figures to real complexity figures, the complexity of a slicer, and the complexity of finding a minimum value from a set of values.

The complexity of a matrix product is determined as follows. Suppose two matrices  $A$  and  $B$  (real or complex) with dimensions  $C \times E$  and  $D \times E$  are multiplied, then the  $(i, j)^{th}$  element of the resulting matrix is given by

$$a^i b_j = \sum_{k=1}^D a_{ik} b_{kj}$$

where  $a^i$  represents the  $i^{th}$  row of matrix  $A$ ,  $b_j$  denotes the  $j^{th}$  column of  $B$  and  $a_{ik}$  and  $b_{kj}$  stand for the  $k^{th}$  element of this row and column, respectively. Thus, in order to obtain one element of the resulting matrix,  $D-1$  additions and  $D$  multiplications need to be performed. The resulting matrix is  $C \times E$  dimensional and, therefore, a total of  $C(D-1)E$  additions and  $CDE$  multiplications are needed to multiply the two  $A$  and  $B$ .

To write complex additions and complex multiplications in terms of real additions and real multiplications, it is easily verified that one complex addition consists of two real additions; the real and the imaginary part of the two complex numbers are added. Furthermore, a complex multiplication can be rewritten in the following two ways:

$$(a + jb)(c + jd) = (ac - bd) + j(bc + ad) \quad (4.1)$$

$$(a + jb)(c + jd) = (ac - bd) + j((a+b)(c+d) - ac - bd) \quad (4.2)$$

The first option consists of 4 real multiplications,  $ac$ ,  $bd$ ,  $bc$  and  $ad$ , and 2 real additions,  $ac - bd$  and  $bc + ad$ . A subtraction is counted as an addition and the addition before the  $j$  does not count because the real and imaginary parts are stored separately. The second option has only three real multiplications

$(ac, bd, (a+b)(c+d))$ , plus five real additions. Compared with the first case, the total operations count is higher by two, but in a number of hardware implementations, a multiplication is a more complex operation. In the remainder of this section, however, the first option will be used.

The complexity of a slicer is minimal in terms of additions and/or multiplications. For an M-PSK constellation scheme, the phase range  $[-p, p]$  is divided in  $M$  equal parts. In such a regular structure, a recursive search is done in which half of the (remaining) range the phase of the estimated symbol best fits. This results in a complexity equivalent to  $\log_2(M)$  comparisons. For an M-QAM constellation diagram, the real and imaginary parts are split. Each of these parts is regularly divided in  $\sqrt{M}$  slicing ranges. Also in this case, a recursive search is achieved in which half of the (remaining) range the real or imaginary part of the estimated symbol best fits, and the complexity is equal to  $\log_2(\sqrt{M})$  comparisons for the real and for the imaginary part, or  $2\log_2(\sqrt{M})$  comparisons in total. It is reasonable to assume that a comparison is as complex as a real addition and, therefore, the slicing of the  $Nt$ -dimensional vector  $x_{est}$  requires at most  $N_t \log_2(M)$  R-adds.

In order to find the minimum of  $N$  numbers in hardware, the easiest thing to do is start with the first two elements, subtract the second number from the first, and compare the result with zero. If the result is larger than zero, the second number is the smallest; otherwise the first number is the smallest, etc. Obviously, finding the minimum between two real numbers has the complexity of one real addition. As a

result, determining the minimum of  $N$  values has a complexity of  $N - 1$  real additions.

The computational complexity of several arithmetic operations can be summarized in Table 4.1.  $A$  refers to real addition,  $M$  refers to real multiplication,  $A_c$  refers to complex addition,  $M_c$  refers to complex multiplication. In the following, the complexity of the algorithms is given in terms of complex floating point operations (flops). A complex multiplication/ division requires 3 flops, and complex addition requires 1 flop.

Based on these general assumptions given in Table 4.1, the complexities of linear, nonlinear and tree search MIMO SM detection techniques are determined and compared, respectively, in section 4.3, section 4.4 and section 4.5.

**Table 4.1: Computational complexity of arithmetic operations**

Operation	Inputs	Output	Complexity	Flops
<b>Complex multiplication</b>	Two complex	Complex	$4M+2A$	3.0
<b>Complex by real</b>	Complex and real	Complex	$2M$	1.0
<b>Square root</b>	Real	Real	$M$	0.5
<b>Complex power</b>	Complex	Real	$2M+A$	1.5
<b>Real division</b>	Two real	Real	$M$	0.5
<b>Complex division</b>	Two complex	Complex	$8M+3A$	5.5
<b>Complex division</b>	Complex and real	Complex	$2M$	1.0

## 4.3 Complexity analysis of linear detections

### 4.3.1 Complexity of Zero-Forcing

As described in Subsection 3.1.1, the Zero Forcing technique is based on calculation of the pseudo-inverse of the channel transfer matrix  $H$ . Because it is assumed that the MIMO system is operating in a quasi-static environment, i.e.,  $H$  is constant during transmission of symbols, the pseudo-inverse of  $H$  needs to be calculated only once per transmitted MIMO vector. For determining the complexity of the calculation of the pseudo-inverse, the following equation is used

$$H^\dagger = (H^H H)^{-1} H^H \quad (4.3)$$

The dimensions of  $H^\dagger$ ,  $H$  and  $H^H$  are  $N_t \times N_r$ ,  $N_r \times N_t$  and  $N_t \times N_r$  respectively. To find the pseudo-inverse of  $H$ , first, determine the complexity of the matrix product  $H^H H$ . To determine this complexity, the general rule introduced in section 4.1 will be used. These rules state that the complexity of the product of two matrices  $A$  and  $B$  (real or complex) with dimensions  $C \times E$  and  $D \times E$  equals  $C(D-1)E$  additions and  $CDE$  multiplications (real or complex). Hence, the complexity of the matrix product  $H^H H$  yields  $N_t^2(N_r-1) A_c$  and  $N_t^2 N_r M_c$ . The result is a square matrix with dimension  $N_t \times N_t$ . For this square matrix  $H^H H$ , the inverse needs to be determined. It was shown in [57], that the direct inversion of a given square matrix  $A$  (with dimension  $N \times N$ ) has a complexity in the order of  $N^3$  additions and  $N^3$  multiplications in total. So, inverting  $H^H H$  has a complexity of  $N_t^3 A_c$  and  $N_t^3 M_c$ .

Finally, the inverse of  $H^H H$  (which is  $N_t \times N_t$  dimensional) is multiplied by  $H^H$ .

The complexity of this last multiplication is equal to  $N_t (N_t - 1) N_r$  A<sub>c</sub> and  $N_t^2 N_r$  M<sub>c</sub>. (see section 4.2). This leads to a total complexity of  $N_t^3 + N_t^2 (N_r - 1) + N_t (N_t - 1) N_r$  A<sub>c</sub> and  $N_t^3 + 2N_t^2 N_r$  M<sub>c</sub> based on general assumptions introduced in section 4.2, the complexity in terms of real operations (flops) equals

$$C_{ZF\text{-pre}}(\text{flops}) = 7N_t^3 + 7N_t^2 N_r - N_t \quad (4.4)$$

The payload processing for ZF consists of a matrix-vector multiplication per transmitted vector and a slicing step to translate the estimated elements of  $x$  to the possible transmitted symbols. Recalling from Subsection 2.1, the matrix-vector multiplication is given by

$$\hat{x} = H^\dagger r \quad (4.5)$$

The complexity of this product is equal to  $N_t (N_r - 1)$  complex additions and  $N_t N_r$  complex multiplications.

As explained in section 4.2, the complexity of slicing  $N_t$   $M$ -ary constellation points equals  $N_r \log_2(M)$  R-adds.

Summarizing, the complexity of the ZF algorithm per transmitted vector  $x$  equals

$$C_{ZF}(\text{flops}) = 7N_t^3 + 7N_t^2 N_r - 2N_t + 4N_t N_r + \frac{1}{2} N_r \log_2(M) \quad (4.6)$$

### 4.3.2 Complexity of MMSE

The complexity of the MMSE algorithm is almost equal to the complexity of the ZF method described in the previous section.

In the preamble-processing phase, the following MIMO processing matrix needs to be determined:

$$G = \left( \mathbf{S}I_{N_t} + H^H H \right)^{-1} H^H \quad (4.7)$$

The calculation of this matrix has almost the same complexity as the determination of the pseudo-inverse in case of the ZF algorithm. Since  $\mathbf{S}$  is real, the only additional complexity consists of the  $N_t$  real additions of  $\mathbf{S}$  (i.e., the addition of  $\mathbf{S}$  to the real part of the diagonal elements of  $H^H H$ ). This leads to a total complexity in the preamble-processing phase of

$$C_{\text{MMSE-pre}}(\text{flops}) = 7N_t^3 + 7N_t^2N_r \quad (4.8)$$

The complexity of MMSE during the payload processing is equal to that of ZF and consists of a matrix-vector product with the same dimensions and slicing. Recalling from the previous section, the payload complexity for every transmitted vector  $x$  equals

$$C_{\text{MMSE}}(\text{flops}) = 7N_t^3 + 7N_t^2N_r - N_t + 4N_tN_r + \frac{1}{2}N_r \log_2(M) \quad (4.9)$$

## 4.4 Complexity analysis of SIC (VBLAST) detection

### 4.4.1 Complexity of ZF-VBLAST

The processing of the ZF-VBLAST algorithm can be divided into two parts: the processing during the preamble and processing of the payload. Based on the assumption that the MIMO channel is static during a vector transmission, the ordering and the weight vectors can be determined during the preamble processing. During the payload processing the actual detection and SIC is performed.

(1)	input $H, r, U$
(2)	$H^{(1)} = H$
(3)	$r^{(1)} = r$
(4)	for $i = 1, \dots, N_T$
(5)	$G_i = H_i^\dagger$
(6)	$k_i = \arg \min \ g_i^{(j)}\ ^2$
(7)	$W_i = g_i^{(k_i)}$
(8)	Exchange columns $k_i$ and $i$ in $U$
(9)	$\mathcal{X}_i = W_i \cdot r$
(10)	$\hat{x}_i = Q[\mathcal{X}_i]$
(11)	$r_{i+1} = r_i - h_i \cdot \hat{x}_i$
(12)	$H_{i+1} = H_i^{\bar{k}_i}$
(13)	end
(14)	Output $U$

### Complexity in the preamble processing

In order to find the weighting vectors, an iterative algorithm that consists of two steps can be performed. First the steps are described and then the complexity will be determined:

**Step 1:** Compute the pseudo-inverse of  $H$ ,  $H^\dagger$ . Find the minimum squared length row of  $H^\dagger$ . This row is a weight vector. Permute it to be the last row and permute the columns of  $H$  accordingly.

**Step 2:** While  $N_t - 1 > 0$  go back to step 1, but now with :

$$H \rightarrow H^{(N_t-1)} = (h_1, \mathbf{K}, h_{N_t-1}) \text{ and } N_t \rightarrow N_t - 1$$

The complexity of calculating the pseudo-inverse is already determined in Section 4.3. For an  $N_r \times N_t$  dimensional matrix  $H$ , it equals  $7N_t^3 + 7N_t^2N_r - N_t$  flops.

The next steps are the calculation of the squared length of all rows of  $H^\dagger$  and the determination of the minimum squared length row. Note that, according to finding the minimum squared length row of  $H$  is equal to finding the minimum element  $P_{pp}$  on the diagonal of  $P$ ,  $p = 1, \mathbf{K}, N_t$ . Since  $P$  is obtained through the computation of the pseudo-inverse of  $H$ , the complexity of these steps consists only of finding the minimum. As explained in section 4.2, finding a minimum of  $N_t$  values has a complexity of  $N_t - 1$  real additions. The permutations of step 1 are considered to have no complexity. The only thing that needs to be done is exchanging the memory pointers that respectively point to the two rows of  $H^\dagger$  and the two columns of  $H$  that need to be permuted.

Since the algorithm is an iterative algorithm, and the dimensions of the used matrices scale down, the complexity per iteration is reduced. To take along this reduction in complexity during the iterations, the total complexity can be written by using series. The final number of real additions can be shown to be equal to

$$\begin{aligned}
NA &= 4 \sum_{p=1}^{N_t} p^3 + (8N_r - 2) \sum_{p=1}^{N_t} p^2 - 2N_r \sum_{p=1}^{N_t} p + \sum_{p=1}^{N_t} (p-1) \\
&= 4 \sum_{p=1}^{N_t} p^3 + (8N_r - 2) \sum_{p=1}^{N_t} p^2 + (1-2N_r) \sum_{p=1}^{N_t} (p-N_t) \\
&= N_t^2 (N_t + 1)^2 + (8N_r - 2) \frac{N_t (N_t + 1)(2N_t + 1)}{6} + (1-2N_r) \frac{N_t (N_t + 1)}{2} - N_t \\
&= \frac{1}{6} N_t (6N_t^3 + 8N_t^2(1+2N_r) + 3N_t(1+6N_r) + 2N_r - 5) \tag{4.10}
\end{aligned}$$

The total number of real multiplications ( $NM$ ) of the preamble phase of ZF-VBLAST equals

$$\begin{aligned}
NM &= 4 \sum_{p=1}^{N_t} p^3 + 8N_r \sum_{p=1}^{N_t} p^2 \\
&= N_t^2 (N_t + 1)^2 + 8N_r \frac{N_t (N_t + 1)(2N_t + 1)}{6} \tag{4.11}
\end{aligned}$$

### Complexity in the payload processing

During the data processing the weighting vectors are used to first estimate the best element of the transmitted vector  $x$ . The result is sliced to find a hard-decision value of the transmitted constellation symbol and then this symbol is used in the feedback loop in order to find the next estimate. The following steps represent this iterative process:

Step 1: Form the estimate of the best component  $p$  of  $x$ . Due to the permutation the corresponding weight vector equals the  $N_t^{th}$  row of permuted  $H^\dagger$ . In case of ZF:

$$(\mathcal{X})_p = G^{N_t} r$$

Slice  $(\mathcal{X})_p$  to the nearest constellation point  $(\hat{x})_p$

Step 2: While  $N_t - 1 > 0$  go back to step 1, but now with :

$$r \rightarrow r - h_{N_t}(\hat{x})_p \text{ and } N_t \rightarrow N_t - 1$$

The complexity of the first step of this iterative algorithm equals  $N_r - 1$  complex-additions and  $N_r$  complex multiplications, because two  $N_r$ -element vectors are multiplied. The slicing step for an  $M$ -ary constellation has a complexity of  $\log_2(M)$  real additions as explained in Section 4.1. Step 2 consists of a scalar-vector product and a vector subtraction. The scalar-vector product has a complexity that is equal to  $N_r$  complex multiplications and the complexity of the vector subtraction is  $N_r$  complex additions, since the vectors have  $N_r$  elements.

Because above steps are performed  $N_t$  times, it can be said that the complexity of the payload processing of Successive Interference Cancellation (VBLAST) with Zero Forcing equals  $2N_t(4N_r - 1) + N_r \log_2(M)$  real additions and  $8N_t N_r$  real multiplications per transmitted vector  $x$ .

$$\begin{aligned} C_{\text{ZF-VBLAST}}(\text{flops}) &= C_{\text{preamble}} + C_{\text{payload}} \\ &= N_t^4 + \frac{5}{3}N_t^3 + \frac{8}{3}N_t^3 N_r + \frac{3}{4}N_t^2 + \frac{7}{2}N_t^2 N_r \dots \\ &\quad + \frac{55}{6}N_t N_r - \frac{17}{12}N_t + \frac{1}{2}N_t \log_2(M) \end{aligned} \quad (4.12)$$

#### 4.4.2 Complexity of MMSE-VBLAST

The complexity of MMSE-VBLAST can be determined in the same way as is done for ZF-VBLAST in the previous section. Compared to ZF-VBLAST, there is a slight difference in the preamble processing, namely in the determination of the weight vectors. In case of MMSE, the iterative process of the weight calculation is given by

Step 1: Compute the weight matrix  $G = PH^H$ , with  $P = (\mathbf{S}I_{N_t} + H^H H)^{-1}$ . Find the smallest diagonal entry of  $P$  and suppose this is the  $p^{\text{th}}$  entry. Permute the  $p^{\text{th}}$  row of  $G$  to be the last row and permute the columns of  $H$  accordingly. The permuted row of  $G$  is a weight vector.

Step 2: (While  $N_t - 1 > 0$ ) go back to step I, but now with:

$$H \rightarrow H^{(N_t-1)} = (h_1, \mathbf{K}, h_{N_t-1}) \text{ and } N_t \rightarrow N_t - 1$$

Compared to ZF VBLAST, the complexity of step 1 is slightly higher, because of the addition of  $\mathbf{S}I$  to  $H^H H$ . Since  $\mathbf{S}$  is real, the only additional complexity consists of the  $N_t$  real additions of  $\mathbf{S}$  (i.e., the addition of  $\mathbf{S}$  to the real part of the diagonal elements of  $H^H H$ ). This leads to a complexity of  $4N_t^3 + N_t^2(8N_r - 2) - 2N_t N_r + N_t$  real additions and  $4N_t^3 + 8N_t^2 N_r$  real multiplications.

Taking all iterations of the algorithm into account, this leads to a total complexity (including the complexity of finding the minimal diagonal element of  $P$ ) of

$$\begin{aligned}
NA &= 4 \sum_{p=1}^{N_t} p^3 + (8N_r - 2) \sum_{p=1}^{N_t} p^2 + (1 - 2N_r) \sum_{p=1}^{N_t} (p) + \sum_{p=1}^{N_t} (p - 1) \\
&= \frac{1}{3} N_t (3N_t^3 + 4N_t^2 (1 + 2N_r) + 3N_t (1 + 3N_r) + N_r - 1) \quad (4.13)
\end{aligned}$$

and

$$\begin{aligned}
NM &= 4 \sum_{p=1}^{N_t} p^3 + 8N_r \sum_{p=1}^{N_t} p^2 \\
&= N_t^2 (N_t + 1)^2 + 8N_r \frac{N_t (N_t + 1) (2N_t + 1)}{6} \quad (4.14)
\end{aligned}$$

- |      |   |
|------|---|
| (1)  | input $H, r, \mathbf{s}, U$   |
| (2)  | $H^{(1)} = \left[ \left( H^H H + (\mathbf{s}^2 I) \right)^{-1} \right] H^H$ |
| (3)  | $\mathbf{r}^{(1)} = r$  |
| (4)  | for $i = 1, \dots, N_T$   |
| (5)  | $G_i = H_i^\dagger$   |
| (6)  | $k_i = \arg \min \ g_i^{(j)}\ ^2$   |
| (7)  | $W_i = g_i^{(k_i)}$   |
| (8)  | Exchange columns $k_i$ and $i$ in $U$                                       |
| (9)  | $\mathcal{X}_i = W_i \cdot r$   |
| (10) | $\hat{x}_i = Q[\mathcal{X}_i]$  |
| (11) | $r_{i+1} = r_i - h_i \cdot \hat{x}_i$                                       |
| (12) | $H_{i+1} = H_i^{\bar{k}_i}$   |
| (13) | end   |
| (14) | Output $U$  |

For the MMSE-VBLAST algorithm, the payload processing is equivalent to the ZF-VLAST technique, except that in the former case the weight vectors are rows of the processing matrix  $G$  instead of rows of the pseudo-inverse of  $H$ . The last fact is irrelevant for the complexity, thus, the complexity of the payload processing of MMSE-VBLAST equals  $2N_t(4N_r - 1) + N_t \log_2(M)$  real additions and  $8N_t N_r$  real multiplications per transmitted vector  $x$ .

$$\begin{aligned}
C_{\text{MMSE-VBLAST}}(\text{flops}) &= \frac{1}{6}N_t(3N_t^3 + 4N_t^2(1 + 2N_r) + 3N_t(1 + 3N_r) + N_r - 1) \dots \\
&\quad + \frac{1}{2}N_t^2(N_t + 1)^2 + 4N_r \frac{N_t(N_t + 1)(2N_t + 1)}{6} \dots \\
&\quad + 8N_t N_r - N_t + \frac{1}{2}N_t \log_2(M)
\end{aligned}$$

$$\begin{aligned}
C_{\text{MMSE-VBLAST}}(\text{flops}) &= C_{\text{preamble}} + C_{\text{payload}} \\
&= N_t^4 + \frac{5}{3}N_t^3 + \frac{7}{3}N_t^3 N_r + N_t^2 + \frac{7}{2}N_t^2 N_r \dots \\
&\quad + \frac{7}{6}N_t N_r - \frac{1}{6}N_t + \frac{1}{2}N_t \log_2(M)
\end{aligned} \tag{4.15}$$

## 4.5 Complexity of QR Decomposition Based Detection

### 4.5.1 Complexity of Zero-Forcing QR Decomposition (ZF-QRD) Detection

As pointed out in [58], in QR decoding most of the time is spent finding the QR factors of the channel matrix. In this section the complexity of ZF-QRD detection technique will be calculated through two stages, the first is to calculate the complexity of ZF-QRD factorization and the second is for calculating the complexity of detection process.

## I. Complexity of Zero-Forcing QR Decomposition

The following are the required operations to obtain the ZF QR decomposition (ZF-QRD). The detailed algorithm is depicted in Table 4.2.

Complex multiplication:

$$\sum_{i=1}^{N_t} \sum_{k=i+1}^{N_t} (2N_r) = N_t^2 N_r - N_t N_r$$

Complex power:

$$\sum_{i=1}^{N_t} (N_r) = N_t N_r$$

Complex addition:

$$\sum_{i=1}^{N_t} \sum_{k=i+1}^{N_t} (2N_r - 1) = N_t^2 N_r - N_t N_r - \frac{1}{2} N_t^2 + \frac{1}{2} N_t$$

Square root:

$$\sum_{i=1}^{N_t} (1) = N_t$$

Real additions:

$$\sum_{i=1}^{N_t} (N_r - 1) = N_t (N_r - 1)$$

Division of complex number by real number:

$$\sum_{i=1}^{N_t} (N_r) = N_t N_r$$

Now, based on the complexity values listed in Table 4.1, the complexity of ZF-SQRD factorization equals

$$C_{\text{ZF-QRD}}(\text{flops}) = 4N_t^2N_r - \frac{1}{2}N_t^2 - N_tN_r + \frac{1}{2}N_t \quad (4.16)$$

**Table 4.2: ZF QR decomposition procedures**

<pre> input <math>H, p = 1, 2, \dots, N_t</math> <math>R = 0, Q = H</math> for <math>i = 1, \dots, N_T</math> do   <math>R_{i,i} = \sqrt{\ q_i\ ^2}</math>   <math>q_i = q_i / R_{i,i}</math>   for <math>k = i + 1, \mathbf{K}, N_t</math> do     <math>R_{i,k} = q_i^H \cdot q_k</math>     <math>q_k = q_k - R_{i,k} \cdot q_i</math>   end end end Output <math>Q, R</math> </pre>
--

## II. Complexity of detection stage

The required operations to obtain detection stage complexity are the followings. The detection stage sub algorithm is depicted in Table 4.3.

Complex multiplication

$$\begin{aligned}
& N_t N_r + \sum_{i=1}^{N_t} \left( \sum_{k=i+1}^{N_t} 1 \right) \\
&= N_t N_r + \left( \sum_{i=1}^{N_t} i \right) - N_t = N_t N_r + \sum_{i=1}^{N_t} (i-1) = N_t N_r + \frac{1}{2} N_t^2 - \frac{1}{2} N_t
\end{aligned}$$

Complex addition

$$N_t (N_r - 1) + \sum_{i=1}^{N_t} (1) = N_t (N_r - 1) + N_t$$

Complex division

$$\sum_{i=1}^{N_t} (1) = N_t$$

**Table 4.3: Steps of detection stage**

$y = Q^H \mathbf{r}$ <p>for <math>k = N_t, \mathbf{K}, 1</math></p> $\hat{d}_k = \sum_{i=k+1}^{N_t} R_{k,i} \hat{x}_i$ $\hat{x}_k = Q \left( \frac{y_k - \hat{d}_k}{R_{k,k}} \right)$ <p>end</p>
--

And the required operations for slicing  $N_t M$ -ary constellation points are

$N_t \log_2(M)$  real additions.

Then, the complexity of the detection stage is equal

$$C_{\text{Detection stage}} (\text{flops}) = \frac{3}{2} N_t^2 + 4N_t + 4N_t N_r + \frac{1}{2} N_t \log_2(M) \quad (4.17)$$

The total complexity of ZF-QRD detection technique equals

$$\begin{aligned}
 C (\text{flops}) &= C_{\text{ZF-QRD}} + C_{\text{Detection stage}} \\
 &= 4N_t^2 N_r + 3N_t N_r + N_t^2 + \frac{9}{2} N_t + \frac{1}{2} N_t \log_2(M) \quad (4.18)
 \end{aligned}$$

#### 4.5.2 Complexity of Zero-Forcing Sorted QR Decomposition (ZF-SQRD)

Below are the required operations to obtain the ZF sorted QR decomposition (ZF-SQRD). The detailed algorithm is depicted in Table 4.4.

Complex multiplication:

$$\sum_{i=1}^{N_t} \sum_{k=i+1}^{N_t} (2N_r) = N_t^2 N_r - N_t N_r$$

Complex power:

$$\sum_{i=1}^{N_t} (N_r) + \sum_{i=1}^{N_t} \sum_{k=i+1}^{N_t} (1) = N_t N_r + \frac{1}{2} N_t^2 - \frac{1}{2} N_t$$

Complex addition:

$$\sum_{i=1}^{N_t} \sum_{k=i+1}^{N_t} (2N_r - 1) = N_t^2 N_r - N_t N_r - \frac{1}{2} N_t^2 + \frac{1}{2} N_t$$

Square root:

$$\sum_{i=1}^{N_t} (1) = N_t$$

Real additions:

$$\sum_{i=1}^{N_t} (N_r - 1) + \sum_{i=1}^{N_t} \sum_{k=i+1}^{N_t} (1) = N_t N_r + \frac{1}{2} N_t^2 - \frac{3}{2} N_t$$

Division of complex number by real number:

$$\sum_{i=1}^{N_t} (N_r) = N_t N_r$$

**Table 4.4: ZF-SQRD algorithm**

```

input  $H, p = 1, 2, \dots, N_t$ 
 $R = 0, Q = H$ 
for  $i = 1, \dots, N_T$ 
    norms $_i = \|q_i\|^2$ 
end
for  $i = 1, \dots, N_T$ 
     $k_i = \arg \min_{j=i, \dots, N_T} (\text{norms}_j)$ 
    Exchange  $i$ -th  $k$ -th columns in  $R, Q, p$  and norms
     $R_{i,i} = \sqrt{\text{norms}_i}$ 
     $q_i = q_i / R_{i,i}$ 
    for  $k = i + 1, \mathbf{K}, N_t$ 
         $R_{i,k} = q_i^H \cdot q_k$ 
         $q_k = q_k - R_{i,k} \cdot q_i$ 
        norms $_k = \text{norms}_k - |R_{i,k}|^2$ 
    end
end, Output  $Q, R$ 

```

Real comparisons (equivalent to real subtraction):

$$\sum_{i=1}^{N_t} (N_t - i) = \frac{1}{2} N_t^2 - \frac{1}{2} N_t$$

Based on the complexity assumption listed in Table 4.1, the total complexity of ZF-Sorted QR Decomposition in terms of flops equals:

$$C_{\text{ZF-SQRD}} (\text{flops}) = 4N_t^2 N_r + \frac{3}{4} N_t^2 - N_t N_r - \frac{3}{4} N_t \quad (4.19)$$

And the total complexity of ZF-SQRD detection technique equals

$$C (\text{flops}) = C_{\text{ZF-SQRD}} + C_{\text{Detection satge}}$$

$$= 4N_t^2 N_r + 3N_t N_r + \frac{9}{4}N_t^2 + \frac{13}{4}N_t + \frac{1}{2}N_t \log_2(M) \quad (4.20)$$

### 4.5.3 MMSE Sorted QR Decomposition (MMSE-SQRD)

The detailed algorithm of MMSE-SQRD is depicted in Table 4.5. Below are the required operations to obtain the complexity of this algorithm:

Complex multiplication:

$$\sum_{i=1}^{N_t} \sum_{k=i+1}^{N_t} (2N_r + 2i) = N_t^2 N_r - N_t N_r + \frac{1}{3}N_t^2 - \frac{1}{3}N_t$$

Complex power:

$$\sum_{i=1}^{N_t} (N_r) + \sum_{i=1}^{N_t} \sum_{k=i+1}^{N_t} (1) = N_t N_r + \frac{1}{2}N_t^2 - \frac{1}{2}N_t$$

Complex addition:

$$\sum_{i=1}^{N_t} \sum_{k=i+1}^{N_t} (2N_r + 2i - 1) = \frac{1}{3}N_t^3 + N_t^2 N_r - N_t N_r - \frac{1}{2}N_t^2 + \frac{1}{6}N_t$$

Square root:

$$\sum_{i=1}^{N_t} (1) = N_t$$

Real additions:

$$\sum_{i=1}^{N_t} (N_r) + \sum_{i=1}^{N_t} \sum_{k=i+1}^{N_t} (1) = N_t N_r + \frac{1}{2}N_t^2 - \frac{1}{2}N_t$$

**Table 4.5: MMSE-SQRD algorithm**

```

input  $H, p = 1, 2, \dots, N_t$ 
 $R = 0, Q = H$ 
for  $i = 1, \dots, N_T$ 
    norms $_i = \|q_i\|^2$ 
end
for  $i = 1, \dots, N_T$ 
     $k_i = \arg \min_{j=i \dots N_T} (\text{norms}_j)$ 
    Exchange  $i$ -th  $k$ -th columns in R, p, norms, and the  $(m+i-1)$  rows of  $Q$ 
     $R_{i,i} = \sqrt{\text{norms}_i}$ 
     $q_i = q_i / R_{i,i}$ 
    for  $k = i+1, \mathbf{K}, N_t$ 
         $R_{i,k} = q_i^H \cdot q_k$ 
         $q_k = q_k - R_{i,k} \cdot q_i$ 
        norms $_k = \text{norms}_k - |R_{i,k}|^2$ 
    end
end
Output  $Q, R$ 

```

Real comparisons

$$\sum_{i=1}^{N_t} (N_t - i) = \frac{1}{2} N_t^2 - \frac{1}{2} N_t$$

The total complexity MMSE-Sorted QR Decomposition in terms of flops equals:

$$C_{\text{MMSE-SQRD}} (\text{flops}) = \frac{4}{3} N_t^3 + 4 N_t^2 N_r + \frac{5}{4} N_t^2 - N_t N_r - \frac{13}{12} N_t \quad (4.21)$$

And the total complexity of MMSE-SQRD detection technique equals

$$C (\text{flops}) = C_{\text{MMSE-SQRD}} + C_{\text{Detection satge}}$$

$$= \frac{4}{3}N_t^3 + 4N_t^2N_r + \frac{11}{4}N_t^2 + 3N_tN_r + \frac{35}{12}N_t + \frac{1}{2}N_t \log_2(M) \quad (4.22)$$

#### 4.5.4 MMSE QR Decomposition (MMSE-SQRD)

The complexity of MMSE-QRD is equivalent to that of the MMSE-SQRD algorithm without the sorting operations; that is,

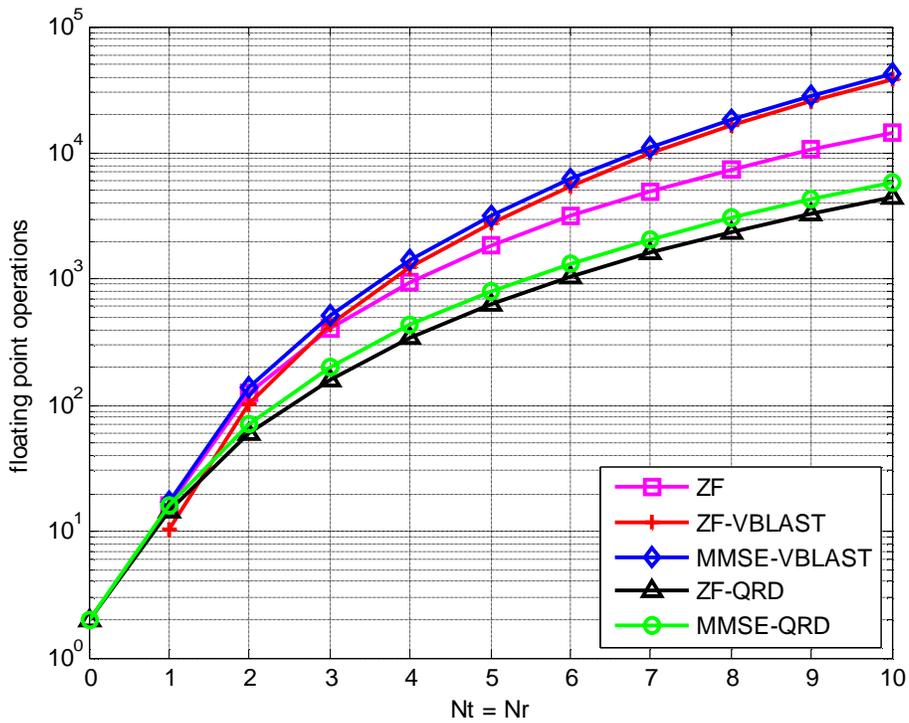
$$C_{\text{MMSE-QRD}}(\text{flops}) = C_{\text{MMSE-SQRD}} - \left( \frac{5}{4}N_t^2 - \frac{5}{4}N_t \right)$$

$$= \frac{4}{3}N_t^3 + 4N_t^2N_r + \frac{3}{2}N_t^2 + 3N_tN_r + \frac{25}{6}N_t + \frac{1}{2}N_t \log_2(M) \quad (4.23)$$

## 4.6 Complexity comparison of linear, SIC and QRD detection techniques

In this section, the computational complexities of linear, VBLAST and QRD detection techniques are compared. The computational complexities of these algorithms were analyzed and discussed in terms of floating point operations (flops) in sections 4.3, section 4.4, and section 4.5 respectively. Figure 4.1 presents a comparison of ZF, ZF-VBLAST, MMSE-VBLAST, ZF-QRD and MMSE-QRD algorithms in terms of computational complexity. In particular, it illustrates that the VBLAST detection schemes has the highest computational complexity. On the other hand the QRD techniques have lower number of floating point operations if compared with linear and VBLAST detections. This demonstrates the fact that VBLAST

detections endure much computational effort in the calculations of the pseudo-inverse of the channel matrix (see section 3.2). It also noticed that MMSE based detections have slightly higher number of flops than those ZF based detection because MMSE based detection has only additional  $Nt$  real additions to ZF based detection complexity.



**Figure 4.1** Number of floating point operations for linear, VBLAST and QRD detection techniques of a MIMO system with  $N_t = N_r$  antennas

#### 4.7 Complexity analysis of tree search algorithms

Linear detectors usually have lower complexity at the cost of performance degradation. Tree search detectors; SD and QRD-M theoretically can achieve optimal performance. However, the complexity of these detectors may become prohibitive at low SNR [59]. Most tree-search-based MIMO detection techniques often have limited

performance because of the required hardware implementations; large memory requirement or high computational complexity of sophisticated algorithms. In this section, the complexity of two tree search based detection algorithms are presented and compared.

#### 4.7.1 Complexity of sphere decoding detection

The complexity of the SD algorithm and its variations has been discussed extensively in literature [42]. As shown, the complexity of ML detection given by solving (2.2) is exponential or NP-hard [41]. This as discussed earlier is not feasible to implement in a practical system, especially when high rate lattice constellations are used and multiple antennas are utilized. Therefore, it is very important that the SD algorithm provides a low complexity alternative to exhaustive ML detection. It has been shown in fact that the average complexity of sphere decoding is polynomial in the number of unknowns [42], [41], roughly  $O(m^3)$ . However, in worst case conditions the complexity of sphere decoding is still exponential, making it inefficient in these conditions. To understand this, it has to be stated that the complexity of SD is highly dependent on many factors, namely the SNR and the choice of initial radius. Therefore in worst case conditions of these parameters, SD will have an exponential complexity. One perception about the SD complexity is that it is a random variable, with an expected complexity proven to be polynomial [42].

There are various ways to analyze and measure the complexity of the SD algorithm. The one most commonly used is to measure the number of *flops* (floating point operations). This is the approach taken in most papers dealing with the complexity of SD [32], [42], [41]. Other ways also include calculating the expected number of visited nodes in the algorithm [59]. Counting the number of flops may

seem to be the most popular way to measure the complexity; however one has to be sure of the accuracy of the count. Most times the assumption is made that the multiplication and addition operations are equivalent in complexity measure. This is inaccurate as one multiplication operation is far more complex than an addition operation. Nevertheless the number of flops is still an acceptable qualitative measure of the complexity. The simple and brief notation used to state the complexity of SD without rigorously analyzing it would be the big  $O$  notation, which is used to roughly estimate the complexity.

The complexity of SD will also depend on the pre-processing steps taken before the actual recursive search. Pre-processing steps include the QR decomposition of the channel matrix  $H$  and initial radius selection. If a quasi-static channel is assumed, that is the channel matrix  $H$  only changes every block of transmitted symbols, then the QR decomposition has to be calculated for every block. As given in [19], the QR decomposition algorithm requires  $2nm^2$  flops for an  $n \times m$  matrix. This means that for the system model adopted here where  $n \geq m$ , then the QR decomposition itself is lower bounded by  $O(m^3)$ . If the channel is quickly varying, then the block length for which the channel matrix  $H$  is static becomes very small. This would mean that  $H$  would change more often requiring the calculation of the QR decomposition every time. The complexity of the QR decomposition becomes significant at this point.

If a deterministic approach is used for the initial radius selection, then the complexity of the sub-optimal estimate of the radius has to be taken into account. For example, using the ZF or the MMSE estimates will require algorithms with complexity on the order of  $O(m^3)$  due to the matrix multiplications involved. If you

take an  $n \times m$  matrix and multiply it by an  $m \times n$  matrix, then the operation has a complexity on the order of  $O(m^3)$ , which is also lower bounded by  $O(m^3)$ . Since the ZF and the MMSE estimates are a series of matrix multiplications and one inversion, then their complexities are also at least on the order of  $O(m^3)$ .

It is easy to see that the complexity of the actual SD search algorithm (excluding the preprocessing steps) is proportional to the number of nodes visited within the sphere since more visited nodes will mean more computations [42]. A very useful expression for the complexity is given in [42] by

$$C(m, s^2, d) = \sum_{i=1}^m (\text{expected \# points } i\text{-dimensional sphere}(d)) \cdot f_p(i) \quad (4.24)$$

where  $f_p(i)$  represents the number of flops per visited point and is given by

$$f_p(i) = 2i + 11 \quad (4.25)$$

The complexity expression in (4.24) is stated as a function of the number of dimensions, noise variance, and sphere radius to emphasize its dependency upon these parameters.

It is clear to see the dependency of SD complexity on the initial radius selection as discussed earlier since the initial radius will determine how many points lie inside the sphere. Therefore the worst case scenario would be choosing a large initial radius which will mean more points than needed inside the sphere, and the complexity will be exponential. The dependency of the complexity on the SNR is also easy to see. In low SNR conditions, points inside the sphere will be tightly spaced. This will naturally show that for the same initial radius, more points will lie inside the sphere for low SNR conditions than high SNR conditions.

It is very difficult to fully analyze the complexity mathematically and come up with closed form expressions for its statistics. A good analysis of the complexity of the SD algorithm was carried out in [42] and [59]; however the analysis had to undertake a lot of simplifying assumptions. It remains to say that the most accurate way of actually calculating the complexity is only through empirical results.

In conclusion, SD will have an expected polynomial complexity for a wide range of system parameters [41]. However, this is still not sufficient to fully implement SD in practical systems as SD will perform poorly with exponential complexity in worst case SNR conditions and bad initial radius choices. It is therefore essential to find variations of the SD algorithm to provide low complexity at all times.

#### **4.7.2 Complexity of QRD-M algorithm detection**

QR-decomposition with M-algorithm was introduced to overcome the random complexity of the SD by retaining a fixed number of candidates per detection level. The original QRD-M has fixed complexity regardless of the channel environment due to the constant selection of survival branches at each stage. The complexity is defined by the total number of branch metric calculations. The parameters which determine the complexity of the QRD-M algorithm; are the number of transmit antennas  $N_t$ , the number of survival candidates vector at each stage  $M$ , and modulation scheme.

#### **4.7.3 Complexity Comparison of QRD-M and SD**

In this section, further comparison of the implementation complexity of the QRD-M and SD algorithms. These comparisons have been achieved through computer simulations performed by Dai *et al* in [51]. The comparison results are listed in Table 4.6 for QPSK and Table 4.7 for 16-QAM modulated 4 x 4 systems, respectively.

To ensure a fair comparison, for the QRD-M algorithm,  $M = 4$  is used for QPSK modulation and  $M = 16$  is used for 16-QAM modulation, in which cases QRD-M has the same ML-achieving performance as the SD algorithm. It is clear from results attained in Tables 4.6 and 4.7 that SD always performs lower average complexity. Its complexity advantage is observed for 16-QAM modulation in particular. The problem of SD lies in that its worst case complexity is significantly higher than the average one.

**Table 4.6: Complexity comparison for a QRSK MIMO system with  $N_t = N_r = 4$**

	Mean # of real multiplications	Max # of real multiplications	Mean # of visited nodes	Max # of visited nodes
<b>ML</b>	2272	2272	85(1+4+16+64)	85
<b>QRD-M (M=4)</b>	304	304	13(1+4+4+4)	13
<b>SD</b>	150	2052	14	51

**Table 4.7: Complexity comparison for 16 QAM MIMO system with  $N_t = N_r = 4$**

	Mean # of real multiplications	Max # of real multiplications	Mean # of visited nodes	Max # of visited nodes
<b>ML</b>	330880	330880	4369(1+16+256+4096)	4369
<b>QRD-M (M=16)</b>	3520	3520	49(1+16+16+16)	49
<b>SD</b>	200	8248	19	676

SD achieves a tremendous reduction in the average complexity; its instantaneous complexity depends on the noise variance and channel conditioning. Thus, for ill-

conditioned matrix or high noise power, the complexity of SD can be comparable with that of the ML algorithm [60]. The second scheme in this category is the QRD-M, which was proposed to achieve quasi-ML performance while requiring fixed computational effort. After the QR-decomposition of the channel matrix into unitary and upper triangular matrices, the M-algorithm is applied to retain the best M symbol replicas candidates at each detection stage. At the last detection stage, the branch with the smallest accumulative metric is considered as the estimate of the transmitted vector. Thus, QRD-M algorithm overcomes the extreme case complexity of SD and become more amenable to hardware implementation than SD. While the number of survival candidates at each detection layer increases, the QRD-M performance approaches that of MLD detection as shown in [61]. However, it should be noted that this improvement come at the cost of increased computational complexity.

#### **4.8 Summary**

In this chapter, the complexity of all detection techniques described in chapter 3 has been analyzed and calculated. This computational complexity could allow to estimate the potential cost of the algorithm and to identify possible bottlenecks for the hardware implementation. The linear detection techniques have an efficient computational complexity but with low performance. But the VBLAST techniques endure a computational bottleneck due to the multiple calculation of pseudo-inverse in detection procedures. This computational bottleneck has been avoided by QR-factorization involved in QRD detection techniques. The tree-search techniques have significant achievements in reducing computational complexity. SD achieved a lower average complexity; but its worst case complexity is comparable to that of MLD when channel is ill-conditioned. The more amenable detection algorithm was the QRD-M as it requires a fixed level of computational complexity.

## CHAPTER 5

### CONCLUSION AND FUTURE WORK

---

#### 5.1 Conclusion

In recent years, MIMO wireless communication systems have exploited spatial multiplexing (SM) approach to increase the channel capacity and improve spectral efficiency as well. Therefore, the MIMO SM-based system has been one of currently promising techniques that could realize Gbps high-speed wireless transmission for future communications networks. The main challenge of MIMO SM-based system resides in designing signal processing techniques, i.e., detection techniques. Those are capable of separating the parallel transmitted signals with acceptable computational complexity and achieved performance. An intensive work is being done in this field to investigate several MIMO SM detection techniques such linear, nonlinear and tree-based detections.

In this study, several MIMO detection techniques have been successfully described, analyzed and compared. In general, linear detection techniques such as ZF and MMSE have an efficient computational complexity; however, the BER performance plots of these techniques demonstrated their relatively poor performance. In an attempt to improve the poor performance of the linear detections, VBLAST have been proposed. It was shown that the ordering strategy over Successive Interference Cancellation (VBLAST) has important benefits. This strategy was applied to the general V-BLAST code and got a higher performance gain. However, performance improvement with SIC techniques is limited due to error propagation,

particularly with the same number of transmitter antennas as receiver antennas. Additionally, the main drawback of the VBLAST detection algorithms lies in the computational complexity, because multiple calculations of the pseudo-inverse of the channel matrix are required. This involves expensive computational requirements and makes VBLAST algorithms enduring computational bottleneck. This computational bottleneck can be avoided using QR Decomposition based algorithm such as ZF-QRD and MMSE-QRD.

In the MLD detection, corresponding metrics are generated for all possible transmitted symbols, and the vector with the least MLD metric is considered as the estimate of the transmitted vector. Although MLD achieves the best performance and diversity order, it requires a brute-force search which is exponential in the number of transmit antennas and constellation set size. Thus, for high problem size, i.e., high modulation order and high  $N_t$ , MLD becomes infeasible from a hardware implementation perspective. In order to achieve quasi-MLD performance with a computationally feasible level of complexity, SD and QRD-M detection techniques have been proposed. These two techniques are the most appealing ones among tree-search based detection schemes. With SD only searching through those candidates falling inside a hypersphere, thus reduces the high complexity of MLD. However, the main drawback of SD, a depth-first search algorithm, is that despite the low average complexity, the worst case complexity of the SD is identical to that of MLD. On the other hand, QRD-M, a breadth-first search algorithm, requires a fixed level of computational complexity by only keeping those candidates with the best accumulated metric values at each step; therefore, QRD-M is more amenable to hardware implementation compared to SD. while the number of survival candidates at each detection layer increases, the QRD-M performance approaches that of MLD

detection, however, this improvement come at the cost of increased computational complexity.

## 5.2 Future work

This study has addressed a number of important issues associated with MIMO SM detection techniques. In particular, provided a detailed description, analysis and comparison of performance and computational complexity of several detection techniques and gave a recommendation for those promising techniques that are potentially amenable to hardware implementation. However, there are still a number of issues that require further analysis and investigation, these include but are not limited to the followings:

- To further reduce the computational complexity of tree-search based detection techniques i.e. SD and QRD-M, future research could investigate further development and improvement of the complexity and performance limiting factors such as best initial radius selection for the case of SD and the number of survival candidates at each detection level for the case of QRD-M.
- Among all detection techniques described in this study, the QRD-M algorithm was extensively studied in the literature and introduced as a potential candidate for signal detection in the future communication systems. Further work could be conducted to improve the efficiency of the QRD-M algorithm by reducing its complexity, processing it in an iterative way to reduce the hardware requirements, or in parallel to reduce the detection latency.
- Generally in MIMO SM detection research and throughout this study, an assumption made is that channel coefficients are known at the receiver and that

this information is always correctly known. Future research could include a comparison of linear, SIC and tree search based techniques and previous detections for the scenarios where the channel knowledge at the receiver is not known.

- This study deal with the general model of MIMO system, further investigations of MIMO SM detection techniques could be achieved for the MIMO OFDM. Because OFDM provides an attractive and practical solution for future high-speed indoor wireless data communication networks. It combines the data-rate and spectral-efficiency enhancements of SDM with the relatively high spectral efficiency and the robustness against different channel impairments.
- The work conducted in this study was independent of channel coding and modulation; hence more investigation could be performed for the MIMO system with different channel coding and several modulation schemes
- Further investigation could be conducted for a detection technique that could utilize a combination of detection schemes described throughout this study, i.e. developing a hybrid MIMO SM detection algorithm
- A prototype implementation of MIMO SM detection system could be designed and implemented on FPGA kit.

## References

- [1] P. W. Wolniansky, G. J. Foschini, G. D. Golden, and R. A. Valenzuela, "V-BLAST: An architecture for realizing very high data rates over the rich-scattering wireless channel," *URSI Int. Symp. on Signals, Systems and Electronics*, (Pisa, Italy), pp. 295-300, Sep. 1998.
- [2] G.J.Foschini. and M.J.Gans, "On the Limits of Wireless Communication in a Fading Environment When Using Multiple Antennas," *Wireless Personal Communications*, vol. 6, no. 3, pp. 311-355, 1998.
- [3] G. J. Foschini, "Layered space-time architecture for wireless communication in a fading environment when using multi-element antennas," *Bell Labs Tech. J.*, vol. 1, no. 2, pp. 41-59, 1996.
- [4] Jeffrey G.Andrews, Arunabha Ghosh, and Rais Mohamed," *Fundamentals of WiMAX: Understanding Broadband Wireless Networking*", Prentice Hall, 2007.
- [5] S. M. Alamouti, "A simple transmit diversity technique for wireless communications", *IEEE Journal on Select Areas in Communications*, vol.16, pp.1451–1458, August 1998.
- [6] V. Tarokh, H. Jafarkhani, and A. Calderbank, "Space-time block codes from orthogonal designs," *IEEE Trans. On Information Theory*, vol. 45, no. 5, pp. 1456-1467, July 1999.
- [7] A. Kaye and D. George, "Transmission of multiplexed PAM signals over multiple channel and diversity systems," *IEEE Trans. On Communication Technology*, vol. 18, no. 5, pp. 520-526, October 1970.
- [8] W. Van Etten, "An optimum linear receiver for multiple channel digital transmission systems," *IEEE Trans. On Communications*, vol. 23, pp. 828-834, August 1975.
- [9] Simon Haykin, and Michael Moher, "Modern Wireless Communications", Prentice-Hall Inc., 2005.
- [10] Choo C. Chiau, "Study of the Diversity Antenna Array for the MIMO Wireless Communication Systems ", PHD thesis, University of London, 2006.
- [11] E. Telatar, "Capacity of multi-antenna Gaussian channels," *European Trans. on Telecommunications*, vol. 10, pp. 585-595, Dec. 1999.

- [12] Ezio Biglieri et al., "MIMO Wireless Communications", Cambridge University Press, 2007.
- [13] Z. Xu and R. D. Murch, "Performance analysis of maximum likelihood detection in a MIMO antenna system," *IEEE Trans. Commun.*, vol. 50, no. 2, pp. 187–191, Feb. 2002.
- [14] G. D. Golden, G.J. Foschini, R. A. Valenzuela, and P. W. Wolniansky, "Detection algorithm and initial laboratory results using V-BLAST space-time communication architecture," *Electron. Lett.*, vol. 35, pp. 14–16, Jan. 1999.
- [15] D. Gore, R. W. Heath Jr., and A. Paulraj, "On performance of the zero forcing receiver in presence of transmit correlation," in *Proc. Int. Symp. Inf. Theory*, 2002, p. 159.
- [16] G. G. Raleigh and J.M. Cioffi, "Spatio-temporal coding for wireless communications," *IEEE Trans Commun.*, vol. 46, pp. 357-366, Mar. 1998.
- [17] H. Bolcskei, D. Gesbert, and A.J. Paulraj, "On the capacity of OFDM-based spatial multiplexing systems," *IEEE Trans. Commun.*, vol. 50, pp. 225-234, Feb. 2002.
- [18] J. W. Wallace and M.A Jensen, "MIMO capacity variation with SNR and multipath richness from full-wave indoor FDTD simulations," *IEEE Antenna and Propagation Society International Symposium*, vol. 2, pp. 523-526, 22-27 June 2003.
- [19] A. F. Molish and M. Z. Wi, "MIMO systems with antenna selection - An overview," copyright Mitsubishi Electric Research Laboratories, Inc., 2004.
- [20] Horst j. Bessai, "MIMO Signals and Systems", Springer Science-HBusiness Media, 2005.
- [21] H. Bolcskei, D. Gesbert, C. Papadias, and A. J. van der Veen, Eds., "Space-Time Wireless Systems: From Array Processing to MIMO Communications", Cambridge Univ. Press, 2006.
- [22] H. Bolcskei and A. j. Paulraj, "Multiple-input multiple-output (MIMO) wireless systems", Chapter in "The Communications Handbook", 2nd edition, J. Gibson, ed., CRC Press, pp. 90.1 - 90.14, 2002.
- [23] D. Wubben, R. Bohnke, V. Khun, and K.-D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Proc. IEEE Vehicular Tech. Conf.*, October 2003, pp. 508-512.

- [24] M. Jankiraman, "Space-time codes and MIMO systems", Artech House, Boston, 2004.
- [25] David Tse, Pramod Viswanath, "Fundamentals of Wireless Communication" Cambridge University Press, 2005
- [26] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. D. Kammeyer, "Efficient algorithm for decoding layered space time codes," IEE Electronic Letters, vol. 37, no. 22, pp. 1348- 1350, 2001.
- [27] D. Wubben, J. Rinas, R. Bohnke, V. Kuhn, and K. D. Kammeyer, "Efficient algorithm for detecting layered space time codes," in Proc. ITG Conference on Source and Channel Coding, Berlin, Germany, January 2002, pp. 399-405.
- [28] E. Biglier, G Taricco, and A. Tulino, "Decoding Space-Time Codes With BLAST Architectures," IEEE Transactions on Signal Processing vol. 50, no. 10, pp. 2547-2551, October 2002.
- [29] R. Bohnke, D. Wubben, V. Khun, and K.-D. Kammeyer, "Reduced Complexity MMSE Detection for BLAST Architectures," in Proc. IEEE Global Communication Conference (Globecom'03), San Francisco, California, USA, December 2003.
- [30] D. Shiu and J. M. Kahn, "Layered space-time codes for wireless communications using multiple transmit antennas," in proc. IEEE International Conference on Communication (ICC'99), pp. 436-440, June 1999.
- [31] E. Viterbo and E. Biglieri, " A universal decoding algorithm for lattice codes," in Proc. GRETSI, Juan- les- Pins, France, pp. 611- 614, September 1993.
- [32] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced bases with applications," ACM SIGSAM, pp.37-44, 1981.
- [33] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in lattice, including a complexity analysis", AMS Mathematics of Computation, vol. 44, no. 170, pp. 463-471, April, 1985
- [34] M. O. Damen, A. Chkeif, and J. C. Belfiore, "Lattice code decoder for space-time codes," IEEE Communications Letters, vol. 4, pp. 161-163, May 2000.
- [35] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," IEEE Transactions on Information Theory, vol. 48, no. 8, pp. 2201-2214, August 2002.

- [36] B. Hassibi and H. Vikalo, "On the expected complexity of sphere decoding," in Proc. Asilomar Conference on Signals, Systems and Computers, pp. 1051–1055, November 2001.
- [37] H. Vikalo, B. Hassibi, and U. Mitra, "Sphere-constrained ML detection for frequency-selective channels," IEEE Transactions on Communications, vol. 54, no. 7, pp. 1179 – 1183, July 2006.
- [38] L. Qingwei, "Efficient VLSI Architectures for MIMO and Cryptography Systems", PhD Dissertation, Oregon State University, January 2008.
- [39] Z. Lei, Y. Ting-ting, Z. Xin, Y. Da-cheng " Combined simplified maximum likelihood and sphere decoding algorithm for MIMO system" Journal of China Universities of Posts and Telecommunications, Vol. 15, no. 2, June 2008.
- [40] E. Viterbo and J. Boutros, " A universal lattice code decoder for fading channels," IEEE Transaction on Information Theory, vol. 45, no.5, pp. 1639-1642, July 1999.
- [41] C. P. Schnorr and M. Euchner. "Lattice basis reduction: improved practical algorithms and solving subset sum problems", Mathematical Programming, vol. 66, pp.181–191, 1994.
- [42] M. O. Damen, H. El Gamal, and G. Caire. "On maximum-likelihood detection and the search for the closest lattice point", IEEE Transactions on Information Theory, vol. 49, no. 10, pp. 2389–2401, October 2003.
- [43] B. Hassibi and H. Vikalo, " On the Sphere Decoding Algorithm: Part I, The Expected Complexity", IEEE Trans. Sig. Process., vol. 53, no. 8, pp. 2806–2818, August 2005.
- [44] H. S. Dhillon, H. Aggarwal and A. Mitra, "A Simple Direction-Based Lattice Decoding Algorithm for Golden Codes", National Conference on Communications (NCC2009), pp. 128-131, January 2009.
- [45] A. D. Murugan, H. E. Gamal, M. O. Damen, and G. Caire," A unified framework for tree search decoding: rediscovering the sequential decoder", IEEE Transactions on Information Theory, 52(3):933–953, March 2006.
- [46] K. Radhakrishnan "Implementation of a Soft Output Sphere Decoder by Rapid Prototyping Methodology " Technical University of Vienna , November, 2007
- [47] A. Burg, M. Borgmann, M. Wenk, M. Zellweger, W. Fichtner, H. Boelcskei : "VLSI Implementation of MIMO Detection Using the Sphere Decoding Algorithm", IEEE-JSSC, July 2005.

- [48] K. J. Kim and R. A. Iltis, "Joint Detection and Channel Estimation Algorithms for QS-CDMA Signals Over Time-Varying Channels," *IEEE Transaction on Communications*, vol. 50, pp. 845- 855, May 2002.
- [49] J. Yue, K. J. Kim, J. D. Gibson, and R. A. Iltis, "Channel Estimation and Data Detection for MIMO-OFDM Systems," in *Proc. GLOBECOM 2003*, pp. 581–585, 2003.
- [50] K. J. Kim, J. Yue, R. A. Iltis, and J. D. Gibson, "A QRD-M/Kalman filter based detection and channel estimation algorithm for MIMO-OFDM systems," *IEEE Transactions on Wireless Communications*, vol. 4, no. 2, pp. 710-721, Mar. 2005.
- [51] J. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Transactions On Communications*, vol. COM- 32, no. 2, pp. 169- 176, February 1984.
- [52] Y. Dai, S. Sun and Z. Lei, "A comparative study of QRD-M detection and sphere decoding for MIMO-OFDM systems," in *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, 2005.
- [53] L. Barbero and J. Thompson, "Fixing the complexity of the sphere decoder for MIMO detection," *IEEE Trans. On Wireless Communications*, vol. 7, no. 6, pp. 2131-2142, June 2008.
- [54] H. Kawai, K. Higuchi, N. Maeda, and M. Sawahashi, "Adaptive control of surviving symbol replica candidates in QRM-MLD for OFDM-MIMO multiplexing," *IEEE Journal of Selected Areas in Communications*, vol. 24, no. 6, pp. 1130-1140, Jun. 2006.
- [55] F. Hasegawa, J. Luo, K.R. Pattipati, P. Willett, and D. Pham, "Speed and accuracy comparison of techniques for multiuser detection in synchronous CDMA", *IEEE Transactions on Communications*, vol. 54, no. 4, pp. 540–545, April 2004.
- [56] C. Windpassinger, L. Lampe, R. F. H. Fischer, and T. Hehn, "A performance study of MIMO detectors", *IEEE Transactions on Wireless Communications*, vol.5, no. 8, pp. 2004–2008, August 2006.
- [57] G. H. Golub and C. F. Van Loan, "Matrix Computations", Johns Hopkins University Press, Baltimore, MD, USA, 3rd edition, 1996.

- [58] M. O. Damen, K. A. Meraim and S. Burykh, "Iterative QR Detection for BLAST," In the Proceedings of 38th Allerton Conference of Communications, Illinois, October 2000.
- [59] J. Jalden and B. Ottersten, "On the complexity of sphere decoding in digital communications," IEEE Transactions on Signal Processing, vol. 53, no. 4, pp. 1474-1484, April 2005.
- [60] Qingwei Li and Zhongfeng Wang, "New Sphere Decoding Architecture for MIMO Systems" School of EECE, Oregon State University, Corvallis, Oregon 97331, USA.
- [61] Yuanbin Guo and Dennis McCain, "Reduced QRD-M Detector in MIMO-OFDM Systems With Partial and Embedded Sorting" Nokia Research Center, IEEE Trans .,2005

## Appendix A

### SD Pseudocode

The pseudocode for the sphere decoding algorithm is shown in Table 1.

**Table 1: Pseudocode for sphere decoding algorithm form [34] (modifications for finite constellations,  $x_k \in \{0, \dots, Q_{\max}\}$ )**

```
function SD( $H, r$ )
1   $Q, R = qr(H)$ 
2   $w = Q^T r$ 
3   $G = R^{-1}$ 
4   $bestdist = \infty$ 
5   $k := K$ 
6   $dist_{k+1} := 0$ 
7   $e_k := Gw$ 
8   $x_k := \lfloor e_{kk} \rfloor$ 
9   $x_k := \max(x_k, 0); x_k := \min(x_k, Q_{\max})$ 
10  $y := (e_{kk} - x_k) / g_{kk}$ 
11  $step_k = \text{sgn}^*(y)$ 
12 while(1){
13    $newdist := dist_k + y^2$ 
14   if ( $newdist < bestdist \wedge (x_k \geq 0 \wedge x_k \leq Q_{\max})$ ){
15     if ( $k \neq 1$ ){
16       for ( $i = 1, \mathbf{K}, k - 1$ ){ $e_{k-1,i} := e_{ki} - yg_{ki}$ }
17        $dist_k := newdist$ 
18        $k := k - 1$ 
19        $x_k := \lfloor e_{kk} \rfloor$ 
20        $x_k := \max(x_k, 0); x_k := \min(x_k, Q_{\max})$ 
21        $y := (e_{kk} - x_k) / g_{kk}$ 
22        $step_k = \text{sgn}^*(y)$ 
23     }
24   else {
25      $\hat{x} := x$ 
26      $bestdist := newdist$ 
27      $k := k + 1$ 
28      $x_k := x_k + step_k$ 
```

```

29     stepk = -stepk - sgn*(stepk)
30     if (xk < 0 ∨ xk > Qmax) {
31         xk := xk + stepk
32         stepk = -stepk - sgn*(stepk)
33     }
34     y := (ekk - xk) / gkk
35 }
36 }
37 else {
38     if (k = K) { return x̂ }
39     else {
40         k := k + 1
41         xk := xk + stepk
42         stepk = -stepk - sgn*(stepk)
43         if (xk < 0 ∨ xk > Qmax) {
44             xk := xk + stepk
45             stepk = -stepk - sgn*(stepk)
46         }
47         y := (ekk - xk) / gkk
48     }
49 }
50 }

```