DETECTION SCHEMES IN MIMO SYSTEMS

A DISSERTATION

Submitted in partial fulfillment of the requirements for the award of the degree

of

MASTER OF TECHNOLOGY

in

ELECTRONICS AND COMMUNICATION ENGINEERING

(With Specialization in Communication Systems)

By

ANILKUMAR SISTU

DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE - 247 667 (INDIA)

JUNE, 2008

CANDIDATE'S DECLARATION

I hereby declare that the work, which is presented in this dissertation report entitled, "**Detection Schemes in MIMO Systems**" towards the partial fulfillment of the requirements for the award of the degree of **Master of Technology** with specialization in **Communication Systems**, submitted in the Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, Roorkee (India) is an authentic record of my own work carried out during the period from June 2007 to June 2008, under the guidance of **Dr. D. K. MEHRA, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee.**

I have not submitted the matter embodied in this dissertation for the award of any other Degree or Diploma.

Date: 24-06-2008.

Place: Roorkee

ANILKUMAR SISTU

CERTIFICATE

This is to certify that the above statement made by the candidate is correct to the best of my knowledge and belief.

Date:

Place: Roorkee

Dmehra

Dr. D. K. MEHRA,
Professor, E&C Department,
IIT Roorkee,
Roorkee – 247 667 (India).

ACKNOWLEDGEMENTS

I would like to extend my heartfelt gratitude to my guide, **Dr. D. K. MEHRA** for his able guidance, valuable suggestions and constant attention. It is his constant encouragement that inspired me throughout my dissertation work. I consider myself fortunate to have my dissertation done under him.

I am indebted to all my teachers who shaped and moulded me. I would like to express my deep sense of gratitude to all the authorities of Dept. of Electronics and Computer Engineering, IIT Roorkee for providing me with the valuable opportunity to carry out this work and also providing me with the best of facilities for the completion of this work.

I am very thankful to the staff of Signal Processing Lab for their constant cooperation and assistance.

I would like to say that I am indebted to my parents Saradhinaidu and Damayanthi for everything that they have given to me. They always supported me with their love, care and valuable advices. My hearty thanks to my friends Swaroop, Navven, DJ, Amit for their encouragement and moral support which lead me to a path of success in completing this dissertation. My hearty thanks to my classmates for all the support.

Last, but not the least my deepest gratitude to the Almighty God whose Divine grace provided me guidance, strength and support always.

ANILKUMAR SISTU

ABSTRACT

In today's society, a growing number of users are demanding more sophisticated services from wireless communication devices. In order to meet these rising demands, it has been proposed to increase the capacity of the wireless channel by using more than one antenna at the transmitter and receiver, thereby creating multiple-input multiple-output (MIMO) channels. Using MIMO communication techniques is a promising way to improve wireless communication technology because in a rich-scattering environment the capacity increases linearly with the number of antennas. However, increasing the number of transmit antennas also increases the complexity of detection at an exponential rate. So while MIMO channels have the potential to greatly increase the capacity of wireless communication systems, they also force a greater computational burden on the receiver. Even suboptimal MIMO detectors that have relatively low complexity have been shown to achieve unprecedented high spectral efficiency. However, their performance is far inferior to the optimal MIMO detector, meaning they require more transmit power. The fact that the optimal MIMO detector is an impractical solution due to its prohibitive complexity leaves a performance gap between sub-optimal detectors (ZF, MMSE,SIC) that require reasonable complexity and the optimal detector.

In this thesis work, the performance comparison of different MIMO detectors namely, optimal, sub-optimal and near-optimal detectors are done. The MIMO detection is then extended to iterative decoding scheme to improve the performance.

CONTENTS

LIST OF FIGURES

Chapter 1

Introduction

Wireless communications is one of the big engineering success stories of the last 20 years not only from a scientific point of view, where the progress has been phenomenal, but also in terms of market size and impact on society. In fact wireless permeates every aspect of our lives. The demands on bandwidth and spectral availability are endless as the wireless systems continue to strive for ever higher data rates. Multiple access wireless communications is being deployed for both fixed and mobile applications. In fixed applications, the wireless networks provide voice or data for fixed subscribers. Mobile networks offering voice and data services can be divided in to two classes: high mobility, to serve high speed vehicle borne users, and low mobility, to serve pedestrian users.

The gradual evolution of mobile communication systems follows the quest for high data rates, measured in bits/sec (bps) and with a high spectral efficiency, measured in bps/Hz. The first mobile communications systems were analog and are today referred to as systems of the *first generation.* In the beginning of 1990s, the first digital systems emerged, denoted as *second generation* (2G) systems, the most popular 2G system introduced was the *global system for mobile communications* (GSM)[1], which operates in the 900MHz or the 1800MHz band and supports data rates up to 22.8bbit/s. Another popular 2G system is the TDMA/136, which is also a digital cellular system. To accomplish higher data rates, two add-ons were developed for GSM, namely *high-speed circuit switched data* (HSCSD) and the *general packet radio service* (GPRS), providing data rates up to 38.4 kbit/s and 172.2 kbit/s, respectively.

The demand for yet higher data rates forced the development of a new generation of wireless systems, the so- called *third generation* (3G). 3G systems are characterized by a maximum data rate of at least 384kbit/s for mobile and 2Mbit/s for indoors.

One of the leading technologies for 3G systems is the now well-known universal mobile telephone system (UMTS)[also referred to as *wideband code-division multiplex*

(WCDMA) or ULTRA FDD/TDD].UMTS represents a revolution in terms of services and data speeds from today's "second generation" mobile networks. UMTS and WCDMA are already a reality and have been used in many parts of the world. To yield 3G data rates, an alternative approach was made with the *enhanced data rates for GSM evolution* (EDGE) concept. The EDGE system is based on GSM and operates in the same frequency bands. The significantly enhanced data rates are obtained by means of a new modulation scheme, which is more efficient than the GSM modulation scheme. As for GSM, two add-ons were developed for EDGE, namely *enhanced circuit switched data* (ECSD) and the *enhanced general packet radio service* (EGPRS). The maximum data rate of the EDGE system is 473.6kbit/s, which is accomplished by means of EGPRS. 2.5G systems, based on GPRS technology, a natural evolutionary stepping stone towards UMTS also provided faster data services.

The new IEEE and High Performance Radio Local Area Network (HIPERLAN) standards specify bit rates up to 54Mbit/s, although 24Mbit/s will be the typical rate used in most applications. Such high data rates impose large bandwidths, thus pushing carrier frequencies for values higher than the UHF band.

The goal of the next generation of wireless systems-the *fourth generation* (4G) is to provide data rates yet higher than the ones of 3G while granting the same degree of user mobility. 4G is expected to deliver more advanced versions of the same improvements provided by 3G, such as enhanced multimedia, smooth streaming video, universal access and portability across all types of devices. 4G enhancements are expected to include world wide "roaming" capability. As was projected for the ultimate 3G system, 4G might actually connect the entire globe and be operable from any location on-or above-the surface of earth. This aspect makes it distinctly different from the technologies developed until now.

In addition to 3G's technical challenges, there are problems from a financial aspect, such as justifying the large expense of building systems based on less-than-compatible 2G technologies. In contrast, 4G wireless networks that are Internet Protocol (IP)-based have an intrinsic advantage over their predecessors. IP tolerates a variety of radio protocols. It allows you to design a core network that gives you complete flexibility as to what shape the access network will take. A 4G IP network has also certain financial advantages. Equipment costs are much lower than what they used to be for 2G and 3G systems.

1.1 Multiple-Input Multiple-Output System[2]

Perhaps one of the most interesting trends in wireless communication is the proposed use of multiple input multiple output (MIMO) systems. A MIMO system uses multiple transmitter antennas and multiple receiver antennas to break a multipath channel into several individual spatial channels. Now MIMO systems represent a huge change in how wireless communication systems are designed. This change reflects how we view multipath in a wireless system.

The Prospects of MIMO

From an information theoretic perspective, increasing the number of antennas essentially allows to achieve higher spectral efficiency compared to single-input single-output (SISO) systems. Actual transmission schemes exploit this higher capacity by leveraging three types of partially contradictory gains:

- *Array gain* refers to picking up a larger share of the transmitted power at the receiver which mainly allows to *extend the range* of a communication system and to suppress interference..
- *Diversity gain* counters the effects of variations in the channel, known as fading, which *increases link-reliability and QoS.*.
- *Multiplexing gain* allows for a *linear increase in spectral efficiency and peak data rates* by transmitting multiple data streams concurrently in the same frequency band. The number of parallel streams is thereby limited by the number of transmit or receive antennas, whichever is smaller.

***The Old Perspective*:** The ultimate goal of wireless communications is to combat the distortion caused by multipath in order to approach the theoretical limit of capacity for a band-limited channel.

***The new Perspective*:** Since multipath propagation actually represents multipath channels between a transmitter and receiver, the ultimate goal of wireless communications is to use multipath to provide higher total capacity than the theoretical limit for a conventional bandlimited channel.

The basic idea is to usefully exploit the multipath rather than mitigate it, considering the multipath itself as a source of diversity that allows the parallel transmission of N independent substreams from the same user. The exploitation of diversity and parallel transmission of several data streams on different propagation paths at the same time and frequency allows for extremely large capacities compared to conventional wireless systems. The prospect of many orders of magnitude improvement in wireless communication performance at no cost of extra spectrum (only hardware and complexity are added) is largely responsible for the success of MIMO as a topic for new research. Pioneering work by Foschini [3], and Telatar [4] ignited much interest in this area by predicting remarkable spectral efficiencies for wireless systems with multiple antennas when the channel exhibits rich scattering and its variations can be accurately tracked.

The large spectral efficiencies associated with MIMO channels are based on the premise that a rich scattering environment provides independent transmission paths from each transmit antenna to each receive antenna. Therefore, for single-user systems, a transmission strategy that exploits this structure achieves capacity on approximately $\min(N,M)$ separate channels, where N is the number of transmit antennas and M is the number of receive antennas. Thus, capacity scales linearly with $\min(N,M)$ relative to a system with just one transmit and one receive antenna. This capacity increase requires a scattering environment such that the matrixes of channel gains between transmit and receive antenna pairs has full rank and independent entries, and that perfect estimates of these gains are available at the receiver.

1.1.1 Detection in MIMO Systems

Of course, the benefits of using multiple antennas at the transmitter and receiver do not come without costs. One fundamental obstacle for MIMO systems is the increased complexity of recovering the transmitted information. As the capacity increases linearly with the number of antennas, the complexity of the detection problem increases exponentially with the number of transmit antennas. Among the various popular MIMO wireless communication schemes, the BLAST (Bell Labs Layered Space Time) approaches are particularly attractive. BLAST attempts to achieve the potentially large channel capacity offered by the MIMO system. *Diagonal Bell Labs Layered Space-Time* (DBLAST) algorithm has been proposed by Foschini for this purpose, which is capable of achieving a substantial part of the MIMO capacity [5]. However, a high complexity of the algorithm implementation is its substantial drawback. A simplified version of the BLAST algorithm is known as *Vertical Bell Labs Layered Space-Time* (VBLAST). It is capable of achieving high spectral efficiency while being relatively simple to implement.

The optimal detection is performed by the maximum-likelihood (ML) detector, which finds the best symbol vector from among an exponential number of possibilities, is prohibitively complex even for small numbers of channel inputs. Suboptimal detectors can achieve the same spectral efficiency as the ML detector, but they need more transmit power to do so. In fact, the performance of MIMO detectors is measured by the amount of transmit power, or signal-to-noise ratio (SNR), they require to recover the transmitted data. The ML detector has optimal performance, but requires exponential complexity in return. Some suboptimal detectors like the zero-forcing (ZF) detector and the minimum mean square error (MMSE) detector require only linear complexity, but they cannot achieve optimal performance. This gives rise to an inherent trade-off between performance and complexity in MIMO detection.

However, recent advances in signal processing techniques have led to the development of the Sphere Decoder (SD) which is based on the enumeration of points in the search set that are located with in a sphere of some radius centered at a target. It is also called 'Lattice decoder' which offers near-ML performance for MIMO channels at an average case with polynomial time complexity. Lattice (sphere) decoders are used to simplify the exponentially complex search problem in ML decoders for MIMO systems with higher modulation constellations. Two types of lattice decoding algorithms are available in the literature, namely Fincke-Pohst algorithm [6][7] called Sphere decoder (SD)and Schnorr-Euchner algorithm[8][9] called (SE).The performance of these two algorithms are equal but they differ in the search method employed.

A new approach to solving the detection problem is created by viewing the channel output as a point in the lattice generated by the channel matrix. This approach helps the detector because the matrix that generates this lattice is not unique, and the receiver can find “better” matrices that generate the same lattice. Lattice-aided detectors achieve near- ML performance by using a lattice-reduction algorithm [10] (such as the LLL algorithm, KZ algorithm) to create a more orthogonal *effective* channel. However, finding the best lattice-reduction is in general an NP-complete problem, and the viability of lattice-aided detection is limited in practice by the high complexity of lattice-reduction algorithms. Particularly on wireless channels that vary rapidly with time, the high overhead of lattice reduction can waste much of the computational savings.

1.2 Statement of problem

This work is aimed at performance study of signal detection strategies in MIMO system.

The work is presented as follows

- Capacity of MIMO channels , MIMO detection schemes and their complexity
- Performance evaluation of iterative detection and decoding of MIMO channel.

1.3 Organization of the Report

Chapter one gives an overview of the evolution of wireless systems through 2G, 3G and 4G systems. It summarizes the problem statement for the thesis work.

Chapter two reviews the MIMO system capacity which achieves large spectral efficiencies so as to meet high bit rate demand in wireless communications, and discus the MIMO detection schemes which includes optimal detector, near-optimal detectors and sub-optimal detectors. Detailed analysis of near-optimal detectors and their complexity will also be discussed.

Chapter three discusses the application of MIMO detection to iterative decoding which improves the performance.

Chapter four presents the implementation details of different MIMO detection schemes and iterative decoding of MIMO detection. Simulation results are also included and the related issues are studied.

Chapter 2

Capacity of MIMO Channels and Detection Schemes

Multiple-input multiple-output (MIMO) systems are today regarded as one of the most promising research areas in wireless communications. This is due to the fact that a MIMO channel can offer a significant capacity gain over a traditional Single-Input Single-Output (SISO) channel. In this chapter we present the capacity of MIMO channels and we then introduce different MIMO decoding solutions. Sphere decoding algorithm and Schnorr-Euchner strategy are presented in detail.

2.1 MIMO System Model

The idea behind MIMO is that the signals on the transmit antennas at one end and the receive antennas at the other end are "combined" in such a way that the quality (bit-error rate or BER) or the data rate (bits/sec) of the communication for each MIMO user will be improved. Consider a MIMO system with a transmit array of *M* antennas and a receive array of *N* antennas. The block diagram of such a system is shown in Fig.2.1.

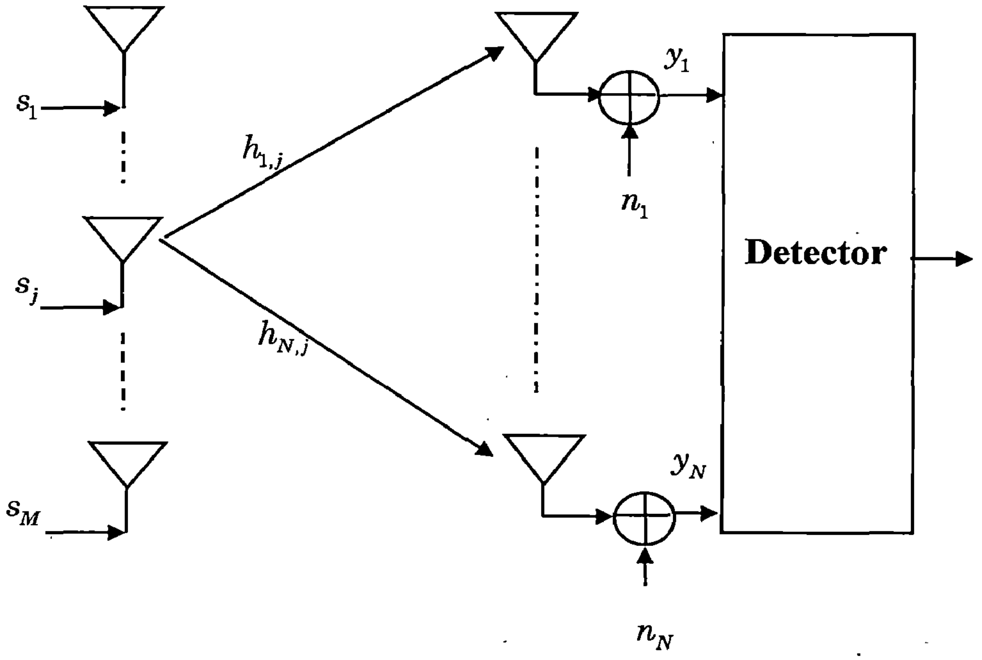


Figure 2.1: *A MIMO system model*

The MIMO channel model can be represented as

$$\tilde{\boldsymbol{y}} = \tilde{\boldsymbol{H}}\tilde{\boldsymbol{s}} + \tilde{\boldsymbol{n}} \tag{2.1}$$

where $\tilde{s} = \left[s_{1,}s_2,.........s_M\right]^T$ is the transmitted symbol vector, $\tilde{y}=[y_1,y_2,.........,y_N]^T$is the received symbol vector, and $\tilde{\boldsymbol{n}} = \left[n_1, n_2,.......n_N\right]$is an independent identically distributed (i.i.d) complex zero–mean Gaussian noise vector with variance σ^2per dimension. Moreover $\tilde{\boldsymbol{H}}$denotes the $N \times M$channel matrix ,whose elements h_{ij}represent the complex transfer functions from the jth transmit antenna to the ith receive antenna, and are all i.i.d. complex zero-mean Gaussian with variance $1\!/\!_2$per dimension . it is assumed that the channel matrix is random and that the receiver has perfect channel knowledge. It is also assumed that the channel is memoryless, i.e., for each use of the channel an independent realization of $\tilde{\boldsymbol{H}}$ is drawn.

A general entry of the channel matrix is denoted by $\{h_{ij}\}$. This represents the complex gain of the channel between the jth transmitter and the ith receiver. With a MIMO system consisting of M transmit antennas and N receive antennas, the channel matrix is written as

$$\tilde{\boldsymbol{H}} = \begin{pmatrix} h_{11} & \cdots & h_{1M} \\ \vdots & \ddots & \vdots \\ h_{N1} & \cdots & h_{NM} \end{pmatrix}$$

$$\begin{aligned} h_{ij} &= \alpha + j\beta \\ &= \sqrt{\left(\alpha^2 + \beta^2\right)}\square e^{-j\arctan(\beta/\alpha)} \\ &= \left|h_{ij}\right| \cdot e^{j\phi_{ij}} \end{aligned}$$

In a rich scattering environment with no line-of-sight (LOS), the channel gains $\left|h_{ij}\right|$ are usually Rayleigh distributed.

2.2 SISO Channel Capacity [11][12]

If the input and output of a memoryless wireless channel are the random variables $X$and Yrespectively, then the channel capacity is defined as

$$C = \max_{p(x)} I(X;Y) \tag{2.2}$$

where $I(X;Y)$ represents the mutual information between X and Y. Equation (2.2) states that the mutual information is maximized with respect to all possible transmitter statistical distributions *p*(*x*). Mutual information is a measure of the amount of information that one random variable contains about another variable. The mutual information between X and Y can also be written as

$$I(X;\ Y) = H(Y) - H(Y/X),$$

where $H(Y \mid X)$ represents the conditional entropy between the random variables X and Y. The entropy of a random variable can be described as a measure of the amount of information required on average to describe the random variable. It can also described as a measure of the uncertainty of the random variable. Note that the mutual information between X and Y depends on the properties of the channel (through a channel matrix $\tilde{\boldsymbol{H}}$) and the properties of X (through the probability distribution of X).

The ergodic (mean) capacity of a random channel with $M = 1$, and $N = 1$ and an average transmit power constraint P_T can be expressed as

$$C = E_{\tilde{H}}\left\{\max_{p(X):P<P_T} I(X;Y)\right\} \tag{2.3}$$

where P is the average power of a single channel codeword transmitted over the channel and E_H denotes the expectation over all channel realizations. The capacity of the channel is now defined as the maximum of the mutual information between the input and the output over all statistical distributions on the input that satisfy the power constraint., If each channel symbol at the transmitter is denoted by s, the average power constraint can be expressed as

$$P = E\left[|s|^2\right] \le P_T$$

Using (2.3), the ergodic (mean) capacity of a SISO system ($M = N = 1$) with a random complex channel gain h_{11} is given by

$$C = E_H\left\{\log_2\left(1 + \rho \cdot |h_{11}|^2\right)\right\} \tag{2.4}$$

where ρ is the average signal-to-noise (SNR) ratio at the receiver branch, If $|h_{11}|$ is Rayleigh, $|h_{11}|^2$ follows a chi-squared distribution with two degrees of freedom. Eq.(2.4) can then be written as

$$C = E_{\tilde{H}}\left\{\log_2\left(1+\rho\cdot\chi_2^2\right)\right\}$$

where χ_2^2 is a chi-square distributed random variable with two degrees of freedom.

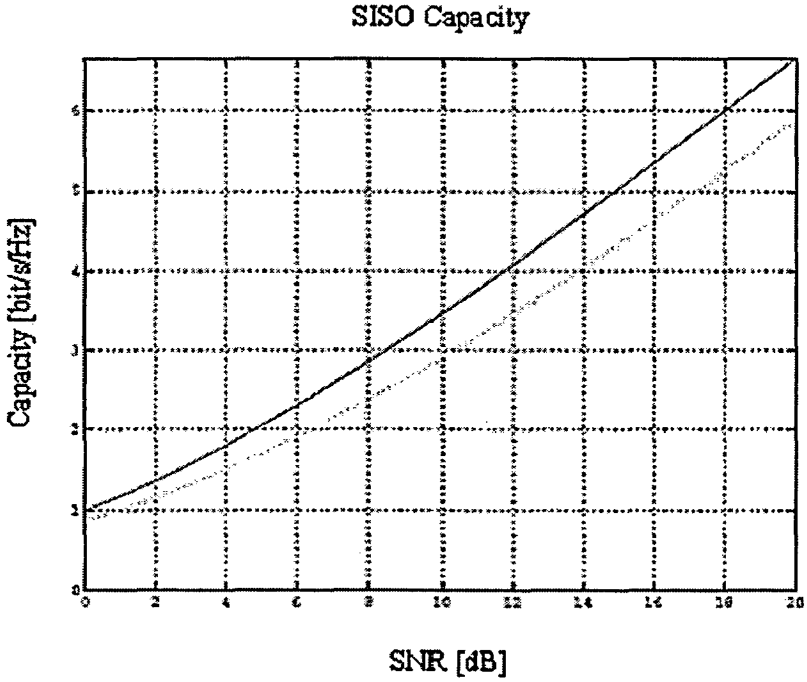


Figure 2.2: Ergodic capacity a Rayleigh fading SISO channel (dotted line) compared to the Shannon capacity of a SISO channel (solid line).

Fig.2.2 shows the Shannon capacity of a Gaussian channel (solid line) and the capacity of a Rayleigh fading channel (dotted line). The capacity of Rayleigh fading channel approaches Shannon capacity by a difference of 1to 2 dB. As we deploy more receiver antennas the statistics of capacity improve and with N receiver antennas, we have a SIMO system with capacity given by [11]

$$C = E_H\left\{\log_2\left(1+\rho\sum_{i=1}^{N}|h_i|^2\right)\right\} \tag{2.5}$$

where h_i are the entries of a column vector represents a channel matrix. Contrasting with Equation (2.4), we see that $|h_{11}|^2$ is replaced by a sum of squares. We call this system an optimum combining system (OC (*N*)).Optimum refers to taking full advantage of what the received vector tells us about the transmitted signal. The crucial feature of Eq.(2.5) is that increasing the value of *N* only results in a logarithmic increase in average capacity. Similarly, if we opt for transmit diversity, in the common case, where the transmitter does not have channel knowledge, we have a multiple-input–single-output (MISO) system with *M* transmitter antennas and the capacity is given by

$$C = \log_2\left(1 + \frac{\rho}{M}\sum_{i=1}^{M} |h_i|^2\right)$$

here h_i represents the entries of a row vector represents the channel matrix, the normalization by *M* ensures a fixed total transmitter power. Here again the capacity has a logarithmic relationship with *M*.

We next consider the use of diversity at both transmitter and receiver giving rise to a MIMO system.

2.3 MIMO Channel Capacity[11][12]

The capacity of a random MIMO channel with power constraint P_T can be expressed as

$$C = E_H\left\{\max_{p(X):tr(\Phi)\le P_T} I(\tilde{s}\,;\tilde{y})\right\} \tag{2.6}$$

where $\Phi = E\left\{\tilde{s}\tilde{s}^{\dagger}\right\}$[1] is the covariance matrix of the transmit signal vector $\tilde{s}$. Irrespective of the number of transmit antennas, the total transmit power is limited to P_T . By using Eq (2.1) and the relation between mutual information and entropy, (2.6) can be expanded as follows for a given $\tilde{\boldsymbol{H}}$

$$\begin{aligned} I(\tilde{s}\,;\tilde{y}) &= h(\tilde{y}) - h(\tilde{y}/\tilde{s}) \\ &= h(\tilde{y}) - h(\tilde{\boldsymbol{H}}\tilde{\boldsymbol{s}} + \tilde{\boldsymbol{n}}/\tilde{s}) \\ &= h(\tilde{y}) - h(\tilde{\boldsymbol{n}}/\tilde{s}) \end{aligned}$$

[1] Subscript † denotes the Hermitian transpose

$$= h(\tilde{\boldsymbol{y}}) - h(\tilde{\boldsymbol{n}}) \tag{2.7}$$

where $h(\cdot)$ in this case denotes the differential entropy of a continuous random variable. It is assumed that the transmit vector $\tilde{\boldsymbol{s}}$ and the noise vector $\tilde{\boldsymbol{n}}$ are independent. Eq. (2.7) is maximized when $\tilde{\boldsymbol{y}}$ is Gaussian, since the normal distribution maximizes the entropy for a given variance. The differential entropy of a real Gaussian vector $\tilde{\boldsymbol{y}} \in R^n$ with zero mean and covariance matrix K is equal to $\frac{1}{2}\log_2\left((2\pi e)^n \det K\right)$. For a *complex* Gaussian vector $\tilde{\boldsymbol{y}} \in \boldsymbol{C}^n$, the differential entropy is less than or equal to $\log_2 \det(\pi e K)$ with equality if and only if $\tilde{\boldsymbol{y}}$ is a circularly symmetric complex Gaussian with $K = E\{\tilde{\boldsymbol{y}}\tilde{\boldsymbol{y}}^\dagger\}$.Assuming the optimal Gaussian distribution for the transmit vector $\tilde{\boldsymbol{s}}$, the covariance matrix of the received complex vector $\boldsymbol{y}$ is given by [10]

$$\begin{aligned} E\{\tilde{\boldsymbol{y}}\tilde{\boldsymbol{y}}^\dagger\} &= E\left\{(\tilde{\boldsymbol{H}}\tilde{\boldsymbol{s}} + \tilde{\boldsymbol{n}})(\tilde{\boldsymbol{H}}\tilde{\boldsymbol{s}} + \tilde{\boldsymbol{n}})\right\} \\ &= E\left\{\tilde{\boldsymbol{H}}\tilde{\boldsymbol{s}}\tilde{\boldsymbol{s}}^\dagger \tilde{\boldsymbol{H}}^\dagger\right\} + E\left\{\tilde{\boldsymbol{n}}\tilde{\boldsymbol{n}}^\dagger\right\} \\ &= \tilde{\boldsymbol{H}}\Phi\tilde{\boldsymbol{H}}^\dagger + K^n \\ &= K^d + K^n \end{aligned} \tag{2.8}$$

where $K^d = \tilde{\boldsymbol{H}}\Phi\tilde{\boldsymbol{H}}^\dagger$, and the subscript d and n denote the desired part and noise part respectively of Eq.(2.8). The maximum mutual information of a random MIMO channel is then given by

$$\begin{aligned} I(\tilde{\boldsymbol{s}}\,;\tilde{\boldsymbol{y}}) &= h(\tilde{\boldsymbol{y}}) - h(\tilde{\boldsymbol{n}}) \\ &= \log_2\left[\det\left(\pi e\left(K^d + K^n\right)\right)\right] - \log_2\left[\det\left(\pi e K^n\right)\right] \\ &= \log_2\left[\det\left(K^d + K^n\right)\right] - \log_2\left[\det\left(K^n\right)\right] \\ &= \log_2\left[\det\left(\left(K^d + K^n\right)\left(K^n\right)^{-1}\right)\right] \\ &= \log_2\left[\det\left(K^d\left(K^n\right)^{-1} + I_N\right)\right] \\ &= \log_2\left[\det\left(\tilde{\boldsymbol{H}}\tilde{\boldsymbol{\Phi}}\tilde{\boldsymbol{H}}^\dagger\left(K^n\right)^{-1} + I_N\right)\right] \end{aligned}$$

When the transmitter has no knowledge about the channel, it is optimal to use a uniform power distribution. The transmit covariance matrix is then given by

$$\Phi = \frac{P_T}{M} I_N$$

It is also common to assume uncorrelated noise in each receiver branch described by the covariance matrix $K^n = \sigma^2 I_N$. The ergodic (mean) capacity for a complex AWGN MIMO channel can then be expressed as

$$C = E_{\tilde{\boldsymbol{H}}}\left\{\log_2\left[\det\left(I_N + \frac{P_T}{\sigma^2 M}\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^\dagger\right)\right]\right\}$$

This can also be written as

$$C = E_{\tilde{\boldsymbol{H}}}\left\{\log_2\left[\det\left(I_N + \frac{\rho}{M}\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^\dagger\right)\right]\right\} \tag{2.9}$$

where $\rho = \frac{P_T}{\sigma^2}$ is the average signal-noise ratio(SNR) at each receiver branch. By the law of large numbers, the term $\frac{1}{M}\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^t \to I_N$ as M gets large and N gets fixed. Thus the capacity in the limit of large M is

$$C = E_{\tilde{\boldsymbol{H}}}\left\{N \cdot \log_2(1+\rho)\right\} \tag{2.10}$$

Further analysis of the MIMO channel capacity given in (2.9) is possible by diagonalizing the product matrix $\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^t$ either by eigenvalue decomposition or singular value decomposition. By using eigenvalue decomposition, the matrix product is written as

$$\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^t = E\Lambda E^\dagger \tag{2.11}$$

where E is the eigenvector matrix with orthonormal columns and Λ is a diagonal matrix with the eigenvalues on the main diagonal. Using this notation (2.9) can be written as

$$C = E_{\tilde{\boldsymbol{H}}}\left\{\log_2\left[\det\left(I_N + \frac{\rho}{M}E\Lambda E^\dagger\right)\right]\right\} \tag{2.12}$$

The matrix product $\tilde{\boldsymbol{H}}\tilde{\boldsymbol{H}}^t$ can also be described by using singular value decomposition on the channel matrix $\tilde{\boldsymbol{H}}$ written as

$$\tilde{\boldsymbol{H}} = U\Sigma V^\dagger \tag{2.13}$$

where U and V are unitary matrices of left and right singular vectors respectively, and Σ is a diagonal matrix with singular values on the main diagonal. All elements on the diagonal are zero except for the first k elements. The number of non-zero singular values k equals the rank of the channel matrix. Using (2.9) and (2.13), the MIMO channel capacity can be written as

$$C = E_{\tilde{H}}\left\{\log_2\left[\det\left(I_N + \frac{\rho}{M} U\Sigma\Sigma^{\dagger} U^{\dagger}\right)\right]\right\} \tag{2.14}$$

After diagonalizing the product matrix $\tilde{H}\tilde{H}^{\dagger}$, the capacity formulas of the MIMO channel now includes unitary and diagonal matrices only. It is then easier to see that the total capacity of a MIMO channel is made up by the sum of parallel AWGN SISO subchannels. The number of parallel subchannels is determined by the rank of the channel matrix. In general, the rank of the channel matrix is given by

$$rank\left(\tilde{H}\right) = k \le \min\left(M, N\right) \tag{2.15}$$

Using (2.15) together with the fact that the determinant of unitary matrix is equal to 1, (2.12) and (2.14) can be expressed respectively as

$$C = E_{\tilde{H}}\left\{\sum_{i=1}^{k} \log_2\left(1 + \frac{\rho}{M}\lambda_i\right)\right\} \tag{2.16}$$

$$= E_{\tilde{H}}\left\{\sum_{i=1}^{k} \log_2\left(1 + \frac{\rho}{M}\sigma_i^2\right)\right\} \tag{2.17}$$

In (2.16), λ_i are the eigenvalues of the diagonal matrix Λ and in (2.18) σ_i^2 are the squared singular values of the diagonal matrix Σ. The maximum capacity of a MIMO channel is reached in the unrealistic situation when each of the M transmitted signals is received by the same set of N antennas without interference. With optimal combining at the receiver and receive diversity only (M = 1), the channel capacity
can be expressed as

$$C = E_{\tilde{H}}\left\{\log_2\left(1 + \rho \cdot \chi_{2N}^2\right)\right\}$$

where χ^2_{2N} is the chi-distributed random variable with $2N$ degrees of freedom. If there are M transmit antennas and optimal combining between N antennas at the receiver, the capacity can be written as

$$C = E_{\bar{H}}\left\{M \cdot \log_2\left(1+\frac{\rho}{M}\cdot \chi^2_{2N}\right)\right\} \tag{2.19}$$

Eq.(2.19) represents the upper bound of a Rayleigh fading MIMO channel. Fig.2.3 shows the comparison between Shannon capacity of SISO channel and the upper bound of (2.19) with $M = N = 6$ [11]. The Figure clearly shows the potential of MIMO technology.

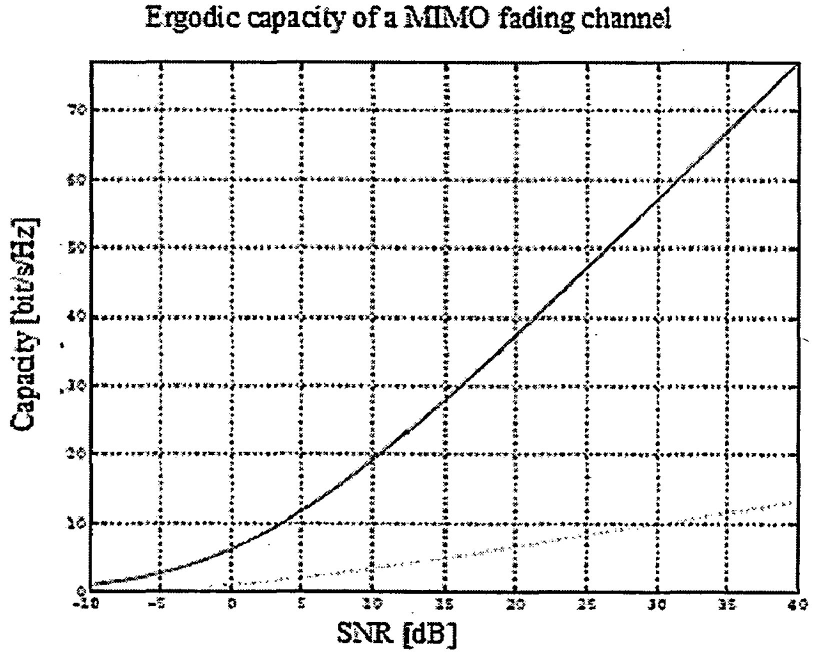


Fig.2.3. The Shannon capacity of a SISO channel (dotted line) compared to the ergodic capacity of a Rayleigh fading MIMO channel (solid line) with $M = N = 6$.

2.3.1 Antenna Selection

The capacity of the MIMO channel is reduced with a rank deficient channel matrix. A rank deficient channel matrix means that some columns in the channel matrix are linearly dependent. When they are linearly dependent, they can be expressed as a linear combination of the other columns in the matrix. The information within these columns is then in some way redundant and is not contributing to the capacity of the channel. The idea of transmit antenna selection is to improve the capacity by not using the transmit antennas that correspond to the linearly dependent columns, but instead redistributing the power among the other antennas. The optimal choice is to distribute the transmit power on a subset of k transmit antennas that maximizes the channel capacity.

2.3.2 Tradeoff between Capacity and Complexity

The capacity of MIMO systems increases linearly with $\min(M,N)$, but along with capacity ,the complexity at the receiver also increases. This gives inherent trade-off between channel capacity and complexity of MIMO detection. Linear detectors exists in the literature for MIMO detection which offers poor performance with less computational complexity. Though ML gives the optimal performance, its complexity increases exponentially as the number of antennas increase. Near-optimal detectors produce near-ML performance with reduced complexity for small constellation sizes. We discuss the detection schemes in detail in the next section.

2.4 MIMO Detection Schemes

Consider a symbol synchronized MIMO system shown in fig 2.1 with M transmit antennas and N receive antennas. The baseband equivalent model for MIMO channel is given by (2.1). For the purpose of normalization, the received power at each receive antenna is assumed to be equal to the total transmitted power. Physically, it means that we ignore signal attenuations and amplifications in the propagation process, including shadowing, antenna gain etc. Thus the SNR is equal to the ratio of total transmitted power and the noise power per receive antenna which is independent of M.

For computational reasons, we shall replace the complex-valued model (2.1) by its real-valued equivalent in the usual way. To this end, we define the $m = 2M$, dimensional vector $\boldsymbol{s}$, and the $n = 2N$ dimensional vectors $\boldsymbol{y}$ and $\boldsymbol{n}$, composed of real, imaginary parts of $\tilde{\boldsymbol{s}}, \tilde{\boldsymbol{y}}$ and $\tilde{\boldsymbol{n}}$ respectively, The real valued equivalent model of (2.1) is given by

$$\boldsymbol{y} = \boldsymbol{Hs} + \boldsymbol{n} \tag{2.20}$$

i.e

$$\begin{bmatrix} \mathcal{R}(\tilde{y}) \\ \mathcal{I}(\tilde{y}) \end{bmatrix} = \begin{bmatrix} \mathcal{R}(\tilde{H}) & -\mathcal{I}(\tilde{H}) \\ \mathcal{I}(\tilde{H}) & \mathcal{R}(\tilde{H}) \end{bmatrix} \begin{bmatrix} \mathcal{R}(\tilde{s}) \\ \mathcal{I}(\tilde{s}) \end{bmatrix} + \begin{bmatrix} \mathcal{R}(\tilde{n}) \\ \mathcal{I}(\tilde{n}) \end{bmatrix}$$

where $\mathcal{R}(\cdot)$ and $\mathcal{I}(\cdot)$ denote the real and imaginary parts $(\cdot)$ respectively . Since the elements of $\tilde{\boldsymbol{H}}$ are assumed to be i.i.d complex Gaussian , $\boldsymbol{H}$ has a full rank of $2M$. Therefore , the set $\{\boldsymbol{Hs}\}$ can be considered as the lattice $\Lambda(\boldsymbol{H})$ generated by $\boldsymbol{H}$. The rows of $\boldsymbol{H}$ are called basis vectors for $\Lambda(\boldsymbol{H})$, $2M$ is said to be the dimension of $\Lambda(\boldsymbol{H})$, and the transmitted vector $\boldsymbol{s}$ acts as the coordinates of a lattice point.

At a receiver, a detector forms an estimate of the transmitted symbol, $\hat{s}$. The optimal detector minimizes the average probability of error, i.e., it minimizes $p(\hat{s} \neq s)$ This is achieved by the maximum-likelihood (ML) design, which performs the non-linear optimization.

There are three categories of solutions to MIMO decoding, the optimal Maximum Likelihood decoder (MLD), the near-optimal sphere decoder and the sub-optimal decoder.

2.4.1 The Optimal Maximum Likelihood Decoder

Consider a linear MIMO system shown in fig 2.1 To communicate over this channel, we are faced with the task of detecting a set of $m = 2M$ transmitted symbols from a set of $n = 2N$ observed signals. Observations are corrupted by the non-ideal communication channel, typically modelled as a linear system followed by an additive noise vector. We take the transmitted symbols from a known finite alphabet Ω of size B . The detector role is to choose one of the B^m possible transmitted symbol vectors based on the available data. If $\hat{s}$ is the estimated symbol vector ,then the symbol vector whose (posterior) probability of having been sent, given the observed signal vector $\mathbf{y}$, is the largest:

$$\hat{\mathbf{s}} \square \arg\max_{s \in \Omega^m} p\ (\ \mathbf{s}\ \textit{was sent} \mid \mathbf{y}\ \textit{is observed}\ ) \tag{2.21}$$

$$= \arg\max_{s \in \Omega^m} \frac{p\ (\mathbf{y}\ \textit{is observed} \mid \mathbf{s}\ \textit{was sent})\, p(\mathbf{s}\ \textit{was sent}\ )}{p(\mathbf{y}\ \textit{is observed}\ )} \tag{2.22}$$

Equation (2.21) is known as the Maximum A posteriori Probability (MAP) detection rule. Making the standard assumption that the symbol vectors $\mathbf{s} \in B^m$ are equiprobable, i.e., that $p($ s *was sent* $)$ is constant ,the optimal MAP detection rule can be written as

$$\hat{s} \square \underset{s\in\Omega^{m}}{\arg\max}\, p\left(y \textit{ is observed} \mid s \textit{ was sent}\right) \tag{2.23}$$

A detector always returns an optimal solution satisfying (2.22) is called *Maximum Likelihood (ML)* detector. Since $\boldsymbol{n}$ is assumed as additive white Gaussian noise, the probability density function of $\boldsymbol{n}$ is

$$p\left(y \textit{ is observed} \mid \boldsymbol{s} \textit{ was sent}\right) = \frac{1}{(2\pi\sigma^{2})^{n}} e^{-\frac{1}{2\sigma^{2}}\left\|y\text{-}\boldsymbol{Hs}\right\|^{2}} \tag{2.24}$$

and consequently the maximum likelihood estimate (2.23) for $\boldsymbol{s}$ given $\boldsymbol{y}$ is

$$\hat{\boldsymbol{s}} = \underset{s\in\Omega^{m}}{\arg\max}\left(\frac{1}{(2\pi\sigma^{2})^{n}} e^{-\frac{1}{2\sigma^{2}}\left\|y\text{-}\boldsymbol{Hs}\right\|^{2}}\right)$$

$$= \underset{s\in\Omega^{m}}{\arg\min}\left(\left\|y\text{-}\boldsymbol{Hs}\right\|^{2}\right) \tag{2.25}$$

Thus, the ML detector chooses the message $\hat{s}$ which yields the smallest distance between the received vector, y , and hypothesized message, $\boldsymbol{Hs}$.

The ML detector of equation (2.25) represents a discrete non linear optimization problem over Ω^{m} candidate vectors $\hat{s} \in \Omega^{m}$. Unfortunately such problems are hard to solve and for general $\boldsymbol{y}$ and $\boldsymbol{Hs}$, the problem (2.25) is called NP-hard. However, for moderate sizes, M, there are efficient algorithms available for the solution of (2.25).

2.4.2 Sub-Optimal Decoder

Linear MIMO detection methods start by considering the input-output relation of a MIMO system in (2.20) as an unconstrained linear estimation problem, which can be solved according to a least-squares (i.e., zero-forcing (ZF)) or minimum mean squared error (MMSE) criterion. For sub-optimal detection , we consider the model presented in section (2.1)

Zero-Forcing detector[13]

ZF detection aims at a perfect separation of the parallel data streams. It solves the unconstrained least squares estimation problem to obtain

$$\hat{s} = H^{\dagger} y, \tag{2.26}$$

where $H^{\dagger}$ denotes the Moore-Penrose pseudo-inverse of the channel matrix H and is defined as

$$H^{\dagger} = \left(H^{H} H\right)^{-1} H^{H} \tag{2.27}$$

In the special case, if the number of transmitting antennas are equal to number of receiving antennas $(m = n)$the Moore-Penrose pseudo-inverse is identical to the straightforward inverse of H, which may be obtained immediately with lower complexity as

$$H^{\dagger} = H^{-1} \tag{2.28}$$

The application of (2.27) or (2.28) to (2.26) yields $\hat{s}$, whose entries will not necessarily be integers, round them of to closest integer (a process referred to as slicing) to obtain

$$\hat{s}_B = \left\lceil H^{\dagger} y \right\rfloor \tag{2.29}$$

The above $\hat{s}_B$ is called Babai estimate.

Now considering the equation (2.27) and replacing the received vector y with the model (2.20) yields,

$$\hat{s} = H^{\dagger}\left(Hs+n\right)$$

$$= I s + n_{ZF} \quad \text{where } n_{ZF} = H^{\dagger} n$$

So that the effective channel between the transmitter and slicer at the receiver now corresponds to a identity matrix I. Hence, the interference from all other parallel streams has been eliminated completely as desired. However, the drawback of the ZF estimate is that perfect separation of the transmitted data streams entails an enhancement of the additive noise, which is now given by n_{ZF}

MMSE estimator[13]

Instead of forcing the interference terms to zero, regardless of the noise, MMSE detection minimizes the *overall expected error* by taking the presence of the noise into account. The optimum tradeoff between interference cancellation and noise enhancement is achieved by setting

$$G = \left(H^H H + 2\sigma^2 I\right)^{-1} H^H \qquad (2.30)$$

$$\text{and } \hat{s} = G\,y \qquad \text{results}$$

$$\hat{s} = \tilde{H} s + n_{MMSE}$$

Where $\tilde{H}$=$G\,H$ is the effective channel after MMSE equalization, $n_{MMSE} = G\,n$.

As opposed to the ZF case, the off-diagonal elements of $\tilde{H}$ are no longer zero, which leads to the expected residual interference. However, the MMSE estimator is also a biased estimator which causes the diagonal entries of the effective channel to be smaller than one ($\tilde{H}_{i,i}$<1). The result is shrinkage of the constellation after MMSE equalization.

Successive Interference Cancellation[13]

SIC is based on the previously described linear estimation algorithms. However, a nonlinear interference cancellation stage partially exploits the knowledge that the entries of the transmitted vector s have been chosen from a finite set of constellation points. The symbols of the parallel data streams are no longer all detected at once. Instead, they are considered one after another and their contribution is subtracted (removed) from the received vector before proceeding to detect the next stream.

For the mathematical description of basic SIC algorithm without ordering we assume that the first stream is detected first, followed by the second, and so forth until the last. It is convenient to represent the channel matrix H into number of columns and rows

$$H = \begin{bmatrix} h_1 & h_2 & \cdots & h_m \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{bmatrix}$$

The SIC algorithm can be stated by the following pseudo code

```
$\boldsymbol{y}_1 = \boldsymbol{y}$
for $k = 0$ to $m-1$
    find weight vector $\boldsymbol{w}_{m-k}$
    $\hat{s}_{m-k} = slice(\boldsymbol{w}_{m-k}\boldsymbol{y}_{k+1})$
    $\boldsymbol{y}_{k+2} = \boldsymbol{y}_{k+1} - \boldsymbol{h}_{m-k}\hat{s}_{m-k}$
end
```

In the above algorithm, for each value of the index k, the entries of the auxiliary vector $\boldsymbol{y}_{k+1}$ are weighted by the components of the weight vector $\boldsymbol{w}_{n-k}$ and linearly combined to account for the effect of the interference. The weight vector $\boldsymbol{w}_{n-k}$ can be calculated from the following two cases

*ZF nulling:*In this case, interference from the yet undetected symbols is nulled.

$$\boldsymbol{H}_{m-k} = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \cdots \quad \boldsymbol{h}_{m-k}]$$

$$\boldsymbol{w}_{m-k} = \boldsymbol{H}^{\dagger}_{m-k}\boldsymbol{e}_{m-k}$$

where $\boldsymbol{H}^{\dagger}_{m-k} = \boldsymbol{H}_{m-k}\left(\boldsymbol{H}^{*}_{m-k}\boldsymbol{H}_{m-k}\right)^{-1}$ is the pseudo inverse of $\boldsymbol{H}_{m-k}$, and $\boldsymbol{e}_{m-k}$ is a $(m-k)\times 1$ column vector that consists of all zeros except for the $(m-k)$ th entry whose value is 1.

MMSE nulling: The weight vector using MMSE nulling is

$$\boldsymbol{w}_{m-k} = \left(\boldsymbol{H}^{*}_{m-k}\boldsymbol{H}_{m-k} + 2\sigma^2 I\right)\boldsymbol{h}_{m-k}$$

2.4.3 Near-optimal detectors

Near-optimal detectors gives near-ML performance with reduced complexity compared to optimal detection. Sphere(Lattice) decoders are called near-optimal detectors, which we describe below.

Sphere (Lattice) Decoders:

Sphere decoding is based on the enumeration of points in the search set that are located within a sphere of some radius centered at a target , e.g., the received signal point. the Fincke-Pohst (F-P)[5] and Schnorr-Euchner (S-E)[6] techniques are two computationally efficient means of realizing this enumeration. To avoid confusion, the lattice decoder using Fincke-Pohst strategy is called Sphere Decoder(SD), and the lattice decoder using the Schnorr-Euchner strategy is called (SE.)

Redefining the integer-least squares problem

$$\hat{\boldsymbol{s}} = \underset{\boldsymbol{s} \in D_L^m}{\arg\min} \left(\|\boldsymbol{y}\text{-}\boldsymbol{Hs}\|^2 \right) \tag{2.31}$$

where D_L^m is the m-dimensional square lattice spanned by $L-PAM$ constellation in each dimension . The above problem has a simple geometric interpretation. As the entries of $\boldsymbol{s}$ run over the points in the $L-PAM$ constellation, $\boldsymbol{s}$ spans the 'rectangular' m-dimensional lattice D_L^m .For any given *lattice- generating matrix* $\boldsymbol{H}$, the n-dimensional vector $\boldsymbol{Hs}$ spans a 'skewed' lattice. Thus given the skewed lattice $\boldsymbol{Hs}$ and the vector $\boldsymbol{y}$, the integer-least squares problem is to find "**closest**" lattice point (in Euclidean sense) to $\boldsymbol{y}$, as shown in Fig.2.4.

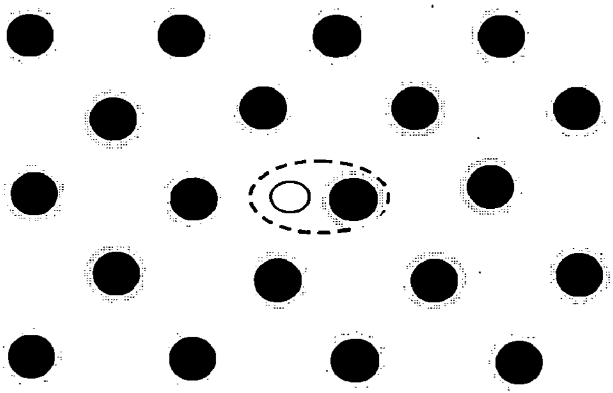


Fig. 2.4 Geometrical interpretation of the integer least-squares problem

2.4.3.1. *Sphere Decoding Algorithm (SD)[13-17]*

The basic premise in sphere decoding is rather simple: attempt to search over only lattice points $s \in D_L^m$ that lie in a certain sphere of radius d around the given received vector y, thereby reducing the search space and hence the required computational effort as shown in Fig 2.5. Clearly, the closest lattice point inside the sphere will also be the closest lattice point for the whole lattice. However, closer scrutiny of this basic idea leads to two key questions.

(1) *How to choose d ?* Clearly, if d is too large, we may obtain too many points and the search may remain exponential in size, whereas if d is too small, we may obtain no points inside the sphere.

A natural candidate for d is the *covering radius* of the lattice, defined to be the smallest radius of spheres centered at the lattice points that cover the entire space. This is clearly the smallest radius that guarantees the existence of a point inside the sphere for any vector y. The problem with this choice of d is that determining the covering radius for a given lattice is itself NP hard.

Another choice is to use d as the distance between the Babai estimate (2.29) and the vector y, i.e., $d = \left\| y - H\hat{s}_B \right\|$, since this radius guarantees the existence of at least one lattice point (here the Babai estimate) inside the sphere. However, it may happen that this choice of radius will yield too many lattice points lying inside the sphere.

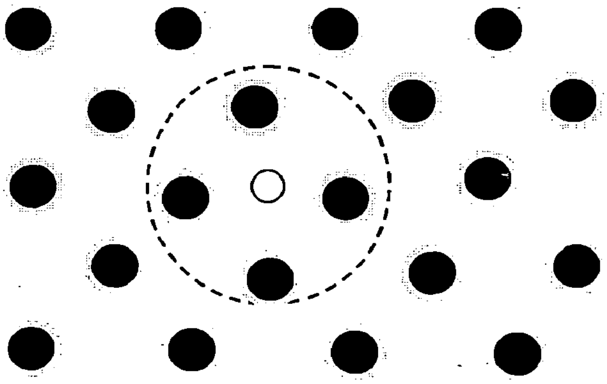


Fig. 2.5 Idea behind sphere decoder

(2) *How can we tell which lattice points are inside the sphere?* If this requires testing the distance of each lattice point from y (to determine whether it is less than d), then there is no point in sphere decoding as we shall still need an exhaustive search.

Sphere decoding does not really address the first question. However, it does propose an efficient way to answer the second one. The basic observation is the following. Although it is difficult to determine the lattice points inside a general m dimensional sphere, it is trivial to do so in the (one-dimensional) case of $m=1$.The reason is that a one-dimensional sphere reduces to the endpoints of an interval and so the desired lattice points will be the integer values that lie in this interval. We can use this observation to go from dimension k to dimension$k+1$. Suppose we have determined all k-dimensional lattice points that lie in a sphere of radius$d$. Then for any such k-dimensional point, the set of admissible values of the $k+1$-th dimensional coordinate that lie in the higher dimensional sphere of the same radius d forms an interval.

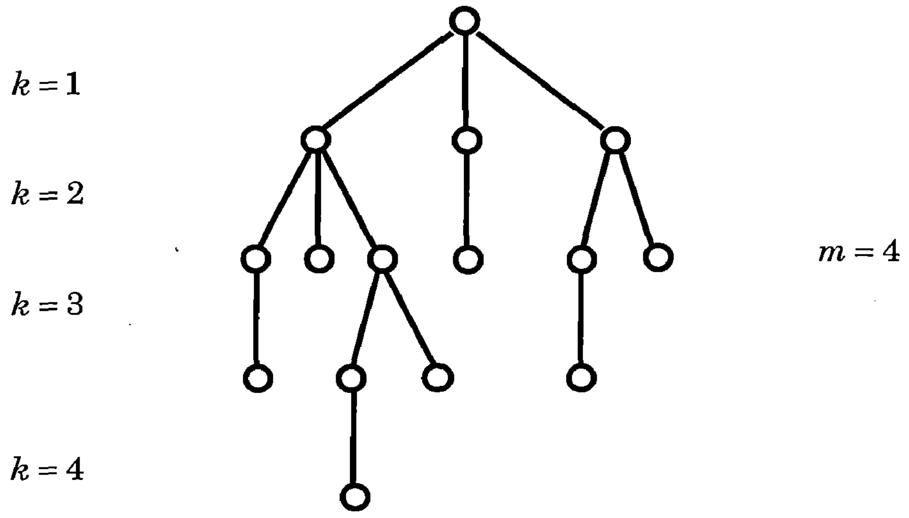


Fig.2.6. Sample tree generated to determine lattice points in a four-dimensional sphere

The above means that we can determine all lattice points in a sphere of dimension m and radius d by successively determining all lattice points in spheres of lower dimensions $1,2,....m$ and the same radius d. Such an algorithm for determining the lattice points in an m-dimensional sphere essentially constructs a tree where the branches in the kth level of the tree correspond to the lattice points inside the sphere of radius d and dimension k as shown in Fig 2.6. Moreover, the complexity of such an algorithm will depend on the *size* of the tree,i.e ,on the number of lattice points visited by the algorithm in different dimensions.

Assuming $n \geq m$, i.e, there are at least as many equations as unknowns in $\boldsymbol{y} \approx \boldsymbol{Hs}$ Note that the lattice point $\boldsymbol{Hs}$ lies inside a sphere of radius d centered at y if, and only if,

$$d^2 \geq \|y - \boldsymbol{Hs}\|^2 \tag{2.32}$$

In order to break the problem into the subproblems described above, it is useful to introduce the QR factorization of the matrix $\boldsymbol{H}$.

$$\boldsymbol{H} = Q\begin{bmatrix} R \\ 0_{(n-m)\times m} \end{bmatrix} \tag{2.33}$$

where R is an $m \times m$ upper triangular matrix and $Q = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}$ is an $n \times n$ orthogonal matrix. The matrices Q_1 and Q_2 represent the first m and last $n-m$ orthonormal columns of Q, respectively. The condition (2.32) can, therefore be written as

$$\begin{aligned} d^2 &\geq \left\| y - \begin{bmatrix} Q_1 & Q_2 \end{bmatrix}\begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2 \\ &\geq \left\| \begin{bmatrix} Q_1^* \\ Q_2^* \end{bmatrix} y - \begin{bmatrix} R \\ 0 \end{bmatrix} s \right\|^2 \\ &\geq \left\| Q_1^* y - Rs \right\|^2 + \left\| Q_2^* y \right\|^2 \end{aligned}$$

where $(\square)^*$ here denotes matrix transpose . In other words

$$d^2 - \left\| Q_2^* y \right\|^2 \geq \left\| Q_1^* y - Rs \right\|^2 \tag{2.34}$$

Defining $y = Q_1^* y$ and $d'^2 = d^2 - \left\| Q_2^* y \right\|^2$ allows us to rewrite this as

$$d'^2 \geq \sum_{i=1}^{m} \left(y_i - \sum_{j=i}^{m} r_{i,j} s_j \right)^2 \tag{2.35}$$

where $r_{i,j}$ denotes an (i,j) entry of R. Here is where the upper triangular property of R comes in handy. The right -handed side(RHS) of the above inequality can be expanded as

$$d'^2 \geq (y_m - r_{m,m}s_m)^2 + (y_{m-1} - r_{m-1,m}s_m - r_{m-1,m-1}s_{m-1})^2 + \ldots \tag{2.36}$$

where the first term depends only on s_m, the second term on $\{s_m, s_{m-1}\}$,and so on. Therefore, a necessary condition for $\boldsymbol{Hs}$ to lie inside the sphere is that

$$d'^2 \geq \left(y_m - r_{m,m}s_m\right)^2$$

This condition is equivalent to s_m belonging to the interval

$$\left\lceil \frac{-d' + y_m}{r_{m,m}} \right\rceil \leq s_m \leq \left\lfloor \frac{d' + y_m}{r_{m,m}} \right\rfloor \qquad (2.37)$$

where $\lceil \Box \rceil$ denotes rounding to the nearest larger element in the set of numbers that spans the lattice. Similarly, $\lfloor \Box \rfloor$ denotes rounding to the nearest smaller element in the set of numbers that spans the lattice. But the condition (2.37) is by no means sufficient. For every s_m satisfying (2.37), defining

$$d'^2_{m-1} = d'^2 - \left(y_m - r_{m,m}s_m\right)^2$$

and $y_{m-1/m} = y_{m-1} - r_{m-1,m}s_m$, a stronger necessary condition can be found by looking at the first two terms in (2.36), which leads to s_{m-1} belonging to the interval

$$\left\lceil \frac{-d'_{m-1} + y_{m-1/m}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \frac{d'_{m-1} + y_{m-1/m}}{r_{m-1,m-1}} \right\rfloor \qquad (2.38)$$

One can continue in a similar fashion for s_{m-2} and so on until s_1, there by obtaining all lattice points belonging to (2.32).

Sphere-Decoding Algorithm [15]:

Input: $Q, R, y, y{=}Q_1^{*}y, d$

```
1. set $k = m$, $d'^{\,2}_m = d^2 - \left\|Q_2^{*}y\right\|^2$, $y_{m/m+1} = y_m$
2. (Bounds for $s_k$) Set $UB(s_k) = \left\lfloor \frac{\left(d'_k + y_{k/k+1}\right)}{r_{k,k}} \right\rfloor$, $s_k = \left\lceil \frac{\left(-d'_k + y_{k/k+1}\right)}{r_{k,k}} \right\rceil - 1$
3. (Increase $s_k$) $s_k = s_k + 1$. If $s_k \leq UB(s_k)$, go to 5; else, go to 4.
4. (Increase $k$) $k = k + 1$; if $k = m + 1$, terminate algorithm; else, go to 3.
5. (Decrease $k$) if $k = 1$, go to 6; else $k = k - 1$,
   $y_{k/k+1} = y_k - \sum_{j=k+1}^{m} r_{k,j}s_j$, $d'^2_k = d'^2_{k+1} - \left(y_{k+1/k+2} - r_{k+1,k+1}s_{k+1}\right)^2$ and go to 2.
```

6. Solution found. Save s and its distance from y, $d_m'^2 - d_1'^2 + (y_1 - r_{1,1}s_1)^2$ and go to 3.

The subscript $k/k+1$ in $y_{k/k+1}$ above is used to denote received signal y_k adjusted with the already estimated symbol components $s_{k+1.}, \ldots\ldots s_m$. Furthermore, in steps 2 and 3 of the code , there is unit spacing between any two nearest elements of the set spanning the lattice. If the lattice is scaled, i.e if the spanning between two neighbors in the set spanning the lattice is different from 1, those algorithm steps needs to be adjusted accordingly.

Fincke-Pohst makes use of the unconstrained least-squares solution $\hat{s}=H^{\dagger}y=R^{-1}Q_1^{*}y$. In this case it follows that $\|Q_2^{*}y\|^2 = \|y\|^2 - \|H\hat{s}\|^2$, and so, inequality (2.34) becomes

$$d^2 - \|y\|^2 + \|H\hat{s}\|^2 \geq \|R(\hat{s}-s)\|^2 \tag{2.39}$$

Expanding (2.35) we can write,

$$d'^2 \geq r_{m,m}^2 (s_m - \hat{s}_m)^2 + r_{m-1,m-1}^2 \times \left(s_{m-1} - \hat{s}_{m-1} + \frac{r_{m-1,m}}{r_{m-1,m-1}} \left(s_m - \hat{s}_m \right) \right)^2 + \ldots. \tag{2.40}$$

and using (2.37) and (2.38)

$$\left\lceil \hat{s}_m - \frac{d'}{r_{m,m}} \right\rceil \leq s_m \leq \left\lfloor \hat{s}_m + \frac{d'}{r_{m,m}} \right\rfloor$$

and $$\left\lceil \hat{s}_{m-1/m} - \frac{d'_{m-1}}{r_{m-1,m-1}} \right\rceil \leq s_{m-1} \leq \left\lfloor \hat{s}_{m-1/m} + \frac{d'_{m-1}}{r_{m-1,m-1}} \right\rfloor$$ respectively ,

where we have defined $\hat{s}_{m-1/m} = \hat{s}_{m-1} - \frac{r_{m-1,m}}{r_{m-1,m-1}} \left(s_m - \hat{s}_m \right)$. We can now alternately write the algorithm as follows.

Input: $R, y, \hat{s}, d$

1. Set $k = m$, $d_m'^2 = d^2 - \|y\|^2 + \|H\hat{s}\|^2$, $\hat{s}_{m/m+1} = \hat{s}_m$
2. (Bounds for s_k) Set $z = d_k'/r_{kk}$, $UB(s_k) = \lfloor z + \hat{s}_{k/k+1} \rfloor$, $s_k = \lceil -z + \hat{s}_{k/k+1} \rceil - 1$
3. (Increase s_k) $s_k = s_k + 1$. If $s_k \leq UB(s_k)$, go to 5; else go to 4

4. (Increase k) $k = k+1$; if $k = m+1$, terminate algorithm; else go to 3
5. (Decrease k) if $k = 1$ go to 6; else $k = k-1$,

$$\hat{s}_{k/k+1} = \hat{s}_k - \sum_{j=k+1}^{m} \left({r_{k,j}} \middle/ {r_{k,k}} \right) \left(s_j - \hat{s}_j \right),$$

$$d'^2_k = d'^2_{k+1} - r^2_{k+1,k+1} \left(s_{k+1} - \hat{s}_{k+1/k+2} \right)^2 \text{ and go to 2.}$$

6. Solution found. Save s and its distance from y, $d'^2_m - d'^2_1 + r^2_{1,1}\left(s_1 - \hat{s}_{1/2}\right)^2$ and go to 3.

2.4.3.2. ***Schnorr-Euchner Strategy (SE)[9][18][19]***

This algorithm has the same principle as the SD , which means searching for the closest point inside a sphere.

Redefining the closest lattice point problem

$$\hat{\boldsymbol{s}} = \underset{s \in \Omega^m}{\arg\min} \left(\|\boldsymbol{y}\text{-}\boldsymbol{Hs}\|^2 \right)$$

where Ω is the set of real entries in the constellation , e.g $\Omega = \{-3,-1,1,3\}$ in the case of 16-QAM. This algorithm is based on two stages. The first stage consists in searching the "Babai point" (BP), which represents a first estimation, but is not necessarily, the closest point. Finding the BP gives us a bound of the error. In the second stage, we modify the BP until the closest point is reached. We oscillate in turn each component to build the closest point (unlike the sphere decoder, we don't have a minimum and maximum bound for each BP component). The time needed to finish the search for the closest point is closely related to the BP, which means related to the SNR. In fact, if the BP is very far from the closest point, i.e for low SNRs, the algorithm is slow to converge. However,if the BP is close to the closest point, i.e for high SNRs, the algorithm converges rapidly.

In SE algorithm, from the perspective of lattice, an $p = 2M$-dimensional lattice is decomposed into p k-dimensional $(k = 1,2,....p)$ sublattices. The algorithm calculates the orthogonal distance y between two points in the adjacent sublattices , and tries to find the smallest possible accumulated distance *bestdist* between the p-dimensional sublattice and the one dimensional sublattice. The basic SE algorithm is given in [9] for infinite lattice , which is not suitable to employ in MIMO systems , since the finite lattice constellation is used in MIMO systems.

Modified SE algorithm for the QAM constellations:

A SE algorithm tailored to the $q-QAM$ ($q=4,16,...$) constellation is presented in [9]. To avoid an infinite loop or an incorrect result due to finite constellation used, the algorithm [20] adopts a search method which allows an over flow . However only the lattice vector $\boldsymbol{u}$ belonging to the constellation is kept. There are probabilities in this search method where most of the elements in $\boldsymbol{u}$ belong to Ω, but remaining elements do not belong to Ω. In such a case, the lattice vector $\boldsymbol{u}$ is not kept and it has to be recalculated, which increases the algorithm complexity.

Another reduced complexity SE algorithm for MIMO systems [19] overcomes the above problem in which only those lattice points u_k $(k=1,2,...n)$ belonging to Ω are investigated and kept. The new reduced-complexity SE algorithm is sub-optimal to [20] and its pseudo code is listed below. The matrix L is the inverse and transpose of matrix R, i.e $L=R^{-T}$. The matrix R and Q are the upper triangular matrix and the orthogonal matrix in the QR-decomposition of the channel matrix $H=QR$, respectively.

Algorithm SE1: $\hat{\boldsymbol{s}} = \boldsymbol{SE1}(L,Q,\boldsymbol{y})$

```
$n = 2M$ i.e size of $L$                 /* dimension */
$bestdist = 2^{10}$                      /* current distance record */
$k = p$                                  /* dimension of examined layer */
$dist_k = 0$                             /* distance to examined layer */
$\boldsymbol{e}_k = \boldsymbol{y}^T QL$
$u_k = rint(e_{kk})$                     /* examined lattice point */
$y = (e_{kk} - u_k)/l_{kk}$
$step_k = sgn(y)$                        /* offset to next layer */
loop
$newdist = dist_k + y^2$
```

```
11. if $newdist < bestdist$ then
12.   if $k > 1$ then
13.     for $i = 1,2...k-1$ do
14.       $e_{k-1,i} = e_{k,i} - y * l_{ki}$
15.     end for
16.     $k = k - 1$                          /* move down */
17.     $dist_k = newdist$
18.     $u_k = rint(e_{kk})$                 /* closest layer */
19.     $y = (e_{kk} - u_k)/l_{kk}$
20.     $step_k = sgn(y)$
21.   else
22.     $\hat{s} = \boldsymbol{u}$           /* Best lattice point so far */
23.     $bestdist = newdist$                 /* update record */
24.     $k = k + 1$                          /* move up */
25.     $y = 2^5$
26.     for $j = 1$ to $2$ do
27.       $u_k = u_k + 2 * step_k$           /* next layer */
28.       $step_k = -step_k - sgn(step_k)$
29.       if $u_k \in constellation$ then
30.         $y = (e_{kk} - u_k)/l_{kk}$
31.         goto $\langle 10 \rangle$
32.       end if
33.     end for
34.   end if
35. else
36.   if $k == p$ then
37.     return $\hat{\boldsymbol{s}}$
38.   else
```

```
39.         goto ⟨24⟩
40.     end if
41    end if
42. end loop
```

the SE algorithm actually constructs a tree of p levels, where the branches in the kth level of the tree correspond to the lattice points in the k-dimensional sublattice. If y^2 is considered as the metric of each branch, the objective of SE is to find the path with the smallest accumulated metric *bestdist* between the first and p th level of the tree.

2.5 Complexity Analysis

2.5.1 Sphere Decoding Algorithm:

Fincke-Pohst give the complexity analysis of the SD algorithm[6]. Their main result is that the number of arithmetic operations of the aforementioned Sphere Decoding (SD) algorithms in section 2.4.3.1(excluding steps 1-3) is at most

$$\frac{1}{6}\left(2m^3+3m^2-5m\right)+\frac{1}{2}\left(m^2+12m-7\right)\times\left(\left(2\left\lfloor\sqrt{d^2t}\right\rfloor+1\right)\binom{\left\lfloor 4d^2t\right\rfloor+m-1}{\left\lfloor 4d^2t\right\rfloor}+1\right) \tag{2.41}$$

where $t=\max\left(r_{1,1}^2,....,r_{m,m}^2\right)$. In practice, t grows proportionally to n ($r_{1,1}^2$, for example is simply the squared norm of the first column of $\boldsymbol{H}$, which has n entries), and d^2 grows proportionally to m, and so the upper bound on the number of computations in (2.41) can be quite large. Although it does depend on the lattice-generating matrix $\boldsymbol{H}$ (through the quantity t), it offers little insight into the complexity of the algorithm.

Vikalo[15][22][23] evaluated the complexity of the sphere-decoding algorithm using the geometric interpretation. As mentioned earlier, the complexity of the sphere-decoding algorithm depends on the size of the generated tree in Fig. 2.6 , which is equal to the sum of the number of lattice points in spheres of radius d and the dimensions $k=1,2,...,m$. The size of this tree depends on the matrix $\boldsymbol{H}$ as well as on the vector $\boldsymbol{y}$. Therefore, unlike the complexity of solving the unconstrained least-squares problem, which only depends on m and n and not on the specific $\boldsymbol{H}$ and $\boldsymbol{y}$, the complexity of the sphere-decoding algorithm is *data dependent.*

1) *Expected Complexity:* Since the integer least-squares problem is NP hard, the worst-case complexity of sphere decoding is exponential. However, if we assume that the matrix $\boldsymbol{H}$ and the vector $\boldsymbol{y}$ are generated randomly (according to some known distributions), then the complexity of the algorithm will itself be a random variable. For any arbitrary point $\boldsymbol{y}$ and an arbitrary lattice $\boldsymbol{H}$, the expected number of lattice points inside the k-dimensional sphere of radius d is proportional to its volume and is given by[24]

$$\frac{\pi^{\frac{k}{2}}}{\Gamma\left(\frac{k}{2}+1\right)}d^k \tag{2.42}$$

Therefore, the expected total number of points visited by the sphere decoding scheme is proportional to the total number of lattice points inside the spheres of dimension $k = 1,2,...,m$

$$P = \sum_{k=1}^{m} \frac{\pi^{\frac{k}{2}}}{\Gamma\left(\frac{k}{2}+1\right)}d^k \tag{2.43}$$

A simple lower bound on P can be obtained by considering only the volume of an arbitrary intermediate dimension, say $\bar{k}$

$$P \geq \frac{\pi^{\frac{\bar{k}}{2}}}{\Gamma\left(\frac{\bar{k}}{2}+1\right)}d^{\bar{k}} \approx \left(\frac{2e\pi d^2}{\bar{k}}\right)^{\frac{\bar{k}}{2}} \frac{1}{\sqrt{\pi \bar{k}}} \tag{2.44}$$

where we have assumed $m \geq \bar{k}$ □ 1 and have used Stirling's formula for the Gamma function. Clearly, P, and its lower bound, depend on the radius d^2. This must be chosen in such a way that the probability of the sphere decoder finding a lattice point does not vanish to zero. This clearly requires the volume of the m-dimensional sphere not to tend to zero, i.e.,

$$\left(\frac{2e\pi d^2}{m}\right)^{\frac{m}{2}} \frac{1}{\sqrt{\pi m}} = \mathrm{O}(1) \qquad (2.45)$$

which for large m implies that $2e\pi d^2 = m^{1+\left(1/m\right)}$. Plugging this into the lower bound for P yields

$$P \geq \left(\frac{m^{1+\left(1/m\right)}}{\bar{k}}\right)^{\frac{\bar{k}}{2}} \frac{1}{\sqrt{\pi \bar{k}}} = \frac{1}{\sqrt{\pi}} \delta^{\frac{m}{2\delta}+\frac{1}{2}} m^{\frac{1}{2\delta}-\frac{1}{2}} \qquad (2.46)$$

where $\delta = \left(m/\bar{k}\right) > 1$.

This last expression clearly shows that the expected number of points P and, hence, the complexity of the algorithm grows exponentially in m.

A Random model:

In communications applications, however, the vector $\boldsymbol{y}$ is not arbitrary, but rather is a lattice point perturbed by additive noise with known statistical properties.

$$\boldsymbol{y} = \boldsymbol{Hs} + \boldsymbol{n}$$

where the entries of $\boldsymbol{n}$ are independent. N $(0, \sigma^2)$ random variables with known variance, and the entries of $\boldsymbol{H}$ are independent N $(0,1)$ random variables. Furthermore, $\boldsymbol{H}$ and $\boldsymbol{n}$ are mutually independent.

Choice of Radius

The first by-product of this assumption is a method to determine the desired radius d. Note that $\left(1/\sigma^2\right)\cdot\|\boldsymbol{n}\|^2 = \left(1/\sigma^2\right)\cdot\|\boldsymbol{y}\text{-}\boldsymbol{Hs}\|^2$ is a χ^2 random variable with n degrees of freedom. Thus, we may choose the radius to be a scaled variance of the noise

$$d^2 = \alpha n \sigma^2 \qquad (2.47)$$

in such a way that with a high probability, we find a lattice point inside the sphere

$$\int_0^{\frac{\alpha n}{2}} \frac{\lambda^{\frac{n}{2}-1}}{\Gamma\left(\frac{n}{2}\right)} e^{-\lambda} d\lambda = 1 - \varepsilon \qquad (2.48)$$

where the integrand is the probability density function of the χ^2 random variable with n degrees of freedom, and $1-\varepsilon$ is set to a value close to 1, say, $1-\varepsilon=0.99$.(If the point is not found, we can increase the probability $1-\varepsilon$, adjust the radius , and search again.)

Now, as mentioned earlier, the complexity of the sphere-decoding algorithm is proportional to the number of nodes visited on the tree in Fig.2.6 and, consequently, to the number of points visited in spheres of radius d and the dimensions $k=1,2....,m$. Hence, the expected complexity is proportional to the number of points in such spheres that the algorithm visits on average. Thus the expected complexity of the sphere-decoding algorithm is given by [15][22]

$$C\left(m,\sigma^2,d^2\right)=\sum_{k=1}^{m}\underbrace{\left(expected\ \#\ of\ points\ in\ k\text{-}dim\ sphere\ of\ radius\ d\right)}_{\square\ E_p\left(k,d^2=\alpha n\sigma^2\right)}\cdot\underbrace{\left(flops\ /\ point\right)}_{\square\ f_p(k)=2k+11} \tag{2.49}$$

The coefficient $f_p(k)=2k+11$ is the number of elementary operations (additions, subtractions, and multiplications) that the Fincke-Pohst algorithm performs per each visited point in dimension k. $E_p(k,d^2)$ is the expected number of points inside the $k-$dimensional sphere of radius d. Suppose that the lattice point $\boldsymbol{s}_t$ was transmitted and that the vector $\boldsymbol{y}=\boldsymbol{H}\boldsymbol{s}_t+\boldsymbol{n}$ was received. The probability that an arbitrary lattice point $\boldsymbol{s}_a$ lies in a sphere of radius d around $\boldsymbol{y}$ can be computed to be

$$\gamma\left(\frac{d^2}{\sigma^2+\|\boldsymbol{s}_a-\boldsymbol{s}_t\|^2},\frac{k}{2}\right)=\int_0^{\frac{d^2}{\sigma^2+\|\boldsymbol{s}_a-\boldsymbol{s}_t\|^2}}\frac{\lambda^{\frac{k}{2}-1}}{\Gamma(k/2)}e^{-\lambda}d\lambda$$

Note that the above probability depends only on $\|\boldsymbol{s}_a-\boldsymbol{s}_t\|^2$, i.e on the squared norm of an arbitrary lattice point in the k -dimensional lattice. Therefore

$$E_p(k,d^2)=\sum_{l=0}^{\infty}\gamma\left(\frac{d^2}{\sigma^2+l},\frac{k}{2}\right)\cdot\left(\#\ of\ lattice\ points\ with\ \|s_a-s_t\|^2=l\right)$$

$$\text{Let }\left(\#\ of\ lattice\ points\ with\ \|s_a-s_t\|^2=l\right)=r_k(n)\ ,$$

Then $r_k(n)$ is given by the coefficient of x^n in the expansion $\left(1+2\sum_{m=1}^{\infty} x^{m^2}\right)^k$

The above argument leads to the following result.

***Theorem1 (Expected complexity for infinite lattices)*:**

Under the aforementioned assumptions, the expected complexity of the sphere decoder is given by

$$C\left(m,\sigma^2\right)=\sum_{k=1}^{m}(2k+17)\sum_{l=0}^{\infty} r_k(n)\gamma\left(\frac{\alpha m\sigma^2}{\sigma^2+l},\frac{k}{2}\right) \tag{2.50}$$

where α *is such that* $\gamma(\alpha m,m)=1-\varepsilon$

It is often useful to look at the complexity exponent $\dfrac{\log C\left(m,\sigma^2\right)}{\log m}$ which approaches a constant if the expected complexity is polynomial, and grows as $\dfrac{m}{\log m}$ if it is exponential.

In communication problems, we are usually concerned with $L-PAM$ constellations

$$D_L^m=\left\{-\frac{L-1}{2},-\frac{L-3}{2},\ldots,\frac{L-3}{2},\frac{L-1}{2}\right\}^m$$

In this case, rather than noise variance σ^2, we are interested in SNR, $\rho=\dfrac{m\left(L^2-1\right)}{12\sigma^2}$.For such constellations, computing the expected complexity is more involved than for infinite lattices.

Theorem2 (Expected complexity for finite lattices):

Under the aforementioned assumptions, the expected complexity of the sphere decoder for a $2-PAM$ constellation is

$$C\left(m,\rho\right)=\sum_{k=1}^{m} f_p(k)\sum_{l=0}^{k}\binom{k}{l}\gamma\left(\frac{\alpha m}{1+\dfrac{12\rho l}{m\left(L^2-12\right)}},\frac{k}{2}\right) \tag{2.51}$$

For a $4-PAM$ constellation it is

$$\sum_{k=1}^{m} f_p(k)\sum_{l}\frac{1}{2^k}\sum_{h=0}^{k}\binom{k}{h}g_{kh}(l)\gamma\left(\frac{\alpha m}{1+\dfrac{12\rho l}{m\left(L^2-12\right)}},\frac{k}{2}\right) \tag{2.52}$$

where $g_{kh}(l)$ is the coefficient of x^l in the polynomial

$$\left(1+x+x^4+x^9\right)^h\left(1+2x+x^4\right)^{k-h}$$

The number of elementary operations per visited point in above equations

$$f_p(k)=2k+9+2L$$

2.5.2 *Schnorr-Euchner Strategy*:

Referring to SE algorithm[8-9][18-21], the algorithm complexity is given by the number of searched sublattices, i.e the number of evaluation on line $\langle 9\rangle$ in algorithm SE1(section 2.4.3.2), It is observed that the complexity becomes excessively high when signal-to-noise ratio (SNR) is low, since algorithm search oscillates too frequently among the sublttices.

Fano-Like Metric Bias:

A Fano-like metric bias can be applied for SE to alleviate the complexity problem mentioned above. With this fano-like metric bias, the branches in higher levels should have a larger metric bias than those in lower levels, reflecting the fact that they are far away from the end of the tree and hence less likely to be part of the smallest path.

From (2.20) and (2.25), the average value of the smallest path is

$$E\left\{\|\boldsymbol{y}\text{-}\boldsymbol{Hs}\|^2\right\}=E\left\{\|\boldsymbol{n}\|^2\right\}=\alpha n\sigma^2 \tag{2.42}$$

It is therefore reasonable to choose $\alpha\sigma^2$ as the metric bias for one level of the tree, where $0\le\alpha\le1$ is a constant. The metric bias for the k-level tree is simply the sum of the biases for the following i-level $(i=1,2,3,\ldots,k-1)$ trees. Moreover, the squared orthogonal distance y^2 is expressed as a portion of l_{kk} for k-dimensional sublattice [9], the Fano-like metric bias for SE is thus to be:

$$F_k=F_{k-1}+\alpha\sigma^2 l_{k-1,k-1} \qquad (k=2,3,\ldots,n)$$

where F_1 □ 0. With this metric bias, the SE can be modified as below

Algorithm SE_FM: $\hat{s} = \text{SE\_FM}\ (L, Q, y, F)$

- In Algorithm SE1 (section 2.4.3.2), replace Line $\langle 11 \rangle$ with :

```
$fanodist = newdist + F_k$
**if** $fanodist < bestdist$ **then**
```

Early Termination:

The eq. (2.42) also implies that the loop in SE can be terminated early as soon as the currently small distance *bestdist* is smaller than a pre-calculated distance

$$D = \beta n \sigma^2$$

where β is a noise level dependent constant that needs to be estimated for each SNR point. The algorithm complexity thus can be further reduced. The pre-calculated distance D is simply the average value of the smallest distance *bestdist* at each SNR point, which can be determined by simulating the algorithm in [18]. With this early termination criteria, the SE1 is modified as [19]

Algorithm SE_ET: $\hat{s} = \text{SE\_ET}\ (L, Q, y, D)$

- In Algorithm SE1 (section 2.4.3.2), insert the following lines between Line $\langle 23 \rangle$ and $\langle 24 \rangle$

```
**If** $bestdist < D$ **then return** $\hat{s}$
**end if**
```

Furthermore, The algorithm SE can be improved to be SE2 which combines the algorithm with Fano-like metric bias and early termination.

2.6 Comparison of the SD and the SE[20-21]

Both SD and SE are ML decoders, which enables us to conclude that the two algorithms perform well. The two algorithms have the same principle, the search for the closest point, but differ mainly in the search method. In the following we will compare the complexities of these two algorithms.

Since the multiplications are the most expensive operations in terms of machine cycles compared to addition and comparison, only multiplications will be taken in account to measure the complexity. The complexity of the algorithm is defined by the number of multiplications carried out until convergence.

However both algorithms, before attacking the closest point searching phase, need a preparation phase, which we will qualify by pre-decoding phase, and also an initialization phase. To study the complexity of both algorithms, it is worth studying and comparing first their respective pre-decoding and initialization phases, and subsequently their respective closest point searching method. Finally we will compare their respective total complexity [20].

2.6.1Comparison of pre-decoding and initialization phases[20-21]

As shown in the flow-charts of the SD and the SE, in pre-decoding and initialization phases we have essentially two operations : the first one consists in the calculation of a triangular form of the matrix $\boldsymbol{H}$.. For that we can use either QR decomposition or Cholesky decomposition. The second one consists in the calculation of the Zero Forcing point (ZF). When using QR decomposition, we decompose $\boldsymbol{H}$, and then we define $G = R^T$, where G is a lower triangular matrix, this needs $\frac{2}{3}N^3$ operations. When using Cholesky decomposition, we have first to calculate the Gram matrix of $\boldsymbol{H}$, defined as $Gram = \boldsymbol{HH}^T$, and so we decompose $Gram$ to obtain $Gram = U^TU$, where U is an upper triangular matrix, and then $G = U^T$. The total number of operations needed is $\frac{7}{6}N^3 + N$. By comparing the number of operations needed for each decomposition, we remark immediately that the QR decomposition is less expensive in terms of operations.

For second operation, we need first to calculate the inverse of the transfer matrix of the channel $\boldsymbol{H}$ .The Zero Forcing point is defined by the equation (2.29), which is also called the " Babai point" .

$$\hat{s} = \boldsymbol{H}^{\dagger} y$$

For the SD, we will use matrix G to build matrix Q. Using the matrix Q and the ZF point we calculate the minimum and maximum bound of each closest point component. For the SE also, we do not need the matrix G but its inverse $L = G^{-1}$. For SE, the ZF point represents the first point found by the algorithm which will be adjusted to obtain the closest point.

The total number of multiplications necessary to carry out the pre-decoding and the initialization phases using QR decomposition for the SD and the SE respectively are:

$$\frac{5}{3}N^3+\frac{3}{2}N^2,\ \frac{11}{6}N^3+\frac{3}{2}N^2+\frac{1}{3}N$$

We remark immediately that the pre-decoding and initialization phases of SE are heavier than that of the SD. In fact the SE uses $\frac{1}{6}N^3+\frac{1}{3}N$ more multiplications than SD. How crucial is this disadvantage depends on the lattice dimension N.

In fact, for small lattice dimensions, the number of multiplications in the pre-decoding phase is of the same order of magnitude as that of the searching phase, so the pre-decoding phase has an influence on the total complexity of both algorithms. This influence is more significant for fast fading channels, where the pre-decoding phase are made more frequently. For large lattice dimensions, the number of multiplications in the pre-decoding phase is very small compared to those in the searching phase, and we can say that the pre-decoding phase doesn't influence the total complexity of the algorithm.

Chapter 3

Iterative Detection and Decoding

In the previous chapter, we have discussed different MIMO detection schemes and their complexity analysis. In this chapter, we present the MIMO transmission model for iterative decoding using modified sphere decoding.

3.1 Transmission Scheme:

Fig.3.1 shows the iterative decoding scheme using MIMO channel with M-transmitting antennas and N-receiving antennas. The vector of information bits $\boldsymbol{b}$ is encoded with convolutional code [25] to obtain the vector of coded bits $\boldsymbol{c}'$, which is then interleaved to result in the vector $\boldsymbol{c}$. The vector $\boldsymbol{c}$ is modulated onto a quadrature amplitude modulation (QAM)-constellation. Assume that each constellation symbol represents p_m modulation bits (e.g., for a $Q-$QAM constellation, $p_m = \log_2 Q$).Then the modulation is performed by taking blocks of vector $\boldsymbol{c}$ of length $p_m M$ and mapping them (e.g., by means of a simple Gray mapping) into $M-$dimensional symbol vectors. The resulting symbols are transmitted across the channel as given by the model

$$\boldsymbol{y} = \boldsymbol{Hs} + \boldsymbol{n} \tag{3.1}$$

Therefore, a block of $p_m M$ coded bits (corresponding to a single symbol vector) is transmitted per each channel use. Let us denote these blocks of coded bits as $\boldsymbol{c}^{[1]}, \boldsymbol{c}^{[2]}, ..., \boldsymbol{c}^{[p_c]}$. Assume that the total length of the vector $\boldsymbol{c}$ is $p_c p_m M$. Then the entire vector $\boldsymbol{c}$ can be blocked as

$$\boldsymbol{c} = \left[\boldsymbol{c}^{[1]} \quad \boldsymbol{c}^{[2]} \quad \cdots \quad \boldsymbol{c}^{[p_c]} \right] \tag{3.2}$$

and transmitted in p_c channel uses.

Consider that the k^{th} channel use (i.e., the block $\boldsymbol{c}^{[k]}$ has been modulated onto symbol vector $\boldsymbol{s}$ and transmitted across the channel). On the receiver side, the received vector $\boldsymbol{y}$ and a priori probabilities of the components of the symbol vector $\boldsymbol{s}$,

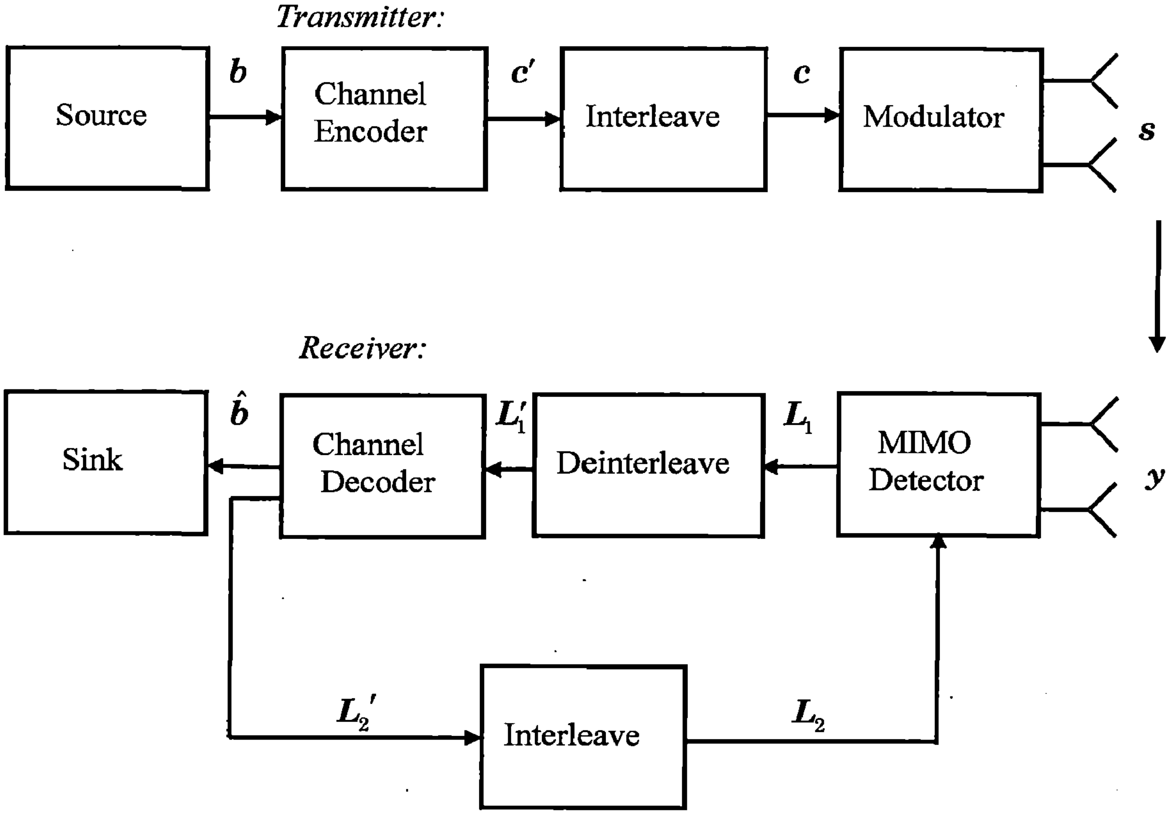


Fig.3.1 MIMO transmission and iterative receiver model

$\{p(s_1), p(s_2),..., p(s_M)\}$, are processed by an MIMO detector in order to obtain both the estimated bits in the current block $\boldsymbol{c}^{[k]}$ and the reliability information about these decisions. Let us denote bits in the block $\boldsymbol{c}^{[k]}$by c_i, $i = 1,2..., p_m M$. The reliabilities of the decisions for the coded bits $c_i$can be expressed in the form of log-likelihood ratio (LLR) as

$$L_1(c_i/\boldsymbol{y}) = \log\frac{p[c_i = +1/\boldsymbol{y}]}{p[c_i = -1/\boldsymbol{y}]} \tag{3.3}$$

(we represent logical 1 with amplitude level+1, and logical 0 with amplitude level -1). Let us denote the reliability information for the block $\boldsymbol{c}^{[k]}$ by

$$\boldsymbol{L}_1^{[k]} = \left[L_1(c_1/\boldsymbol{y}) \quad L_1(c_2/\boldsymbol{y}) \quad \cdots \quad L_1(c_{p_m M}/\boldsymbol{y})\right]$$

and let $\boldsymbol{L}_1$ denote a vector of concatenated blocks of reliabilities

$$L_1 = \left[L_1^{[1]} \quad L_1^{[2]} \quad \cdots \quad L_1^{[p_c]} \right]$$

collected over all p_c uses of the channel. Then L_1 is a vector of LLRs corresponding to all the bits in the vector c.

The vector L_1 is deinterleaved to obtain vector L_1', which is then used by a channel decoder to form the estimate of the information bit vector $\hat{b}$, as well as to provide L_2', the a posteriori reliability information for the coded bits vector c'.

A posteriori reliability information for the vector c is obtained by interleaving L_2' into L_2. Let us denote the a posteriori reliability information for the block $c^{[k]}$ by $L_2^{[k]}$. Furthermore, assume that the bits $c_i, i = 1,2....,p_m M$, in the block $c^{[k]}$ are independent. Then the posteriori probabilities for the components of the symbol vector s (symbol vector corresponding to the block $c^{[k]}$) can easily be found from $L_2^{[k]}$ using the modular mapping function. These probabilities $\{p(s_1), p(s_2),....,p(s_M)\}$ can now be used to run the MIMO detector algorithm [i.e evaluate (3.3)] once again.Hence, the MIMO detector is an iterative one, and we use the described scheme for iterative joint detection and decoding in MIMO systems.[Note that for first iteration of the MIMO detector, we assume that all symbols are equally likely]. For a simple convolutional code at encoder, the channel decoder is a simple soft-in soft-out detector ,such as Bhal, Cocke, Jelinek, and Raviv algorithm of [25][26].

The computational complexity of traditional algorithms for evaluation (3.3) can be prohibitive for applications in multi antenna systems. Since the sphere decoding algorithm of Fincke and Phost described in chapter 2 results in the ML estimate of s with reasonable complexity, a modification to the sphere decoding algorithm yields soft information with low complexity.

3.2 Modified F-P Algorithm for MAP Detection [27][28]

Redefining the MAP detection rule which maximizes the posterior probability $p_{s/y}(s/y)$

$$\max_{s\in D_L^m} p_{s/y}(s/y) \tag{3.4}$$

Using Bayes' rule

$$\begin{aligned} arg\max p_{s/y}(s/y) &= arg\max \frac{p_{y/s}(y/s)p_s(s)}{p_y(y)} \\ &= arg\max p_{y/s}(y/s)p_s(s) \end{aligned}$$

Further, by assuming that the symbols $s_1, s_2, ..., s_m$ are independent, we can write

$$p_s(s) = \prod_{k=1}^{m} p(s_k) = e^{\sum_{k=1}^{m} \log p(s_k)}$$

Then, for a known channel in additive white Gaussian noise (AWGN), (3.4) is equivalent to optimization problem

$$\min_{s\in D_L^m}\left[\|y - Hs\|^2 - \sum_{k=1}^{m} \log p(s_k)\right]. \tag{3.5}$$

For an iterative decoding scheme, we also require soft information, i.e., the probability that each bit is decoded correctly. To this end, consider the LLR defined in (3.3) and, consider the k^{th} channel use (that is , the current symbol vector s is obtained by modulating coded block $c^{[k]} = \begin{bmatrix} c_1 & c_2 & \cdots & c_{p_m M}\end{bmatrix}$ onto an $L-$PAM constellation)

$$\begin{aligned} L_1(c_i/y) &= \log \frac{p[c_i = +1/y]}{p[c_i = -1/y]} \\ &= \log \frac{p[y, c_i = +1]}{p[y, c_i = -1]} \\ &= \log \frac{\sum_{c^{[k]}:c_i=+1} p[y/c^{[k]}]p[c^{[k]}]}{\sum_{c^{[k]}:c_i=-1} p[y/c^{[k]}]p[c^{[k]}]} \end{aligned} \tag{3.6}$$

Assuming independent bits $c_1, c_2, ..., c_{p_m M}$, (3.6) becomes

$$L_1(c_i/y) = \underbrace{\log \frac{p[c_i = +1]}{p[c_i = -1]}}_{L_{1a}(c_i)} + \underbrace{\log \frac{\sum_{c^{[k]}:c_i=+1} p[y/c^{[k]}]\prod_{j,j\neq i} p[c_j]}{\sum_{c^{[k]}:c_i=-1} p[y/c^{[k]}]\prod_{j,j\neq i} p[c_j]}}_{L_{1e}(c_i)}$$

where $L_{1a}(c_i)$ and $L_{1e}(c_i)$ denote so-called *a priori* and *extrinsic* parts of the total soft information, respectively. [Note that, when used in an iterative decoding scheme, it is only $L_{1e}(c_i)$ that is passed to the other decoding block(s) in the scheme.] Since the block $c^{[k]}$ is uniquely mapped into the symbol vector s, it follows that for an AWGN channel

$$L_1(c_i/y) = \log\frac{\sum_{s:c_i=+1} p[y/s]\prod_j p[s_j]}{\sum_{s:c_i=-1} p[y/s]\prod_j p[s_j]}$$

$$= \log\frac{\sum_{s:c_i=+1} e^{-\|y-Hs\|^2+\sum_j \log p[s_j]}}{\sum_{s:c_i=-1} e^{-\|y-Hs\|^2+\sum_j \log p[s_j]}} \tag{3.7}$$

Computing (3.7) over the entire signal space D_L^m is of prohibitive complexity. Instead, we constrain ourselves to those $s \in D_L^m$ for which the argument in (3.5) is small. [Note that these are the signal vectors whose contribution to the numerator and denominator in (3.7) is significant.]

Applying the idea of the Fincke-pohst algorithm, we search for the points s that belong to the geometric body described by

$$r^2 \geq (s-\hat{s})^* R^* R(s-\hat{s}) - \sum_{k=1}^{n} \log p(s_k) \tag{3.8}$$

where R is the lower triangular matrix obtained from the QR factorization of H. (Note that this is no longer a hypersphere.) The search radius r in (3.8) can be chosen according to the statistical properties of the noise and the a priori distribution of s.

A necessary condition for s_m to satisfy (3.8) follows:

$$r_{mm}^2(s_m - \hat{s}_m)^2 - \log p(s_m) \leq r^2 \tag{3.9}$$

Moreover, for every s_m satisfying (3.9), we define

$$r_{m-1}^2 = r^2 - r_{mm}^2(s_m - \hat{s}_m)^2 + \log p(s_m)$$

and obtain a stronger necessary condition for (3.8) to hold

$$r_{m-1,m-1}^2\left(s_{m-1} - \underbrace{\hat{s}_{m-1} + \frac{r_{m-1,m}}{r_{m-1,m-1}}(s_m - \hat{s}_m)}_{\hat{s}_{m-1/m}}\right)^2 - \log p(s_{m-1}) \leq r_{m-1}^2$$

The procedure continues until all the components of vector s are found. The FP-MAP algorithm can be summarized as follows

Input: $R, \boldsymbol{y}, \hat{\boldsymbol{s}}, r, p_s(\boldsymbol{s})$

1. Set $k = m$, $r_m'^2 = r^2 - \|\boldsymbol{y}\|^2 + \|\boldsymbol{H}\hat{\boldsymbol{s}}\|^2$, $\hat{s}_{m/m+1} = \hat{s}_m$
2. (Bounds for s_k) Set $z = {r_k'}/{r_{k,k}}$, $UB(s_k) = \lfloor z + \hat{s}_{k/k+1} \rfloor$, $s_k = \lceil -z + \hat{s}_{k/k+1} \rceil - 1$
3. (Increase s_k) $s_k = s_k + 1$. If $r_{kk}^2(s_k - \hat{s}_{k/k+1})^2 > r_k'^2 + \log p(s_k)$ and $s_k \leq UB(s_k)$, go to (3), else proceed. If $s_k \leq UB(s_k)$ go to (5), else go to (4)
4. (Increase k) $k = k+1$; if $k = m+1$, terminate algorithm; else go to (3)
5. (Decrease k) if $k = 1$ go to (6). Else $k = k-1$,

   $$\hat{s}_{k/k+1} = \hat{s}_k - \sum_{j=k+1}^{m} \left({r_{k,j}}/{r_{k,k}} \right) \left(s_j - \hat{s}_j \right),$$

   $$r_k'^2 = r_{k+1}'^2 - r_{k+1,k+1}^2 \left(s_{k+1} - \hat{s}_{k+1/k+2} \right)^2 + \log p(s_{k+1}), \text{ and go to 2.}$$
6. Solution found. Save $\boldsymbol{s}$ and go to (3).

Assume that the search yields the set of points $\mathcal{S} = \left\{ \boldsymbol{s}^{(1)}, \boldsymbol{s}^{(2)}, \ldots, \boldsymbol{s}^{(l_s)} \right\}$. The vector $\boldsymbol{s} \in \mathcal{S}$ that minimizes (3.5) is the solution to the MAP detection problem (3.4).The soft information for each bit c_i can be estimated from (3.7), by only summing the terms in the numerator and denominator such that $\boldsymbol{s} \in \mathcal{S}$. Now (3.7) can be approximated using Max-Log algorithm as

$$L_1(c_i/\boldsymbol{y}) = \max_{s:c_i=+1} \left(-\|\boldsymbol{y} - \boldsymbol{Hs}\|^2 + \sum\nolimits_j \log p\left[s_j\right] \right) - \max_{s:c_i=-1} \left(-\|\boldsymbol{y} - \boldsymbol{Hs}\|^2 + \sum\nolimits_j \log p\left[s_j\right] \right)$$

3.3 Channel Decoder

The ouput of MIMO detector is deinterleaved and given to a channel decoder to form the estimate of the information bit vector, as well as to provide the a posteriori reliability information of coded bits. The channel decoder is a soft-in soft-out decoder (BCJR algorithm) and is described below .In the following, symbols in ( ) refer to Fig.3.1

BCJR Algorithm [25]

Let $\boldsymbol{u}$ $(\boldsymbol{b})$ be the information sequence and $\boldsymbol{v}$ $(\boldsymbol{c}')$ be the coded information sequence and $\boldsymbol{r}$ $\left(\boldsymbol{L}_1^1\right)$ be the received sequence. Then BCJR algorithm calculates the a posteriori L-values

$$L(u_l) = \ln\left[\frac{p(u_l = +1/r)}{p(u_l = -1/r)}\right] \tag{3.10}$$

(Note: from the Fig.3.1, $L_2' = L(u_l)$)

called the APP L – values, of each information bit, and the decoder output is given by

$$u_l = \begin{cases} +1 if\ L(u_l) > 0 \\ -1 if\ L(u_l) < 0 \end{cases}, l = 0,1,\ldots,h-1. \tag{3.11}$$

Rewriting the APP value $p(u_l = +1/r)$ as follows

$$p(u_l = +1/r) = \frac{p(u_l = +1, r)}{p(r)} = \frac{\sum_{u \in U_l^+} p(r/v)p(u)}{\sum_u p(r/v)p(u)} \tag{3.12}$$

where U_l^+ is the set of all information sequences $\boldsymbol{u}$ such that $u_l = +1$, $\boldsymbol{v}$ is the transmitted codeword corresponding to the information sequence $\boldsymbol{u}$, and $p(r/v)$ is the pdf of the received sequence $\boldsymbol{r}$ given $\boldsymbol{v}$. Rewriting $p(u_l = -1/r)$ in the same way, we can write the expression in (3.10) for the APP L-values as

$$L(u_l) = \ln\left[\frac{\sum_{u \in U_l^+} p(r/v)p(u)}{\sum_{u \in U_l^-} p(r/v)p(u)}\right] \tag{3.13}$$

where U_l^- is the set of all information sequence $\boldsymbol{u}$ such that $u_l = -1$. MAP decoding can be achieved by computing the APP L -values $L(u_l), l = 0,1,\cdots,h-1$ directly from eq.(3.13) and then applying (3.11); however, except for short block lengths h, the amount of computation required is prohibitive. For codes with trellis structure and a reasonable number of states, such as short constraint length convolutional codes, eq. (3.13) can be simplified by making use of the trellis structure of the code as

$$p(u_l = +1/r) = \frac{p(u_l = +1, r)}{p(r)} = \frac{\sum_{(s',s) \in \Sigma_l^+} p(s_l = s', s_{l+1} = s, r)}{p(r)} \tag{3.14}$$

where Σ_l^+ is the set of all state pairs $s_l = s'$ and $s_{l+1} = s$ that corresponds to the input bit $u_l = -1$ at time l. Reformulating the expression $p(u_l = -1/r)$ in the same way, we can now write (3.10) for the APP L – values as

$$L(u_l) = \ln\left\{\frac{\sum_{(s',s)\in\Sigma_l^+} p(s_l = s', s_{l+1} = s, \boldsymbol{r})}{\sum_{(s',s)\in\Sigma_l^-} p(s_l = s', s_{l+1} = s, \boldsymbol{r})}\right\}, \tag{3.15}$$

where Σ_l^- is the set of all state pairs $s_l = s'$ and $s_{l+1} = s$ that correspond to the input bit $u_l = -1$ at time l.. Equations (3.13) and (3.15) are equivalent expressions for APP $L-$ value $L(u_l)$, but whereas the summation in (3.13) extend over a set of 2^{h-1} information sequences, the summations in (3.15) extend only over a set of 2^{υ} state pairs. Hence for large block lengths h, (3.15) is considerably simpler to evaluate.

The joint pdf's $p(s',s,\boldsymbol{r})$ in (3.15) can be evaluated recursively as follows

$$p(s',s,\boldsymbol{r}) = p(s',s,\boldsymbol{r}_{t<l},\boldsymbol{r}_l,\boldsymbol{r}_{t>l}), \tag{3.16}$$

where $\boldsymbol{r}_{t<l}$ represents the portion of received sequence $\boldsymbol{r}$ before time l, and $\boldsymbol{r}_{t>l}$ represents the portion of the received sequence $\boldsymbol{r}$ after time l, application of Baye's rule yields

$$\begin{aligned} p(s',s,\boldsymbol{r}) &= p(\boldsymbol{r}_{t>l}/s',s,\boldsymbol{r}_{t<l},\boldsymbol{r}_l)\,p(s',s,\boldsymbol{r}_{t<l},\boldsymbol{r}_l) \\ &= p(\boldsymbol{r}_{t>l}/s',s,\boldsymbol{r}_{t<l},\boldsymbol{r}_l)\,p(s,\boldsymbol{r}_l/s',\boldsymbol{r}_{t<l})\,p(s',\boldsymbol{r}_{t<l}) \\ &= p(\boldsymbol{r}_{t>l}/s)\,p(s,\boldsymbol{r}_l/s')\,p(s',\boldsymbol{r}_{t<l}) \end{aligned} \tag{3.17}$$

where the last equality follows from the fact that the probability of the received branch at time l depends only on the state and input bit at time l. Defining

$$\alpha_l(s') \equiv p(s',\boldsymbol{r}_{t<l}) \tag{3.18a}$$

$$\gamma_l(s',s) \equiv p(s,\boldsymbol{r}_l/s') \tag{3.18b}$$

$$\beta_{l+1}(s) = p(\boldsymbol{r}_{t>l}/s), \tag{3.18c}$$

we can write (3.17) as

$$p(s',s,\boldsymbol{r}) = \beta_{l+1}(s)\gamma_l(s',s)\alpha_l(s'). \tag{3.19}$$

We can now rewrite the expression for the probability $\alpha_{l+1}(s)$ as

$$\alpha_{l+1}(s) = p(s,\boldsymbol{r}_{t<l+1}) = \sum_{s'\in\sigma_l} p(s',s,\boldsymbol{r}_{t<l+1})$$

$$
\begin{aligned}
&= \sum_{s'\in\sigma_l} p\left(s,\boldsymbol{r}_l/s',\boldsymbol{r}_{t<l}\right)p\left(s',\boldsymbol{r}_{t<l}\right) \\
&= \sum_{s'\in\sigma_l} p\left(s,\boldsymbol{r}_l/s'\right)p\left(s',\boldsymbol{r}_{t<l}\right) \\
&= \sum_{s'\in\sigma_l} \gamma_l\left(s',s\right)\alpha_l\left(s'\right),
\end{aligned}
\tag{3.20}
$$

where σ_l s the set of all states at time l. Thus we can compute a *forward metric* $\alpha_{l+1}\left(s\right)$ for each state s at time $l+1$ using the forward recursion (3.20). Similarly we can write the expression for the probability $\beta_l\left(s'\right)$

$$
\beta_l\left(s'\right) = \sum_{s\in\sigma_{l+1}} \gamma_l\left(s',s\right)\beta_{l+1}\left(s\right)
\tag{3.21}
$$

where σ_{l+1} is the set of all states at time $l+1$, and we can compute a *backward metric* $\beta_l\left(s'\right)$ for each state s' at time l using the *backward recursion* (3.21). The forward recursion begins at time $l=0$ with initial condition

$$
\alpha_0\left(s\right) = \begin{cases} 1, s = \boldsymbol{0} \\ 0, s \neq \boldsymbol{0} \end{cases}
\tag{3.22}
$$

Since the encoder starts in all-zero state $S_0 = \boldsymbol{0}$, and we use (3.20) to recursively compute $\alpha_{l+1}\left(s\right), l = 0,1,\cdots,K-1$, where $K = h+m$ is the length of the input sequence. Similarly, the backward recursion begins at time $l = K$ with initial condition

$$
\beta_K\left(s\right) = \begin{cases} 1, s = \boldsymbol{0} \\ 0, s \neq \boldsymbol{0} \end{cases},
\tag{3.23}
$$

since the encoder also ends in the all-zero state $S_0 = \boldsymbol{0}$, and we use (3.21) to recursively compute $\beta_l\left(s\right), l = K-1, K-2,\cdots,0$.

We can write the *branch metric* $\gamma_l\left(s',s\right)$ as

$$
\begin{aligned}
\gamma_l\left(s',s\right) = p\left(s,\boldsymbol{r}_l/s'\right) &= \frac{p\left(s',s,\boldsymbol{r}_l\right)}{p\left(s'\right)} \\
&= \left[\frac{p\left(s',s\right)}{p\left(s'\right)}\right]\left[\frac{p\left(s',s,\boldsymbol{r}_l\right)}{p\left(s',s\right)}\right] \\
p\left(s/s'\right)p\left(\boldsymbol{r}_l/s',s\right) &= p\left(u_l\right)p\left(\boldsymbol{r}_l/\boldsymbol{v}_l\right)
\end{aligned}
\tag{3.24}
$$

where u_l is the input bit and v_l the output bits corresponding to the state transition $s' \rightarrow s$ at time l. For a continuous output AWGN channel, if $s' \rightarrow s$ is a valid state transition,

$$\gamma_l(s',s) = p(u_l)p(r_l/v_l) = p(u_l)\left(\sqrt{\frac{E_s}{\pi N_0}}\right)^n e^{-\frac{E_s}{N_0}\|r_l - v_l\|^2}, \tag{3.25}$$

where $\|r_l - v_l\|^2$ is the squared Euclidean distance between the (normalized by $\sqrt{E_s}$) received branch r_l and the transmitted branch v_l at time l; however , if $s' \rightarrow s$ is not a valid state transition, $p(s/s')$ and $\gamma_l(s',s)$ are both zero. The algorithm that computes the APP $L-$value $L(u_l)$ using (3.15),(3.19),and the metrics defined in (3.20)-(3.23) and (3.25) is called MAP algorithm.

We introduce some modifications to above algorithm that result in greater computational efficiency. First, we note from (3.19)-(3.21) and (3.25) that the constant term $\left(\sqrt{\frac{E_s}{\pi N_0}}\right)^n$ always appears raised to the power h in the expression for the pdf $p(s',s,r)$. Thus , $\left(\sqrt{\frac{E_s}{\pi N_0}}\right)^{nh}$ will be a factor of every term in the numerator and denominator summations of (3.15), and its effect will cancel. Hence, the modified branch metric

$$\gamma_l(s',s) = p(u_l)e^{-E_s/N_0\|r_l - v_l\|^2} \tag{3.26}$$

$$\begin{aligned} p(u_l = \pm 1) &= \frac{[p(u_l = +1)/p(u_l = -1)]^{\pm 1}}{\{1 + [p(u_l = +1)/p(u_l = -1)]^{\pm 1}\}} \\ &= \frac{e^{\pm L_a(u_l)}}{\{1 + e^{\pm L_a(u_l)}\}} \\ &= \frac{e^{-L_a(u_l)/2}}{\{1 + e^{-L_a(u_l)}\}} e^{u_l L_a(u_l)/2} \\ &= A_l e^{u_l L_a(u_l)/2}, \end{aligned} \tag{3.27}$$

We use (3.27) to replace $p(u_l)$ in (3.26) for $l = 0,1,2,\cdots h-1$, that is , for each information bit. For the termination bits u_l, $l = h,h+1,\cdots,h+m-1 = K-1$, however, where $p(u_l) = 1$ and $L_a(u_l) = \pm\infty$ for each valid state transition, (3.26) can be written as

$$\begin{aligned}\gamma_l(s',s) &= A_l e^{u_l L_a(u_l)/2} e^{-(E_s/N_0)\|r_l - v_l\|^2}, \\ &= A_l e^{u_l L_a(u_l)/2} e^{(2E_s/N_0)(r_l \cdot v_l) - \|r_l\|^2 - \|v_l\|^2} \\ &= A_l e^{-(\|r_l\|^2 + n)} e^{u_l L_a(u_l)/2} e^{(L_c/2)(r_l \cdot v_l)} \\ &= A_l B_l e^{(u_l L_a(u_l)/2)} e^{(L_c/2)(r_l \cdot v_l)}, l = 0,1,\cdots h-1\end{aligned} \tag{3.28a}$$

$$\begin{aligned}\gamma_l(s',s) &= p(u_l) e^{-E_s/N_0\|r_l - v_l\|^2} \\ &= e^{-(E_s/N_0)\|r_l - v_l\|^2}, \\ &= B_l e^{(L_c/2)(r_l \cdot v_l)}, l = h,h+1,\cdots K-1\end{aligned} \tag{3.28b}$$

where $B_l \equiv \|r_l\|^2 + n$ is a constant independent of the codeword u_l, and $L_c = 4E_s/N_0$ is the channel reliability factor.

From the eq.(3.19)-(3.23) and (3.28), the pdf $p(s',s,r)$ contains the factors $\prod_{l=0}^{h-1} A_l$ and $\prod_{l=0}^{K-1} B_l$ an these will be factors for every term in the numerator and denominator summations of (3.15), and their effect will be cancel. Hence,

$$\gamma_l(s',s) = e^{u_l L_a(u_l)/2} e^{(L_c/2)(r_l \cdot v_l)}, l = 0,1,\cdots h-1, \tag{3.29a}$$

$$\gamma_l(s',s) = e^{(L_c/2)(r_l \cdot v_l)}, l = h,h+1,\cdots K-1 \tag{3.29b}$$

Note that when the input bits are equally likely, $L_a(u_l) = 0$, and simplified branch metric is given by

$$\gamma_l(s',s) = e^{(L_c/2)(r_l \cdot v_l)}, l = 0,1,2,\cdots,K-1 \tag{3.30}$$

By using the following identity , we can simplify the equations (3.20),(3.21) and (3.29)

$$\max{}^*(x,y) = \ln(e^x + e^y) = \max(x,y) + \ln\left(1 + e^{-|x-y|}\right) \tag{3.31}$$

$$\gamma_l^*(s',s) = \ln\gamma_l(s',s) = \begin{cases} \dfrac{u_l L_a(u_l)}{2} + \dfrac{L_c}{2} r_l \cdot v_l, l = 0,1,\cdots h-1 \\ \dfrac{L_c}{2} r_l \cdot v_l, l = h,h+1,\cdots,K-1 \end{cases} \tag{3.32a}$$

$$\begin{aligned} \alpha_{l+1}^*(s) = \ln\alpha_{l+1}(s) &= \ln\sum_{s'\in\sigma_l}\gamma_l(s',s)\alpha_l(s') \\ &= \ln\sum_{s'\in\sigma_l} e^{\left[\gamma_l^*(s',s)+\alpha_l^*(s')\right]} \qquad (3.32\text{b}) \\ &= \max\nolimits^*_{s'\in\sigma_l}\left[\gamma_l^*(s',s)+\alpha_l^*(s')\right], l = 0,1,\cdots,K-1 \end{aligned}$$

$$\alpha_0^*(s) = \ln\alpha_0(s) = \begin{cases} 0, & s = \boldsymbol{0} \\ -\infty, & s \neq \boldsymbol{0} \end{cases} \tag{3.32c}$$

$$\begin{aligned} \beta_l^*(s') \equiv \ln\beta_l(s') &= \ln\sum_{s\in\sigma_{l+1}}\gamma_l(s',s)\beta_{l+1}(s) \\ &= \ln\sum_{s\in\sigma_{l+1}} e^{\left[\gamma_l^*(s',s)+\beta_{l+1}^*(s)\right]} \qquad (3.32\text{d}) \\ &= \max\nolimits^*_{s'\in\sigma_l}\left[\gamma_l^*(s',s)+\beta_{l+1}^*(s')\right], l = K-1,K-2,\cdots,0 \end{aligned}$$

$$\beta_K^*(s) = \ln\left(\beta_K(s)\right) = \begin{cases} 0, & s = \boldsymbol{0} \\ -\infty, & s \neq \boldsymbol{0} \end{cases} \tag{3.32e}$$

The use of $\max^*$ function in (3.32b) and (3.32d) follows from the fact that these recursive equations involve calculating the log of a sum of two exponential functions, one corresponding to each valid state transition. Further we can now write the expressions for the pdf $p(s',s,\boldsymbol{r})$ in (3.19) and the APP $L-$value $L(u_l)$ in (3.15) as

$$p(s',s,\boldsymbol{r}) = e^{\beta_{l+1}^*(s)+\gamma_l^*(s',s)+\alpha_l^*(s')} \tag{3.33}$$

and

$$L(u_l) = \ln\left\{\sum_{(s',s)\in\Sigma_l^+} e^{\beta_{l+1}^*(s)+\gamma_l^*(s',s)+\alpha_l^*(s')}\right\} - \ln\left\{\sum_{(s',s)\in\Sigma_l^-} e^{\beta_{l+1}^*(s)+\gamma_l^*(s',s)+\alpha_l^*(s')}\right\} \tag{3.34}$$

Each of two terms in eq.(3.34) involves calculating the log of a sum of 2^{υ} exponential terms, one corresponding to each state in trellis. So we can apply the $\max^*$ function defined in (3.31) to (3.34) and the APP $L-$value can be expressed as

$$L(u_l) = \max\nolimits^{*}_{(s',s)\in\Sigma_l^{+}}\left[\beta^{*}_{l+1}(s)+\gamma^{*}_{l}(s',s)+\alpha^{*}_{l}(s')\right]-\max\nolimits^{*}_{(s',s)\in\Sigma_l^{-}}\left[\beta^{*}_{l+1}(s)+\gamma^{*}_{l}(s',s)+\alpha^{*}_{l}(s')\right] \tag{3.35}$$

The decision is made according to (3.11) to get the decoded information bits.

A posteriori reliability information for the vector $\boldsymbol{c}$ is obtained by interleaving eq.(3.35) L_2' into L_2 . Let us denote the a posteriori reliability information for the block $\boldsymbol{c}^{[k]}$ by $\boldsymbol{L}_2^{[k]}$.Furthermore, assume that the bits $c_i, i = 1,2....,p_m M$, in the block $\boldsymbol{c}^{[k]}$ are independent. Then the posteriori probabilities for the components of the symbol vector $\boldsymbol{s}$ (symbol vector corresponding to the block $\boldsymbol{c}^{[k]}$) can easily be found from $\boldsymbol{L}_2^{[k]}$ by using equation (3.27) and these probabilities can be used to run MIMO detector (FP-MAP algorithm) once again.

3.4 Computational Complexity of FP-MAP Algorithm[27]

The complexity of the FP-MAP algorithm can be found following the outline of the calculation of the complexity of original Fincke-Pohst algorithm in chapter.2. However, the probability that an arbitrary point $\boldsymbol{s}_a$ belongs to a k – dimensional sphere of radius r around the transmitted point $\boldsymbol{s}_t$ (which we need to compute the expected number of points the FP-MAP algorithm visits) now becomes

$$p_{s_a} = \gamma\left(\frac{\alpha n + \dfrac{12\rho}{m(L^2-1)}\sum_{j=1}^{m}\log p(s_j)}{2\left(1+\dfrac{12\rho}{m(L^2-1)}\left\|\boldsymbol{s}_a - \boldsymbol{s}_t\right\|^2\right)}, \frac{n-m+k}{2}\right). \tag{3.36}$$

First and foremost, (3.36) is a function of the a priori probabilities, which are generally not known in advance to iterations. Second, since each point $\boldsymbol{s}_a$ in a lattice has a distinct a priori probability affiliated with it, argument of the probability function (3.36) will, in general, be different for each pair of points $(\boldsymbol{s}_t, \boldsymbol{s}_a)$. Hence, to compute the expected number of points, one needs to consider all the possible pairs of points $(\boldsymbol{s}_t, \boldsymbol{s}_a)$and the corresponding probabilities (3.36) which , as the size of the problem increases, clearly becomes rather cumbersome. However we note that since $\log p(s_j) \le 0, j = 1,2,...,m$.

we have

$$\alpha n + \frac{12\rho}{m(L^2-1)}\sum\nolimits_{j=1}^{m}\log p(s_j) \le \alpha n$$

Hence, from the complexity of original Finke-Pohst and FP-MAP, it follows that for the same choice of radius r

$$p_{s_a}^{FP-MAP} \le p_{s_a}^{FP}$$

and we conclude that, for same choice of r, the expected number of points that the FP-MAP algorithm visits is upper bounded by the expected number of points visited by the original sphere decoding algorithm. Thus, the expected complexity of the FP-MAP is roughly upper bounded by the expected complexity of the sphere decoding , for same choice of r.["Roughly" upper bounded because since the a priori probabilities enter the algorithm, there are two additional operations per each visited point; this is accounted for by changing $(2k+17)$to $(2k+19)$ in the original FP complexity.]

Theorem

The complexity of FP-MAP algorithm for a 2-PAM constellation is

$$C(m,\rho) = \sum_{k=1}^{m} f_p(k) \sum_{l=0}^{k} \binom{k}{l} \gamma\left(\frac{\alpha n + \frac{12\rho}{m(L^2-1)}\sum\nolimits_{j=1}^{m}\log p(s_j)}{1+\frac{12\rho l}{m(L^2-12)}}, \frac{n-m+k}{2} \right). \tag{3.37}$$

For a 4-PAM constellation it is

$$C(m,\rho) = \sum_{k=1}^{m} f_p(k) \sum_{l} \frac{1}{2^k} \sum_{h=0}^{k} \binom{k}{h} g_{kh}(l) \gamma\left(\frac{\alpha n + \frac{12\rho}{m(L^2-1)}\sum\nolimits_{j=1}^{m}\log p(s_j)}{1+\frac{12\rho l}{m(L^2-12)}}, \frac{n-m+k}{2} \right) \tag{3.38}$$

where $g_{kh}(l)$is the coefficient of x^l in the polynomial

$$\left(1+x+x^4+x^9\right)^h \left(1+2x+x^4\right)^{k-h}$$

Chapter 4

Simulation Results

This chapter presents the performance comparison of different signal detection strategies for MIMO systems .The simulation results for the complexity analysis of the algorithms are also included. Finally the performance results of an iterative detection and decoding for MIMO channel are presented.

4.1 Simulation results on the error performance of MIMO Detectors

As mentioned in previous chapters, there are three categories of solutions to MIMO decoding, the optimal Maximum Likelihood decoder (MLD), near-optimal sphere decoder, and the sub-optimal decoder

A MIMO system with $M = 4$ transmit and $N = 4$ receive antennas is considered in the simulation. The entries of channel matrix $\boldsymbol{H}$ are assumed to be i.i.d. zero-mean complex Gaussian variables. The channel is assumed to be known at the receiver, and 16-QAM symbols are used for transmission.

The following are the common steps and simulation parameters for all detection strategies :

Step.1 Generate a random data stream and demultiplex in to M substreams

Step.2. The demultiplexed data is mapped in to 16-QAM symbols and is modulated.

Step.3. Generate complex Gaussian channel matrix $\boldsymbol{H}$. The channel is assumed to be known at the receiver.

Step.4. Generate zero-mean complex Gaussian noise vector with variance σ^2 . Add noise to the channel impaired transmit vector $\boldsymbol{s}$ to obtain the receive vector $\boldsymbol{y}$.

- Number of transmit antennas $M = 4$, and receive antennas $N = 4$
- Modulation Scheme - $16 - QAM$
- Average energy per bit $E_b = 1$
- Variance $\sigma^2 = (ME_s/2\log_2(q))10^{-SNR/10}$, $E_s = 2(q-1)/3$, $q = 16$;($q - QAM$)

4.1.1. Sub-optimal detectors

Zero-Forcing Detector: This is a liner detector, which is basically multiplication of the spatial filter with the received vector. The spatial filter is typically the inverse of the channel matrix. The zero-forcing detector solution is

$$\hat{\boldsymbol{s}} = \boldsymbol{H}^{-1}\boldsymbol{y} \tag{4.1}$$

MMSE Detector: Taking additive noise into account, The MMSE detector produces the following estimate

$$\hat{\boldsymbol{s}} = \left(\boldsymbol{H}^H\boldsymbol{H} + 2\sigma^2\boldsymbol{I}\right)^{-1}\boldsymbol{H}^H\boldsymbol{y} \tag{4.2}$$

Successive Interference Cancellation: SIC technique is based on removing the interfering signal from the received signal, one at a time as they are nulled by nulling vectors. The nulling vectors can be derived using either ZF or MMSE criterion. The symbols of the parallel data streams are no longer all detected at once. Instead, they are considered one after another and their contribution (after slicing) is subtracted (removed) from the received vector before proceeding to detect the next stream.

SIC is performed on the received vector $\boldsymbol{y}$ as follows

It is convenient to represent the channel matrix $\boldsymbol{H}$ into number of columns and rows

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}_1 & \boldsymbol{h}_2 & \cdots & \boldsymbol{h}_m \end{bmatrix} = \begin{bmatrix} H_1 \\ H_2 \\ \vdots \\ H_n \end{bmatrix}$$

The SIC algorithm can be stated by the following pseudo code

```
$y_1 = y$
for $k = 0$ to $m-1$
find weight vector $w_{m-k}$
$\hat{s}_{m-k} = slice(w_{m-k} y_{k+1})$
$y_{k+2} = y_{k+1} - h_{m-k}\hat{s}_{m-k}$
end
```

In the above algorithm, for each value of the index k, the entries of the auxiliary vector $\boldsymbol{y}_{k+1}$ are weighted by the components of the weight vector $\boldsymbol{w}_{n-k}$ and linearly combined to account for the effect of the interference. The weight vector $\boldsymbol{w}_{n-k}$ can be calculated from the following two cases

*ZF nulling:*In this case, interference from the yet undetected symbols is nulled.

$$\boldsymbol{H}_{m-k} = [\boldsymbol{h}_1 \quad \boldsymbol{h}_2 \quad \cdots \quad \boldsymbol{h}_{m-k}]$$

$$\boldsymbol{w}_{m-k} = \boldsymbol{H}^{\dagger}_{m-k}\boldsymbol{e}_{m-k}$$

where $\boldsymbol{H}^{\dagger}_{m-k} = \boldsymbol{H}_{m-k}\left(\boldsymbol{H}^{*}_{m-k}\boldsymbol{H}_{m-k}\right)^{-1}$ is the pseudo inverse of $\boldsymbol{H}_{m-k}$, and $\boldsymbol{e}_{m-k}$ is a $(m-k)\times 1$ column vector that consists of all zeros except for the $(m-k)$ th entry whose value is 1.

MMSE nulling: The weight vector using MMSE nulling is

$$\boldsymbol{w}_{m-k} = \left(\boldsymbol{H}^{*}_{m-k}\boldsymbol{H}_{m-k} + 2\sigma^2 I\right)\boldsymbol{h}_{m-k}$$

The estimated symbol vector is demodulated and demultiplexed into binary data.

Fig.4.1 compares the BER performance between the linear detectors (ZF detector, MMSE detector) and successive interference cancellation (SIC). Among the three, ZF detector has poor BER performance due to the fact that the perfect separation of transmitted data streams entails the enhancement of the additive noise. MMSE detector performs better compared to ZF, because it minimizes the overall expected error by taking the presence of noise into account. SIC results in better performance as compared to the other two methods.

Fig.4.2 shows the performance in terms of BER for MMSE detector in case of non constant modulus modulation schemes like $16-QAM$ and $64-QAM$. Figure clearly shows the performance degradation of 15dB at 10^{-1} BER for $64-QAM$ compared to $16-QAM$.

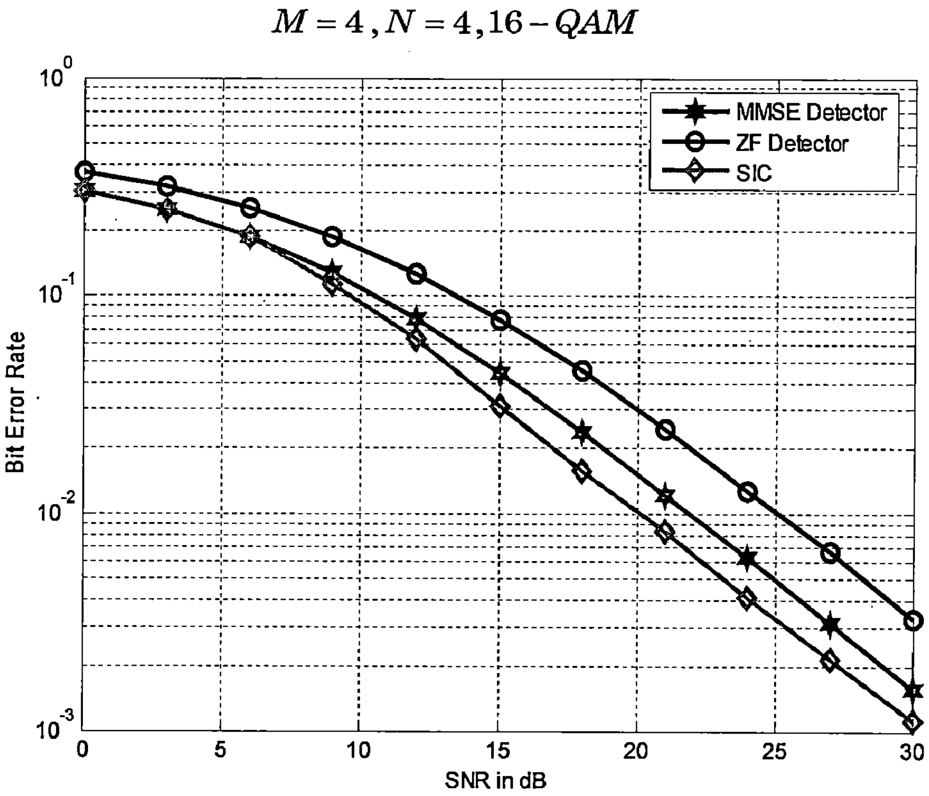


Fig.4.1. BER performance comparison of ZF, MMSE, SIC in a 4×4 *MIMO system with* $16-QAM$ *modulation*

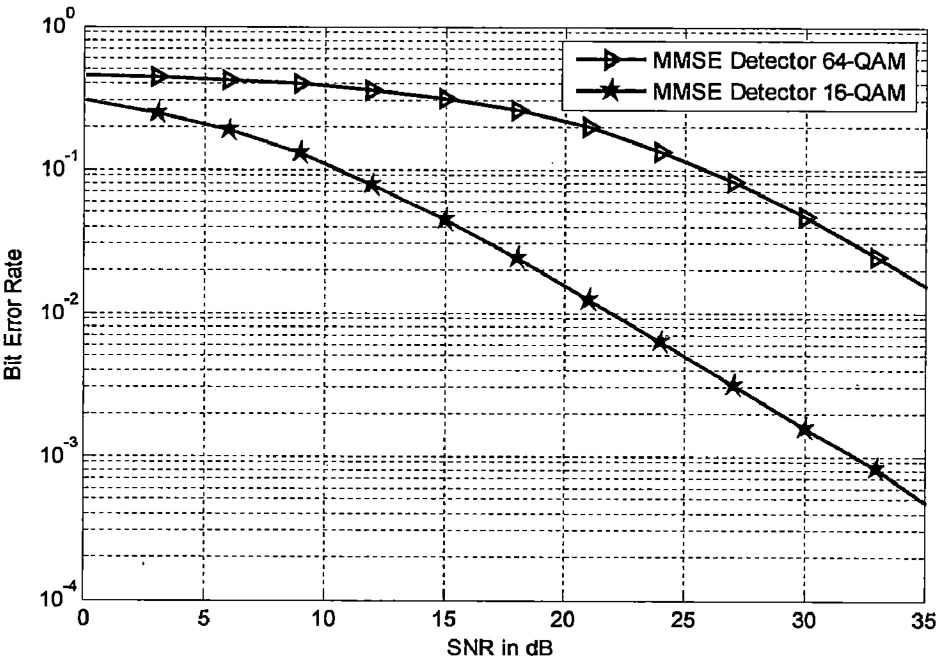


Fig.4.2. BER performance comparison of MMSE Detector in a 4×4 *MIMO system with* $16-QAM$ *and* $64-QAM$ *modulation*

4.1.2. Optimal and near-optimal Detectors

Optimal detector:

The essential of MIMO decoding is to solve (2.1). The optimal solution is the ML decoding.

$$\hat{s} = \underset{s \in \mathcal{S}}{\arg\min}\left(\|y\text{-}Hs\|^2\right) \tag{4.3}$$

where $\mathcal{S}$ consists of all possible vectors of s.

The procedure to perform exhaustive ML decoding is as follows

Generate $\mathcal{S}$ which consists of all possible vectors of s. Perform exhaustive search (4.3) for each symbol vector and obtain best $\hat{s}$. The estimated symbols are 16-QAM demodulated and demapped to bits. The N-dimensional decoded data is multiplexed in to a single bit stream.

Near-optimal decoders:

As discussed in chapter.2, the sphere decoding algorithms can achieve near-ML performance for MIMO decoding for reasonable complexity. The sphere decoding algorithms have two kinds of implementation strategies, The FIncke-Pohst strategy called SD and the Schnorr-Euchner strategy called SE.

(1) Sphere decoder

Sphere decoder is based on the enumeration of points in the search set that are located with in a sphere of some radius centered at a target. The following are the steps to perform Sphere decoding.

Step.1 *Pre-decoding phase*: The inputs to the sphere decoder are H, y, d. QR-decomposition is performed to calculate the triangular form of matrix H.

Step.2 *Initialization phase:* Calculate the Zero-Forcing point (ZF-point) as an initial estimate, and initialize the sphere dimension.

Step.3 *Search phase:* Calculate the bounds for each component of the estimate. Search for the closest point using bounds and get the best estimate.

The detailed flowchart is given below

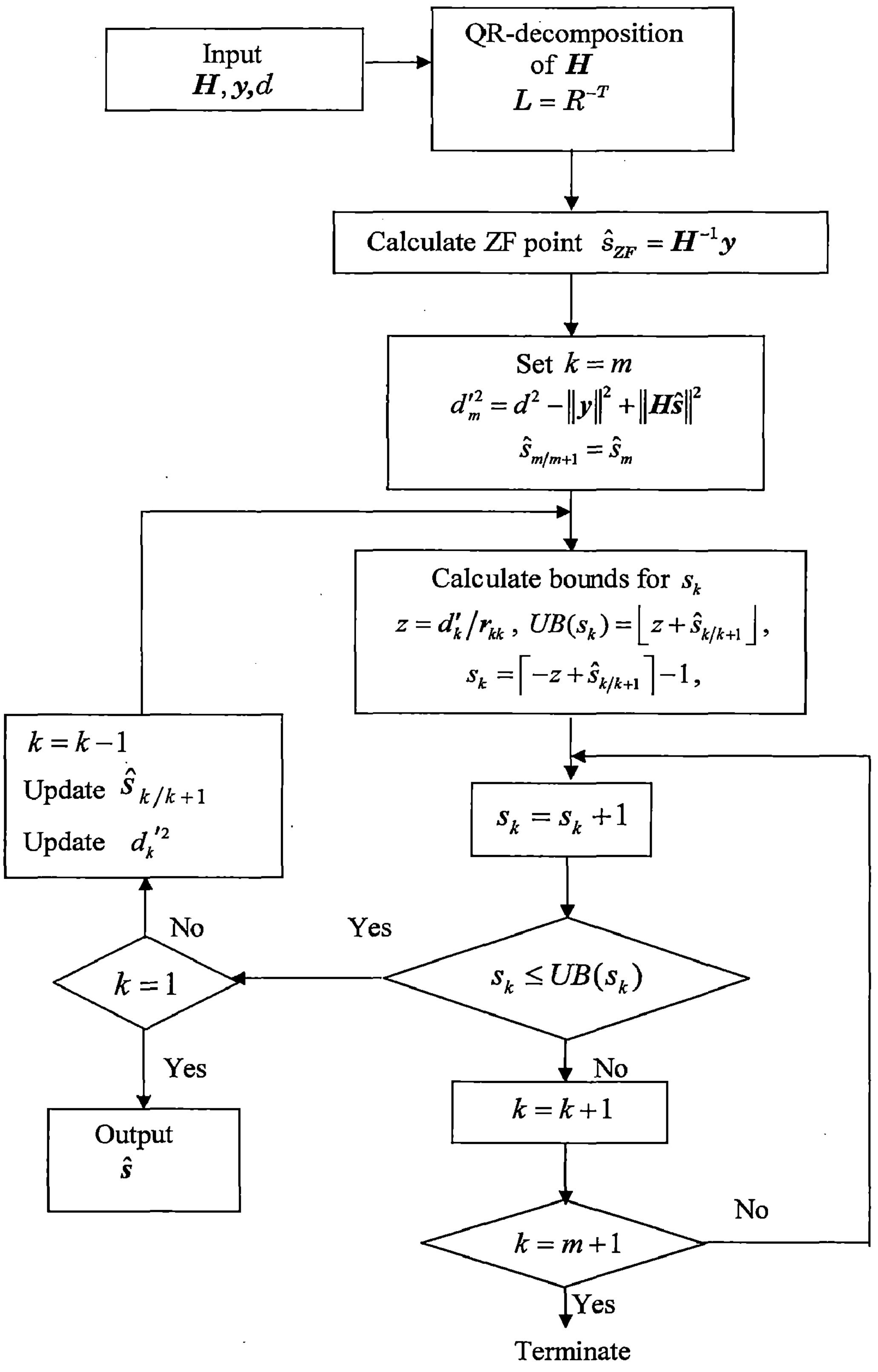
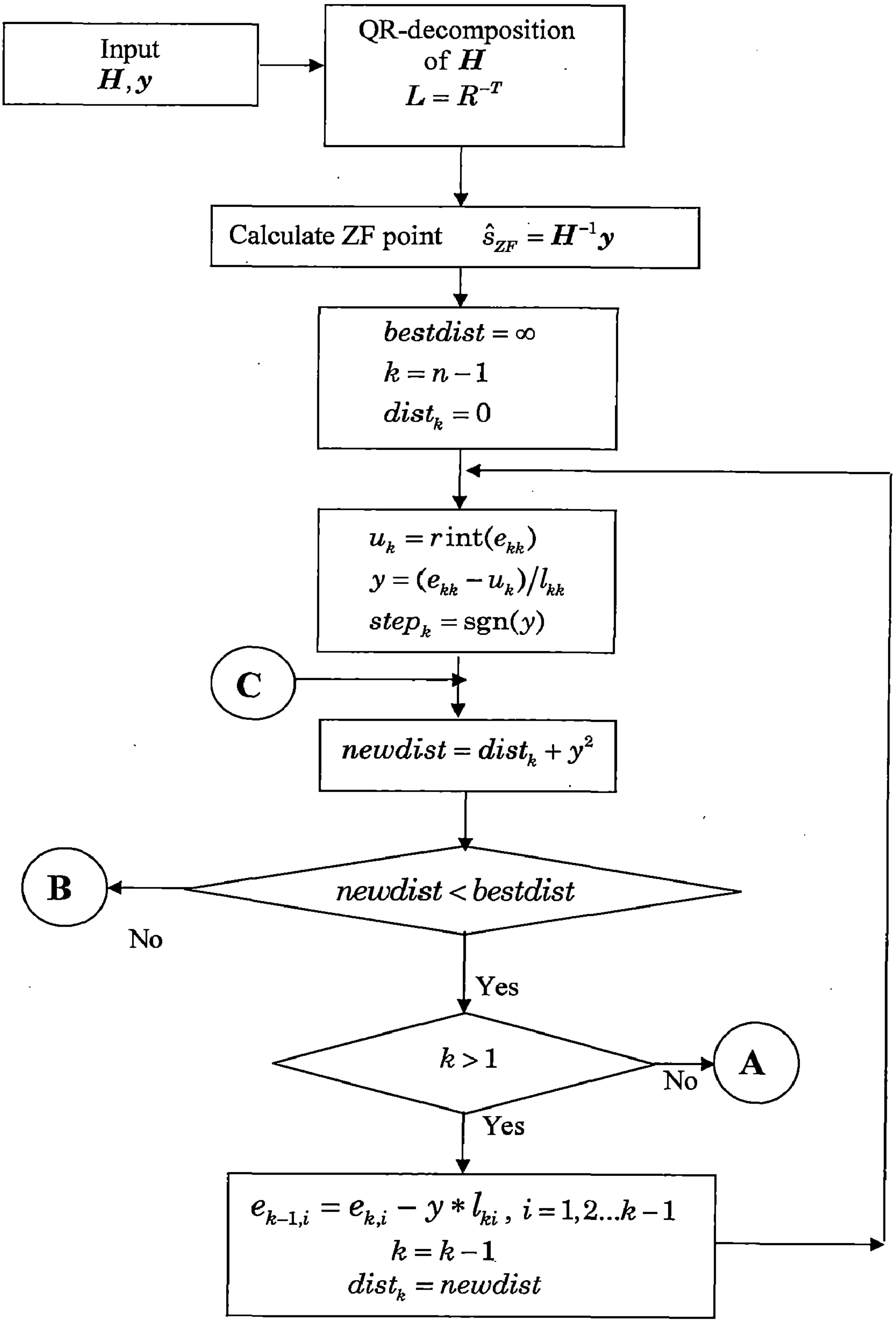

Input
H, y, d
QR-decomposition
of H
$L = R^{-T}$
Calculate ZF point $\hat{s}_{ZF} = H^{-1}y$
Set $k = m$
$d'^2_m = d^2 - \|y\|^2 + \|H\hat{s}\|^2$
$\hat{s}_{m/m+1} = \hat{s}_m$
Calculate bounds for s_k
$z = d'_k / r_{kk}$, $UB(s_k) = \lfloor z + \hat{s}_{k/k+1} \rfloor$,
$s_k = \lceil -z + \hat{s}_{k/k+1} \rceil - 1$,
$k = k - 1$
Update $\hat{s}_{k/k+1}$
Update $d_k'^2$
$s_k = s_k + 1$
No
Yes
$k = 1$
$s_k \le UB(s_k)$
Yes
No
Output
$\hat{s}$
$k = k + 1$
No
$k = m + 1$
Yes
Terminate

Fig.4.3. Flowchart for Sphere Decoding Algorithm

(2) Schnorr-Euchner strategy

This algorithm has the same principle as the SD: the search for the closest point. This algorithm is based on two stages. The first stage consists in searching for the "Babai point" (BP), which represents a first estimation, but is not necessarily, the closest point. Finding the BP gives us a bound on the error. In the second stage, we modify the BP until the closest point is reached. We zigzag around each BP component in turn to build the closest point (unlike the sphere decoder, there is no minimum and maximum bound for each BP component). The time needed to find the closest point is closely related to BP, which means closely related to the SNR. In fact, if the BP is very far from the closest point, i.e for low SNRs, the algorithm takes much more time to converge. However, if the BP is close to the closest point, i.e for high SNRs, the algorithm converges rapidly. This algorithm searches for entire vector in the tree at same time, even if one component of the estimate is not belongs to the constellation, it once again calculates all the components of the estimate.This algorithm is called as the Schnorr-Euchner reference algorithm.

A modified Schnorr-Euchner algorithm in contrast to the above algorithm searches for individual components of the estimate which belongs to constellation. The following flow charts give the detailed description.


Input
H, y
QR-decomposition of H
$L = R^{-T}$
Calculate ZF point $\hat{s}_{ZF} = H^{-1}y$
$bestdist = \infty$
$k = n-1$
$dist_k = 0$
$u_k = rint(e_{kk})$
$y = (e_{kk} - u_k)/l_{kk}$
$step_k = sgn(y)$
C
$newdist = dist_k + y^2$
B
$newdist < bestdist$
No
Yes
$k > 1$
No
A
Yes
$e_{k-1,i} = e_{k,i} - y * l_{ki}, \; i = 1,2...k-1$
$k = k-1$
$dist_k = newdist$

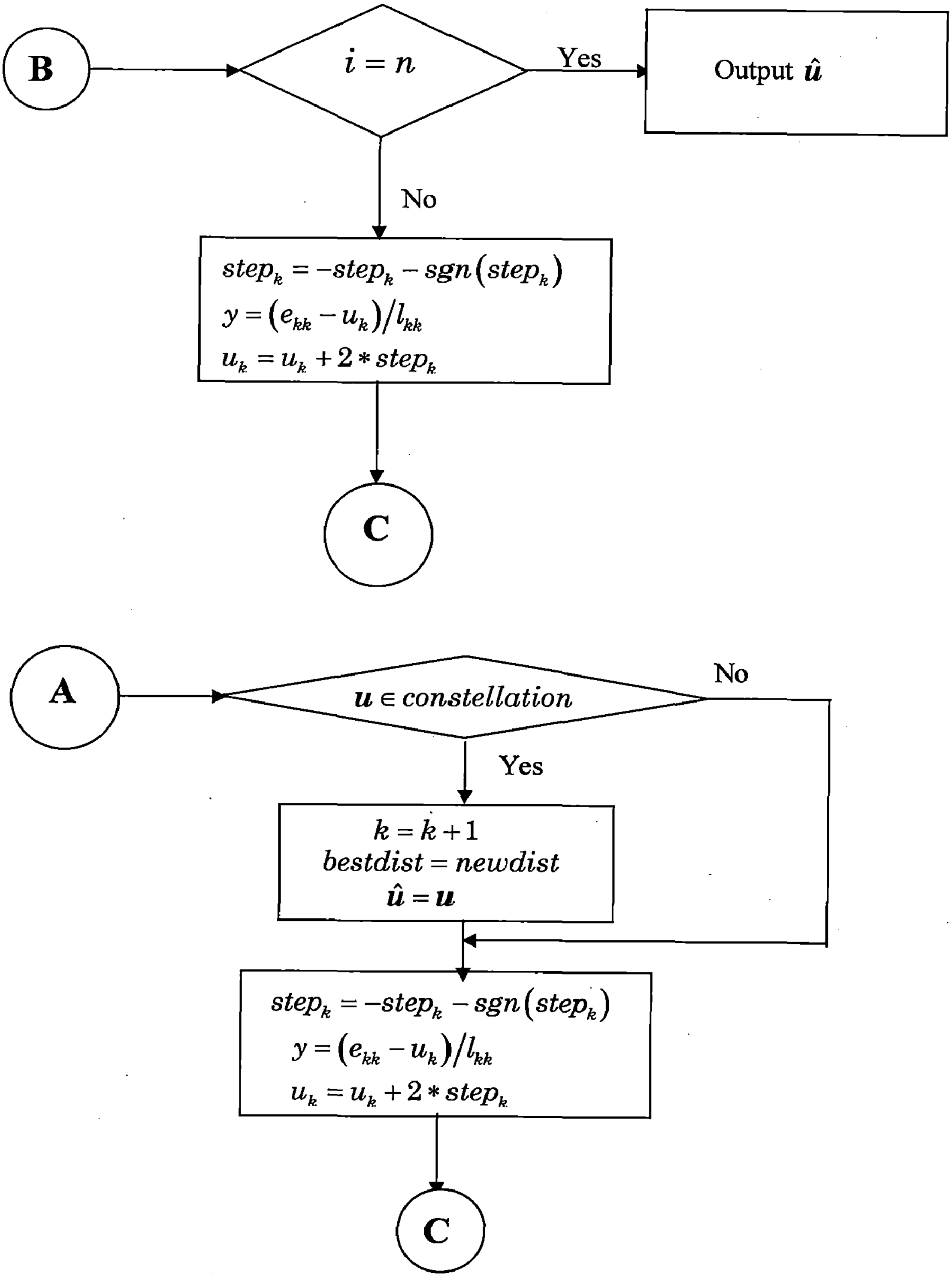
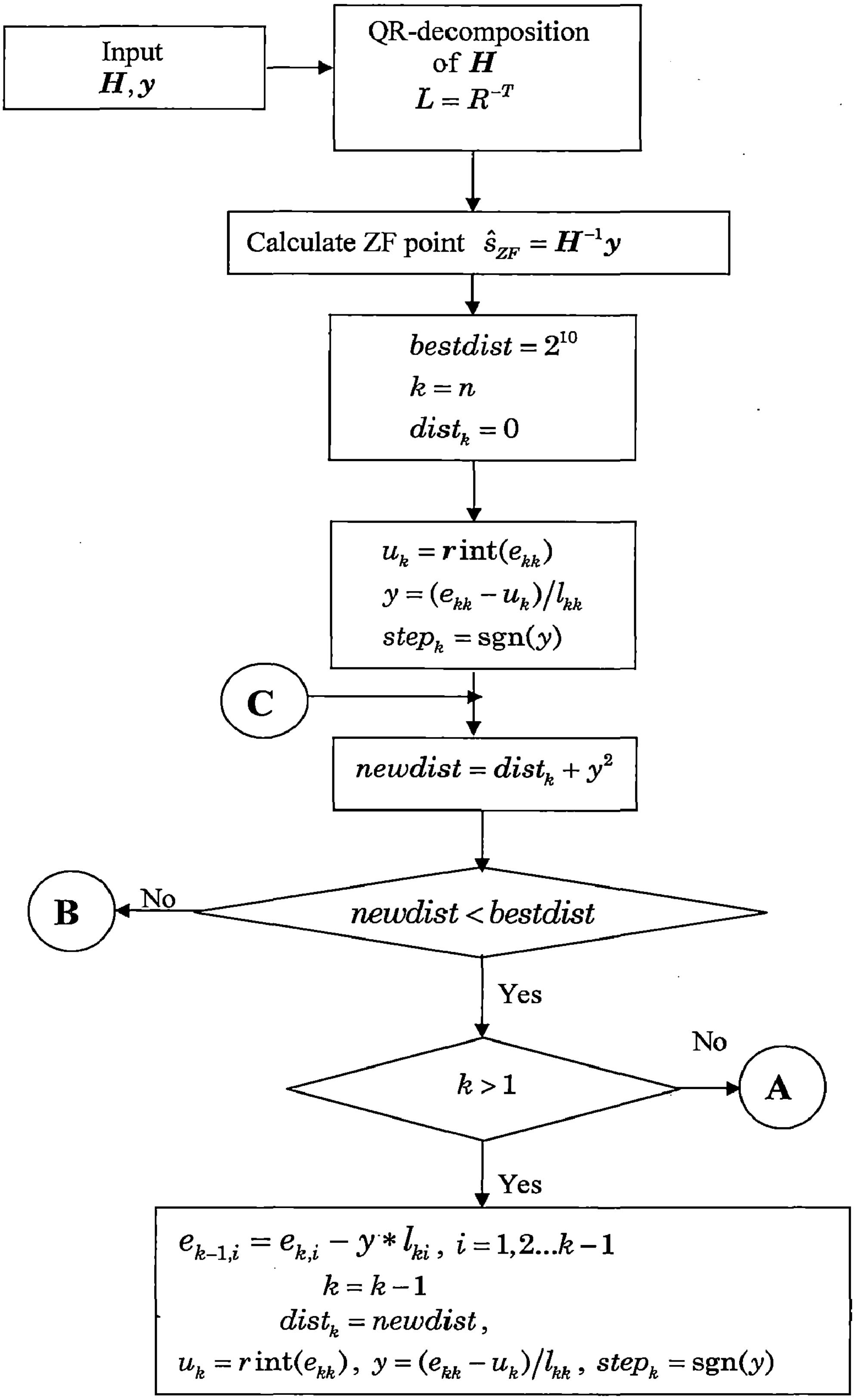

B
$i = n$
Yes
Output $\hat{\boldsymbol{u}}$
No
$step_k = -step_k - sgn(step_k)$
$y = (e_{kk} - u_k)/l_{kk}$
$u_k = u_k + 2 * step_k$
C
A
$\boldsymbol{u} \in constellation$
No
Yes
$k = k + 1$
$bestdist = newdist$
$\hat{\boldsymbol{u}} = \boldsymbol{u}$
$step_k = -step_k - sgn(step_k)$
$y = (e_{kk} - u_k)/l_{kk}$
$u_k = u_k + 2 * step_k$
C

Fig.4.4. Flowchart for Schnorr-Euchner algorithm


Input
$\boldsymbol{H},\boldsymbol{y}$
QR-decomposition of $\boldsymbol{H}$
$L = R^{-T}$
Calculate ZF point $\hat{s}_{ZF} = \boldsymbol{H}^{-1}\boldsymbol{y}$
$bestdist = 2^{10}$
$k = n$
$dist_k = 0$
$u_k = r\text{int}(e_{kk})$
$y = (e_{kk} - u_k)/l_{kk}$
$step_k = \text{sgn}(y)$
C
$newdist = dist_k + y^2$
B
No
$newdist < bestdist$
Yes
No
$k > 1$
A
Yes
$e_{k-1,i} = e_{k,i} - y * l_{ki}$, $i = 1,2...k-1$
$k = k-1$
$dist_k = newdist$,
$u_k = r\text{int}(e_{kk})$, $y = (e_{kk} - u_k)/l_{kk}$, $step_k = \text{sgn}(y)$

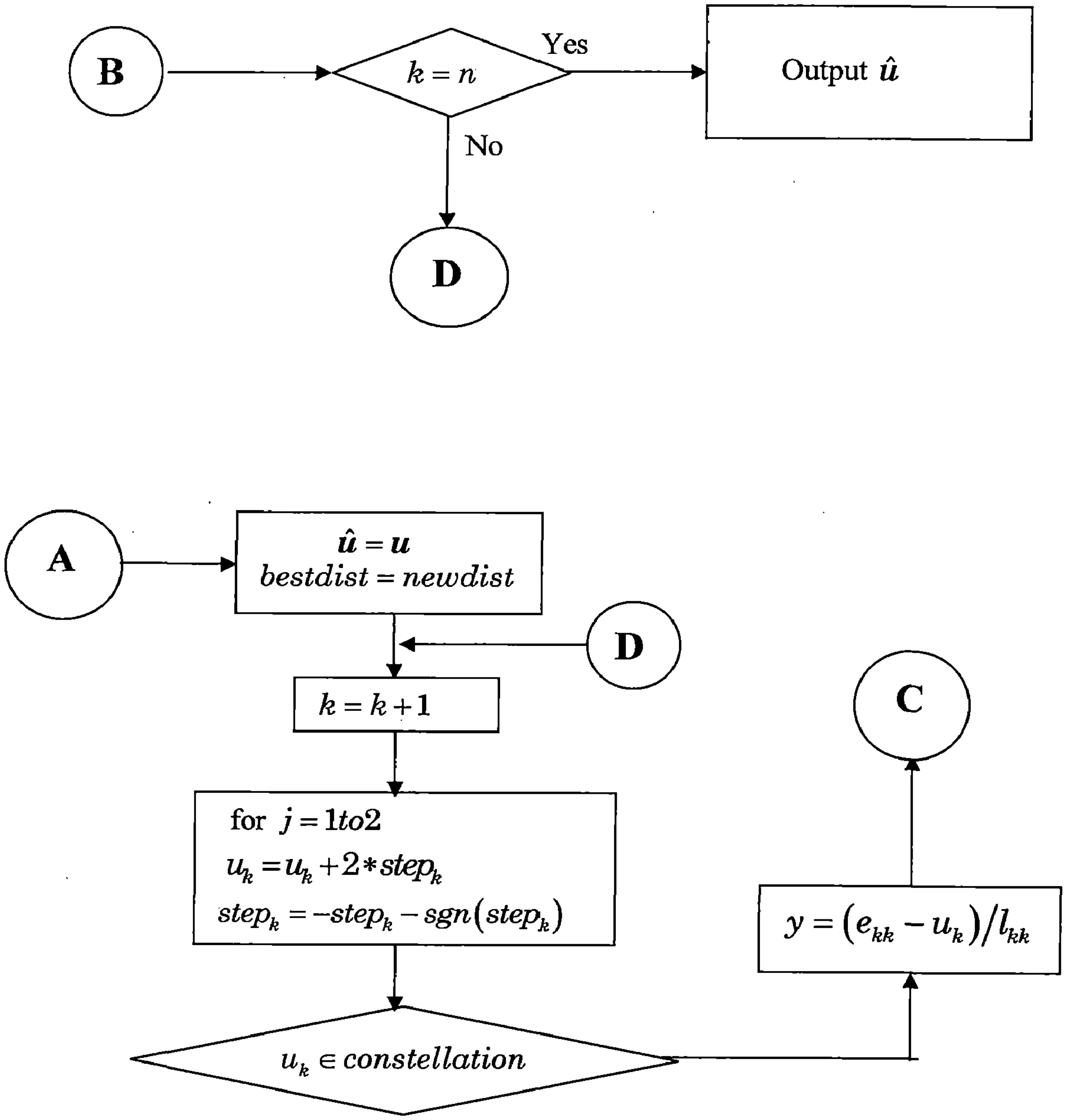


Fig.4.5. Flowchart for Modified Schnorr-Euchner strategy

Fig.4.6. gives the BER performance comparison of Exhaustive ML, Sphere decoding algorithm (SD) for a 4×4 MIMO system. BER of Sphere decoder is evaluated over 10,00,000 independent channel realizations and BER of exhaustive ML is evaluated over 10,000 independent channel realizations. The performance of sphere decoder approaches exhaustive ML search with reasonable complexity.

Fig.4.7. shows the BER performance comparison of different detection methods namely ZF detector, MMSE detector, SIC, Exhaustive ML, and Sphere decoder. Clearly exhaustive ML and SD outperforms by nearly 10dB at BER of 10^{-3} compared to other techniques but there is a trade off between BER performance and computational complexity. In VLSI implementation point of view, we need systems with low complexity so that the design may become easy, for which we can effort to lose the performance but when high performance is needed, optimal, near-optimal detectors play major role with high computational cost.

Fig.4.8 compares the BER performance of Schnorr-Euchner strategy and its variants by considering 4×4 MIMO system with $16-QAM$ modulation. Simulations are carried out over 10,00,000 independent channel realizations. Clearly, a BER of 10^{-5} can be attained at SNR$\approx$22dB by all Schnorr-Euchner algorithms, while the performance deference between SE-reference and SE1 (Modified SE) is minor. It is also clear from figure that SE-reference outperforms SE1 by about 1dB when BER $=10^{-6}$. However, the performance difference between SE-reference and SE2 is about 0.5 dB when BER$=10^{-6}$, which shows that suitable early termination criteria could improve the BER performance of SE1 at high SNR.

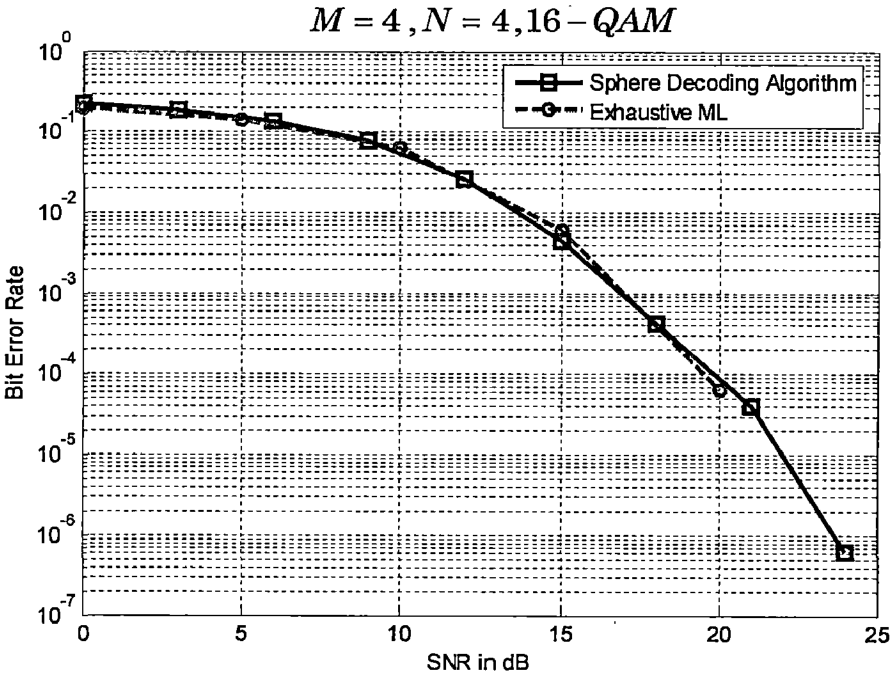


Fig.4.6. BER performance comparison of Exhaustive ML, Sphere Decoding Algorithm in a 4×4 *MIMO system with* $16-QAM$.

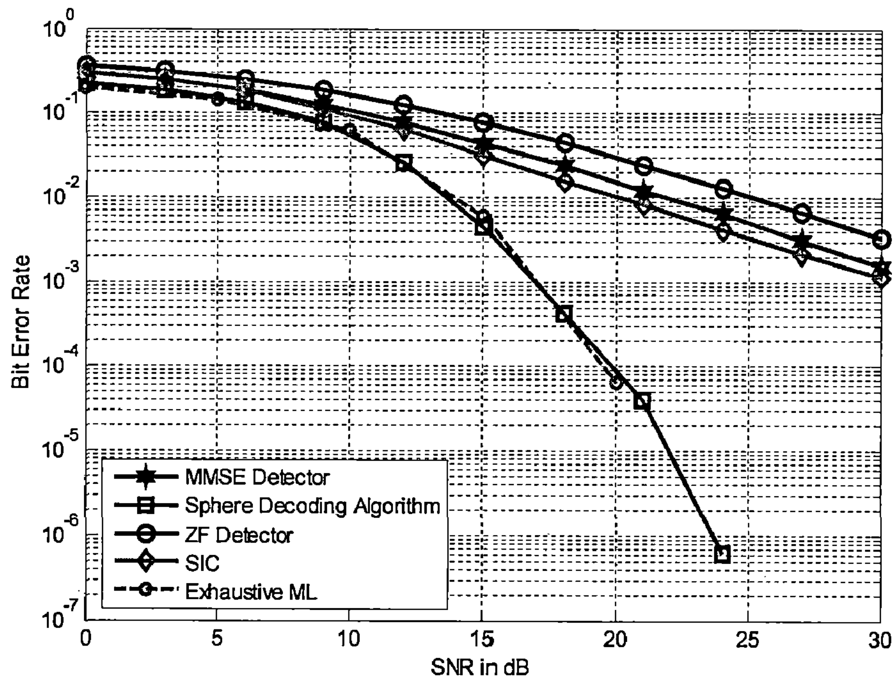


Fig.4.7. BER performance comparison of different MIMO detection schemes in a 4×4 *MIMO system with* $16-QAM$.

$M = 4\,, N = 4, 16 - QAM$

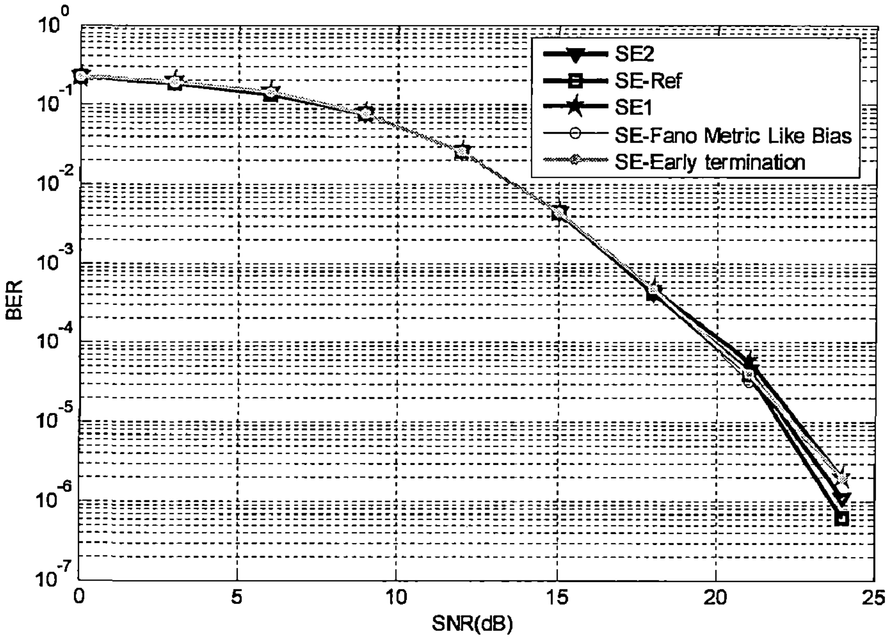


Fig.4.8. BER performance of Schnorr-Euchnerr strategies in a 4×4 *MIMO system with* $16 - QAM$ *modulation.*

4.2 Comparison of Computational Complexity

ML detector gives the optimum performance for small number of antennas and for small constellations, but if the number of antennas increases at transmitter or at receiver or if the constellation size increases, the complexity of ML detector increases exponentially. Near-optimal detectors give near-ML performance with polynomial complexity.

Fig.4.9. shows the performance in terms of average number of multiplications needed to detect a symbol vector. Considering a 4×4 MIMO system with $16-QAM$ modulation, 16 bits will be transmitted per each channel realization. Now exhaustive ML detector performs (4.3) by searching over 2^{16} possible transmitted vectors for each channel realization which results in a huge complexity problem. Now consider a real MIMO channel model in which the channel matrix is of size 8×8 and the received vector is of size 8×1. If we perform exhaustive ML search given by (4.3) over all possible symbol vectors, the detector needs $(64+1)16^4=4259840$ multiplications to detect a symbol vector which is infeasible with current technologies. If we increase the number of antennas or the constellation size, the complexity of ML detector will increase exponentially. On the other hand, sphere decoder searches for the estimate with reasonable complexity which varies with SNR, because as SNR increases, the radius gets adapted there by the search space gets reduced, which means the perturbed point gets closer to the estimate. The average number of multiplications can be found by counting the multiplications in the search phase of sphere decoding algorithm described in section 2.4.3.1. For simplicity, the preprocessing computations are not included. In case of SIC also, the ZF nulling vector computations are neglected, but indeed they results in heavy computational complexity as the pseudo inverse requires more multiplications. It is clear from the Fig.4.9 that SIC results in less number of multiplications compared to exhaustive ML and SD but it has poor BER performance.

Fig.4.10. compares the average number of multiplications required for the sphere decoder (searching phase), when the radius is fixed to 1 and when radius is a function of SNR. The number of multiplications can be calculated through simulation of sphere decoding algorithm described in section 2.4.3.1. When the SNR is low, the search oscillates in between the layers which results in more multiplications, which can be found by incrementing the count each time the search visits step.5. Clearly the Fig.4.10 confirms the need to adapt the radius according to SNR. Note that by using as adapted radius, we obtain smallest number of multiplications especially at small SNRs, for example at 5dB, we have around 750 multiplications less than the other case. This improvement of the SD will obviously improve the total complexity of the SD.

Fig.4.11. plots the algorithm complexity of Schnorr-Euchner and its variants. The algorithm complexity is defined as the average number of searched sublattices per symbol vector, i.e number of evaluations on Line $\langle 9 \rangle$ in algorithm SE1 described in section 2.4.3.2.. So the term average number of sublattices per symbol vector comes from the fact that the number of times the loop is repeated before termination of the algorithm. It is clear from the figure that SE1 reduces the complexity of SE-reference significantly at low and moderate SNR. Moreover the algorithm complexity is further reduced in SE2, which combines SE1 with Fano like metric bias and SE1-Early termination. The effects of SE1 with Fano like metric bias and SE1-Early termination are different on complexity reduction. SE1-Fano like metric bias is more effective at low SNR than at high SNR, since the value of σ^2 is too small at high SNR to affect the path metric of the higher level of the tree.

Fig.4.12 shows the complexity exponent for sphere decoder as a function of number of number of transmit antennas m for SNR=20dB, for $L-PAM$ constellation with $L=2$. It is plotted for eq.(2.50) as a complexity exponent $\frac{\log C(m,\rho)}{\log m}$ It is clear from the figure that for small constellations the expected complexity is polynomial, where as for large constellations the expected complexity would be exponential.

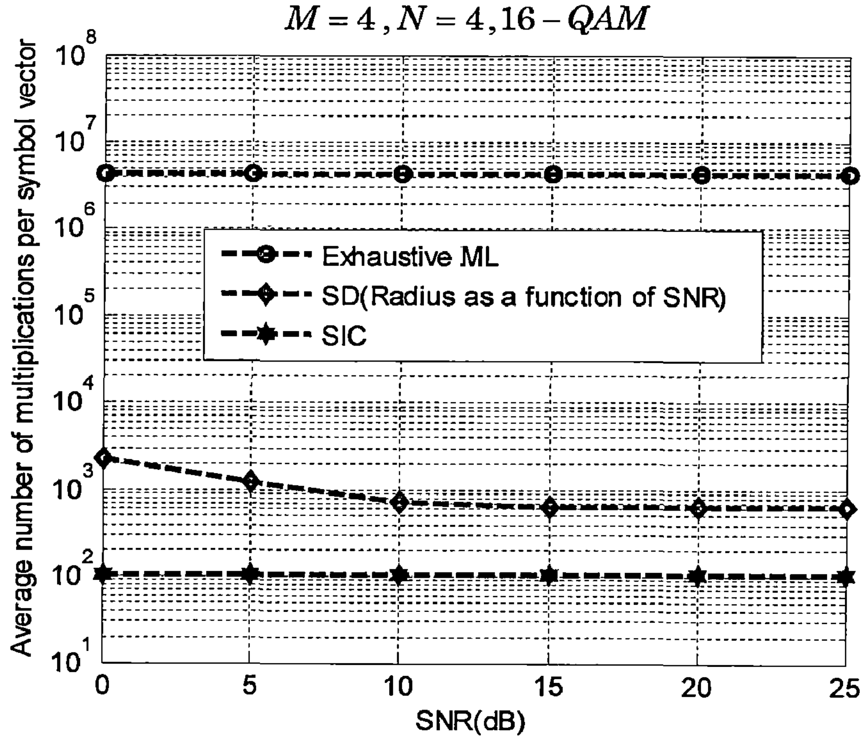


Fig.4.9. Computational complexity comparison of Exhaustive ML, SD (Radius as a function of SNR), and SIC.

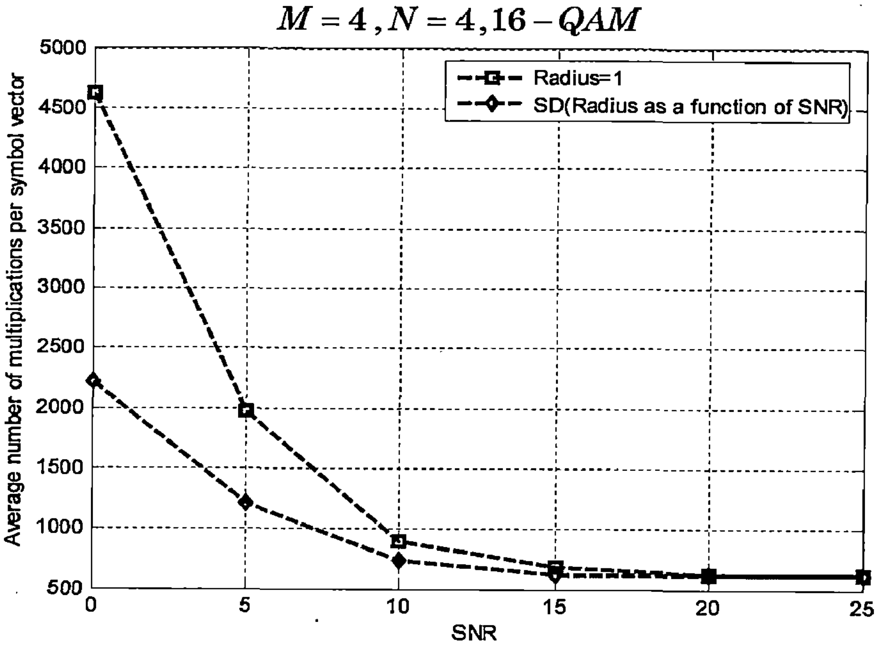


Fig.4.10. Number of multiplications required for Sphere Decoder as a function SNR in a 4×4 MIMO system with $16 - QAM$

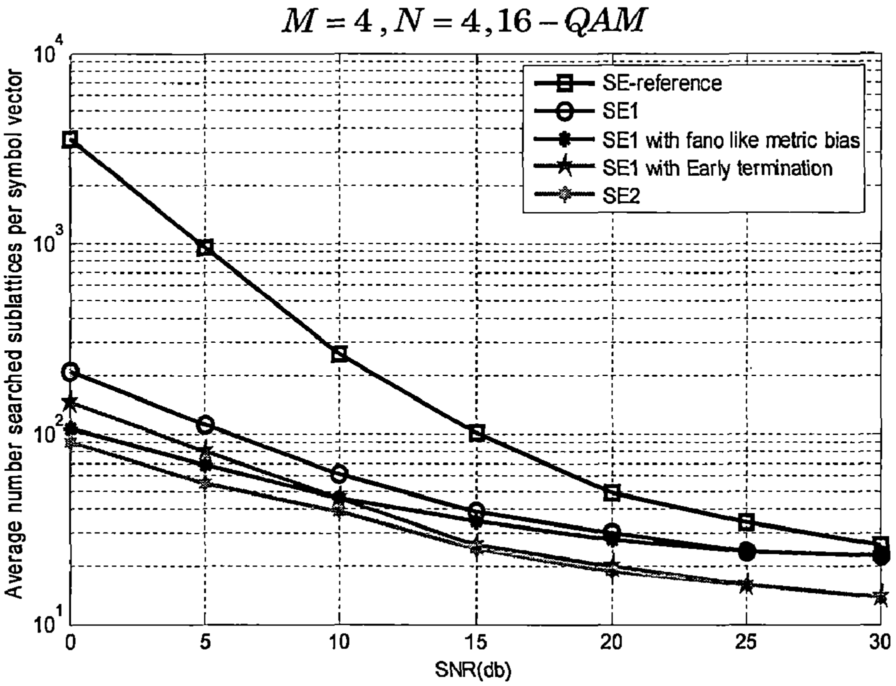


Fig.4.11.Algorithm complexity of Schnorr-Euchner strategy in a 4×4 *MIMO system with* $16 - QAM$.

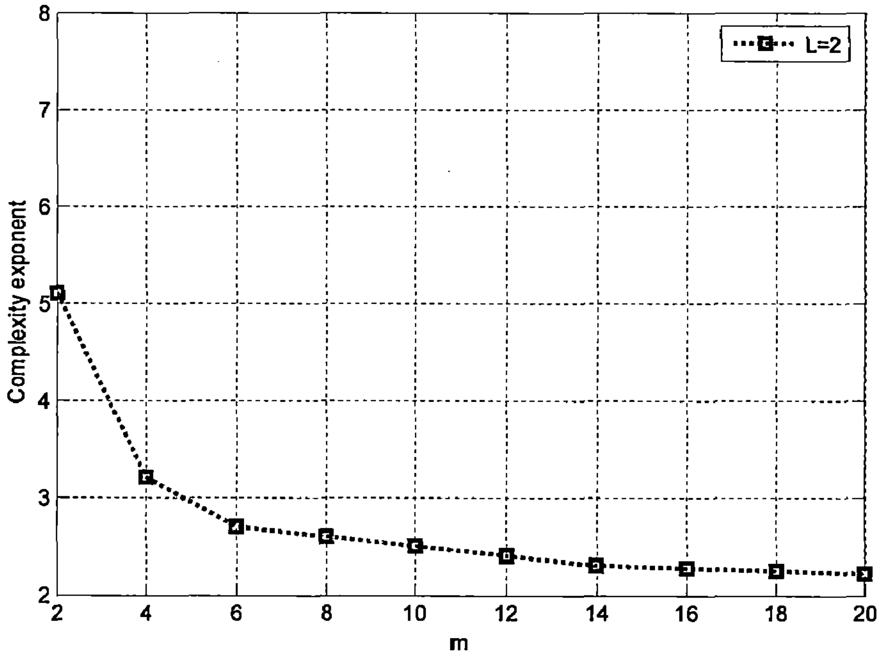


Fig.4.12.The complexity exponent of the sphere decoder as a function of m for SNR=20dB and 2 – PAM constellation.

4.3 Simulation for Iterative MIMO detection using modified Sphere decoding algorithm

An iterative detection and decoding process is performed which improves the performance of coded MIMO detection. Here we use an optimal soft input-soft output detector for channel decoding .The IDD approach improves the detection performance by utilizing the detector output.

Simulation steps required for the IDD scheme by referring Fig.3.1 is outlined as follows

Step.1. Generate binary data of 1000 bits per one block.

Step.2. The information is encoded by $R = 1/2$ rate non-systematic convolutional code with memory length 2. The generating polynomials are $G_1(D) = 1 + D + D^2$ and $G_2(D) = 1 + D^2$. The coded sequence is interleaved and modulated by means of simple Gray mapping onto a 16-QAM modulation scheme, and transmitted through 4×4 MIMO channel. The trellis for the above convolutional code is given by

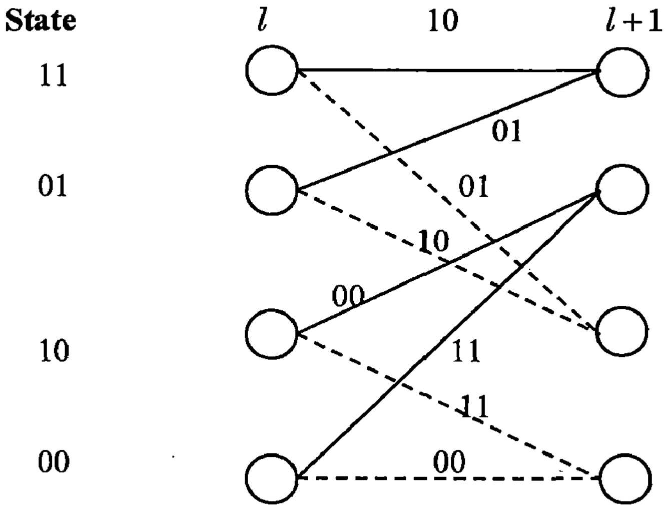


Fig. 4.13 Trellis for rate $\frac{1}{2}$-convolutional encoder with memory 2

In Fig.4.13 the transition, corresponds to zero information bit is shown by dotted line and the transition, correspond to one information bit is shown by thick line.

Step.3. On the receiver side, by using FP-MAP algorithm described in section 3.2, we compute the soft bit information as follows

$$L_1(c_i/y) = \log \frac{\sum_{s:c_i=+1} e^{-\|y-Hs\|^2+\sum_j \log p[s_j]}}{\sum_{s:c_i=-1} e^{-\|y-Hs\|^2+\sum_j \log p[s_j]}}$$

where c_i is the coded information from which current symbol vector s is obtained.

Step.4. The vector L_1 is deinterleaved which is then used by channel decoder (soft input-soft output decoder) to form the estimate of the information bit vector $\hat{b}$, as well as to provide L_2', the a posteriori reliability information for the coded bits vector c'. The channel decoder finds the APP $L-$values by using the above trellis and the equations described in section 3.3.

Step.5. A posteriori reliability information for the vector c is obtained by interleaving L_2' into L_2,which will converted to the a posteriori probabilities for the components of symbol vector s and those probabilities will be used by FP-MAP algorithm .

Step.6. steps3 to 5 can be repeated to improve the performance.

Fig.4.14 shows the BER performance comparison of IDD based MIMO detection using modified sphere decoding algorithm (FP-MAP) for iteration1 and interation3. The performance is improved for iteration 3 by 1.5 dB at BER10^{-3}. Thus the IDD based MIMO detection gives better performance compared to normal MIMO detection.

$M = 4, N = 4, 16 - QAM$

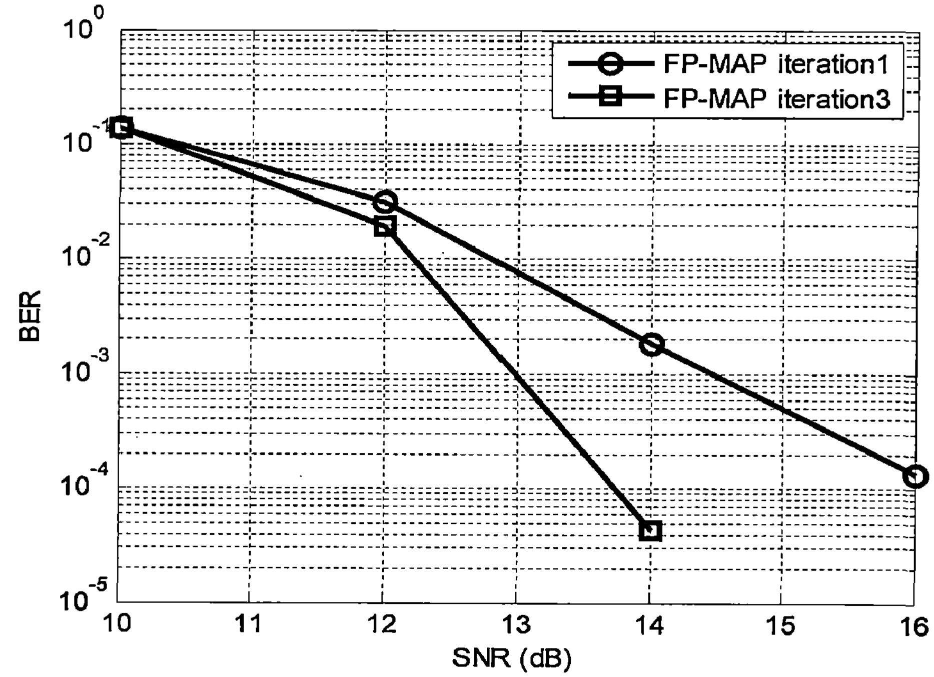


Fig.4.14. BER performance curve for IDD based MIMO detection in 4×4 *MIMO system with* $16 - QAM$ *modulation*

Chapter 5

Conclusion

This dissertation work is aimed at the performance comparison of different detection schemes in multiple input multiple output (MIMO) systems, namely optimal (ML), sub-optimal (ZF, MMSE, SIC)and near-optimal (Sphere) detectors . The MIMO detection is then extended to iterative decoding scheme which results in improved performance. The simulation results can be summarized as follows

❖ Among the sub-optimal detectors (ZF, MMSE, SIC), the performance of SIC is better compared to the linear detectors such as ZF, MMSE. Linear detectors can achieve the diversity order only $N-M+1$, but compared to linear detectors, SIC achieves increased diversity order with each iteration. While the first detected stream sees a diversity order $N-M+1$, the second achieves $N-M+2$ and so forth.

❖ The optimal performance is achieved by ML detectors but its complexity increases exponentially with number of antennas or the constellation size. Near-ML performance is achieved by near optimal decoding schemes such as sphere decoding algorithm and Schnorr-Euchner algorithm at the cost of polynomial complexity for smaller constellations. But if the constellation size increases, the complexity of these two algorithms increases exponentially. For same specifications, the BER performances of sphere decoding algorithm and Schnorr-Euchner algorithm are same, but differ in the complexity.

❖ Sphere decoding algorithm performance is dependent on the choice of radius. When the radius is fixed , algorithm requires more number of multiplications specially at low SNRs, but when the radius is adapted according to SNR, the complexity is reasonable even at low SNRs. The complexity of Schnorr-Euchner algorithm is reduced by its variants such as Fano like metric bias, early termination and the combination of these two , with only a small degradation in BER performance at high SNR.

❖ A further improvement in performance is achieved for a coded MIMO detection using iterative decoding scheme, in which the modified sphere decoding algorithm (FP-MAP) reduces the complexity of MIMO detection. It has been proved that, for the same choice of radius, the expected number of points that the FP-MAP algorithm visits is upper bounded by the expected number of points visited by original sphere decoding algorithm. This simplified IDD scheme is quiet important as the demand grows for higher spectral efficiency in future wireless systems.

Future work

In evaluating the performance of different detection schemes in MIMO systems, it is assumed that the channel is perfectly known to the receiver. Effect of imperfect channel estimation at the receiver on the performance of near optimal ML decoder needs to be investigated.. A lattice aided reduction algorithms can also be implemented to reduce the lattice size. Orthogonal frequency division multiplexing (OFDM) may be combined with MIMO to increases the diversity gain and enhance the system capacity on time variant and frequency selective channels. A direct application of the MIMO detectors can be applied to each subcarrier in a MIMO-OFDM system. The number of MIMO detectors required to implement a MIMO-OFDM system is the number of subcarriers that the OFDM system employ. Recently developed fixed complexity sphere decoders (FSD) can also be applied to the case where an outer code is used in the MIMO system (Turbo-MIMO system).

References

1. T.Rappaport, "Wireless Communications: Principles and Practice, 2nd Edition, 2001.
2. A. J. Paulraj, D. Gore, R. U. Nabar, H. Bolcskei, "An Overview of MIMO communications - A Key to Gigabit Wireless. Nov.2003
3. G. J. Foschini and M. J. Gans, "On limits of wireless communications in a fading environment when using multiple antennas," Wireless Personal Communications: Kluwer Academic Press, vol. 6, no. 3, pp. 311–335, 1998.
4. I. E. Telatar, "Capacity of multi-antenna Gaussian channels," AT*&T Bell Lab. Internal Tech. Memo.*, June 1995.
5. G. J. Foschini, "Layered space-time architecture for wireless communication in fading environments when using multi-element antennas," *Bell Labs Technical Journal*, vol. 1 no. 2, pp. 41-59, Autumn 1996.
6. U. Fincke, M. Pohst, "Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis", *Mathematics of Computation*, Vol. 44, No. 170, pp. 463-471, April 1985
7. E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels," IEEE *Trans. Inform. Theory*, vol. 45, pp. 1639–1642, July 1997.
8. C.P.Schnorr, M.Euchner, " Lattice basis reduction: Improved practical algorithms and solving subset sum problems."Math.Programming,vol.66,pp.181-191,1994.
9. E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inform. Theory*, vol. 48, pp. 2202–2214, Aug.2002.
10. W. H. Mow, C. W. Bay, "Universal Lattice Decoding: Principle and Recent Advances", Hong Kong Univ. of Science and Technology, 2002.
11. B. Holter, " On the capacity of the MIMO channel-A tutorial introduction", Norwegian University of Science and Technology.

12. A. Goldsmith, S.A. Jafar, N. Jindal, and S. Vishwanath, “Fundamental Capacity of MIMO Channels” Department of Electrical Engineering, Stanford University, Stanford, CA 94305, 2002.

13. T. Kailath, H. Vikalo, and B. Hassibi, "MIMO Receive Algorithms," in *Space-Time Wireless Systems: From Array Processing to MIMO Communications*, (editors H. Bolcskei, D. Gesbert, C. Papadias, and A. J. van der Veen), Cambridge University Press, 2005.

14. B. Hassibi and H. Vikalo, "Maximum-Likelihood Decoding and Integer Least-Squares: The Expected Complexity," in Multiantenna Channels: Capacity, Coding and Signal Processing, (editors J. Foschini and S. Verdu), American Mathematical Society (AMS) 2003, pp. 161-191.

15. H. Vikalo ,B.Hassibi, “On the Sphere-Decoding Algorithm I. Expected Complexity”, IEEE Transactions on Signal Processing, Vol. 53, No. 8, pp.2806-2818, August 2005.

16. C.Y. Hung , T.H. Sang, “ A Sphere Decoding Algorithm for MIMO Channels,” IEEE International Symposium on Signal Processing and Information Technology, pp.502-506, 2006.

17. A. M. Chan , I. Lee, “A new reduced-complexity sphere decoder for multiple antenna system,” *IEEE Conference on Communications*, pp.460-464, 2002.

18. Z. Guo, P.Nilsson, “Algorithm and implementation of K-best Sphere decoding for MIMO detection”, IEEE Journal on selected areas in communications, Vol.24, No.3, pp. 491-503, March 2006.

19. Z.Guo, P.Nilsson , “Reduced complexity Schnorr-Euchner decoding algorithms for MIMO systems,” IEEE Communication Letters, Vol.8,No.5,pp.286-268,May 2004.

20. G. Rekaya and J. Belfiore, “Complexity of ML lattice decoders for the decoding of linear full rate space-time codes,” IEEE *Trans. Wireless Commun.*, to be published.

21. G. Rekaya , J. Belfiore, “On the complexity of ml lattice decoders for decoding linear full rate space-time codes”, IEEE International Symposium on, Proceedings, Information Theory , page.206, July 2003.

22. H. Vikalo ,B.Hassibi, "On the Sphere-Decoding Algorithm II, Generalizations, second order statistics, and applications to communications", IEEE Transactions on Signal Processing,Vol.53, No.8, pp.2819-2834, August 2005.

23. B. Hassibi and H.Vikalo, On the expected complexity of sphere decoding, *Conference Record of the Thirty-Fifth Asilomar Conference on Signals, Systems and Computers* ,, pages 1051-55, 2001.

24. J. Conway, N. Sloane, *Sphere Packings,Lattices and Graphs*. New York: springer-Verlag, 1993.

25. S.Lin, D.J.Costello, "Error Control Coding-Fundamental and Applications", Second edition, Pearson Prentice Hall, 2004.

26. L.R.Bahl, J.Cocke, F.Jelinek, J.Raviv, "Optimal decoding of linear codes for minimizing symbol error rate, IEEE Transactions on Information Theory, Volume 20, No.2, pp.284 – 287, March 1974.

27. H.Vikalo, B.Hassibi, T.Kailath, " Iterative decoding for MIMO channels via modified Sphere decoding", IEEE transactions on wireless communications, Vol.3, No.6, pp.2299-2310, November 2004.

28. H. Vikalo , B. Hassibi, " Towards closing the capacity gap on multiple antenna channels," Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, pages 2385-88 ,2002.