Master’s Thesis

A Study of Lattice Reduction Detection Techniques for LTE Systems

Liping Wang

Department of Electrical and Information Technology,
Faculty of Engineering, LTH, Lund University, 2016.

A Study of Lattice Reduction Detection Techniques for LTE Systems

Liping Wang
wir11wli@student.lu.se

Department of Electrical and Information Technology
Lund University

Advisor: Fredrik Rusek

May 3, 2016

Printed in Sweden
E-huset, Lund, 2016

Abstract

MIMO (Multiple Transmit and Multiple Receive) antenna techniques are widely used in the most recent wireless communication standards. For example, LTE (Long-Term Evolution), WiMAX (Worldwide Interoperability for Microwave Access) and IEEE 802.11n have recently been rolled out across the world. In any communication system, the ML (Maximum Likelihood) receiver provides optimal error rate performance, but it turns out to be difficult to implement in system with 4 antennas or more.

This thesis studies ways to approach the optimal performance for MIMO channels with M-QAM (4-QAM, 16-QAM and 64-QAM) in Rayleigh MIMO channels with low complexity. While linear detectors like ZF (Zero Forcing) and MMSE (Minimum Mean Square Error) have very low computational complexity, they suffer from noise enhancement and ISI (Inter Stream Interference) respectively.

LR (Lattice Reduction) coupled with ZF or MMSE will give us near optimal results close to that of the ML receiver. Same diversity order like ML is also found. Basically, LR is aiming to find a basis of the channel matrix as orthogonal as possible. A basis is composed by a set of linearly independent vectors. A big improvement can be achieved by replacing linear detectors with LR techniques.

Correlated channels will be studied for comparison reasons. To evaluate the impact of channel correlation, Kronecker Model with three different correlation factors will be discussed.

In future work section, some cutting edge algorithms will be mentioned, for example: BKZ (Block Korkine Zolotarev) Algorithm.

Acknowledgments

First and foremost, my deepest gratitude goes to Supervisor Fredrik Rusek, for his great encouragement and guidance through whole process of the master thesis. To write this master thesis is one of the greatest academic challenges that I have ever faced. At the same time, I would like to express my heartfelt gratitude to Ph.D. student Meifang Zhu, who has instructed and helped me with my academic research and master thesis.

In addition, I would like to give my thanks to teachers that have gave me enormous help and advisors for the past few years during my master. And sincere thanks to the technical staff who help me to get access to the latest version of essential soft wares.

Here are special thanks to my beloved family. With their great support both financially and mentally, I can chase my life goal without hesitation, fear and loss. And many thanks to all my dearest friends and all my classmates, who give me best life experiences ever, hope our friendship last forever.

Acronyms

MIMO Multiple Input and Multiple Output

ZF Zero Forcing

MMSE Minimum Mean Square Error

BER Bit Error Rate

CVP Closest Vector Problem

AWGN Additive White Gaussian Noise

NLOS Non Line of Sight

i.i.d Independent and Identically Distributed

LR Lattice Reduction

ML Maximum Likelihood

ISI Inter-Symbol Interference

MSI Multi-Symbol Interference

LLL Lenstra–Lenstra–Lovász

QAM Quadrature Amplitude Modulation

OFDM Orthogonal Frequency-division Multiplexing

PSK Phase Shift Keying

ASK Amplitude Shift Keying

SU-MIMO Single User Multiple Input and Multiple output

MU-MIMO Multi User Multiple Input and Multiple output

LLL-SIC Lenstra–Lenstra–Lovász with Successive Interference Cancellation

SIC Successive Interference Cancellation

LAN Local Area Network

MMSE-SIC Minimum Mean Square Error with Successive Interference Cancellation

BKZ Block Korkine Zolotarev

SNR Signal to Noise Ratio

BW Bandwidth

BS Base Station

MS Mobile Station

CPU Central Processing Unit

Contents

List of Figures

List of Tables

List of Algorithms

Chapter 1

Introduction

1.1 Problem description

For LTE and LTE-Advanced protocols, high-speed transmission and larger capacity will be the future trends. In consideration of fulfilling the connections between wireless networks, LAN (Local Area Network), blue-tooth and satellite communication, are applied to 4G networks in pursuance of increasing the spectrum efficiency. OFDM (Orthogonal Frequency-Division Multiplexing) is the key point to avoid frequency selective fading. To achieve high spectrum efficiency and to achieve larger capacity, MIMO technology is applied. MIMO has eminent features which offer significant increment in data throughput and link range without additional bandwidth or increased transmit power [14]. When it comes to MIMO, there are MU-MIMO (Multi User Multiple Input and Multiple Output) and SU-MIMO (Single User Multiple Input and Multiple output). SU-MIMO, MU-MIMO suffer from co-channel interference in general. Even though ZF and MMSE can help to avoid MUI, they result in a reduced throughput or require higher power at the transmitter [26] due to the lack of shared information in MU-MIMO. In this thesis, only SU-MIMO will be studied.

In this chapter, the main methods have been used for reducing the BER (Bit Error Rate) and complexity will be introduced. ML is closely related to the CVP (Closest Vector Problem) in a lattice. The most common used method for CVP is sphere decoding. The CVP is the problem to find the smallest Euclidean distance from the received signal to the signal constellation [14]. In [14], 3 MIMO configurations are studied: 2x2, 2x3 and 2x4 MIMO for Rayleigh fading channels with QPSK (Quadrature Phase Shift Keying). For a modified MMSE detector with SIC (Successive Interference Cancellation) proven in [14] that BER achieves optimal performance without requiring large number of signals. With the utilization of sphere decoding in CVP, the complexity at high SNR (Signal to Noise Ratio) is quite low, but the complexity grows exponentially with the system size M_T [23], where M_T is the number of transmit antennas.

We introduce LR combined with ZF or MMSE. The reduced lattice will highly decrease BER for each channel. The LR method makes the column vectors of the MIMO channel matrix close to mutually orthogonal [24].

ZF or SIC will not preserve the diversity order of the system [20]. However, the augmented LR will have better performance with LLL-SIC (Lenstra–Lenstra–Lovász with Successive Interference Cancellation), the price is to increase the complexity [20], [25], and [30]. This is the reason why we are going to investigate LR further in this thesis.

A correlated model in MIMO for ultra wideband is proposed in [3]. It's for the purpose of robustness and delivering higher data rates without extra power, BW (Bandwidth), and time slots. For indoor environment, it is assumed in [3] that the correlation among the receive antennas and the correlation among the transmit antennas are independent of each other, with the justification that merely ambient environment of antenna array brings to bear on the correlation between array elements; however the coupling effect of antenna is ignored in our model, meanwhile the spatial correlation is our only concern. Hence, the well-known Kronecker model is introduced with fixed transmit and receive correlation matrices. The correlation factors are chosen from 0 to 0.5 (highly correlated). The BER performances of LLL-SIC and MMSE-SIC (MMSE with Successive Interference Cancellation) are compared with ML's BER performance in [30].

The complexity for each algorithm is measured by the average number of flops. Which can be either a multiplication, a division, an addition or a subtraction [28]. For LLL, an integral part of LR, the complexity is polynomial in the size of the input; while for LLL has n-dimonsional lattices with integer input basisvectors with bounded length B, it is terminated after at most $O(n^2 \log B)$ iterations. For a real-valued Gaussian channel model, it is upper bounded in the dimension of the lattice, not applicable for a broadcast precoding case [31].

In the future, SA (Seysen's Algorithm) should be brought into consideration experimental results show in [17] that SA requires significantly fewer iterations than the LLL algorithm. SA implements global vectors searching simultaneously which reduces the lattice basis and its dual, it also leads to a more orthogonal lattice basis compared with LLL. However this case not applicable for LR SIC situation, there will be no such advantage to diminish the number of iterations while having SA algorithm in LLL-SIC.

In this thesis, the basic idea is to combine LR with linear detectors, which gives us the sub-optimal performance with reasonable complexity. For the sake of comparison, the correlated channels with different factors will be introduced.

1.2 Basic MIMO channel models

In order to get an $M_T \times M_R$ channel, we assume M_T transmitter antennas and M_R receiver antennas in our channels as illustrated in Figure 1.1. The input-output relation of the channel can be mathematically described as:

$$\mathbf{x} = \mathbf{Hs} + \mathbf{n}. \tag{1.1}$$

In this channel, we take $\mathbf{s}$ as the complex transmit signal, and $\mathbf{n}$ represents AWGN (Additive White Gaussian Noise) with variance of σ_{n}^2 per element. The channel $\mathbf{H}$ is assumed to be uncorrelated flat fading, which means that it will remain constant for each frame, and independent.

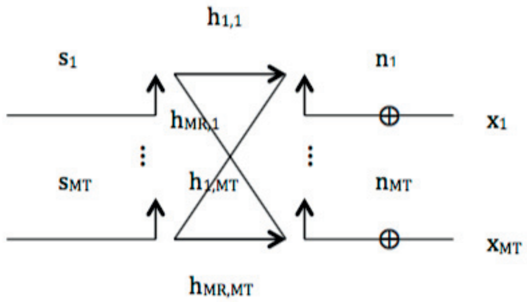


Figure 1.1: MIMO Channel with M_T transmit and M_R receive antennas

In 1.1 quantities are defined as follows $\mathbf{x} = \left[\mathbf{x}_1 \ \mathbf{x}_2 \cdots \ \mathbf{x}_{M_R} \right]^T$, $\mathbf{s} = \left[\mathbf{s}_1 \ \mathbf{s}_2 \cdots \ \mathbf{s}_{M_T} \right]^T$, $\mathbf{n} = \left[\mathbf{n}_1 \ \mathbf{n}_2 \cdots \ \mathbf{n}_{M_R} \right]$, Where $(.)^T$ denotes the transpose of a vector.

$$\mathbf{H} = \begin{bmatrix} \mathbf{h}_{11} & \mathbf{h}_{12} & \dots & \mathbf{h}_{1M_T} \\ \mathbf{h}_{21} & \mathbf{h}_{22} & \dots & \mathbf{h}_{2M_T} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{h}_{M_R1} & \mathbf{h}_{M_R2} & \dots & \mathbf{h}_{M_RM_T} \end{bmatrix}. \tag{1.2}$$

For the data vector $\mathbf{s}$, we use M-QAM (M-Quadrature Amplitude Modulation) information symbols, the transmit power is normalized to 1 for each antenna. I_m is identity matrix, M_T equals to M_R, so the dimension of I_m is $M_T \times M_R$.

$$I_m = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}. \tag{1.3}$$

1.3 Rayleigh Fading Channels with AWGN

Table 1.1: Simulation Parameters

Antenna Configurations	1×1 - 8×8
SNR(dB)	0-40
Channel type	Rayleigh Flat Fading, Kronecker Model
Constellations	4-QAM, 16-QAM, 64-QAM

The simulation parameters that are used in this thesis are illustrated in Table 1.1. The number of antenna varies from 1 to 8, the SNR value ranges from 0 dB to

40 dB; two types of channels are probed: Rayleigh channel and Kronecker model; three kinds of signals are chosen: 4-QAM, 16-QAM and 64-QAM.

Three types of constellation diagrams 4-QAM, 16-QAM and 64-QAM will be discussed, on Figure 1.2, Figure 1.3 and Figure 1.4, respectively. By simply combining PSK (Phase shift keying) with ASK (Amplitude shift keying), an M-QAM symbol will be obtained. QAM symbols have been enormously used in telecommunication systems.

Basically, a Rayleigh fading channel is an i.i.d (independent and identically distributed) Gaussian channel with zero mean and NLOS (Non-line-of-sight). It usually applies to environments like urban areas with many buildings, where NLOS happens, which means that no dominant component between the transmitter and receiver exists, and only scattered signals are included.

The i.i.d channel is composed by real and imaginary parts, each part has variance 0.5, that is to say the total variance of each element $h_{m,n}$ is 1. Meanwhile, the phase of each element is uniformly distributed between 0 to 2π. In other words, the channel is an uncorrelated complex Gaussian channel with unit variance and zero mean, which can be used in urban areas with NLOS.

1.4 Kronecker Model

In general, the correlated channels can be defined as follows:

$$H_{Cor} = R_R^{1/2}\mathbf{H}(R_T^{1/2})^T \tag{1.4}$$

where R_R and R_T are spatial correlation matrices. They are defined differently in 4×4 MIMO and 2×2 MIMO. The covariance matrix of BS and MS are called R_T and R_R separately. The correlation scaling factor for the BS and MS are α and β. The definition of channel spatial correlation matrixis is according to the LTE specification in [35], where various set of antenna combinations (1×1, 2×2, 4×2 and 4×4). However, only 2×2 and 4×4 are taken into consideration in this thesis.

For 2×2 MIMO:

$$R_T = R_R = \begin{bmatrix} 1 & \alpha \\ \alpha^* & 1 \end{bmatrix} \tag{1.5}$$

when it comes to 4×4 MIMO:

$$R_T = R_R = \begin{bmatrix} 1 & \alpha^{1/9} & \alpha^{4/9} & \alpha \\ (\alpha^{1/9})^* & 1 & \alpha^{1/9} & \alpha^{4/9} \\ (\alpha^{4/9})^* & (\alpha^{1/9})^* & 1 & \alpha^{1/9} \\ \alpha^* & (\alpha^{4/9})^* & (\alpha^{1/9})^* & 1 \end{bmatrix} \tag{1.6}$$

1.5 Signal to Noise Ratio

In this section, SNR and BER will be described. For SNR, as it is shown on Table 1.1, it varies from 0 dB to 40 dB to see the trend of changes for different numbers of antennas and all the kinds of linear detection methods to be studied. BER will be shown in logarithmic scale from 10^{-5} to 10^{0}. The overall SNR at the receiver for MIMO is defined as:

$$\rho \triangleq \frac{E\{\|\mathbf{x}\|^2\}}{E\{\|\mathbf{n}\|^2\}} = \frac{E\{\|\mathbf{Hs}\|^2\}}{E\{\|\mathbf{n}\|^2\}} = \frac{E\{\sum_{i=1}^{M_T}\sum_{j=1}^{M_R}|h_{i,j}s_i|^2\}}{E\{\sum_{i=1}^{M_R}n_i^2\}} = \frac{M_T M_R \sigma_{\mathbf{s}}^2}{M_R \sigma_{\mathbf{n}}^2} = \frac{M_T \sigma_{\mathbf{s}}^2}{\sigma_{\mathbf{n}}^2}, \quad (1.7)$$

while

$$E\{| h_{i,j} |^2\} = 1 \quad (1.8)$$

$$E\{| s_i |^2\} = \sigma_{\mathbf{s}}^2 \quad (1.9)$$

$$E\{| n_i |^2\} = \sigma_{\mathbf{n}}^2. \quad (1.10)$$

The SNR per transmit antenna is computed as follows:

$$\rho_{average} = \frac{\rho}{M_T} = \frac{\sigma_{\mathbf{s}}^2}{\sigma_{\mathbf{n}}^2} = \frac{E_s}{N_0}. \quad (1.11)$$

For MIMO channels, the SNR equals M_T times the SNR per transmit antenna, because the antennas receive the imcoming power from M_T number of antennas:

$$\rho_{MIMO} = M_T\frac{E_s}{N_0} = M_T \log_2(M)\frac{E_b}{N_0}, \quad (1.12)$$

where the average energy per bit is $E_b = E_s/\log_2(M)$.

1.6 Average Energy

In order to get the average energy of each signal for 16-QAM, let's calculate the total energy in each quadrant, $E_{16-QAM} = (1^2+1^2)+(1^2+3^2)+(3^2+3^2)+(3^2+1^2) = 40$, since there are 4 in each constellation, the average energy of each symbol is $E_s = 10$. The normalization factor for each symbol then is $1/\sqrt{E_s} = 1/\sqrt{10}$. By proceeding in the same way, the average energy for 4-QAM is 2; when it comes to 64-QAM, the average energy is 42 for each symbol.

1.7 Modulation

Different modulation schemes can carry various types of signals; the information which needs to be transmitted will be carried by modulated carrier signals. QAM is the most well known modulation type used among all the modulation schemes. By increasing the modulation order M, of course more bits per symbol will be conveyed and higher spectral efficiency will be obtained, however, if the mean energy remains constant, this will lead to dissatisfactory results since it will be: less resilient to noise and interference.

So high modulated symbols like 64-QAM and 256-QAM are more often used in downstream channels in both cable modem systems and digital cable systems. In 64-QAM and 256-QAM, they are more sensitive to noise and other sources of impairments which makes the system less reliable and generate higher BER. In this thesis, we will take a close look at the performance of 4-QAM, 16-QAM and 64-QAM, respectively.

1.7.1 4-QAM signals

When it comes to 4-QAM, the definition of the signal set is: A=$\{a + bj\}$ where $a, b \in \mathcal{B}$ where $\mathcal{B} = \{\pm 1\}$. There are 2 bits in each symbol which will illustrated in Figure 1.2

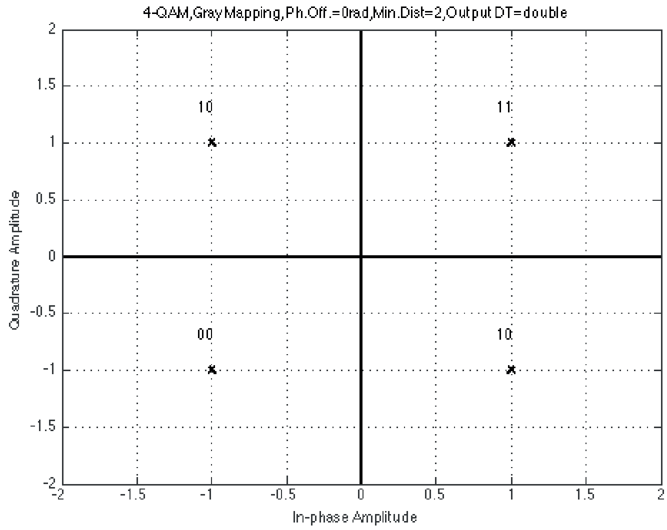


Figure 1.2: 4-QAM Constellation Diagram with Bit Mapping

1.7.2 16-QAM signals

The 16-QAM signal constellation is defined as: A=$\{a + bj\}$ where $a, b \in \mathcal{B}$ where $\mathcal{B} = \pm 1,\ \pm 3$. For each symbol, there are 4 bits, Gray mapping is used to encode and decode both the transmitted and received signals as shown in Figure 1.3.

1.7.3 64-QAM signals

In 64-QAM, each symbol contains 6 bits. The coefficient of the 64-QAM constellation is shown in Figure 1.4. They are defined as follows: A=$\{a+bj\}$ where $a, b \in \mathcal{B}$ where $\mathcal{B} = \pm 1,\ \pm 3,\ \pm 5,\ \pm 7$. 6-bits Gray code as shown in Figure 1.4.

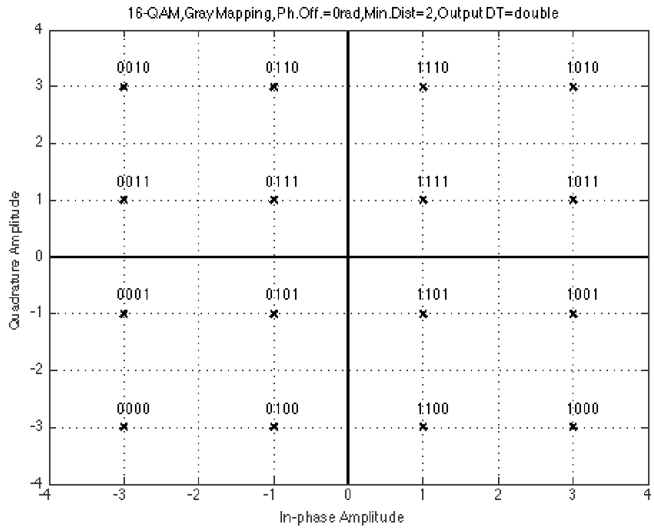


Figure 1.3: 16-QAM Constellation Diagram with Bit Mapping

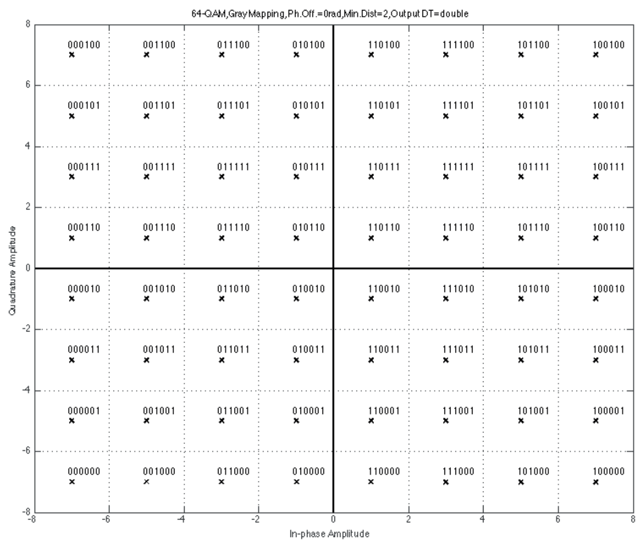


Figure 1.4: 64-QAM Constellation Diagram

1.8 Complex channels to real channels

To get the best results from Lattice reduction, we have to convert the complex channel system to a real-valued system, and the system can be rewritten as:

$$x = Hs + n \tag{1.13}$$

$x = \left[R(\mathbf{x}),\, I(\mathbf{x}) \right]^T$, $H = \left[R(\mathbf{H}),\, -I(\mathbf{H});\, I(\mathbf{H}),\, R(\mathbf{H}) \right]^T$, $s = \left[R(\mathbf{s}),\, I(\mathbf{s}) \right]^T$, $n = \left[R(\mathbf{n}),\, I(\mathbf{n}) \right]^T$.

The dimension of the channel will be doubled to $2M_T \times 2M_R$, $I_{m_{real}}$ is an identity matrix in real-valued system, so the dimension of $I_{m_{real}}$ will be increased to $2M_T \times 2M_R$. $I_{m_{real}}$ is a square matrix since $M_T = M_R$, $I_{m_{real}}$ will be used in our algorithms since real-valued channels are implemented.

$$I_{m_{real}} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix}. \tag{1.14}$$

1.9 Detection Methods

In MIMO, MSI will be the main issue that we will have to deal with. Frequency flat channels are applied in our MIMO system. The number of transmit and receive antennas will be the same for simplicity. ISI is assumed here. In order to cancel MSI, let's discuss linear detection methods like ZF and MMSE. Then, we can move on to non-linear methods and in this thesis we will study lattice aided reduction combined with linear detection in closer detail. We will include channel correlation according to a Kronecker model.

1.10 Modulation and Demodulation Methods

Let's take 4-QAM as an example to iillustrate modulation 1 and demodulation 2 algorithms. By subtracting the received bits with the transmitted bits, BER can be easily obtained by comparing how different those two streams of bits are. This filter will balance the multi-stream interference and noise enhancement very well, in the meantime, the total amount of errors are largely decreased. So it is quite obvious that the MMSE filter is superior to the ZF filter.

During 4-QAM modulation process in Algorithm 1, the first step taken into consideration is to generate corresponding random bits; secondly, We get all possbile symbol conbinations in decimal for both real and imaginary parts; thirdly, the random bits are converted to binary bits; afterwards binary bits are converted to decimal bits; finally they will be converted to Gray coded symbols. After symbols have been through the channel, we would need to consider the process in Algorithm 2.

To simplify the calculation process in Algorithm 2, we treat the real and imaginary equally as long as the same decimal options are provided. Since there are only two possibilities in decimal for 4-QAM in either real or imaginary part: 1 or -1; so as a result, 4 different symbols in total will be involved in the process.

Algorithm 1 Modulation for 4-QAM - s = Mod-QAM(bits, K, D)

```
% create all possible 2-bit sequences for both real and imaginary parts
r = [−1 : 2 : 1];
im = [−1 : 2 : 1];
bits_reshape = reshape(bits, K, D).′;
bin2dec = ones(D, 1) ∗ (2.^[(K/2−1):−1:0]); %binary to decimal
% Real part for 4-QAM
Re_bits = bits_reshap(:, [1 : K/2]);
Re_decbits = sum(Re_bits. ∗ bin2dec, 2);
Re_graydec = bitxor(Re_decbits, floor(Re_decbits/2));
% Imaginary part for 4-QAM
Im_bits = bits_reshape(:, [K/2 + 1 : K]);
Im_decbits = sum(Im_bits. ∗ bin2dec, 2);
Im_graydec = bitxor(Im_decbits, floor(Im_decbits/2));
Re_s = r(Re_graydec + 1);
Im_s = im(Im_graydec + 1);
s = Re_s + 1j ∗ Im_s; % complex signals with gray coding
s=s.'; % s are signals without normalization
end
```

For 4-QAM demodulation algorithm, the received signal will be estimated to it's closest coordinate value in decimal. Thus if the coordinate value is close to 1, then the corresponding Gray bits ='1'; otherwise if the coordinate value is close to -1, the Gray bits ='0'; the final step is to reconstruct it into the right order for received Gray coded bits. By comparing the received bits with the original random bits, the Bit Error Rate can be obtained, which will be discussed later.

1.10.1 Zero Forcing Equalizer

In ZF detection, if the signal is transmitted through MIMO channels with noise, the noise will surely be enhanced. The good thing is that the complexity is low compared with ML. The filter matrix is as follow:

$$s_{ZF} = H^{\dagger} x = (H^T H)^{-1} H^T (Hs + n) \tag{1.15}$$

$$= s + (H^T H)^{-1} H^T n \tag{1.16}$$

where $H^{\dagger} = (H^T H)^{-1} H^T$ represents the Pseudo-inverse of channel H. By multiplying the received signal with the Pseudo-inverse of the channel $H^{\dagger}$, we can get the signal after ZF equalizer s_{ZF} as the signal that has been through ZF filter. Then we can use bit decoder to achieve the received bits. Comparing the received bits with the original ones, the number of wrong bits will be obtained. The bit error rate is calculates as follows:

$$BER = \frac{Total\ errors}{Number\ of\ realizations \times Total\ bits\ in\ one\ realization}. \tag{1.17}$$

Algorithm 2 Demodulation for 4-QAM - [bits] = Demod 4-QAM(S_{hat}, M_T)

Symbollength = length(S_{hat})
temp = [ ]
$r = [-1 : 2 : 1]$ %get all possible coordinate values
$im = [-1 : 2 : 1]$
$s\_dem = 2 * floor(s\_hat/2) + 1;$
$s\_dem(find(s\_dem > max(r))) = max(r);$
$s\_dem(find(s\_dem < min(r))) = min(r);$
$s\_dem = s\_dem.';$
for $i = 1 : symbollength$ **do**
  if $s\_dem(i) == 1;$ **then**
    $temp = [temp; 1];$ % if coordinate value=1, output is 1
  else
    $temp = [temp; 0];$ % if coordinate value=-1, output is 0
  end if
end for
$bits\_hat\_temp = temp;$
$bits\_temp\_Re = bits\_hat\_temp(1 : symbollength/2, :);$
$bits\_temp\_Im = bits\_hat\_temp(symbollength/2 + 1 : end, :);$
$bits\_temp = [bits\_temp\_Rebits\_temp\_Im];$
$bits\_out = reshape(bits\_temp.', 2 * M_T, 1)';$

1.10.2 Minimum Mean-Square error Equalizer

As a well-known method of measuring the mean square error in signal processing, the MMSE equalizer reaches a better performance compared with ZF. MMSE keeps good balance between complexity and BER.

Let's take the expectation of product of the received signal x and the conjugate transpose of x^T, which is $E(xx^T)$, because $E(H^T s^T n) = 0$, so the expectation can be written as:

$$E(xx^T) = E[(Hs+n)(Hs+n)^T] \tag{1.18}$$

$$= HE(ss^T)H^T + \frac{N_0}{2} I_{Mt} \tag{1.19}$$

$$= HH^T + \frac{N_0}{2} I_{m_{real}}. \tag{1.20}$$

The signal after MMSE equalizer s_{MMSE} is defined as:

$$s_{MMSE} = (H^T H + \sigma_n^2 I_{m_{real}})^{-1} H^T (Hs+n) \tag{1.21}$$

$$= (H^T H + \sigma_n^2 I_{m_{real}})^{-1} H^T x \tag{1.22}$$

where $\sigma_n^2 = N_0/2$. In MMSE equalizer, the demodulation function must be called for the purpose of decoding s_{MMSE} into bits, which gives us the received bits $Bits_{MMSE}$. As in Figures 1.2, 1.3 and 1.4, the 4-QAM is a 2-bit Gray-code sequence; similarily, the 16-QAM symbols are combined by 3-bit Gray-code sequence and 64-QAM symbols are constructed with 4-bit Gray-code sequence. In Figures 1.2, 1.3 and 1.4, the Gray coded constellations are exhaustively described for 4-QAM, 16-QAM and 64-QAM.

1.10.3 Maximum Likelihood Equalizer

As it's acknowledged, ML obtains the highest complexity but the best performance, which makes it the optimal method in case of performance. ML detection calculates the Euclidean distance between the received signal vector and the product of all possible transmitted signal vectors with the channel H and finds the one with the minimum distance [6].

$$\hat{s}_{ML} = \arg\min_{s} \parallel x - Hs \parallel^2 . \tag{1.23}$$

In Algorithm 3 and Figure 1.5, the process and BER performance of ML are described. ML detection will add complexity as we increase the modulation order or the number of antennas [6]. However, the performance should be superior to the MMSE equalizer. For example, in 4-QAM maximum likelihood, let's take all possible constellations as samples: A = {1+1j, 1-1j, -1+1j, -1-1j}. Even though we don't know which symbols are transmitted, the coordinate value table can be assessed to decoded the received symbols by ML algorithm in Algorithm 3. Once we get the estimation, the exact symbols can also be estimated, which will be examined in the following paragraph.

In maximum likelihood algorithm for 4-QAM, the metric is set as a fairly large number, then all possible coordinate value combinations for it are listed for comparison factor. As is shown in the 4-QAM maximum likelihood algorithm, we start with taking the first column of the received signal, subtract it from the product of first column from the channel and the one possible symbol chosen from the coordinate value table, this subtraction gives us the estimated received symbol, for more accuracy, the metric which is related to $||x - Hs||^2$ can be calculated, thus we need to compare the received symbols column by column with the coordinate value table, the complexity is extremely high. The $temp_{M_T}$ is to save the correct symbols after a careful comparison. The number of elements to search over this algorithm is M^{M_T}.

1.10.4 Limitations

The complexity of ML increases exponentially with the modulation order and number of antennas, so it will be harder to implement a higher modulation order and more antennas. Performance of ZF and MMSE are not as good as ML, while ML's is optimal. As we mentioned above, the performance that we expected should be as close to ML's performance as possible. But because of the noise enhancement in ZF and lattice reduction achieves higher complexity than MMSE, we would like to find a better solution, which is why we will take a close look at LR combined with linear detectors in next chapter.

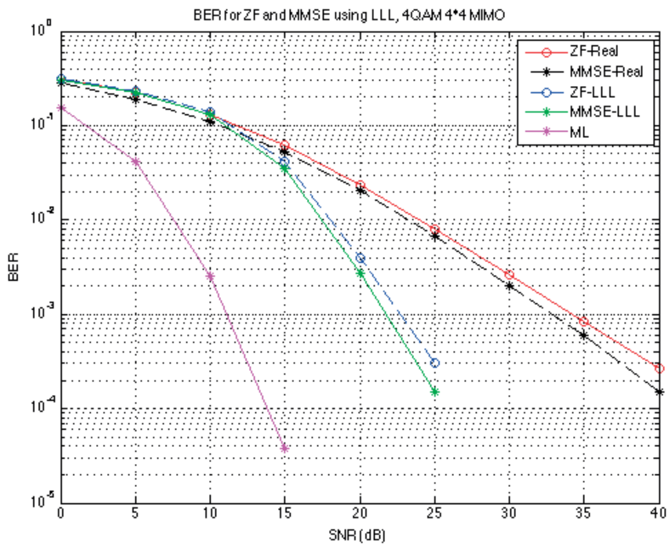


Figure 1.5: Maximum Likelihood for 4-QAM, in 4×4 MIMO channel, 10000 frames

Algorithm 3 ML Algorithm for 4-QAM - $[\hat{x}] \Leftarrow MLdetector(s, H)$

Initialization:

Set $metric \Leftarrow inf$

coordinate values: A = {1+1j, 1-1j, -1+1j, -1-1j}

Iteration:

for all s in A **do**
  $temp(1) = A(s)$
  $est1(:,1) = x - H(:,1). * temp(1)$
  for all s in A **do**
    $temp(2) = A(s)$
    $est2(:,1) = x - H(:,2). * temp(2)$
    for all s in A **do**
      $temp(3) = A(s)$
      $est3(:,1) = x - H(:,3). * temp(3)$
      for all s in A **do**
        $temp(4) = A(s)$
        $est4(:,1) = x - H(:,4). * temp(4)$
        $e \Leftarrow \|x_{real} - H_{real} * shat\|^2$
        $metric_{tmp} = sqrt(est4(:,1)' * est4(:,1))$
        if $metric_{tmp} < metric$ **then**
          $\hat{x} = temp$
          $metric = metric_{tmp}$
        end if
      end for
    end for
  end for
end for
$bits \Leftarrow demod4QAM(s_{hat})$

Chapter 2

Lenstra–Lenstra–Lovász lattice basis reduction

2.1 Brief Introduction

The first time lattice basis reduction introduced was in Lagrange and Hermite. In this chapter, we are aiming to mitigate noise enhancement and to improve the performance closer to ML's. The main reasons why we take further research about lattice basis reduction are for the sake of getting the shortest and the closest vectors. In recent year's research, there are many efficient methods to approach these two goals: LLL, BKZ. However, the most well-known and widely used algorithm is LLL. A lattice is a combination of a group of linearly independent vectors, which are a set of discrete points. For a reduced basis, which directly reduces the vector's lenght to a shorter one; meanwhile, the shorter the basis, the more orthogonal the basis is. In a lattice-reduced matrix for our thesis, only real values are included, which make the dimension of the matrix doubled.

For the purpose of achieving lower BER with lower complexity, we combine LR with linear detection methods: ZF and MMSE. A real-valued system with reduced basis can be defined as follows:

$$x = Hs + n = HTT^{-1}s + n \tag{2.1}$$

$$= \tilde{H}z + n \tag{2.2}$$

$\tilde{H} = HT,\ TT^{-1} = 1,\ z = T^{-1}s$. $\tilde{H}$ is the channel with reduced basis. For the transformation matrix T, $\det(T) = \pm 1$, T is an integer and unimodular matrix, so the inverse of T is also an integer matrix. The channel matrix after LR is a sub-optimal orthogonal matrix with the shortest basis besides ML.

2.2 Lenstra–Lenstra–Lovász Algorithm

In this section, the process of LLL Algorithm will be interpreted. Gram-Schmidt Orthogonalization, Size Reduction, Lovász Condition and Swap process are the four algorithms needed to implement LLL. First, Let's take a look at Gram-Schmidt Orthogonalization.

2.2.1 Gram-Schmidt Orthogonalization

Gram-Schmidt Orthogonalization is a process of transformation, which is converted from sets of linear dependent vectors into unimodular vectors with orthogonal basis, where its process is exhaustively described in Algorithm 4. First, a basis H is defined as

$$H = \{h_1, h_2, ..., h_{M_T}\} \tag{2.3}$$

.

In Algorithm 4, $h_1, h_2, ..., h_{M_T}$ are the columns of H, M_T is the number of transmit antennas, while M(i) are the norm squared of columns of H. For an orthogonal matrix, orthogonal bases are needed, they can be defined as $\mu_1, \mu_2, ..., \mu_{M_T}$.

Here let's first convert the complex channel system into real-valued system by the following formula:

$$H = \left[\, R(\mathbf{H}),\ -I(\mathbf{H});\ I(\mathbf{H}),\ R(\mathbf{H}) \,\right]^T . \tag{2.4}$$

Its Gram-Schmit Orthogonal basis vector is defined as follows:

$$H^T = \{h_1^T, h_2^T, ..., h_{M_T}^T\}, \tag{2.5}$$

which can be written as:

$$\mathcal{H}(:, M_T) = \{\mathcal{H}(:, 1), \mathcal{H}(:, 2), ..., \mathcal{H}(:, M_T)\}. \tag{2.6}$$

The channel after Gram-Schmit orthogonal is $\mathcal{H}$. For the reason of simplicity, $h_1^T, h_2^T, ..., h_{M_T}^T$ will be replaced by $\mathcal{H}(:, 1)$ or $\mathcal{H}(:, 2)$, ..., $\mathcal{H}(:, M_T)$ in Algorithm 8, which actually explains how a specific reduced basis is taken out from the whole channel. Meanwhile, the same expressions like $h_1^T, h_2^T, ..., h_{M_T}^T$ will be remained in the text while we mention reduced basis vector. The coefficients for Gram-Schmidt Orthogonalization are described as below:

$$\mu_{i,j} = \frac{1}{\mathcal{H}_j}\left(\langle h_j,\ h_j\rangle - \sum_{k=1}^{j-1}(\overline{\mu_{j,k}}\mu_{j,k})\mathcal{H}_k\right), 1 < i < j < M_T, \tag{2.7}$$

where $\overline{\mu_{j,k}}$ is the conjugate of $\mu_{j,k}$, $h_j = H(:, j)$,

$$\mathcal{H}_j = H(:, j) - H(:, 1 : (j-1)) * \mu(i, 1 : (j-1)). \tag{2.8}$$

It can also described as:

$$\mu_1 = h_1 \tag{2.9}$$

$$\mu_2 = h_2 - \frac{h_2, \mu_1}{\|\mu_1\|^2}\mu_1 \tag{2.10}$$

$$\mu_3 = h_3 - \frac{h_3, \mu_1}{\|\mu_1\|^2}\mu_1 - \frac{h_3, \mu_2}{\|\mu_2\|^2}\mu_2 \tag{2.11}$$

$$\vdots \tag{2.12}$$

$$\vdots$$

2.2.2 Size Reduction

According to the Gram-Schmidt coefficients, the basis vectors can be judged by an certain condition whether Size Reduction Algorithm 5 should be applied or not. The formula is written as below:

$$| \mu_{k,k-1} |> \frac{1}{2}, \tag{2.13}$$

where this condition in Size Reduction Algorithm 5 is described as:

$$round(\mu(k, k-1)) \sim= 0. \tag{2.14}$$

If the condition above holds, which means its basis we obtained is not the shortest one, at a later time the Size Reduction algorithm should be applied for the sake of achieving a more reduced basis vector; if condition fails then we keep the original one. The basis and the Gram-Schmidt coefficients need to be updated afterwards, basis vectors $\dot{\mathcal{H}}_i$, basis $\dot{h}_i$ and coefficients $\dot{\mu_{i,j}}$ are acquired to save the final data.

The whole process can be stopped by the estimated stage number k,

$$k = max(2, k-1) \tag{2.15}$$

for cases like j= k-2 to 1, we perform action: step -1, the updated Gram-Schmit coefficients will be applied to the updated conditions as follows:

$$| \mu_{k,k-1} |> \frac{1}{2}, \tag{2.16}$$

until the shortest vectors are obtained.

The size reduction process in Algorithm 5 is based on the condition we mentioned above, H can easily converted into a reduced basis. If $q = round(\mu(k, i)) > 0$, subtract each basis with the previous basis; if $q < 0$, the original basis is kept, then we update the coefficient μ. Next step is to check the Locasz Condition, let's move to the next section.

2.2.3 Lovász Condition

In 1982, as the concept of LLL Algorithm was first mentioned in [32], Lovász condition was defined as follow:

$$|h_k^T + \mu_{k,k-1} h_{k-1}^T|^2 \geq \delta |h_{k-1}^T|^2, \ \delta = 0.75. \tag{2.17}$$

Algorithm 4 Gram-Schmidt Orthogonalization - $[H,\ \mu,\ i]$[34]

Initialization:

Lattice Basis H: $H = [\, h_1\ h_2 \cdots\ h_{M_T} \,]$,

Iteration:

$\mathcal{H}(:,1) = H(:,1)$
% set the first column of $\mathcal{H}$ same as H's first column
$M(1) \Leftarrow$ norm squared of the first column of $\mathcal{H}$
for $i = 2 : m$ **do**
  $\mu(i, 1 : (i-1))= (H(:,i)')^{*}\ \mathcal{H}(:,1;(i-1))).\!/M(1 : (i-1))$
  $\mathcal{H}(;,i) = H(:,i) - H(:,1 : (i-1)) * \mu(i, 1 : (i-1))'$
  $M(i) = dot(H(:,i), H(:,i))$
end for

Algorithm 5 Size Reduction Algorithm - $[H,\ k,\ i]$[34]

Initialization:

Lattice Basis H: $H = [\, h_1\ h_2 \cdots\ h_{M_T} \,]$,

Iteration:

***S*ize Reduction**
for $i = k-1 : -1 : 1$ **do**
  q = round(μ(k,i));
  if $q \sim= 0$ **then**
    $H(:,k)$ =$H(:,k)$ - $q * H(:,i)$ // Reduce the size of the k-th basis vector
  end if
end for

h_k^T is defined in Equation 2.5. So as a common sense, when people talk about LLL algorithm, they mean LLL algorithm with factor $\delta = 0.75$. According to [33], the Lovász condition can be rewritten as

$$|h_k^T|^2 \geq (\delta - \mu_{k,k-1}^2)|h_{k-1}^T|^2 \tag{2.18}$$

for each basis vector, where $1 \leq k \leq M_T$, while the corresponding process is shown in Algorithm 6. Where if we take the whole channel into consideration, it can be rewritten as in Algorithm 6:

$$|\mathcal{H}(:,k), \mathcal{H}(:,k)| \geq (\delta - abs(\mu(k,k-1))^2) * |\mathcal{H}(:,(k-1)), \mathcal{H}(:,(k-1))|. \tag{2.19}$$

Algorithm 6 Locasz Condition Algorithm - $[H,\ k,\ \delta]$[34]

Initialization:

1: $|\mathcal{H}(:,k), \mathcal{H}(:,k)| \geq (\delta - abs(\mu(k,k-1))^2) * |\mathcal{H}(:,(k-1)), \mathcal{H}(:,(k-1))|$

The step after Size Reduction is to check if Lovász Condition holds. If the Lovász condition holds, perform this command line: k=k+1; if not, Swap process will be implemented, meanwhile, the coefficient μ must be updated. Otherwise, redo the Size Reduction process for k ranges from $k-1$ to k and update the coefficient μ. For k bigger than 2, the following command line will be executed: k = k-1.

2.2.4 Swap process

As we mentioned above, if the Lovász condition fails, we need to swap the two vectors for basis vectors according to Algorithm 7. Swap process must be applied to ensure the norm of the basis vectors won't decrease too much. After the Swap process, it should continue checking if Lovász condition fulfills . If the Lovász condition successed, ultimately the coefficients will be updated, otherwise it continues with Swap process until the Lovász condition holds.

Algorithm 7 Swap process - $[H,\ k]$[34]

Initialization:

1: ***S*wap the k-th and (k-1)-th basis vector**
2: $V = H(:,k)$
3: $H(:,.k) = H(:,k-1)$
4: $H(:,k-1) = V$

Finally, we will achieve the reduced basises which are shorter and more orthogonal.

2.3 Lenstra–Lenstra–Lovász Algorithm Analysis

Algorithm 8 LLL Reduction Algorithm by K. Shum- $[H,\ \delta]$ [34]

Initialization:

Factor δ:
if $nargin == 1$ **then**
  $\delta = 0.75$
end if
m: Vector space's dimension
$\mathcal{H}$: Vectors after Gram-Schmidt Process
μ : Gram-Schmit coefficients
M: M(i) is the norm squared of i-th column of $\mathcal{H}$

Iteration:

***G*ram-Schmidt Orthogonalization**
$k = 2$
while $k <= 2$ **do**
  ***S*ize Reduction**
  μ(k,1:i)=μ(k,1:i)-q*[μ(i,i:(i-1)) 1]// update Gram-Schmit coefficients
end while
Check the Lovasz Condition
if Lovasz Condition holds **then**
  k = k + 1
else
  The Lovasz Condition fails
  ***S*wap the k-th and (k-1)-th basis vector**
  ***U*pdate the Gram-Schmit coefficients**
  for $s = (k-1) : k$ **do**
    $\mu(s, 1 : (s-1)) = ((H(:, s)') * \mathcal{H}(:, 1 : (s-1)))./M(1 : s-1)$
    $\mathcal{H}(:, s) = H(:, s)$ - $\mathcal{H}(:, 1 : (s-1)) * \mu(s, 1 : (s-1))$
    $M(s) = dot(\mathcal{H}(:, s), \mathcal{H}(:, s))$
  end for
  $\mu((k+1) : m, (k-1) : k) = (H(:, (k+1) : m)' * \mathcal{H}(:, (k-1) : k))/diag(M((k-1) : k))$ // updated Gram-Schmit coefficients
  if $k > 2$ **then**
    k = k-1
  end if
end if

Now we can draw a full picture of the LLL process according to Algorithm 8, before that some initializations must be done: first we initialize $\delta = 0.75$, let's set m as vector space's dimension, $\mathcal{H}$ is the vectors after Gram-Schmit process, and μ is used to save Gram-Schmit coefficients, lastly we treat M as the norm squared of the i-th column of $\mathcal{H}$.

Then we start with the main process, before the Gram-Schmidt Orthogonalization was applied.

1. Set the first column of $\mathcal{H}$ as the same value as $H's$ first column; then M(1) was set to save the norm squared of the first column of $\mathcal{H}$;
2. For i from 2 to M_T, all columns of the vectors of Gram-Schmit process and Gram-Schmit coefficients and norm squared of $\mathcal{H}$ are saved, repectively;
3. When k $\leq$ 2, we apply size reduction to the k-th basis to get the size-reduced vectors and the Gram-Schmit coefficients will also be updated;
4. Lovasz condition will be checked afterwards in order to ensure it holds if we increase k; if Lovasz condition fails, we swap the k-th and (k-1)th basis until the Lovasz condition holds again, then all Gram-Schmidt coefficients will be updated;
5. If k is larger than 2, we apply equation k = k -1; then we continue the Gram-Schmidt process until Gram-Schmit coefficients, $\mathcal{H}$ and M are all updated.

After taking the view of whole LLL's process and functionality, let's move to the application part in the following chapter.

Chapter 3

Linear Detections combined with Lattice Reduction

3.1 Linear Detections combined with LR

In this chapter we combine linear detections with lattice aided reduction to give a lower noise enhancement and better performance.

3.1.1 ZF Estimator with LR

In ZF detector, the ZF pre-filtering was yielded by multiplying the inverse of transformation matrix T with the signal after ZF equalizer:

$$Z_{ZF\_LR} = T^{-1} s_{ZF} \tag{3.1}$$
$$= T^{-1}(s + (\tilde{H}^T \tilde{H})^{-1} \tilde{H}^T n)s. \tag{3.2}$$

As is shown above, we can obtain our desired signal $\hat{s}_{ZF\_LR}$ by multiplying Z_{ZF_LR} with the transformation matrix.

$$\hat{s}_{ZF\_LR} = T Z_{ZF\_LR}. \tag{3.3}$$

Matrix T is an integer and unimodular matrix. After going through the filter of ZF, Z_{ZF_LR} will be received and then rounded to its closest constellation due to the constellation table in related demodulation algorithms: a simple demodulation example is shown Algorithm 2; during demodulation process, we get both received signals s_{ZF_LR} and its originally sent bits.

3.1.2 MMSE Estimator with LR

For the case in MMSE, we would like to take a look at the expectation of following product:

$$E(zz^T) = E[T^{-1}s(T^{-1}s)^T] \tag{3.4}$$
$$= T^{-1}E(ss^T)(T^{-1})^T \tag{3.5}$$
$$= T^{-1}(T^{-1})^T, \tag{3.6}$$

$$E(xx^T) = E[(\tilde{H}z + n)(\tilde{H}z + n)^T] \tag{3.7}$$
$$= \tilde{H}E(zz^T)\tilde{H}^T + \frac{N_0}{2} I_{M_T\_real} \tag{3.8}$$
$$= \tilde{H}T^{-1}(T^{-1})^T\tilde{H}^T + \frac{N_0}{2} I_{M_T\_real} \tag{3.9}$$
$$= \tilde{H}\tilde{H}^T + \frac{N_0}{2} I_{M_T\_real}. \tag{3.10}$$

Then the pre filter of MMSE turns out to be:

$$Z_{MMSE\_LR} = T^{-1}s_{MMSE} \tag{3.11}$$
$$= T^{-1}(H^TH + \sigma_n^2 I_{M_T\_real})^{-1}H^Tx \tag{3.12}$$
$$= (\tilde{H}^T\tilde{H} + TT^T\sigma_n^2 I_{M_T\_real})^{-1}\tilde{H}^Tx. \tag{3.13}$$

The received signal which combines LR with MMSE is shown as follow:

$$\hat{s}_{MMSE\_LR} = TZ_{MMSE\_LR} \tag{3.14}$$
$$= T(\tilde{H}^T\tilde{H} + TT^T\sigma_n^2)^{-1}\tilde{H}^Tx. \tag{3.15}$$

Z_{MMSE_LR} is the signal received after MMSE filter. In order to get the exact sent symbols and bits, Z_{MMSE_LR} must be decoded through corresponding demodulation method. After demodulation process, the estimated received symbol is $\hat{s}_{MMSE\_LR}$ and its received bits.

3.2 Performance analysis

With the calculation of BER, we can organize all the BER performance related to SNR in the same figure for comparison.

3.2.1 4-QAM

The comparison between different number of antennas is shown in Figure 3.1. 10000 realizations of 4-QAM signals are transmitted through channels with various numbers of antennas. The BER and gain are considered in those cases:

1. The BER for linear detection methods decreases with the increase of transmit or receive antenna number, it changes around 10^{-3}. The BER reaches its best performance for 4-QAM in 4×4 MIMO at 40 dB, which is close to 10^{-4}; when the number of antennas is larger than 4, the performance becomes less satisfying. For cases of Lattice Reduction, its best record will be at 31dB in 6×6 MIMO for BER 10^{-5};
2. In cases with only linear detectors, the gain over ZF and MMSE in 2×2 MIMO is close to 0 dB, and in 8×8 case, the gain increases to 3 dB. There is not much differences in all cases of Lattice Reduction;
3. For situations with LR, its BER is much smaller than cases with pure linear detectors, which successfully overcomes the imperfect orthogonalization problem, whose BER performance benefits a lot from the reduced basis vectors;

4. The gap between ML and ZF is the largest, then comes MMSE without LR, the third is ZF with LR, the smallest gap goes to MMSE with Lattice Reduction, which implies it achieves the best performance in this scenario and the closest performance to ML's; ML in 4×4 MIMO with 4-QAM has its SNR around 20 dB while reaching BER 10^{-5}, so MMSE in 6×6 with LR is the sub-optimal solution compared with ML.

3.2.2 16-QAM

The performance of 16-QAM with antenna number ranging from 1 to 8 are illustrated in Figure 3.2.

1. In SISO channel, the performance of ZF and MMSE are the same for both LR and witout LR detectors.
2. When it comes to the MIMO scenario, for those cases without LR, it's best BER for 40 dB is around 10^{-3} for 4×4 MIMO; and for cases with LR, the most eye-catching performance is at 35 dB when the BER is 10^{-5} in 6×6 MIMO.
3. There is only a 1 dB gain between ZF and MMSE in both with and without LR cases.
4. The performance in ML with 16-QAM is a bit worse than ML with 4-QAM, it reaches BER of 10^{-5} around 27 dB, while 20 dB for 4-QAM. And the result from 6×6 MIMO is pretty close to ML's performance, which can be regard as the sub-optimal performance in 16-QAM.

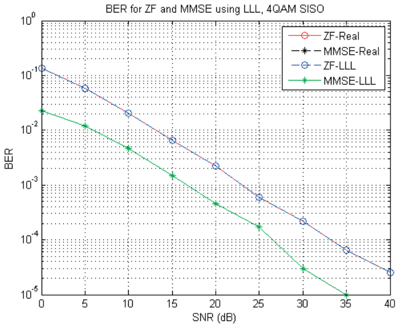
3.2.3 64-QAM

In Figure 3.4, the comparison for different number of antennas with 64-QAM will be shown, where 1 to 8 antennas will be applied in those cases.

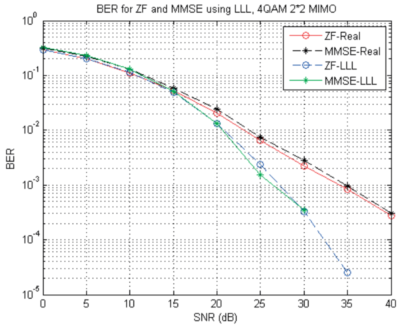
1. For linear detectors, their peak performance arrives at somewhere between 10^{-3} and 10^{-4} for 50 dB in 4×4 MIMO.
2. For LR combined with linear detection methods, the results are improved compared with linear detection methods, the best result for BER is close to 10^{-4} where SNR is 40 dB, while the MMSE with LR was carried out in 6×6 MIMO.
3. The BER for MMSE with LR in 6×6 MIMO can be considered as the sub-optimal solution compared with ML's performance.

3.2.4 Comparison between 4-QAM, 16-QAM and 64-QAM

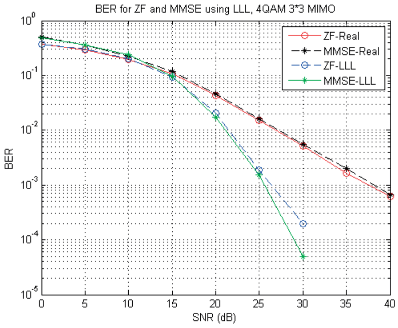
1. Among all those modulation methods, the best performance goes to ML method, and the sub-optimal performance is MMSE combined with Lattice Reduction in 6×6 MIMO with 4-QAM, which is the closest to the performance of ML.



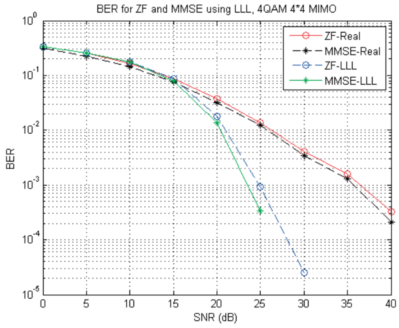
(a) SISO, 1×1



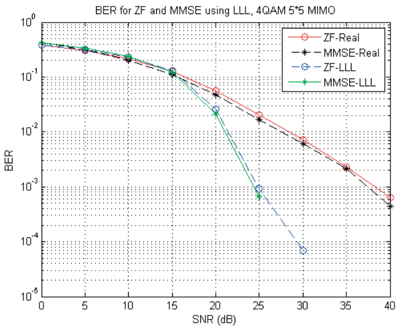
(b) 4-QAM, 2×2



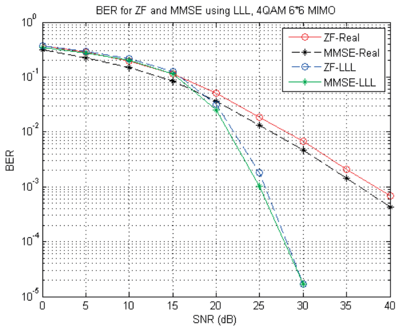
(c) 4-QAM, 3×3



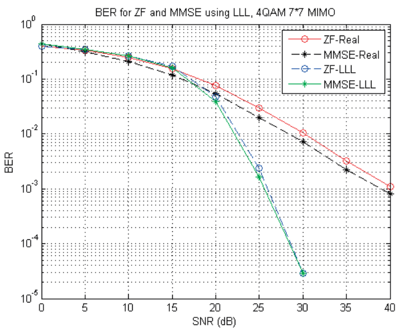
(d) 4-QAM, 4×4



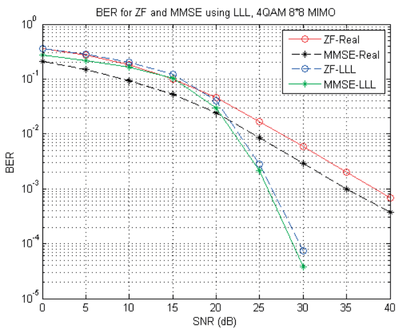
(e) 4-QAM, 5×5



(f) 4-QAM, 6×6

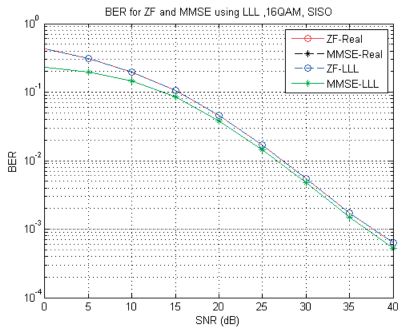


(g) 4-QAM, 7×7

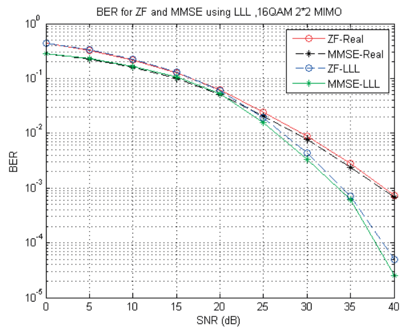


(h) 4-QAM, 8×8

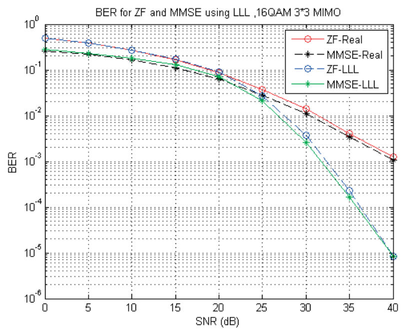
Figure 3.1: Figure for comparison between different amount of antenna numbers for 4-QAM, the number of antennas varies from 1 to 8, 10000 frames



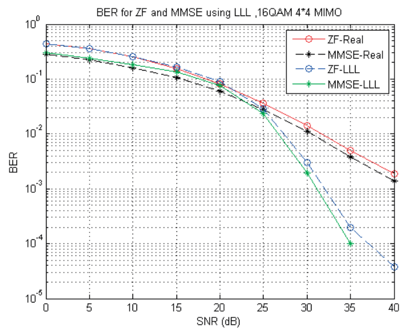
(a) 16-QAM, 1×1



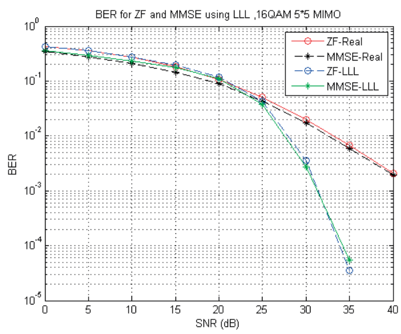
(b) 16-QAM, 2×2



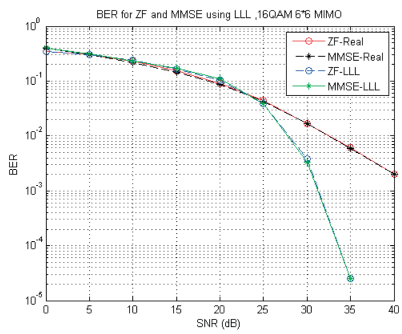
(c) 16-QAM, 3×3



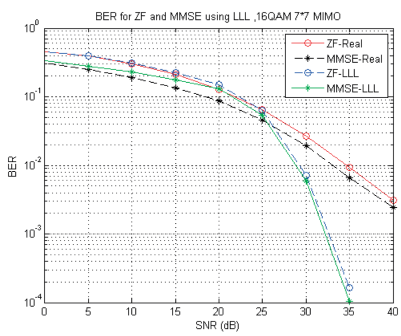
(d) 16-QAM, 4×4



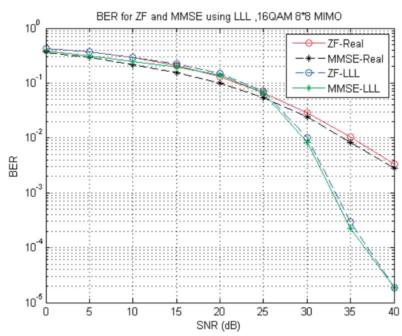
(e) 16-QAM, 5×5



(f) 16-QAM, 6×6



(g) 16-QAM, 7×7



(h) 16-QAM, 8×8

Figure 3.2: Figures for comparison between different amount of antenna numbers for 16-QAM with Maximum Likelihood algorithm, the number of antennas varies from 1 to 8, 10000 frames

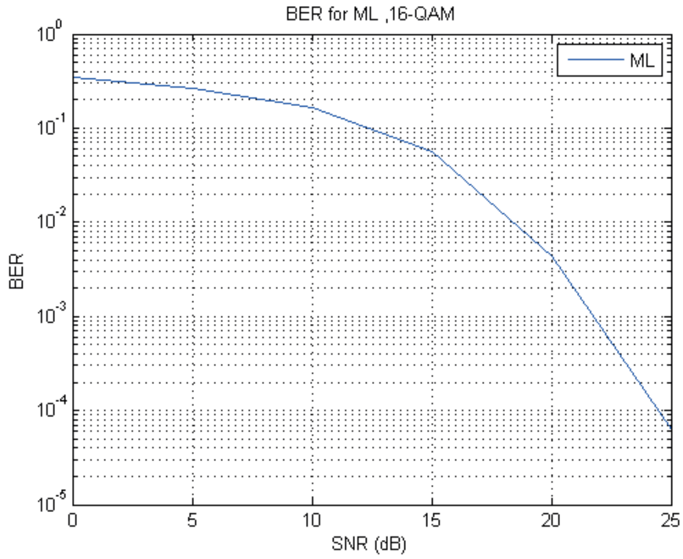
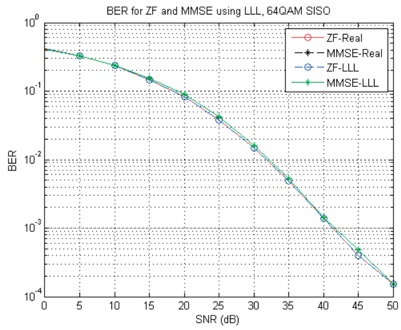
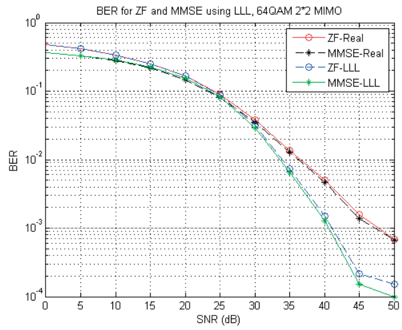


Figure 3.3: Maximum Likelihood for 16-QAM in 4×4 MIMO channel, 10000 frames

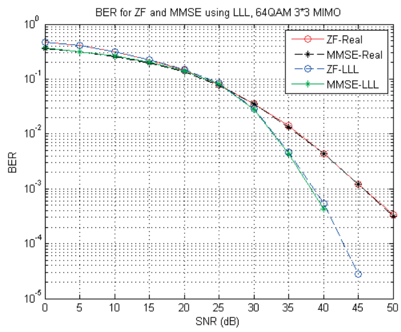
2. The detection methods combined with LR have largely affected how close the performance can reach ML, which are obviously better.
3. The larger modulation, the worse performance, which means the 4-QAM symbols achieve the best performance among 4-QAM, 16-QAM and 64-QAM.



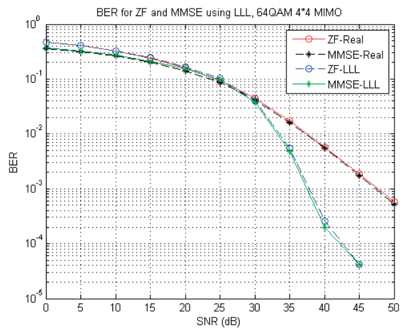
(a) 64-QAM, 1×1



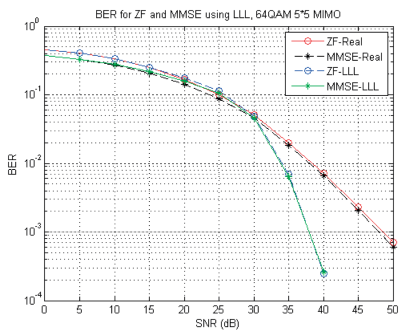
(b) 64-QAM, 2×2



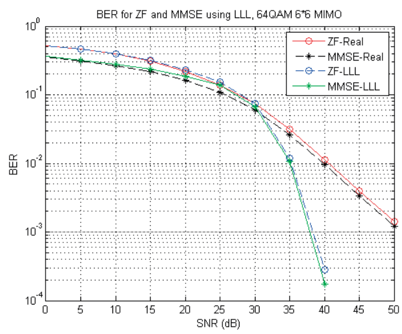
(c) 64-QAM, 3×3



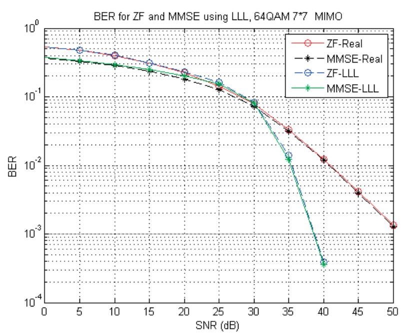
(d) 64-QAM, 4×4



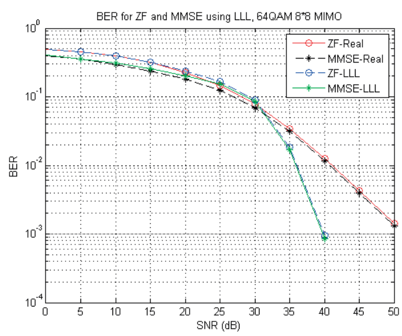
(e) 64-QAM, 5×5



(f) 64-QAM, 6×6



(g) 64-QAM, 7×7



(h) 64-QAM, 8×8

Figure 3.4: Figures for comparison between different amount of antenna numbers for 64-QAM, the number of antenna varies from 1 to 8, 10000 frames

Chapter 4

Correlated LTE channels

4.1 Correlated LTE channels with Kronecker Model

First, we would like to review correlation factors in Table 4.1.

Table 4.1: Correlation levels for MIMO

Correlation Level	Value
No	0
Low	0.1
Medium	0.3
High	0.5

The correlated channels we used is:

$$H_{Cor} = R_R^{1/2}\mathbf{H}(R_T^{1/2})^T \tag{4.1}$$

The covariance matrix of BS (Base Station) and MS (Mobile Station) are as follows: for 2×2 MIMO,

$$R_T = \quad R_R = \quad \begin{bmatrix} 1 & \alpha \\ \alpha^* & 1 \end{bmatrix}, \tag{4.2}$$

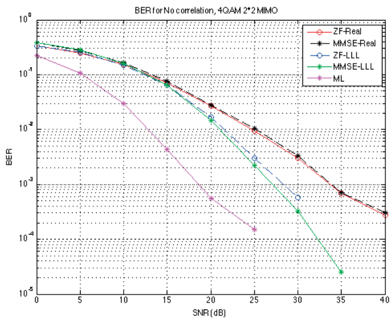
when it comes to 4×4 MIMO:

$$R_T = \quad R_R = \quad \begin{bmatrix} 1 & \alpha^{1/9} & \alpha^{4/9} & \alpha \\ (\alpha^{1/9})^* & 1 & \alpha^{1/9} & \alpha^{4/9} \\ (\alpha^{4/9})^* & (\alpha^{1/9})^* & 1 & \alpha^{1/9} \\ \alpha^* & (\alpha^{4/9})^* & (\alpha^{1/9})^* & 1 \end{bmatrix}. \tag{4.3}$$

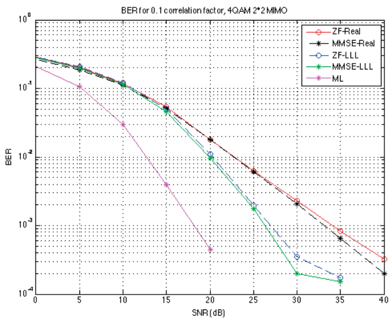
4.2 Performance analysis

4.2.1 2×2 MIMO with 4-QAM symbols in LTE channels

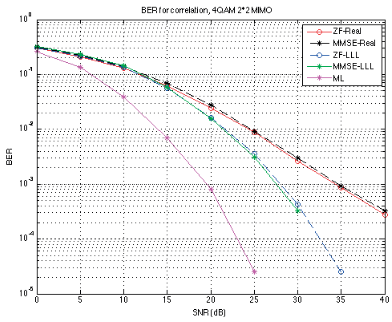
In this section, four typical correlation values will be compared. With different α values, three different R_T and R_R will be obtained. Let's take a look at Figure 4.1:



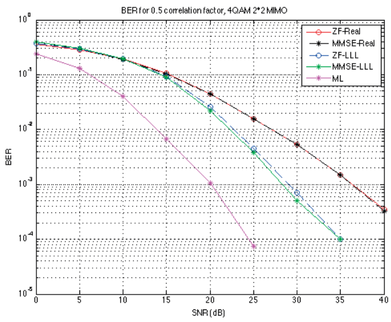
(a) No Correlation



(b) Low Correlation



(c) Medium Correlation



(d) High Correlation

Figure 4.1: The comparison of different coefficient factors range from low to high for 2×2 MIMO with ML, 4-QAM signal, 10000 frames

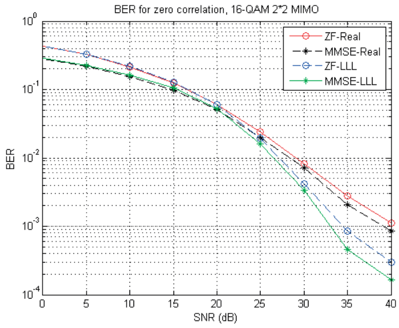
In Figure 4.1, it is a 2×2 MIMO where 4-QAM signals are transmitted, there are no much difference between different coefficient factors, no matter with high correlated factor or low correlated factor. Meanwhile, the high correlated model has the best performance for both linear detectors and the ones with LR, the performance for MMSE with LR and ZF with LR are similar. As acknowledged, the LR will still obtains superior outcomes to the linear detectors. Even the performance of ML didn't vary too much for each case.

4.2.2 2×2 MIMO with 16-QAM symbols in LTE channels

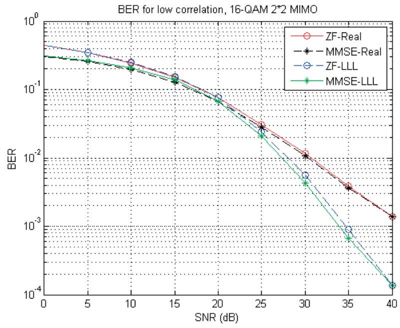
In Figure 4.2, different coefficient factors are applied to 2×2 MIMO with 16-QAM symbols which has 10000 frames.

1. As is shown in Figure 4.2, the MMSE with LR is superior to other detection methods. But the coefficient factors don't influence much of the performance of each method.
2. The results vary from no correlation to high correlation don't affect much between different detection methods. They have similar performance in each coefficient factor, but as for the individual situation, the best award

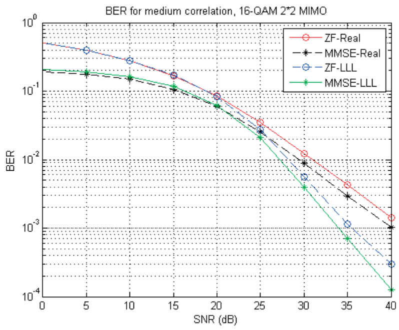
still remains for the Lattice-Aided Reduction combined with MMSE, then LR with ZF, and the worst is with pure ZF detection method.



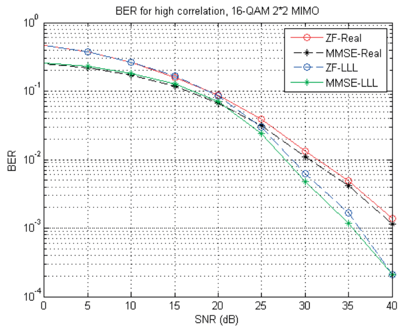
(a) No Correlation



(b) Low Correlation



(c) Medium Correlation



(d) High Correlation

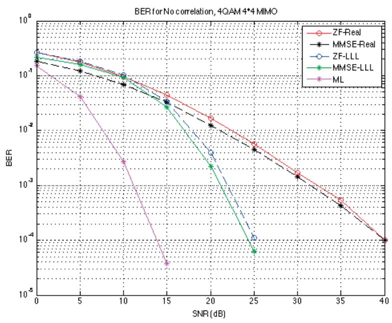
Figure 4.2: The comparison of different coefficient factors range from low to high for 2×2 MIMO with 16-QAM signals, 10000 frames

4.2.3 4×4 MIMO with 4-QAM symbols in LTE channels

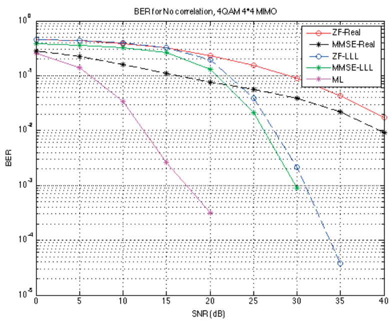
The comparison for 4-QAM and 16-QAM in 4×4 MIMO with various coefficient factors are shown in Figure 4.3.

1. In sub-figures a, b, c and d, the BER increase enormously while larger coefficient factors are added. The less correlated the channel is, the better BER it will obtain, which means the best results are in the cases without any coefficient factors, which means sub-figure a obviously is sub-optimal to ML's performance in this case.

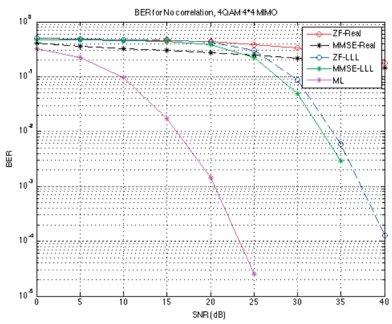
2. For cases with LR, they suffer approximately 25 dB loss if we compare the cases without any correlated channels with the ones with high correlation factors. As for the linear detectors, the gain losses are unpredictable. For example in high correlation case with ZF detector, the BER almost reaches 0.5 which means half of the transmission went wrong.



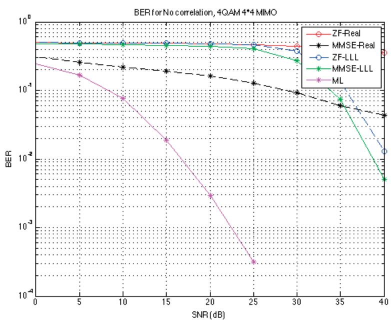
(a) No Correlation



(b) Low Correlation



(c) Medium Correlation

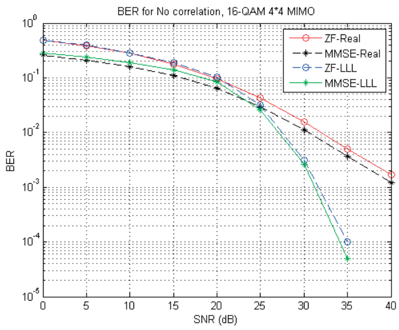


(d) High Correlation

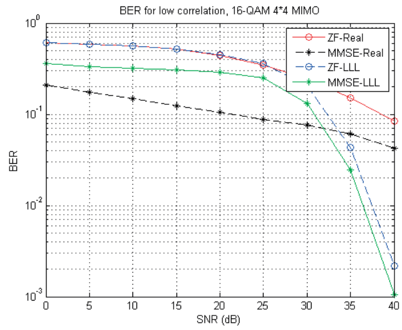
Figure 4.3: BER performance for 4-QAM in LTE channel compared with ML, in 4×4 MIMO channel, 10000 frames

3. The coefficient factors have great impact in 4×4 MIMO, which causes the worse performance in high correlated channels. The 4×4 MIMO channels are very sensitive to the correlation factors.

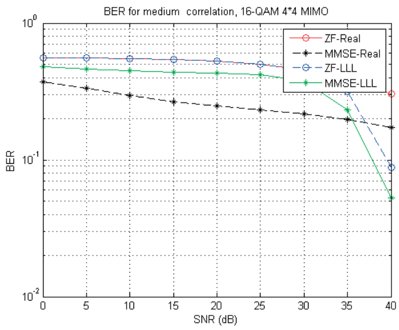
4.2.4 4×4 MIMO with 16-QAM symbols in LTE channels



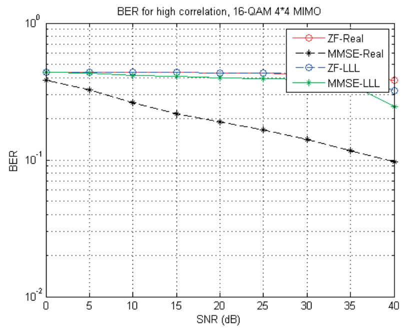
(a) No Correlation



(b) Low Correlation



(c) Medium Correlation



(d) High Correlation

Figure 4.4: The comparison of different coefficient factors range from low to high for 4×4 MIMO with 16-QAM signals, 10000 frames

Then 16-QAM situation in Figure 4.4 can be discussed as shown below:

1. With higher modulation order, the BER increases enormously. In high correlated channels, there is no difference between the four methods we've discussed above. All transmitted bits go wrong. With only linear detectors, half of the received bits can be wrong which makes the channels useless.

2. When the coefficient factor increases, the performance differences between ZF and MMSE are more obvious, which gives MMSE a much better result ever compared with ZF; on the other hand, it makes no difference between the ones with LR, their performance in high correlated channels are almost the same as ZF.

Chapter 5

Complexity Analysis

5.1 Complexity Analysis for MIMO

The complexity for each algorithm basically depends on the average number of flops. It can be either a multiplication, a division, an addition or a subtraction [28]. For LLL, the complexity is polynomial in the size of input; while for termination of LLL, it is bounded to $O(n^2)$ iterations. For real-valued Gaussian channel model, it is uppper bounded polynomial in the dimension of the lattice, not applicable for a broadcast precoding case [31].

	ZF(s)	MMSE(s)	ZF with LR(s)	MMSE with LR(s)
Zero Correlation	118	117	523	530
Low Correlation	117.5	113.1	857	946
Medium Correlation	105.5	106.5	903	1052
High Correlation	120.4	111.9	1063	1270

Table 5.1: Complexity camparison for 4-QAM in 4*4 MIMO channel between different algorithms, Total CPU time

Instead of measuring the number of flops that consumed by each algorithm, we use an easier way of comparison: to use command profile in Matlab to get the self time of each algorithm with different combinations, where their total CPU time is shown in Table 5.1. Since it has 10000 frames, to create 10000 channels itself takes huge part in time consuming, another element which occupies the most of the time is LLL algorithm. Apart from that, the linear detectors' selftimes and linear detectors' selftimes which are combined with LLL algorithm consumes loads of time, the total CPU (Central Processing Unit) time is shown in Table 5.1. Depend on how highly the channels are correlated, the algorithms' CPU time performance vary from each other while they have LTE channels instead of Rayleigh channels.

1. With or without LR, MMSE expends more time than ZF;
2. For both ZF and MMSE, the ones with LR depletes way more time than the ones without;

3. The most time consuming method among all the combinations is MMSE-LLL, and the one with best performance is MMSE, but MMSE-LLL depletes longer time than ZF-LLL;

4. Above all, ML consumes the highest amount of time with the optimal performance. The CPU time of ML is 1161s, while the other detection methods demand much less time if the channels are without correlation; when they're with high correlation factor, ML consumes less time than linear detection methods. The sub-optimal performance is MMSE-LLL, so it's a trade-off.

Chapter 6

Conclusions

In this thesis, two different types of channel models are studied: Rayleigh fading channels and Kronecker model. For both channels, ML's BER performance is superior to all other kinds of methods, including ZF, MMSE, ZF-LLL and MMSE-LLL, where the BER performances can be summarized as:

$$ML > MMSE - LLL > ZF - LLL > MMSE > ZF. \tag{6.1}$$

For Rayleigh fading channels, the linear detection methods achieved their sub-optimal solution at MMSE in 4×4 MIMO for 4-QAM, 16-QAM and 64-QAM; the higher the modulation order is, the worse BER performance can be observed, which means 4-QAM obtains the best performance among 4-QAM, 16-QAM and 64-QAM. When it comes to linear detection methods combined with LR, with reduced basises in 6×6 MIMO, where the performances are highly improved compared with pure linear detectors, MMSE-LLL achieves better performance than ZF-LLL,

For Kronecker model, the same performance rule applies:

$$ML > MMSE - LLL > ZF - LLL > MMSE > ZF. \tag{6.2}$$

Only 2×2 MIMO and 4×4 MIMO are considered in this thesis, where four different coefficient factors are introduced. In 2×2 MIMO, 4-QAM signals' performance is better than the ones in 16-QAM; but the coefficient factors don't have much impact on the performances, so all in all, the MMSE-LLL brings out the sub-optimal solution in this case. While in 4×4 MIMO, the channels are very sensetive to the coefficient factors. The larger coefficient factor, the worse BER performance will be observed, so the one with no correlation brings in the best performance among all four factors, where in no correlated channels, the MMSE-LLL obtains the closest performance to ML.

Consequently, MMSE-LLL with 4-QAM symbols, in 6×6 MIMO Rayleigh fading channels, has the sub-optimal performance compared with ML.

Chapter 7

Future Work

As we mentioned before in [8], 4 basic ways will be introduced to reduce bases: Minkowski Minima bases, LLL-reduced bases, HKZ-reduced bases and optimally reduced bases. It is hard to accompolish ideal results from the Minkowski method, even though it's perfectly defined, but not applicable.

In this paper, there is one improved method which can give us a much shorter basis than HKZ and LLL. It is a suitable method where optimal short basis is required. This method aims to achieve the shortest nonzero lattice vector by a transformed basis which combine the shortest nonzero lattice with another basis.

Another common method to qualify the basis is if the matrix condition number is smaller. But in the future, linear independent number can taken into consideration, if it is close to one, which can give us better performance than the matrix condition number. Another evidence which can prove that the shorter basis vector is more orthogonal than other methods, the determinant value of this lattice basis will not change.

But still due to its high computational cost, some future studies that have to be done to deal with unimodular matrix transformation and linear independent number on applications [8].

One addition way to get improved results is to apply BKZ Algorithm, which is considered as the leading technique in LR area. In [36], experimental results with the application of BKZ are studied with the resource offered by the NTL library.

Bibliography

[1] Dirk Wübben, Ronald Böhnke, Volker Kühn, and Karl-Dirk Kammeyer, *Near-Maximum-Likelihood Detection of MIMO Systems using MMSE-based Lattice-Reduction*, Department of Communications Engineering, University of Bremen

[2] Sabuj Sarkars, Mohammad Shaifur Rahman, *Optimal BER in MIMO Raleigh Fading Channel from QPSK Modulation: Modified MMSE Versus ML Equalizer Evaluation*, International Journal of Electronics and Informatics, Vol. 2, No. 1, 2013

[3] J.Adeane, W.Q Malik, I.J Wassell and D.J Edwards, *Simple correlated channel model for ultrawideband multiple-input multiple-output systems*, IET Microw. Antennas Propag, 2007, 1, (6), pp. 1177-1181

[4] Kamal A. Singhal, Tanumay Datta, and A. Chockalingam, *Lattice Reduction Aided Detection in Large-MIMO Systems*, Department of ECE, Indian Institute of Science, Bangalore

[5] Alan T. Murray and Steven R. Weller, *A Digital Signal Processing Architecture for Soft-output MIMO Lattice Reduction Aided Detection*, http://www.intechopen.com/books//

[6] Yong Soo Cho, Jaekwon Kim, Won Young Yang and Chung G. Kang, *MIMO-OFDM Wireless Communications with MATLAB*, 2010 John Wiley and Sons (Asia) Pte Ltd

[7] Wikipedia, en.wikipedia.org/

[8] Franklin T. Luk, Kowloon Tong, Sanzheng Qiao and Wen Zhang, *A Lattice Basis Reduction Algorithm*

[9] Claud Oestges, *Validity of Kronecker Model for MIMO Correlated Channels*, Microwave Laboratory, Univertsite catholique de Louvain

[10] Arogyaswami Paulraj, Rohit Nabar and Dhananjay Gore, *Introduction to Space-Time Wireless Communications*, Cambridge University Press

[11] Andreas F. Molisch, *Wireless Communications, Second Edition*, A John Wiley and Sons, Ltd, Pubulication

[12] David L. Milliner and John R.Barry, *A Lattice-Reduction-Aided Soft Detector for Multiple-input Multiple-output Channels*, School of ECE, Georgia Insititute of Technology

[13] Ying Hung Gan, Cong Ling and Wai Ho Mow, *Complex Lattice Reduction Algorithm for Low-Complexity MIMO Detection*, IEEE Trans. Signal Processing, vol. 57, pp. 2701-2710, July 2009.

[14] Sabuj Sarkar, Mohammad Shaifur Rahman, *Optimal BER in MIMO Raleigh Fading Channel from QPSK Modulation: Modified MMSE Versus ML Equalizer Evaluation* International Journal of Electronics and Informatics, Vol. 2, No. 1, 2013.

[15] Mathsworks, http://www.mathworks.se/help/comm/ref/comm.ltemimochannel-class.html

[16] Luis G. Barbero, T. Ratnarajah and Colin Cowan, *A COMPARISON OF COMPLEX LATTICE REDUCTION ALGORITHMS FOR MIMO DETECTION*, ECIT, Queen's University of Belfast

[17] Jun Niu and I-Tai Lu, *A Comparison of Two Lattice-Reduction-Based Receivers for MIMO Systems*, Department of Electrical and Computer Engineering, Polytechnic University

[18] David L. Milliner and John R. Barry, *A Lattice-Reduction-Aided Soft Detector for Multiple-Input Multiple-Output Channels*, School of ECE, Georgia Institute of Technology

[19] Dirk Wübben, Dominik Seethaler, Joakim Jaldén and Gerald Matz, *A survey with applications in wireless communications*, IEEE SIGNAL PROCESSING MAGAZINE [70] MAY 2011

[20] Laura Luzzi, Ghaya Rekaya-Ben Othman and Jean-Claude Belfiore, *Augmented Lattice Reduction for MIMO Decoding*, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 9, NO. 9, SEPTEMBER 2010

[21] Mahmoud Taherzadeh, Amin Mobasher and Amir K. Khandani, *Communication Over MIMO Broadcast Channels Using Lattice-Basis Reduction*, IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 53, NO. 12, DECEMBER 2007

[22] Ying Hung Gan, Cong Ling, and Wai Ho Mow, *Complex Lattice Reduction Algorithm for Low-Complexity Full-Diversity MIMO Detection*, IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 57, NO. 7, JULY 2009

[23] Cong Ling and Nick Howgrave-Graham *Effective LLL Reduction for Lattice Decoding*, ISIT2007, Nice, France, June 24 - June 29, 2007

[24] Tetsuyoshi Shimokawa and Tadashi Fujino, *Improved Signal Estimation Based on Combined Forward and Backward Lattice Reduction in MIMO Systems*, Department of Information and Communication Engineering, The University of Electro-Communications

[25] Jaehyun Park, Bruno Clerckx, Joohwan Chun and Byung Jang Jeong, *Lattice Reduction-Aided Successive Interference Cancelation for MIMO Interference Channels*, IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL. 63, NO. 8, OCTOBER 2014

[26] Keke Zu and Rodrigo C. de Lamare, *Low-Complexity Lattice Reduction-Aided Regularized Block Diagonalization for MU-MIMO Systems*, IEEE COMMUNICATIONS LETTERS, VOL. 16, NO. 6, JUNE 2012

[27] Dominik Seethaler, Gerald Matz, and Franz Hlawatsch, *LOW-COMPLEXITY MIMO DATA DETECTION USING SEYSEN'S LATTICE REDUCTION ALGORITHM*, Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology

[28] Hossein Najafi, Mohammad Erfan Danesh Jafari, and Mohamed Oussama Damen, *On Adaptive Lattice Reduction over Correlated Fading Channels*, IEEE TRANSACTIONS ON COMMUNICATIONS, VOL. 59, NO. 5, MAY 2011

[29] Mohamed Oussama Damen, Hesham El Gamal, and Giuseppe Caire, *On Maximum-Likelihood Detection and the Search for the Closest Lattice Point*, IEEE TRANSACTIONS ON INFORMATION THEORY, VOL. 49, NO. 10, OCTOBER 2003

[30] Kyungchun Lee, Joohwan Chun, and Lajos Hanzo, *Optimal Lattice-Reduction Aided Successive Interference Cancellation for MIMO Systems*, IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, VOL. 6, NO. 7, JULY 2007

[31] Joakim Jalden, Dominik Seethaler, and Gerald Matz *WORST- AND AVERAGE-CASE COMPLEXITY OF LLL LATTICE REDUCTION IN MIMOWIRELESS SYSTEMS*, Institute of Communications and Radio-Frequency Engineering, Vienna University of Technology; Communication Technology, Laboratory, ETH Zurich

[32] A. K. Lenstra, H. W. Lenstra, Jr, and L. Lovász, *Factoring Polynomials with Rational Coefficients*, Mathematische Annalen December 1982, Volume 261, Issue 4, pp 515-534

[33] Phong Q. Nguyen and Brigitte Valle, *The LLL Algorithm: Survey and Applications*, Springer Publishing Company, Incorporated ©2009

[34] PK. Shum, *http://home.ie.cuhk.edu.hk/ wkshum/wordpress/?p=442*,

[35] *http://se.mathworks.com/help/lte/ug/propagation-channel-models.html?requestedDomain=www.mathworks.com.*

[36] Michael Schneider and Johannes Buchmann, *Technische Universitat Darmstadt*, {mischnei,buchmann}@cdc.informatik.tu-darmstadt.de

Appendix A

Appendix A

A.1 Complex Lenstra–Lenstra–Lovász Algorithm

Here is the swapping process for complex symbols:

$$\dot{h}_{k-1} = h_k \tag{A.1}$$

$$\dot{h}_k = h_{k-1} \tag{A.2}$$

$$\dot{\mathcal{H}}_{k-1} = \mathcal{H}_k + \mid \mu_{k,k-1} \mid^2 \mathcal{H}_{k-1} \tag{A.3}$$

$$\dot{\mu}_{k,k-1} = \overline{\mu_{k,k-1}} \left(\frac{\mathcal{H}_{k-1}}{\dot{\mathcal{H}}_{k-1}} \right) \tag{A.4}$$

$$\dot{\mathcal{H}}_k = \left(\frac{\mathcal{H}_{k-1}}{\dot{\mathcal{H}}_{k-1}} \right) \mathcal{H}_k \tag{A.5}$$

$$\dot{\mu}_{i,k-1} = \mu_{i,k-1} \dot{\mu}_{k,k-1} + \mu_{i,k} \frac{\mathcal{H}_k}{\dot{\mathcal{H}}_{k-1}}, \qquad k < i \leq n \tag{A.6}$$

$$\dot{\mu}_{i,k} = \mu_{i,k-1} - \mu_{i,k} \mu_{k,k-1}, \qquad k < i \leq n \tag{A.7}$$

$$\dot{\mu}_{k-1,j} = \mu_{k,j}, \qquad 1 \leq j \leq k-2 \tag{A.8}$$

$$\dot{\mu}_{k,j} = \mu_{k-1,j}, \qquad 1 \leq j \leq k-2 \tag{A.9}$$

which is way complicated than the method we offered

A.2 Demodulation Algorithm for 64-QAM

The modulation method for 4-QAM, 16-QAM and 64-QAM are alike. By replacing the bits for real and imaginary parts with $r = im = r = [-3 : 2 : 3]$;, the 16-QAM signals are be obtained; similarly, 64-QAM can be obtained by replacing the bits for real and imaginary parts with $r = im = [-7 : 2 : 7]$.

However, it varies a bit for the demodulation methods of different QAMs. In principle, the general method to decode the received signals is by rounding it to the closest coordinate value, which will give it an output of bits. In the following section, let's take a look at demodulation methods for both 4-QAM and 64-QAM to find out the distinctions between them.

Algorithm 9 Demodulation for 64-QAM - [bits] = Demod 64-QAM(S_{hat}, M_T)

```
Symbollength = length(S_hat)
temp = [ ]
r = [−7 : 2 : 7] %get all possible coordinate values
im = [−7 : 2 : 7]
s_dem = 2 ∗ floor(s_hat/2) + 1;
s_dem(find(s_dem > max(r))) = max(r);
s_dem(find(s_dem < min(r))) = min(r);
s_dem = s_dem.';
for i = 1 : symbollength do
   if s_dem(i) == 7; then
      temp = [temp; 101]; % if coordinate value=7, output is 101
   else if s_dem(i) == 5; then
      temp = [temp; 100]; % if coordinate value=5, output is 100
   else if s_dem(i) == 3; then
      temp = [temp; 110]; % if coordinate value=3, output is 110
   else if s_dem(i) == 1; then
      temp = [temp; 111]; % if coordinate value=1, output is 111
   else if s_dem(i) == −1; then
      temp = [temp; 010]; % if coordinate value=-1, output is 010
   else if s_dem(i) == −3; then
      temp = [temp; 011]; % if coordinate value=-3, output is 011
   else if s_dem(i) == −5; then
      temp = [temp; 001]; % if coordinate value=-5, output is 001
   else if s_dem(i) == −7; then
      temp = [temp; 000]; % if coordinate value=-7, output is 000
   end if
end for
bits_hat_temp = temp;
bits_temp_Re = bits_hat_temp(1 : symbollength/2, :);
bits_temp_Im = bits_hat_temp(symbollength/2 + 1 : end, :);
bits_temp = [bits_temp_Rebits_temp_Im];
bits_out = reshape(bits_temp.', 6 ∗ Mt, 1)';
```

Series of Master's theses
Department of Electrical and Information Technology
LU/LTH-EIT 2016-497

http://www.eit.lth.se