View metadata, citation and similar papers at core.ac.uk

brought to you by CORE

provided by University of Waterloo's Institutional Repository

Adaptive Lattice Reduction in MIMO Systems

by

Mohammad Erfan Danesh Jafari

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Applied Science
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2008

© Mohammad Erfan Danesh Jafari 2008

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

In multiple-input multiple-output (MIMO) systems, the use of lattice reduction methods such as the one proposed by Lenstra-Lenstra-Lovász (LLL) significantly improves the performance of the suboptimal solutions like zero-forcing (ZF) and zero-forcing deceision-feedback-equalizer (ZF-DFE). Today's high rate data communication demands faster lattice reduction methods. Taking advantage of the temporal correlation of a Rayleigh fading channel, a new method is proposed to reduce the complexity of the lattice reduction methods. The proposed method achieves the same error performance as the original lattice reduction methods, but significantly reduces the complexity of lattice reduction algorithm. The proposed method can be used in any MIMO scenario, such as the MIMO detection, and broadcast cases, which are studied in this work.

Acknowledgements

I would like to thank all the people who made this possible. Specially my supervisor Prof. Mohamed Oussama Damen, who helped me a lot in finding my way through difficult moments of research. Also as this is a joint work with my friend and colleague Mr. Hossein Najafi, I would like to thank him for his time and efforts which made this work possible.

Dedication

This is dedicated to my wonderful wife, Masoume, and my caring parents that showed me everything is possible in this world, through patience and hard work.

Contents

1 Introduction **1**

1.1 Thesis Outline 2

2 Channel Model **3**

2.1 Correlated Fading models 4

2.2 AR Modeling of Bandlimited Rayleigh Random Processes 4

3 MIMO Detection System Model **7**

3.1 Pohst and Schnorr-Euchner Enumeration 8

3.2 Babai Decoder 11

3.3 Pre-Processing 12

3.3.1 Taming the channel: left preprocessing 13

3.3.2 Increasing the sparsity: right preprocessing 14

3.4 Reduced Complexity Pre-processing Stage 18

3.4.1 Effective LLL 20

3.4.2 Seysen's Lattice Reduction Algorithm 21

3.5 Proposed Method 22

3.6 Simulations Results 25

4 Broadcast System Model **35**

4.1 Channel Inversion 36

4.2 Search based broadcast precoding 37

4.3 Lattice-Reduction-Aided Broadcast Pre-coding 38
4.4 Proposed Method 38
4.5 Simulation Results 39

5 Conclusions and Future Work **44**

References **45**

List of Tables

3.1 Schnorr-Euchner with boundary check(Input C_0', $\mathbf{y}''$, $\mathbf{R}$) 11

3.2 LLL algorithm details 17

3.3 RED(k, l) sub-algorithm 17

3.4 SWAP(k) sub-algorithm 17

3.5 LLL with deep insertions 19

3.6 INSERT(k, i) sub-algorithm 19

3.7 Seysen lattice reduction algorithm 23

List of Figures

3.1 Reduction Effect, Solid line: Original basis, Dashed Line: Reduced basis 15

3.2 Effective LLL, complexity performance 21

3.3 Error performance of the proposed algorithm 26

3.4 Running Time Comparison of the proposed and conventional reduction methods 27

3.5 Average Number of basis updates for different number of antennas (LLL) 29

3.6 Average number of basis updates for different number of antennas(DILLL) 30

3.7 Average number of flops for different number of antennas (LLL) . . 31

3.8 Average number of flops for different number of antennas (DILLL) . 31

3.9 CDF of number of flops for different number of antennas (DILLL) . 32

3.10 CDF of number of flops for different number of antennas(DILLL) . 33

3.11 Performance of Adaptive DILLL for different f_m(chosen based on GSM model) 34

4.1 Simulation Results for different precoding schemes $N_T = K = 4$, 4-QAM in channel without temporal correlation 39

4.2 Error performance of the proposed method in temporal correlated channel 40

4.3 Average number of basis updates for different number of antennas . 41

4.4 Average Number of flops for Lattice Aided Reduction Precoding . . 42

4.5 CDF of number of flops for different number of antennas 43

Chapter 1

Introduction

In recent years, using multiple antenna for communication in fading channels has attracted many researchers. In these systems, the outputs can be described as a linear combination of the inputs corrupted by additive noise. It is shown in [1], [2] that there are advantages in using multiple antennas, namely they proved that the capacity of a multiple input multiple output system increases linearly with the minimum number of receive and transmit antennas. In point to point systems, which in this work is often referred to as MIMO detection system, for moderate to large values of number of antennas, decoding represents a challenging problem in communication theory. Recently, many researchers have tried to employ some of the methods used in lattice theory, as the solutions to this challenging problem [3], [4], [5]. The MIMO detection problem translates to closest lattice point search problem in lattice theory. Lattice reduction methods have proved themselves to be powerful tools in solving the closest lattice point problems. There is no unique definition for lattice reduction, and therefore, there exist many different methods for lattice reduction. Among the lattice reduction methods, the LLL methods due to Lenstra, Lenstra, and Lovász, [6] is the most practical one, due to its efficiency in finding near orthogonal vectors with short norms. Generally, in most of the recent works like in [7], the complexity of using the LLL algorithm is ignored. This can be justified in a case that, the channel variations are slow enough, to make it possible to use the result of the LLL reduction for quite a large number of received signals. In this way at the beginning of each frame the lattice reduction is performed on the channel, and for the rest of the frame the channel is assumed to be constant. However in many practical situations, to achieve higher data rates, reduction algorithms less complex than the LLL are required. In this work the channel is not assumed to

be constant throughout the frame and the channel realizations have a temporal correlation. The proposed method in this work, takes advantage of this correlation to reduce the complexity of the reduction method. The main idea is to use the results of the previous channel realization to perform an efficient reduction of the new channel realization. The proposed adaptive method can be used along with any reduction method, and it is not restricted to just the LLL algorithm. This makes it a powerful tool in many problems in communication theory that use the lattice reduction as a tool.

In addition, in this work the usage of the proposed method in broadcast scenario is investigated. In broadcast model, in contrast to the MIMO detection, users at the receiver side can not cooperate. In [8] and [9], it is shown that the sum-capacity grows linearly with the minimum number of transmit and receive antennas. In order to achieve the promised rate, some kind of precoding is required at the transmitter side. Lattice reduction methods such as LLL found themselves a great role in broadcast system too [10]. Therefore, the proposed adaptive method can be used to reduce the complexity of the precoding stage in a broadcast system too.

1.1 Thesis Outline

The rest of this work is organized as follows. In chapter 2 the generating algorithms for Rayleigh fading channel, are briefly described. Section 2.2, goes over an auto regressive model used as a generator for our fading channel throughout this work. In chapter 3, the system model for MIMO detection scenario is explained. In sections 3.1, and 3.2, different algorithms for MIMO detection are explored along with their advantages and disadvantages. Section 3.3, studies the preprocessing algorithms used to reduce the complexity of the decoding stage. Details of the LLL, lattice reduction algorithm can be found in this section. Section 3.4, reviews recent efforts to reduce the complexity of the lattice reduction stage. In section 3.5, the new adaptive method is proposed. Section 3.6 contains the simulation results, investigating the error and complexity performance of the proposed algorithm.

Chapter 4 deals with the broadcast scenario, its system model and the precoding methods used in this scenario. Sections 4.1-4.3 describe different algorithms used in the precoding stage. In section 4.4, and 4.5, the proposed method, and the simulation results are presented respectively.

In chapter 5 the reader can find the conclusion and the ideas for future work.

Chapter 2

Channel Model

In this work, we are considering a fading channel, which is assumed to be Rayleigh. The band limited Rayleigh process, which its power spectral density (PSD) is limited by maximum Doppler frequency has an extensive use in today's wireless communication problems. Many different techniques are developed to simulate Rayleigh random process [11], [12],[13]. One of the the pioneer methods was to generate correlated Rayleigh process based upon Clark's wide sense stationary (WSS) isotropic scattering models [14]. The problem with this model is that in many practical situations, the scattering might not be isotropic. This will definitely, affect the second order statistics of the channel. Some of the other popular simulation methods are based on sum of sinusoids, white noise filtering or inverse discrete Fourier transform (IDFT). In recent years some major problems were found in the commonly used sum of sinusoids, namely it was shown that the classical Jakes simulator produces fading signals that are not wide sense stationary [15]. On the other hand, the IDFT method is well known to be accurate and efficient [11]. The major disadvantage of this model is that all the samples are produced with a single Fast Fourier Transform (FFT) operation. This can cause a huge problem when we are dealing with generating a large number of variates.

In this work we use a general autoregressive model which was proposed by K. E. Baddour, in [16].His technique essentially employs the all-pole infinite-impulse response (IIR) filtering to shape the spectrum of uncorrelated Gaussian variates.

2.1 Correlated Fading models

In Rayleigh fading model the variation of the channel is captured by its auto correlation function (ACF) [17]. The second order statistics generally depend on the geometric properties of the area, the pace the mobile user is moving with, and characteristics of the antennas. A common situation (assumption) is that the propagation path consists of a two dimensional isotropic scattering, with a vertical monopole antenna at the receiver [18]. In this case, the theoretical PSD would be of the form [18]

$$S(f) = \frac{1}{\pi f_d \sqrt{1 - \frac{f}{f_d}^2}} \tag{2.1}$$

where f_d is the maximum doppler frequency. The in-phase and quadrature Gaussian processes each should have the autocorrelation sequence

$$R[n] = J_0(2\pi f_m |n|) \tag{2.2}$$

in which $f_m = f_d T$ is the maximum doppler frequency normalized by the sampling rate $1/T$. Furthermore in this model the in-phase and quadrature components should have zero mean and be independent of each other.

2.2 AR Modeling of Bandlimited Rayleigh Random Processes

An autoregression model of order p can be generated in time domain via the recursion [19]

$$x[n] = -\sum_{k=1}^{p} a_k x[n-k] + w[n] \tag{2.3}$$

in which $w[n]$ is a complex white noise Gaussian process with uncorrelated real and imaginary components. In order to produce Rayleigh fading process, $w[n]$ should have zero mean and $|x[n]|$ should be considered as the output. The PSD of the aforementioned AR model can be expressed using the following rational form [19]

$$S_{xx}(f) = \frac{\sigma_p^2}{|1 + \sum_{k=2}^{p} a_k \exp(-j2\pi f k)|^2} \tag{2.4}$$

In order to characterize the model, we should solve for $\{a_1, ..., a_p\}$ and the variance σ_p^2 of the driving noise.

The AR spectrum is used to approximate the Doppler spectrum. If one makes p sufficiently large, the approximation would be quite accurate. The relation between the desired model ACF $R_{xx}[j]$ and the AR parameters is given by [19]

$$R_{xx}[k] = \begin{cases} -\sum_{m=1}^{p} a_m R_{xx}[k-m] & k \geq 1 \\ -\sum_{m=1}^{p} a_m R_{xx}[-m] + \sigma_p^2 & k = 0 \end{cases} \tag{2.5}$$

Considering $k = 1, ..., p$, and combining the equations together, we would have,

$$\mathbf{R}_{xx}\mathbf{a} = \mathbf{v} \tag{2.6}$$

Where

$$\mathbf{R}_{xx} = \begin{bmatrix} R_{xx}[0] & R_{xx}[-1] & \cdots & R_{xx}[-p+1] \\ R_{xx}[1] & R_{xx}[0] & \cdots & R_{xx}[-p+2] \\ \vdots & \vdots & \ddots & \vdots \\ R_{xx}[p-1] & R_{xx}[p-2] & \cdots & R_{xx}[0] \end{bmatrix} \tag{2.7}$$

$$\mathbf{a} = \begin{bmatrix} a_1 & a_2 & \cdots & a_p \end{bmatrix}^T \tag{2.8}$$

$$\mathbf{v} = \begin{bmatrix} R_{xx}[1] & R_{xx}[2] & \cdots & R_{xx}[p] \end{bmatrix}^T \tag{2.9}$$

and considering the case for $k = 0$, it will results in,

$$\sigma_p^2 = R_{xx}[0] + \sum_{k=1}^{p} a_k R_{xx}[-k] \tag{2.10}$$

Therefore using the equations (2.7) to (2.10) and given the desired ACF, we can solve for the unknown parameters. These equations can be efficiently solved using Levinson-Durbin recursion in $O(p^2)$.

It can be proved that for large p the autocorrelation matrix will almost always be ill conditioned [16], and therefore makes the solving process of the mentioned equations numerically impractical. A heuristic approach that can be used to solve the numerical problems is to improve the condition of the autocorrelation matrix by adding a small value ϵ to the diagonal elements of the autocorrelation matrix.

Therefore the first $p+1$ autocorrelation lags of the resulting AR(p) process will be

$$\hat{R}_{xx}[m] = \begin{cases} R_{xx}[0] + \epsilon, & m = 1 \\ R_{xx}[m] & m = 1, 2, \cdots, p \end{cases} \tag{2.11}$$

Chapter 3

MIMO Detection System Model

In this work, we consider a multiple input multiple output (MIMO) system with M transmit, and $N \geqslant M$ receive antennas. If we consider $\mathbf{x}^c = [x_1^c, ..., x_M^c]^T$, $\mathbf{y}^c = [y_1^c, ..., y_M^c]^T$, $\mathbf{w}^c = [w_1^c, ..., w_M^c]^T$ and the $N \times M$ matrix $\mathbf{H}^c$, respectively as the transmitted signal, the received signal, the noise vector and the channel matrix, it will lead to the popular base-band model

$$\mathbf{y}^c = \mathbf{H}^c\mathbf{x}^c + \mathbf{w}^c \tag{3.1}$$

The channel is assumed to be Rayleigh, and the noise is Gaussian, i.e., the elements of $\mathbf{H}$, namely $h_{i,j}^c$, are independent and identically distributed (i.i.d), with zero mean and unit variance complex Gaussian distribution. The complex input signal $\mathbf{x}^c$ is composed of components, c_i^c, chosen from a Q^2-QAM constellation with energy $\frac{\rho}{M}$, in which ρ can be interpreted as the signal-to-noise ratio (SNR) observed at any receive antenna. We can convert the whole system to its real counterpart using the following transformations defined for vectors and matrices,

$$\mathbf{u}^c \mapsto \mathbf{u} = \begin{bmatrix} \Re\{\mathbf{u}^c\}^T & \Im\{\mathbf{u}^c\}^T \end{bmatrix}$$
$$\mathbf{H}^c \mapsto \mathbf{H} = \begin{bmatrix} \Re\{\mathbf{H}^c\} & -\Im\{\mathbf{H}^c\} \\ \Im\{\mathbf{H}^c\} & \Re\{\mathbf{H}^c\} \end{bmatrix}$$

Using the aforementioned transformations, the resulting real model is given by

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{w} \tag{3.2}$$

in which, now the components of $\mathbf{x}$ is chosen with uniform probability from a Q-PAM constellation, with energy $\frac{\rho}{2M}$. We can further simplify the model by mapping the PAM signals to integers using the following mapping.

$$\mathbf{c} \mapsto \mathbf{x} = \kappa\mathbf{c} + \mathbf{v} \tag{3.3}$$

in which elements of $\mathbf{c}$ are in $\{0, 1, ..., Q-1\}$, κ is a constant related to the PAM constellation energy, and $\mathbf{v}$ is a constant vector. Using the (3.3) our system simplifies to,

$$\mathbf{y} = \mathbf{H}(\kappa\mathbf{c} + \mathbf{v}) + \mathbf{w} \tag{3.4}$$

In the resulting model , vector $\mathbf{c}$ has integer elements, therefore our problem changes to detecting a lattice code transmitted over a linear channel with additive white gaussian noise [20].

The maximum-liklihood(ML) solution to this problem is given by,

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathcal{U}} |\mathbf{y} - \mathbf{H}\mathbf{v} - \mathbf{H}\kappa\mathbf{c}| \tag{3.5}$$

$$= \arg\min_{\mathbf{c}\in\mathcal{U}} |\mathbf{y}' - \mathbf{H}'\mathbf{c}| \tag{3.6}$$

in which $\mathcal{U}$ refers to the hypercube $\{0, 1, ..., Q-1\}^n \in \mathbb{R}^n$, $\mathbf{y}' = \mathbf{y} - \mathbf{H}\mathbf{v}$, and $\mathbf{H}' = \kappa\mathbf{H}$. In communication literature the optimization problem in (3.5) is referred to as constrained *Closest Lattice Point Search*(CLPS). Solution to this problem is vastly investigated in lattice theory like in, [3], [4], and [5]. It is shown that for a general $\mathbf{H}$, finding the optimal solution to this problem can be NP-hard. There exist algorithms that deal with finding this optimal solution, such as Sphere Decoding by Pohst [21]. Pohst proposed an efficient method to find all the lattice points within a sphere of certain radius. It is shown that the Pohst algorithm lowers the decoding complexity for higher SNRs, but still the complexity grows exponentially with the system dimension.

3.1 Pohst and Schnorr-Euchner Enumeration

The idea of Post enumeration is as follows. Consider C_0 to be the squared radius of hyper sphere $S(\mathbf{y}', \sqrt{C_0})$ in which the enumeration is trying to find a list of all the points inside the hyper sphere. If a set $\Lambda = \{\mathbf{H}'\mathbf{c} : \mathbf{c} \in \mathbb{Z}^n\}$ is considered, the goal of the Pohst enumeration is to find all the points in $\Lambda \cap S(\mathbf{y}', \sqrt{C_0})$. The first

step is to perform QR decomposition on $\mathbf{H}'$, and we would have,

$$\mathbf{H}' = [\mathbf{Q}, \mathbf{Q}'] \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \tag{3.7}$$

where $\mathbf{R}$ is $m \times m$ upper triangular matrix with positive diagonal elements, $\mathbf{0}$ is $m \times (m-n)$ zero matrix, $\mathbf{Q}$ is a $n \times m$ unitary matrix, and $\mathbf{Q}'$ is a $n \times (n-m)$ unitary matrix. The points inside the hyper sphere can be stated as

$$|\mathbf{y}' - \mathbf{H}'\mathbf{c}|^2 \leqslant C_0 \tag{3.8}$$

for some $\mathbf{c} \in \mathbb{Z}^n$. Using the QR decomposition equation (3.8) can be rewritten as

$$\left| [\mathbf{Q}, \mathbf{Q}']^T \mathbf{y}' - \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{c} \right|^2 \leqslant C_0 \tag{3.9}$$

$$\left| \mathbf{Q}^T \mathbf{y}' - \mathbf{R}\mathbf{c} \right|^2 \leqslant C_0 - |(\mathbf{Q}')^T \mathbf{y}'|^2 \tag{3.10}$$

$$|\mathbf{y}'' - \mathbf{R}\mathbf{c}|^2 \leqslant C_0' \tag{3.11}$$

where $\mathbf{y}'' = \mathbf{Q}^T\mathbf{y}'$, and $C_0' = C_0 - |(\mathbf{Q}')^T\mathbf{y}'|^2$. Taking advantage of the upper triangular property of $\mathbf{R}$ the inequality (3.11) can be rewritten as a set of conditions,

$$\sum_{j=i}^{m} \left| y_i'' - \sum_{l=j}^{m} r_{j,l} c_l \right|^2 \leqslant C_0' \quad i = 1, 2, \cdots, m \tag{3.12}$$

Considering the above conditions starting from the last index(m) until we reach the first index(1), for each c_i we can find an interval in which the possible values of c_i belong. Assuming some value for $c_i, \cdots, c_m$, we can compute the mentioned interval for c_{i-1}. More specifically, assume that $\mathbf{c}_l^m = [c_l, c_{l+1}, \cdots, c_m]^T$ be the last $m - l + 1$ elements of the vector $\mathbf{c}$. For a fixed $\mathbf{c}_{i+1}^m$, c_i can only take the integer values in the interval $\mathcal{I}_i(\mathbf{c}_{i+1}^m) = [A_i(\mathbf{c}_{i+1}^m), B_i(\mathbf{c}_{i+1}^m)]$, where

$$A_i(\mathbf{c}_{i+1}^m) = \left\lceil \frac{1}{r_{i,i}} \left(y_i'' - \sum_{j=i+1}^{m} r_{i,j} c_j - \sqrt{C_0' - \sum_{j=i+1}^{m} \left| y_j'' - \sum_{l=j}^{m} r_{j,l} c_l \right|^2} \right) \right\rceil \tag{3.13}$$

$$B_i(\mathbf{c}_{i+1}^m) = \left\lfloor \frac{1}{r_{i,i}} \left(y_i'' - \sum_{j=i+1}^{m} r_{i,j} c_j + \sqrt{C_0' - \sum_{j=i+1}^{m} \left| y_j'' - \sum_{l=j}^{m} r_{j,l} c_l \right|^2} \right) \right\rfloor \tag{3.14}$$

This assumes that $A_i < B_i$, and $\sum_{j=i+1}^{m} \left| y''_j - \sum_{l=j}^{m} r_{j,l} c_l \right|^2 \leqslant C'_0$, otherwise $\mathcal{I}_i = \emptyset$, which means there are no valid choices for c_i and therefore the choices for $\mathbf{c}_{i+1}^m$ are not correct.

Pohst enumeration examines all the admissible lattice points and finds the one which is closest to the the center of the hyper sphere, $\mathbf{y}''$. At each level i this consists of testing all the admissible values of c_i, which belong to $\mathcal{I}_i(\mathbf{c}_{i+1}^m)$. Starting from the last index(m) and computing each c_i until we reach to the first index(1) if $\mathcal{I}_i$ is non empty, then we have found a lattice point inside the hyper sphere S. This means $\mathbf{c}_1^m \in S(\mathbf{y}'', \sqrt{C_0})$. The euclidean distance of this lattice point from the center of sphere is,

$$d^2(\mathbf{y}'', \mathbf{H}'\mathbf{c}) = \sum_{j=1}^{m} \left| y''_j - \sum_{l=j}^{m} r_{j,l} c_l \right|^2 \tag{3.15}$$

The Pohst enumeration just outputs the point with the least euclidian distance. If after spanning the interval corresponding to the last interval, $\mathcal{I}_m$, no valid lattice point was found, then the radius of the hyper sphere should be increased, in order to contain at least one lattice point. At each level i, Pohst enumeration starts from the lowest integer in the interval, and spans all the integers in the interval. This is called the *natural spanning* of the interval. A more wise method is to start from the middle point of the interval and traverse the whole interval using zig-zag moves. This method of enumeration was first used by Schnorr-Euchner [22] to improve the performance of the Pohst enumeration. The mid point of the interval can be expressed as

$$S_i(\mathbf{c}_{i+1}^m) = \left\lceil \left(y''_i - \sum_{j=i+1}^{m} r_{i,j} c_j \right) \right\rfloor \tag{3.16}$$

If $S_i(\mathbf{c}_{i+1}^m)$ is rounded down meaning,

$$y''_i - \sum_{j=i+1}^{m} r_{i,j} c_j - r_{i,i} S_i(\mathbf{c}_{i+1}^m) \geqslant 0 \tag{3.17}$$

the order that the Schnorr-Euchner traverses the interval would be,

$$\{S_i(\mathbf{c}_{i+1}^m), S_i(\mathbf{c}_{i+1}^m) + 1, S_i(\mathbf{c}_{i+1}^m) - 1, S_i(\mathbf{c}_{i+1}^m) + 2, \cdots\} \cap \mathcal{I}_i(\mathbf{c}_{i+1}^m) \tag{3.18}$$

or in contrast if $S_i(\mathbf{c}_{i+1}^m)$ is rounded up, then the sequence would be

$$\{S_i(\mathbf{c}_{i+1}^m), S_i(\mathbf{c}_{i+1}^m) - 1, S_i(\mathbf{c}_{i+1}^m) + 1, S_i(\mathbf{c}_{i+1}^m) - 2, \cdots\} \cap \mathcal{I}_i(\mathbf{c}_{i+1}^m) \tag{3.19}$$

Table 3.1: Schnorr-Euchner with boundary check(Input C_0', $\mathbf{y}''$, $\mathbf{R}$)

Step 1	(Initialize) $i = m$, $d_c = C_0', T_m = 0, \xi_m = 0$.
Step 2	Set $c_i = \lceil (y_i'' - \xi_i)/r_{i,i} \rfloor$ and $\Delta_i = \text{sgn}(y_i'' - \xi_i - r_{i,i}c_i)$
Step 3	If $d_c < T_i + \lvert y_i'' - \xi_i - r_{i,i}c_i \rvert^2$ go to step 4(Already outside the hyper sphere) Else if $c_i \notin \{0, 1, \cdots, Q-1\}$ go to step 6(Inside the hyper sphere but outside the boundary) Else(We are inside the hyper sphere and boundary set) If $i = 1$ go to step 5 Else $\xi_{i-1} = \sum_{j=i}^{m} r_{i-1,j}c_j$, $T_i = T_{i-1} + \lvert y_i'' - \xi_i - r_{i,i}c_i \rvert^2$, $i = i - 1$, goto step 2.
Step 4	If $i = m$ terminate the algorithm, else $i = i + 1$ and goto step 6
Step 5	(A valid point is found) $d_c = T_1 + \lvert y_1'' - \xi_1 - r_{1,1}c_i \rvert^2$, save $\hat{\mathbf{c}} = \mathbf{c}$, and $i = i + 1$, goto step 6.
Step 6	(Schnorr-Euchner enumeration at level i) $c_i = c_i + \Delta_i$, $\Delta_i = -\Delta_i - \text{sgn}(\Delta_i)$, and goto step 2.

If at any level i the interval $\mathcal{I}_i(\mathbf{c}_{i+1}^m)$ is the empty set, then the enumeration algorithm returns to $(i+1)$th level and chooses the next value in the sequence as the value for c_{i+1}. In order to reduce the complexity of finding the closest lattice point, we can update the radius of the sphere, as soon as a new lattice point is found. We can set the new radius as the euclidean distance between the new found lattice point and center of the sphere. For finding the maximum likelihood solution, at each step i the algorithm should check the c_i under test to make sure it belongs to $\mathcal{U}$. This might be easy for some well shaped $\mathcal{U}$, like the one that is considered here, but it can be quite complex to check for the boundaries in general. The loss of performance for not checking the boundary is shown to be negligible, but the gain in complexity is quite worthy. Detection of the signal without checking boundary is often called *lattice decoding* in communication literature.

When $\mathcal{U} = \{0, 1, \cdots, Q-1\}^n$, the Schnorr-Euchner enumeration with radius update can be written explicitly as in Table 3.1.

3.2 Babai Decoder

As mentioned before the optimal solution is in general complex and not practical to find, in many of today's communication systems. To further reduce the complexity of the solution of the CLPS problem, the problem should be solved in an approximate fashion. The most obvious approximate solution is the linear zero forcing(ZF)

decoder which was first proposed by Babai in [23]. The solution can be written as

$$\hat{\mathbf{c}} = \lceil (\kappa \mathbf{H})^{-1} (\mathbf{y} - \mathbf{H}\mathbf{v}) \rfloor \tag{3.20}$$

In this way the interference is totally suppressed. This causes the noise effect to be amplified. Using the decision feedback equalizer (DFE) can lessen this effect. DFE can be done by performing QR algorithm on $\mathbf{H}'$ to get $\mathbf{H}' = \mathbf{Q}\mathbf{R}$. As mentioned before, using (3.6), and the QR decomposition we can write,

$$\hat{\mathbf{c}} = \arg\min_{\mathbf{c}\in\mathcal{U}} \left| [\mathbf{Q}, \mathbf{Q}']^T \mathbf{y}' - [\mathbf{Q}, \mathbf{Q}']^T \mathbf{H}'\mathbf{c} \right| \tag{3.21}$$

$$= \arg\min_{\mathbf{c}\in\mathcal{U}} \left| [\mathbf{Q}, \mathbf{Q}']^T \mathbf{y}' - \begin{bmatrix} \mathbf{R} \\ \mathbf{0} \end{bmatrix} \mathbf{c} \right| \tag{3.22}$$

$$= \arg\min_{\mathbf{c}\in\mathcal{U}} \left(\left| \mathbf{Q}^T \mathbf{y}' - \mathbf{R}\mathbf{c} \right| + \left| \mathbf{Q}'^T \mathbf{y}' \right| \right) \tag{3.23}$$

$$= \arg\min_{\mathbf{c}\in\mathcal{U}} |\mathbf{y}'' - \mathbf{R}\mathbf{c}| \tag{3.24}$$

Where $\mathbf{y}'' = \mathbf{Q}^T \mathbf{y}'$. Since R is triangular, the last symbol can be estimated as $\hat{c}_n = \lceil y''_n / R_{n,n} \rfloor$. We can then substitute the estimated value to cancel the noise interference in y''_{n-1}. We can continue in this manner until the last symbol is found. The solution can be written as,

$$\hat{c}_i = \left\lceil \frac{y''_i - \sum_{j=i+1}^{n} r_{i,j} \hat{c}_j}{r_{i,i}} \right\rfloor \text{ for } i = n, n-1, \cdots, 1 \tag{3.25}$$

This is exactly what Babai proposed as an appriximate solution to the problem (3.5). This is equivalent to the first point found using the Schnorr-Euchner enumeration with $C_0 = \infty$.

3.3 Pre-Processing

The error performance of Babai is far from ML in high SNRs. Therefore further efforts have been done to develop methods with low complexity and error performance near the ML solution. As it is mentioned in [7] most of sub-optimal algorithms can be divided into two stages. Stage one is the preprocessing, and stage two is to perform the search. The rough idea for the preprocessing stage is to transform the original CLPS problem defined by the lattice $\kappa\mathbf{H}$ and the constraint set $\mathcal{U}$ in a way

which is suitable for the search stage. Complexity of the tree search stage is heavily dependent on the efficiency of the preprocessing stage.

The first cause of complexity in solving the CLPS problem arises when the channel matrix is ill conditioned. This intuitively causes the lattice generated by $\mathbf{H}'$ to have a skewed fundamental cell. This causes some points of the set $\{\mathbf{H}'\mathbf{c} : \mathbf{c} \in \mathcal{U}\}$ be difficult to be distinguished from each other. This problem can be alleviated using the left preprocessing. This stage changes the channel matrix and the noise vector in a way that they are better conditioned for the search algorithms. As the resulting CLPS problem may not be equivalent to the original problem, this transformation causes the overall performance to be suboptimal and not ML.

The second problem is that sometimes it is difficult and complex to check for boundaries. This can be solved be relaxing the boundaries and letting the CLPS problem search in $\mathbb{Z}^m$ instead of $\mathcal{U}$. This relaxation of boundaries again causes the algorithms to be suboptimal, but it makes it possible to replace the lattice basis with one which is more appropriate for the rest of stages (tree search) in the decoding process. As it was mentioned before, this is called *lattice decoding* in communication literature. This change of lattice basis can be done using the right preprocessing algorithms.

The right and left preprocessing algorithms combined with the lattice decoding are effective ways to reduce the complexity of the search stage, but the expense paid is the sub-optimality of these algorithms. I will try to briefly go over some of the preprocessing algorithm in next few subsections.

3.3.1 Taming the channel: left preprocessing

Performing the QR decomposition on the $\mathbf{H}'$, allows to employ a recursive detection of the transmitted signal $\mathbf{c}$. $\mathbf{Q}$ is the feed forward matrix, of the zero forcing, decision feedback equalizer (ZF-DFE) [24]. Sphere decoders can be assumed as a kind of ZF-DFE filtering, with ability to correct their previous decisions. It is well known that the MMSE-DFE decoders, have better performance than ZF-DFE decoders [25], in terms of signal to noise plus interference ratio(SINR). This encourages the use of MMSE filtering in decoders. This can be done by performing the QR decomposition on the augmented channel matrix.

$$\tilde{\mathbf{H}} = \begin{bmatrix} \mathbf{H}' \\ \mathbf{I} \end{bmatrix} = \tilde{\mathbf{Q}}\mathbf{R}_1 \tag{3.26}$$

in which $\tilde{\mathbf{Q}} \in \mathbb{R}^{(n+m)\times m}$ and has orthogonal columns, and $\mathbf{R}$ is upper triangular. Let $\mathbf{Q}_1$ be the upper $n \times m$ sub matrix of $\tilde{\mathbf{Q}}$, then $\mathbf{Q}_1$ and $\mathbf{R}_1$ are MMSE-DFE forward and backward filters respectively. As an approximate solution to the original problem, we can solve the following CLPS problem,

$$\arg\min_{c\in\mathcal{U}} |\mathbf{y}' - \mathbf{R}\mathbf{c}| \tag{3.27}$$

with $\mathbf{y}' = \mathbf{Q}_1^T\mathbf{y} - \mathbf{R}_1\mathbf{v}$. This new CLPS problem is not equivalent to the original problem, because in general $\mathbf{Q}_1$ does not have orthonormal columns. Also the noise in the new CLPS problem is no longer gaussian but it is still white. This causes the solution of the new problem to be sub optimal comparing to the original CLPS problem. The good point about this transformation is that the resulting $\mathbf{Q}_1$ is always well conditioned, and has rank m.

3.3.2 Increasing the sparsity: right preprocessing

Matrix $\mathbf{R}$ is used to form a search tree. The search is performed on this tree. The sparser the matrix $\mathbf{R}$, the less complex would the tree search stage be. For example considering $\mathbf{R}$ to be diagonal, then the symbol by symbol detection would be optimal.

We can define a measure for sparsity of $\mathbf{R}$ as follows,

$$S(\mathbf{R}) = \max_{i\in\{1,\cdots,m\}} \frac{\sum_{j=i+1}^{m} r_{i,j}^2}{r_{i,i}^2} \tag{3.28}$$

Smaller $S(\mathbf{R})$ means sparser $\mathbf{R}$. The goal in this preprocessing is to find a new basis $\mathbf{S}$ for the lattice $\{\mathbf{R}_1\mathbf{c} : \mathbf{c} \in \mathbb{Z}^m\}$, that when QR is performed on $\mathbf{S}$, the resulting $\mathbf{R}$ has the least $S(\mathbf{R})$ among all the possible bases of the original lattice. This means finding a unimodular matrix $\mathbf{T}$(this means the entries of $\mathbf{T}$ and $\mathbf{T}^{-1}$ are integers) that satisfies $\mathbf{R_1} = \mathbf{QRT}$ and $S(\mathbf{R})$ is the smallest possible for all $\mathbf{T}$s. The optimal solution to this minimization is very difficult to find, but fortunately there exist some suboptimal solutions that can increase the sparsity of $\mathbf{R}$. As a few, we can name lattice basis reduction algorithms, column permutations and combinations. Permutation means to change the order of the columns of the lattice, to have a sparser matrix. We can use the V-BLAST greedy ordering method proposed in [24] This method finds a permutation matrix $\mathbf{\Sigma}$ that $\mathbf{R}_1 = \mathbf{QR\Sigma}$ and maximizes $\min_i r_{i,i}^2$. This minimizes $S(\mathbf{R})$ over the group of permutation matrices. The inter-

esting point about all these preprocessing stages is that the lattice decoding is not affected by them. As in lattice decoding we relax the boundaries to be the whole $\mathbb{Z}^m$, the boundary is not affected by left and right preprocessing stages. It worths to say that however the boundary checking for ML detection after preprocessing can impose a lot of complexity on the decoder.

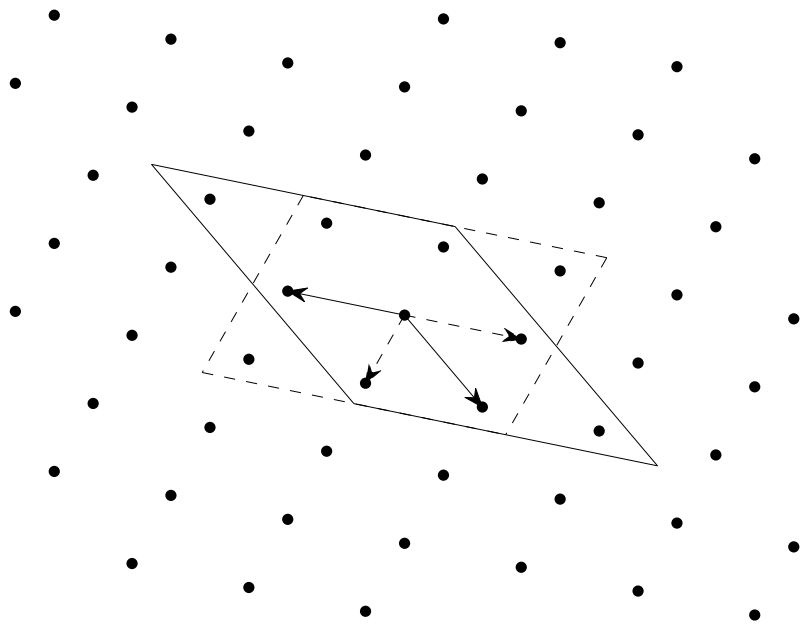


Figure 3.1: Reduction Effect, Solid line: Original basis, Dashed Line: Reduced basis

The goal of the lattice basis reduction is to find a new basis, that the columns of the new generator matrix $\mathbf{S}$ have a small norm, and they are as orthogonal as possible. You can see the effect of reduction in figure 3.1. This concept was proposed more than a century ago. There is no unique definition for lattice reduction. Minkowski proposed a definition in 1890s. A basis is Minkowski reduced if for $i = 1, \cdots, m$, $\mathbf{b}_i$ is the shortest vector that can be extended to a basis having $(\mathbf{b}_1, \cdots, \mathbf{b}_{i-1})$. In this definition $\mathbf{b}_i$s are the columns of the lattice basis $\mathbf{B}$. The problem of finding the shortest vector in general is considered to be NP-hard. In 1982, Lenstra, Lenstra, and Lovász(LLL) [6] proposed a breakthrough algorithm for lattice reduction. A further improved version was developed by Schnorr and Euchner [22] which is called deep insertion LLL. This modification gives significantly shorter vector in comparison to the orignial LLL algorithm. The complexity of the

original LLL is polynomial in lattice dimension. The complexity of the deep insertion LLL in worst case can be exponential, but simulations show that on average it does not require much more iterations than the original LLL [26].

Next, details of the LLL and the deep insertion LLL are explained.

LLL algorithm uses Gram-Schmidt orthogonalization, has a polynomial complexity and guarantees a bounded orthogonality defect. The orthogonality defect is defined as

$$\delta = \frac{(||\mathbf{b}_1||^2||\mathbf{b}_2||^2 \cdots ||\mathbf{b}_m||^2)}{\det \mathbf{B}^H\mathbf{B}} \tag{3.29}$$

As the determinant is constant for all the bases of a lattice, the lattice reduction problem can be viewed as finding the minimum of $(||\mathbf{b}_1||^2||\mathbf{b}_2||^2 \cdots ||\mathbf{b}_m||^2)$. Given the lattice Λ with basis $(\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_m)$, one can compute the Gram-Schmidt mutually orthogonal vectors, $(\mathbf{b}_1^*, \mathbf{b}_2^*, \cdots, \mathbf{b}_m^*)$ using the following approach and performing it for $i = 1, \cdots, m$:

$$\mathbf{b}_i^* = \mathbf{b}_i - \sum_{j=1}^{i-1} \mu_{ij}\mathbf{b}_j^* \tag{3.30}$$

$$\mu_{ij} = \frac{< \mathbf{b}_i, \mathbf{b}_j^* >}{< \mathbf{b}_j, \mathbf{b}_j^* >} \tag{3.31}$$

The $< \cdot, \cdot >$ is the inner product. A basis $(\mathbf{b}_1, \cdots, \mathbf{b}_{i-1})$ is LLL reduced if

- $||\mu_{ij}|| \leqslant \frac{1}{2}$ for $1 \leqslant i < j \leqslant m$, and
- $p \cdot ||\mathbf{b}_i^*||^2 \leqslant ||\mathbf{b}_{i+1}^* + \mu_{i+1,i}\mathbf{b}_i^*||^2$

where $\frac{1}{4} < p < 1$. Choosing larger values for constant p, results in a better reduction but also a higher complexity. At practice to get to best result, it is suggested to choose the value of 0.99 for constant p, because for the ideal value $p = 1$, although in practice the running time seems to be polynomial, there is no analytical guarantee that the running time of the algorithm is polynomial. The detailed LLL algorithm can be written as you can see in Table 3.2 [26].

In the original LLL algorithm, to check if a basis is LLL reduced, only adjacent columns are checked against each other. One can argue that this condition can be strengthened, to take into account the earlier columns too. This leads to a non-polynomial algorithm both in theory and practice. Obviously this was one of the reasons that the authors of LLL chose the relaxed condition. Schnorr and Euchner

Table 3.2: LLL algorithm details

Step 1:	Initialization
	Set $k = 2, k_{max} = 1, \mathbf{b}_1^* = \mathbf{b}_1, B_1 =< \mathbf{b1}, \mathbf{b1} >, \mathbf{H} = \mathbf{I}_n$
Step 2:	Incremental Gram-Schmidt
	if $k < k_{max}$ go to step 3.
	else
	$k_{max} = k, \mathbf{b}_k^* = \mathbf{b}_k$
	for $j = 1, \ldots, k-1$
	$\mu_{k,j} = \frac{<\mathbf{b}_k, \mathbf{b}_j^*>}{B_j}$, $\mathbf{b}_k^* = \mathbf{b}_k^* - \mu_{k,j}\mathbf{b}_j^*$
	$B_k =< \mathbf{b}_k^*, \mathbf{b}_k^* >$
Step 3:	Testing for LLL condition
	Run RED$(k, k-1)$
	if $B_k < (0.75 - \mu_{k,k-1}^2)B_{k-1}$
	Run SWAP(k) (See Table 3.4)
	$k = \max(k-1, 2)$
	Go to step 3
	else
	for $l = k-2, k-3, \ldots, 1$
	Run RED(k, l) (See Table 3.3)
	$k = k + 1$
Step 4:	Test for termination
	if $k \leqslant m$ go to step 2
	Terminate the program and output $\mathbf{b}_i$s and the transformation matrix $\mathbf{H}$

Table 3.3: RED(k, l) sub-algorithm

if $\lvert\mu_{k,l}\rvert \leqslant 0.5$ exit the sub algorithm
else
$q = \lceil m u_{k,l} \rfloor$
$\mathbf{b}_k = \mathbf{b}_k - q\mathbf{b}_l, \mathbf{H}_k = \mathbf{H}_k - q\mathbf{H}_l, \mu_{k,l} = \mu_{k,l} - q$
for $i = 1, \ldots, i-1$
$\mu_{k,i} = \mu_{k,i} - q\mu_{l,i}$
terminate the sub algorithm

Table 3.4: SWAP(k) sub-algorithm

Swap vectors $\mathbf{b}_k$ and $\mathbf{b}_{k-1}$,$\mathbf{H}_k$, and $\mathbf{H}_{k-1}$
if $k > 2$
for $j = 1, \ldots, k-2$, exchange $\mu_{k,j}$ with $\mu_{k-1,j}$
$\mu = \mu_{k,k-1}, B = B_k + \mu^2 B_{k-1}, \mu_{k,k-1} = \mu B_{k-1}/B$, $\mathbf{b} = \mathbf{b}_{k-1}^*, \mathbf{b}_{k-1}^* = \mathbf{b}_k^* + \mu\mathbf{b}$,
$\mathbf{b}_k^* = -\mu_{k,k-1}\mathbf{b}_k^* + (B_k/B)\mathbf{b}, B_k = B_{k-1}B_k/B, B_{k-1} = B$
for $i = k+1, k+2, \cdots, k_{max}$
$t = \mu_{i,k}, \mu_{i,k} = \mu_{i,k-1} - \mu t, \mu_{i,k-1} = t + \mu_{k,k-1}\mu_{i,k}$ Terminate the program.

proposed a method to strengthen the condition without losing much practical speed [22].

In the proposed algorithm it is possible to insert $\mathbf{b}_k$ between $\mathbf{b}_{i-1}$ and $\mathbf{b}_i$ for some $i < k$. In this case the new $||\mathbf{b}_i^*||^2$ will become:

$$||\mathbf{b}_i^*||^2 = ||\mathbf{b}_k||^2 - \sum_{1 \leqslant j < i} \mu_{k,j}^2 ||\mathbf{b}_j^*||^2 = |\mathbf{b}_k^*||^2 + \sum_{i \leqslant j < k} \mu_{k,j}^2 ||\mathbf{b}_j^*||^2 \tag{3.32}$$

If the norm of the vector k inserted at position i is significantly smaller than that of the current vector at position i, then the insertion is performed. Significantly smaller means norm of the new vector at position i to be at most p times the norm of the vector previously at position i. It worth to note that, the insertion of k at $(k-1)$th position is exactly, what was done in the original LLL.

The pseudo code for Deep Insertion LLL due to Schnorr and Euchner can be found in Table 3.5 [26].

3.4 Reduced Complexity Pre-processing Stage

The complexity of the preprocessing is usually ignored, this can be justified by the assumption of a flat-fading channel, that allows us to perform all the preprocessing algorithms at the beginning of a new frame, and to use the results within the whole frame. As it was said before, in many of the preprocessing scenarios, a reduction method is used to reduce the lattice basis vectors. The goal for this reduction is to form a new basis for the lattice, which has shorter basis vectors. Finding the shortest lattice vectors is again NP-hard. Near optimal methods are proposed in [6]. The LLL algorithms have found many usages in different applications. LLL algorithm is still a complex algorithm, therefore many efforts have been done to reduce the complexity of the LLL itself.

It worths to say that the application of the LLL algorithm is not limited to just the MIMO detection problem. This algorithm, also has applications in cryptanalysis of public key encryption schemes, finding integer solutions to problems, and etc. Although the complexity of the LLL algorithm is proved to be bounded with a polynomial, but still the average time it takes to execute is quite high. Therefore, many efforts have been done to reduce the complexity of this algorithm. Effective LLL [27] which is proposed by Cong Ling, and Nick Howgrave-Graham, is one of

Table 3.5: LLL with deep insertions

Step 1:	Initialization
	Set $k = 1$, , $\mathbf{H} = \mathbf{I}_n$
Step 2:	Incremental Gram-Schmidt
	$\mathbf{b}_k^* = \mathbf{b}_k$
	for $j = 1, \ldots, k-1$
	$\quad \mu_{k,j} = \frac{<\mathbf{b}_k, \mathbf{b}_j^*>}{B_j}$, $\mathbf{b}_k^* = \mathbf{b}_k^* - \mu_{k,j}\mathbf{b}_j^*$
	$B_k =< \mathbf{b}_k^*, \mathbf{b}_k^* >$
	if $k = 1$
	$\quad k = 2$, go to Step 5.
Step 3:	Initialize test
	For $l = k-1, k-2, \ldots, 1$
	$\quad$ Execute sub algorithm RED(k, l). See Table 3.3.
	$B =< \mathbf{b}_k, \mathbf{b}_k >, i = 1$
Step 4	Deep LLL test
	If $i = k$
	$\quad k = k + 1$, Goto Step 5
	Else if $\frac{3}{4}B_i \leqslant B$
	$\quad B = B - \mu_{k,i}^2$, $i = i + 1$, Go to Step 4.
	Else
	$\quad$ Execute INSERT(k, i)
	$\quad$ if $i \geqslant 2$
	$\quad\quad k = i - 1, B =< \mathbf{b}_k, \mathbf{b}_k >, i = 1$, Goto Step 4
	$\quad$ if $i = 1$
	$\quad\quad k = 1$, Goto Step 2.
Step 5:	Test for termination
	if $k \leqslant m$ go to step 2
	Terminate the program and output $\mathbf{b}_i$s and the transformation matrix $\mathbf{H}$

Table 3.6: INSERT(k, i) sub-algorithm

$\mathbf{b} = \mathbf{b}_k, \mathbf{V} = \mathbf{H}_k$
for $j = k, k-1, \ldots, i+1$
$\quad \mathbf{b}_j = \mathbf{b}_{j-1}, \mathbf{H}_j = \mathbf{H}_{j-1}$
$\mathbf{b}_i = \mathbf{b}, \mathbf{H}_i = \mathbf{V}$
Terminate the sub-algorithm.

these efforts. They studied the application of LLL in the MIMO detection and reduced the complexity of the LLL algorithm by lowering the number of size reductions done in the LLL algorithm. This was done without any change in the error performance of the MIMO detection.

3.4.1 Effective LLL

In their paper [27] they studied the decision region of the ZF-DFE decoder and the effect of the LLL algorithm on that region. At high SNR the performance is dominated by the minimum distance in the decision region. The decision region of ZF-DFE decoder is a fundamental parallelogram centered at the transmitted codeword. This decision region can be specified by the Gram-Schmidt orthogonal vectors. The size reduction of a vector which is done by sub-algorithm RED(k, l) in Table 3.3, does not affect the size reduction of the other vectors. By studying the LLL algorithm it is not hard to see that, the size reduction does not affect the Gram-Schmidt orthogonal vectors.

As the decision region of the ZF-DFE decoder is determined by Gram-Schmidt orthogonal vectors $\mathbf{b}_i^*$, and the size reduction does not affect these vectors, therefore a weaker version of the LLL algorithm can be used in the MIMO detection applications. This makes it possible to develop a new version of the LLL algorithm, called effective LLL, which is suitable for MIMO detection. Because of the relaxed conditions, the effective LLL runs faster than the original LLL. In their paper, authors discuss that, it is not possible to totally remove the reduction part, and they have to perform some size reductions in order to maintain the same error performance as the original LLL. They prove in their paper that, it is enough to just size reduce adjacent vectors against each other to maintain the same error performance as the original LLL. Therefore a set of basis vectors are called effectively LLL reduced if

- $||\mu_{i-1,i}|| \leqslant \frac{1}{2}$ for $1 < i \leqslant m$, and
- $p \cdot ||\mathbf{b}_i^*||^2 \leqslant ||\mathbf{b}_{i+1}^* + \mu_{i+1,i}\mathbf{b}_i^*||^2$

The effective LLL terminates with the same number of iterations as the original LLL. As you can see the only thing missed in the new algorithm is size reducing against the other vectors which has not any effect on the Lovasz test. This alterations removes the most complex part of the algorithm. With this slight change, as stated before, there is no change in the error performance of the ZF-DFE detection

algorithm. Furthermore, the authors propose a way to convert the effectively LLL reduced basis to LLL reduced basis. They claimed that by performing the full size reduction at the end of the modified algorithm one can get the same output as the original LLL algorithm gives.

The complexity analysis of the Effective LLL figure can be seen in figure 3.2

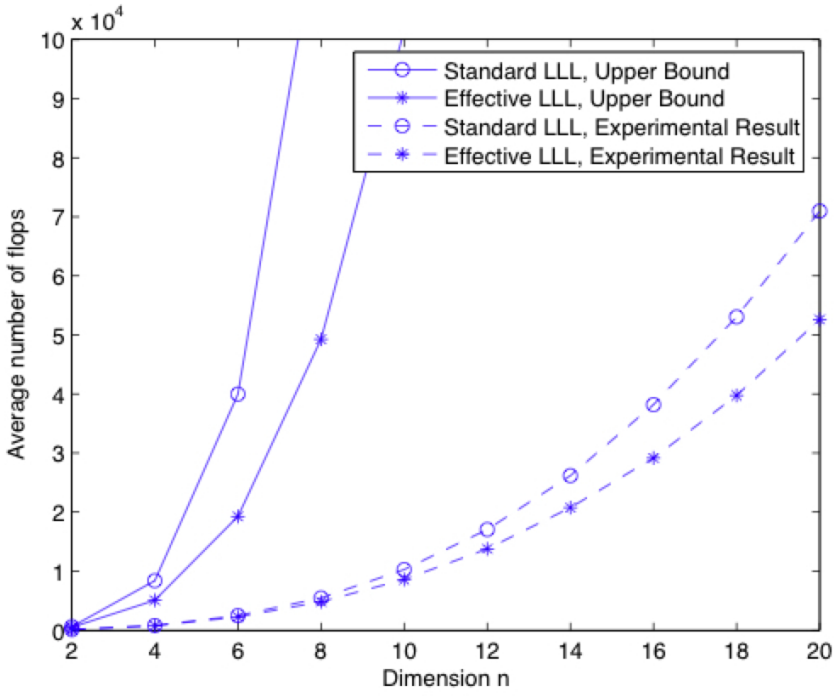


Figure 3.2: Effective LLL, complexity performance

3.4.2 Seysen's Lattice Reduction Algorithm

Authors of [28] proposed using Seysen's lattice reduction algorithm instead of LLL. They claimed that, this algorithm outperforms LLL in terms of error performance and has lower complexity for practical MIMO situations. In the proposed method the lattice basis and its dual are reduced at the same time, using an efficient algorithm. A local minimum for Seysen's Orthogonality measure is achieved using this method. Seysen's Orthogonality measure, like orthogonality defect, reaches the minimum value, if the basis vectors are orthogonal. Seysen's orthogonality measure

is defined as follows [29],

$$S(\tilde{\mathbf{B}}) = \sum_{i=1}^{m} ||\tilde{\mathbf{b}}_i||^2 ||\tilde{\mathbf{b}}_i^{\#}||^2 \tag{3.33}$$

In which $\tilde{\mathbf{b}}_i$ and $\tilde{\mathbf{b}}_i^{\#}$ are the ith basis vectors of the lattice and its dual respectively. The minimum for $S(\tilde{\mathbf{B}})$ is m, and is achieved if and only if, the basis vectors are orthogonal. The proposed method finds a local minimum of $S(\tilde{\mathbf{B}}) = S(\mathbf{BH})$ in an iterative way. In Seysen algorithm a basis update can be performed using the following relation,

$$\tilde{\mathbf{B}} = [\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_{s-1}, \tilde{\mathbf{b}}_s^{'}, \ldots, \tilde{\mathbf{b}}_M] \quad \text{with} \quad \tilde{\mathbf{b}}_s^{'} = \tilde{\mathbf{b}}_s + \lambda_{s,t}\tilde{\mathbf{b}}_t \tag{3.34}$$

$$\mathbf{H} = [\mathbf{h}_1, \ldots, \mathbf{h}_{s-1}, \mathbf{h}_s^{'}, \ldots, \mathbf{h}_M] \quad \text{with} \quad \mathbf{h}_s^{'} = \mathbf{h}_s + \lambda_{s,t}\mathbf{h}_t \tag{3.35}$$

For each pair of (i, j), the update for this pair should be done in a way that minimizes the resulting lattice. This can be done be choosing,

$$\lambda_{i,j} = \left\lfloor \frac{1}{2} \left(\frac{\tilde{\mathbf{b}}_j^{\#H}\tilde{\mathbf{b}}_i^{\#}}{||\tilde{\mathbf{b}}_i^{\#}||^2} - \frac{\tilde{\mathbf{b}}_j^{H}\tilde{\mathbf{b}}_i}{||\tilde{\mathbf{b}}_i||^2} \right) \right\rceil \tag{3.36}$$

If this minimization results in a non-zero value for $\lambda_{i,j}$, the updating results in better lattice. This means $S(\tilde{\mathbf{B}}_{i,j}) < S(\tilde{\mathbf{B}})$, and $\tilde{\mathbf{B}}_{i,j} = [\tilde{\mathbf{b}}_1, \ldots, \tilde{\mathbf{b}}_{i-1}, \tilde{\mathbf{b}}_i^{'}, \tilde{\mathbf{b}}_{i+1}, \ldots, \tilde{\mathbf{b}}_M]$. A basis is called SA-reduced if and only if, it is not possible to find any pair (i, j) that $\lambda_{i,j} \neq 0$. It is possible to find a SA-reduced basis using a greedy algorithm. In each iteration pair (s, t) is chosen in a way to maximize the amount of reduction.

$$(s, t) = \arg\max_{(i,j)} S(\tilde{\mathbf{B}}) - S(\tilde{\mathbf{B}}_{i,j}) \tag{3.37}$$

The basic idea behind Seysen Algorithm is expressed as in table 3.7.

3.5 Proposed Method

In this paper we are trying to deal with a case that the assumption of flat fading is not valid. This means, the channel is changing in a way that, we can not assume it to be fixed for the whole frame. This means that we have to perform the preprocessing stage, every time a set of signal is received in receive antennas. As it was formerly discussed, the preprocessing stage can be quite complex because of the reduction

Table 3.7: Seysen lattice reduction algorithm

Step 1:	Initialize
	$\mathbf{H} = \mathbf{I}_N$ and $\tilde{\mathbf{B}} = \mathbf{B}$
Step 2:	Seysen Reduction Algorithm
	Repeat until $\tilde{\mathbf{B}}$ is SA-reduced
	for each possible pair (i, j) compute $\lambda_{i,j}$
	for each pair (i, j) compute $\Delta_{i,j} = S(\tilde{\mathbf{B}}) - S(\tilde{\mathbf{B}}_{i,j})$
	choose (s, t) that maximizes $\Delta_{i,j}$
	Update the basis according to (3.34) and (3.35)

required to perform in it.

We know that the channel has changed slightly from previous reception of signal. This triggers the idea that the previous steps done for the basis reduction of the previous channel might be quite useful.

As it was mentioned before the output of the LLL algorithm is a reduced matrix and a transformation matrix. The transformation matrix is used to convert the initial channel matrix to the reduced one. This relation can be expressed as follows,

$$\mathbf{H}_{red,1} = \mathbf{H}_1 \mathbf{U}_{trans,1} \tag{3.38}$$

In which $\mathbf{H}_{red,1}, \mathbf{H}_1, \mathbf{U}_{trans,1}$ are the reduced matrix, channel matrix, and the transformation matrix respectively. The reduced matrix has the property that it satisfies the Lovasz and Size Reduce conditions. Studying the Lovasz and Reduce conditions in the LLL algorithm, shows that they merely depend on the Gram-Shmidt vectors of the reduced matrix. Considering a small value for f_m in the MIMO fading channel model, the variations in each element of the channel will be modest through time. This can be written as follows,

$$\mathbf{H}_2 = \mathbf{H}_1 + \Delta\mathbf{H} \tag{3.39}$$

in which $\mathbf{H}_1$ and $\mathbf{H}_2$ are two consecutive channel realizations and $\Delta\mathbf{H}$ is a matrix with small elements. Considering these small changes, it seems quite reasonable to make use of the previous transformations for $\mathbf{H}_1$ in computing the reduced matrix for the new channel $\mathbf{H}_2$. Using the previous transformation on the new matrix

results in,

$$\begin{aligned}\tilde{\mathbf{H}}_2 &= \mathbf{H}_2\mathbf{U}_{trans,1} \\ &= (\mathbf{H}_1 + \Delta\mathbf{H})\mathbf{U}_{trans,1} \\ &= \mathbf{H}_1\mathbf{U}_{trans,1} + \Delta\mathbf{H}\mathbf{U}_{trans,1} \\ &= \mathbf{H}_{red,1} + \Delta\mathbf{H}\mathbf{U}_{trans,1} \end{aligned} \tag{3.40}$$

Using the fact that $\mathbf{U}_{trans,1}$ is a uni-modular matrix, $\tilde{\mathbf{H}}_2$ is still a basis for the space spanned by columns of $\mathbf{H}_2$. As $\mathbf{H}_{red,1}$ is already LLL reduced, it satisfies all the Lovasz and size reduce conditions. If the other term in equation (3.40) is small enough, the Gram-Smidt coefficients of the resulting matrix in the right-hand side of equation (3.40) are close to those of the $\mathbf{H}_{red,1}$. Therefore, performing LLL on $\tilde{\mathbf{H}}_2$, does not require many more swaps and reductions. Thus, using $\tilde{\mathbf{H}}_2$ as the initial input for the LLL algorithm reduces the complexity of the LLL algorithm significantly, in comparison with starting from the original channel matrix $\mathbf{H}_2$. Roughly speaking, starting from an almost LLL reduced matrix results in less complexity for LLL reduction. Performing LLL on $\tilde{\mathbf{H}}_2$ results in,

$$\mathbf{H}_{red,2} = \tilde{\mathbf{H}}_2\tilde{\mathbf{U}}_{trans,2} \tag{3.41}$$

$$= (\mathbf{H}_2\mathbf{U}_{trans,1})\tilde{\mathbf{U}}_{trans,2} \tag{3.42}$$

$$= \mathbf{H}_2\mathbf{U}_{trans,2} \tag{3.43}$$

In which $\mathbf{U}_{trans,2} = \mathbf{U}_{trans,1}\tilde{\mathbf{U}}_{trans,1}$ is the transformation matrix for reducing $\mathbf{H}_2$. Note that, the proposed adaptive method of matrix reduction results in a reduced matrix which might be different from the result of applying the LLL algorithm directly on $\mathbf{H}_2$, but as both resulting matrices are LLL reduced,the error performance of the MIMO decoder should be the same.

For using this adaptive method another input should be added to the LLL algorithm in Table 3.2. This input is used to pass the transformation of the previous channel realization to the LLL algorithm. Furthermore, In Step 1 of the LLL algorithm, the transformation matrix, should be initialized by the new input instead of the identity matrix. As mentioned above, this revised algorithm should be called to reduce $\tilde{\mathbf{H}}_2$.

3.6 Simulations Results

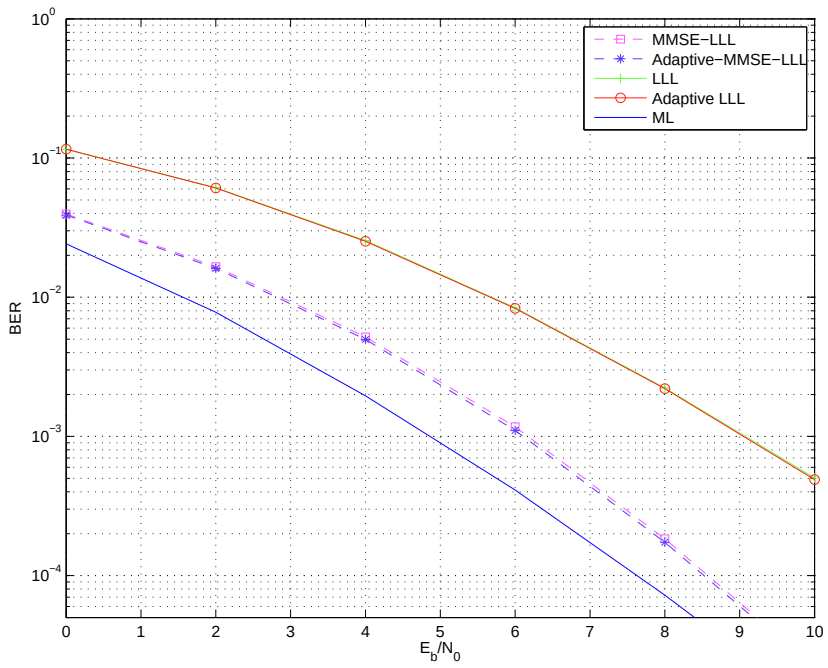
In this section performance and complexity of the adaptive reduction method is studied and compared with other conventional reduction methods. We consider the MIMO channel with $M = N$, transmit and receive antennas. Channel is assumed to be the Rayleigh fading channel explained in section 2.2. A 4-QAM constellation is used to investigate the error performance of the proposed method. Figure 3.3 shows the bit error rate versus the average transmitted energy per information bit E_b, divided by one sided noise power spectral density N_0. In this simulation the proposed method is used to modify the deep insertion LLL("DILLL") and the LLL algorithm as the one in [6]. As it was expected, it can be seen in figure 3.4(b), the proposed method("Adaptive DILLLL") has the same error performance as the original deep insertion LLL algorithm. Also it can be seen from 3.4(a) that using the proposed method does not affect the performance of the LLL algorithm.

We study different aspects of complexity to assess the performance of the proposed method. The number of flops, number of basis updates, and running time are used as measures to compare complexity. Figure 3.4 illustrates the average running time for different number of antennas for the modified deep insertion LLL and the conventional deep insertion LLL. The simulations for running times are run on a Pentium Dual Core 2.4GHz, with 2GB of RAM, and under Fedora Linux 7.0. Furthermore, all the source codes are in mex files to produce better running times. The running time is averaged over one million channel realizations, in which each 10000 channels form a frame and they are correlated with each other. In order to measure time, the MATLAB 7.0 {`tic,toc`} functions are used. It is seen that the adaptive method improves the running time significantly with respect to the conventional method. It is also seen that the time saving from using the adaptive algorithm increases with increasing the lattice dimension in the deep insertion LLL case.

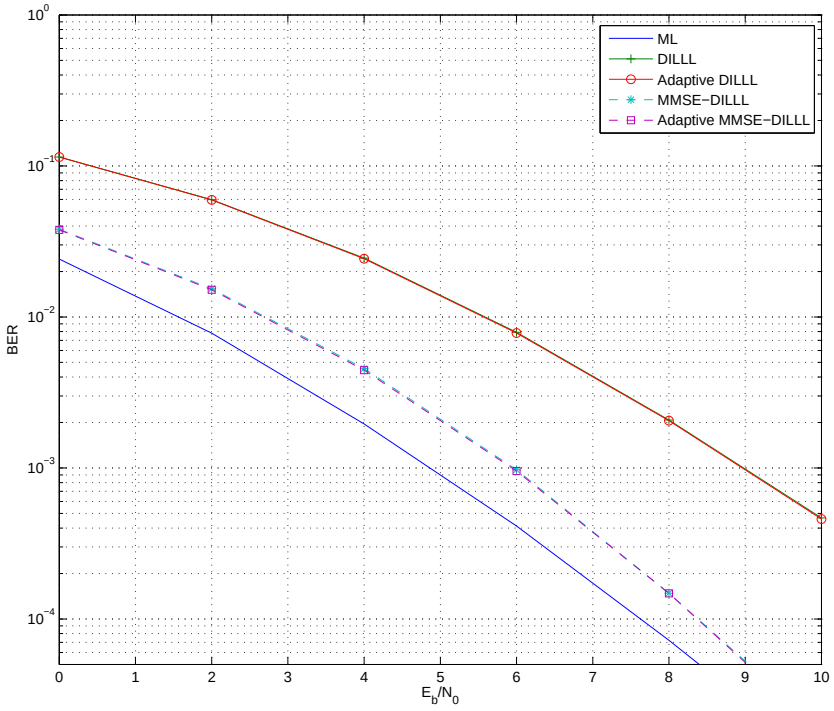
Figure 3.5 compares average number of basis updates for the LLL algorithm and the Adaptive LLL algorithm. As you can see in the figures there exist an obvious gain in using the proposed method without loosing in error performance.

Figure 3.6 shows the average number of required reductions and insertions for both deep insertion LLL and adaptive deep insertion LLL. As it was expected from the adaptive nature of the proposed method, there is a significant gain in terms of number of basis updates. This admits that for the channel we have consid-

Figure 3.3: Error performance of the proposed algorithm



(a) LLL Algorithm



(b) Deep Insertion LLL algorithm

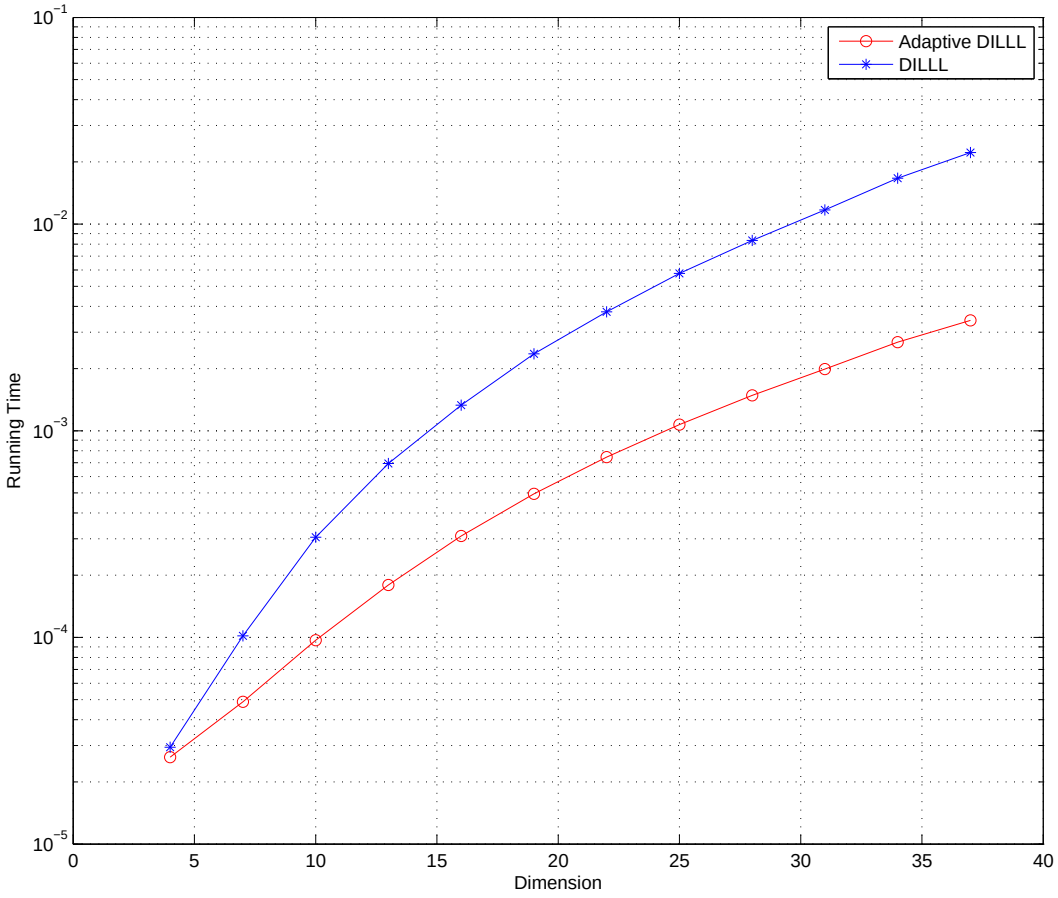


Figure 3.4: Running Time Comparison of the proposed and conventional reduction methods

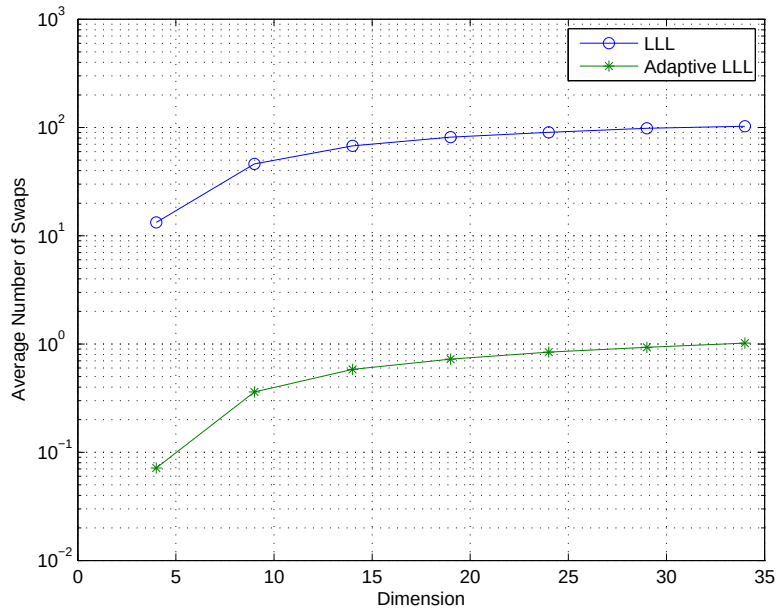
ered in these simulations, using the transformation matrix of the previous channel realization gives, a quite good reduction for the current channel realizations.

As the next step, we use the average number of flops as another measure for complexity analysis as in [30] and [31]. The counting is only done for the lattice reduction part and the other parts such as detection, channel simulations, and etc. are not counted for, in this simulation. Figure 3.7 contains a comparison of the the LLL algorithm and the Adaptive LLL algorithm. Figure 3.8 compares the average number of flops for deep insertion LLL and adaptive deep insertion LLL. This is seen that the adaptive methods outperform the non-adaptive algorithms in terms of flops, and as the number of antennas increases, for the deep insertion case in contrast to the original LLL, the gap between the two algorithms, tends to get larger. This might be justified by the exponential nature of the deep insertion LLL, but further analysis requires a more in depth study of the LLL and DILLL algorithms.

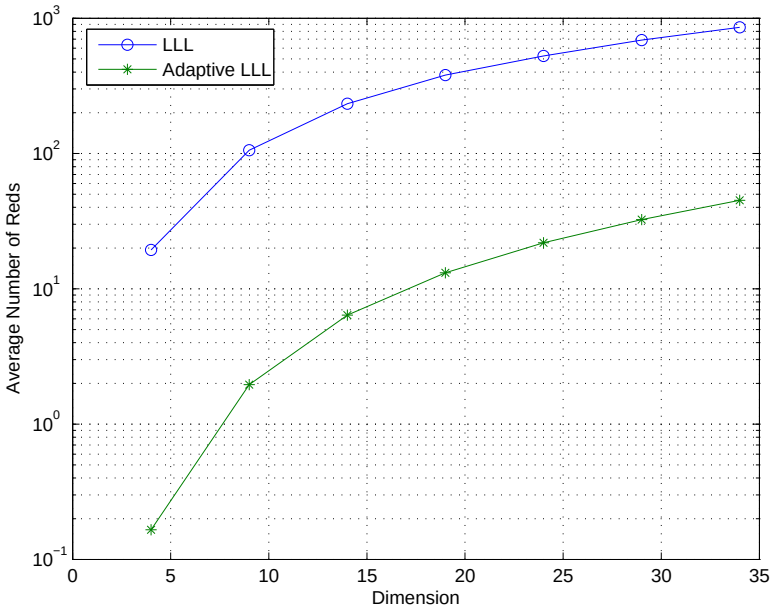
Figure 3.9 compares the cumulative density function (CDF) of adaptive LLL and LLL. Also in figure 3.10 a comparison between the CDF of the number of flops for the deep insertion LLL and the adaptive deep insertion LLL, is made. The empirical CDF is calculated and plotted for different number of antennas. As you can see in the figures the adaptive methods are better than the non-adaptive methods for all the considered number of antennas. Comparing the three plots in each figure, you can see that, the number of flops required to reduce the matrix increases by an increase in number of antennas.

Next in figure 3.11 the behavior of the adaptive deep insertion LLL is investigated for different values of f_m and it is compared to deep insertion LLL. The range used for f_m, is tried to be consistent with the practical data of the GSM standard. Velocities investigated for the moving receiver varies from walking speed to a high speed train. In this plot for each value of f_m, number of reductions(insertions) of the deep insertion LLL over the number of reductions(insertions) of adaptive deep insertion LLL is plotted. The simulation results show that along with increasing the speed of the moving receiver, the complexity gain from using the proposed method gets smaller. Also it can be seen that the proposed algorithm performs better than the deep insertion LLL for all the velocities considered.

Figure 3.5: Average Number of basis updates for different number of antennas (LLL)

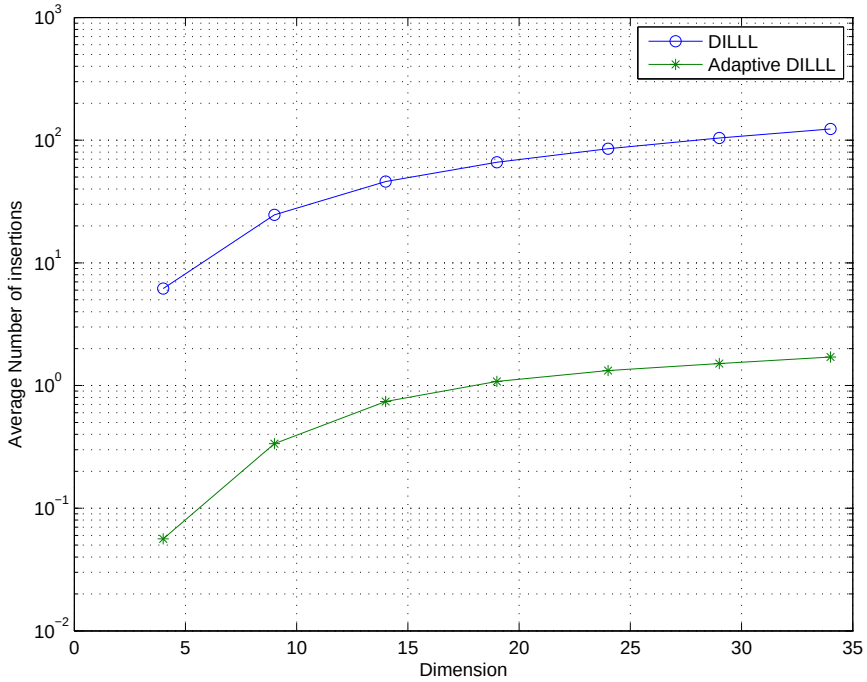


(a) Average Number of Size Reductions

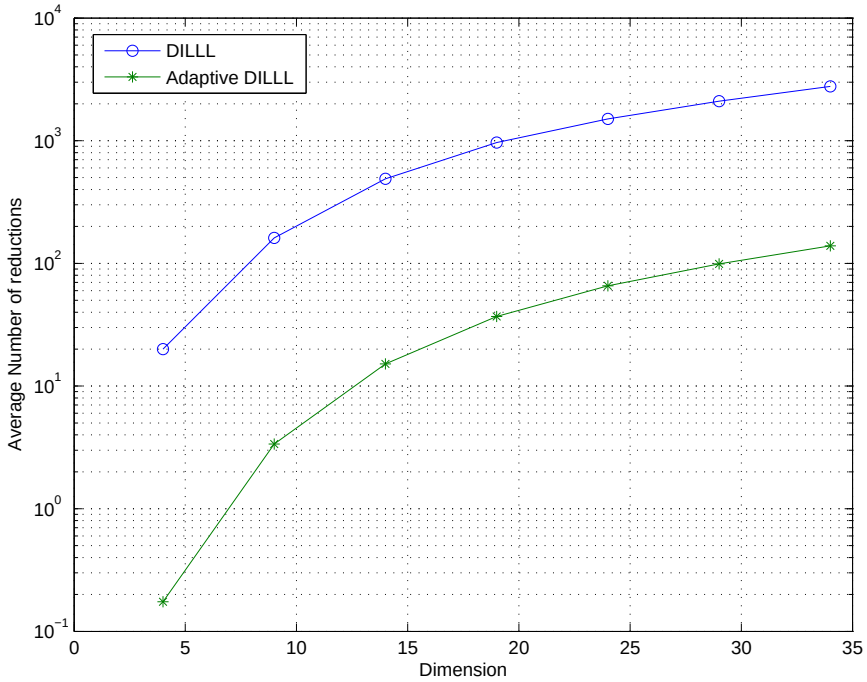


(b) Average Number of Swaps

Figure 3.6: Average number of basis updates for different number of antennas(DILLL)



(a) Average Number of Insertions



(b) Average Number of Size Reductions

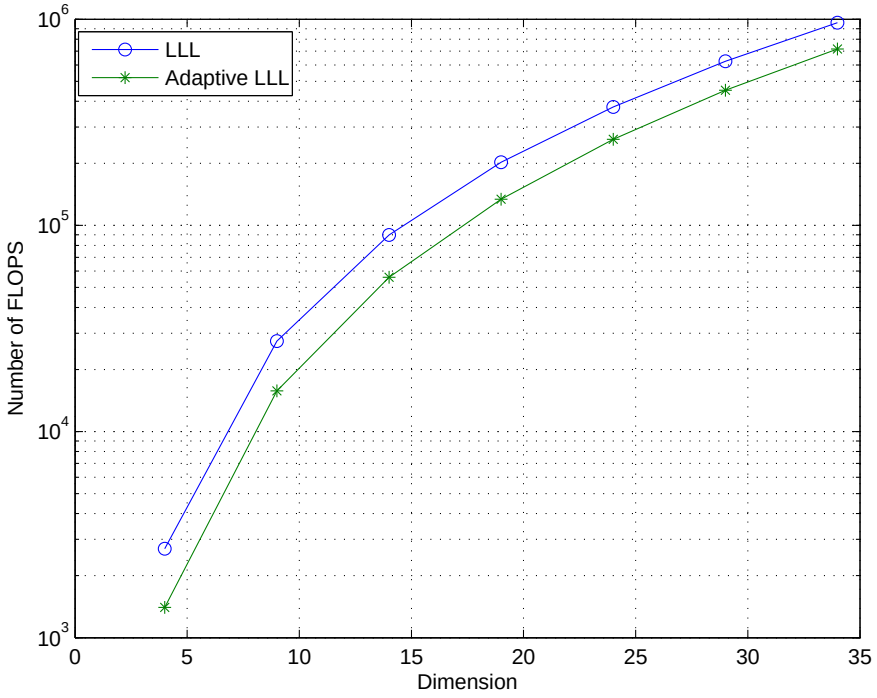


Figure 3.7: Average number of flops for different number of antennas (LLL)

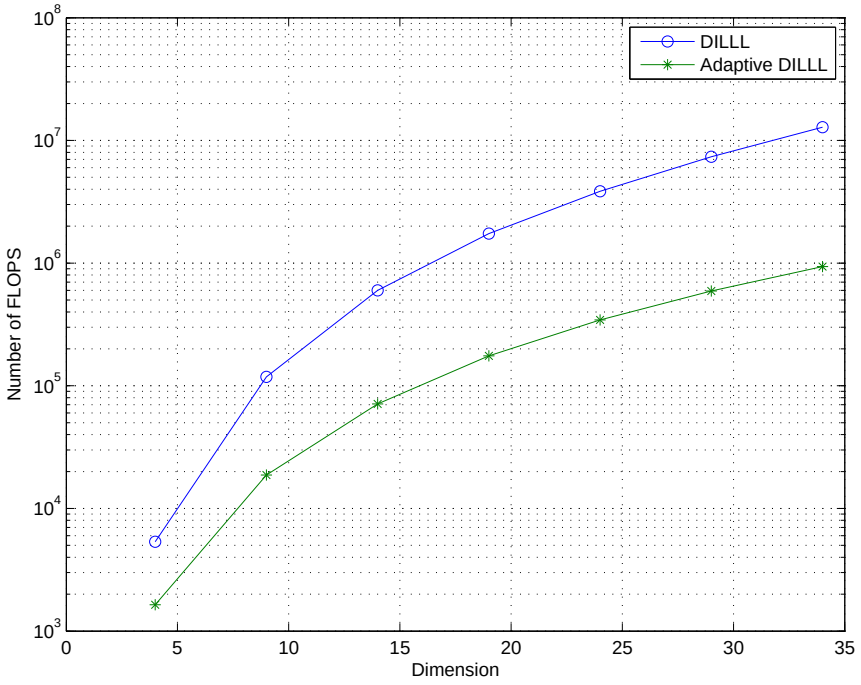
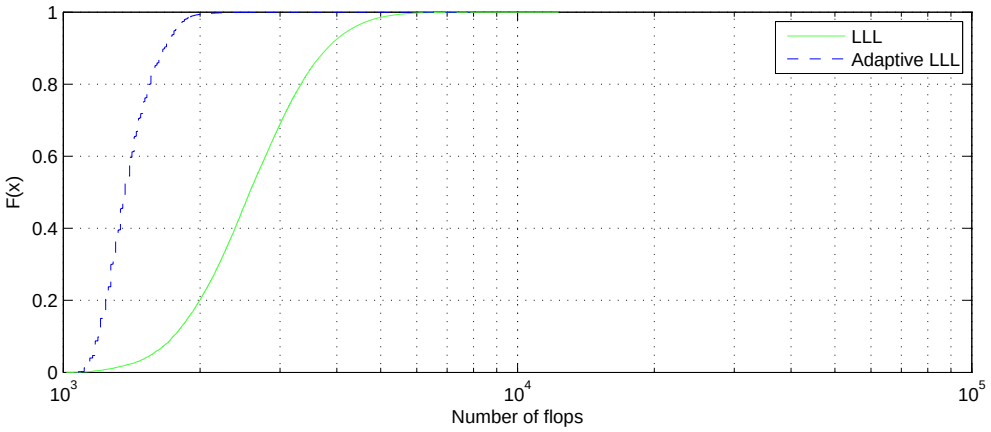
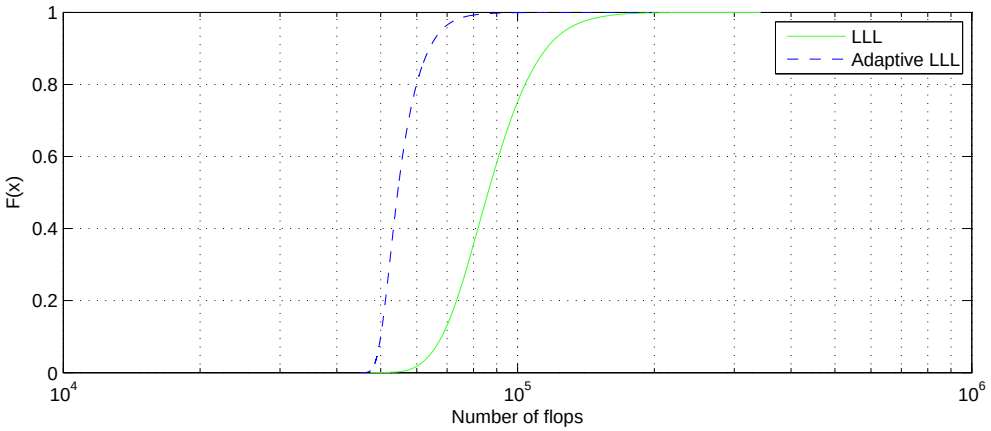


Figure 3.8: Average number of flops for different number of antennas (DILLL)

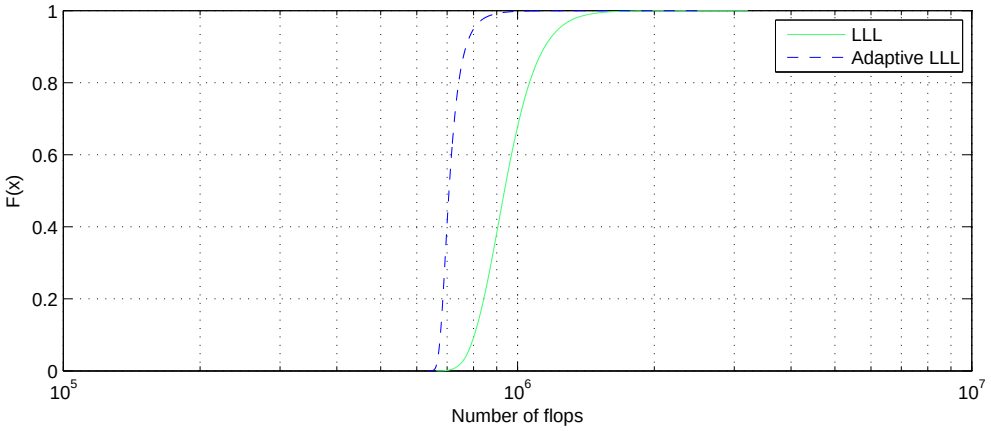
Figure 3.9: CDF of number of flops for different number of antennas (DILLL)



(a) 4×4 4-QAM

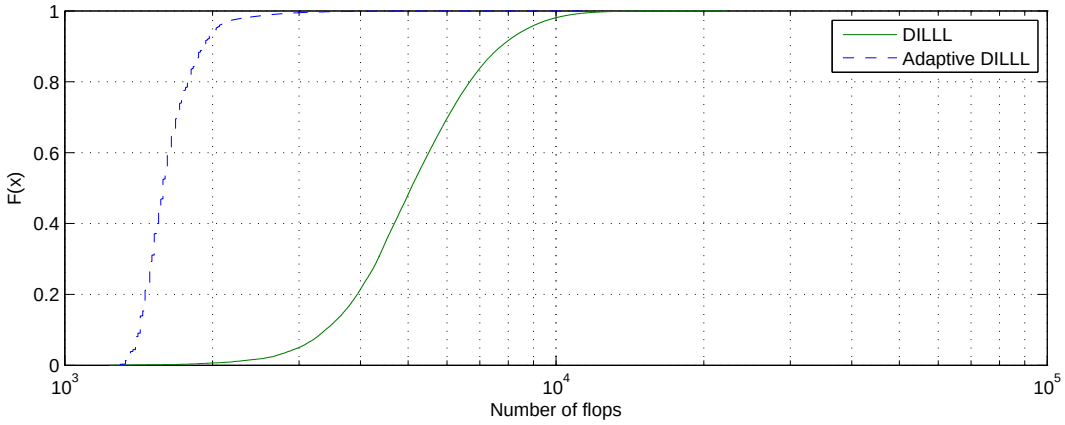


(b) 14×14 4-QAM

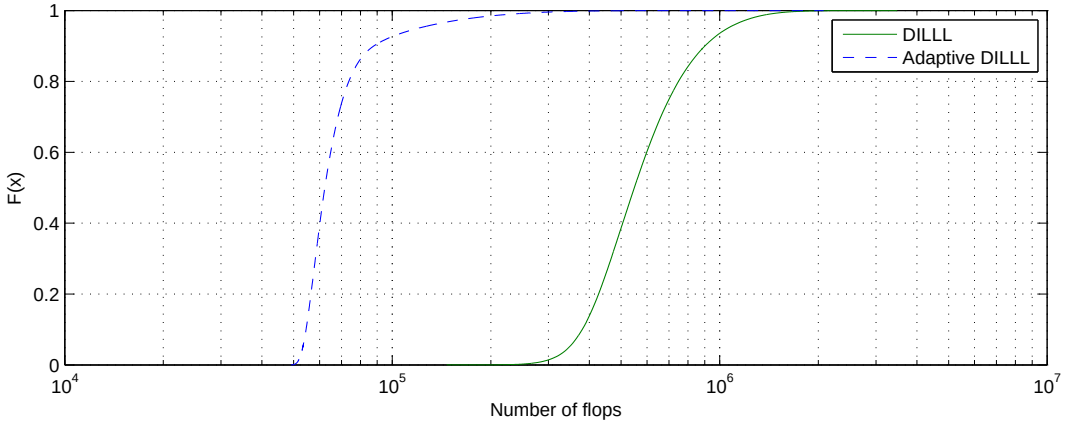


(c) 34×34 4-QAM

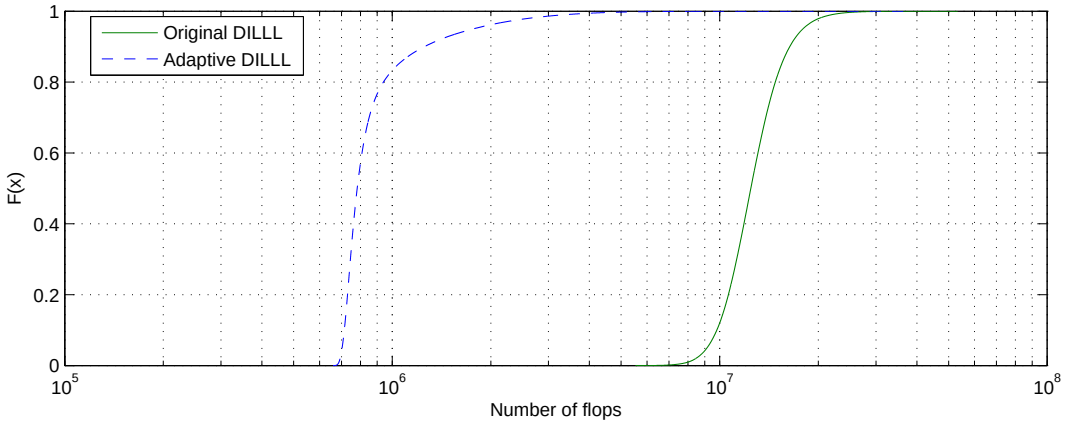
Figure 3.10: CDF of number of flops for different number of antennas(DILLL)



(a) 4×4 4-QAM



(b) 14×14 4-QAM



(c) 34×34 4-QAM

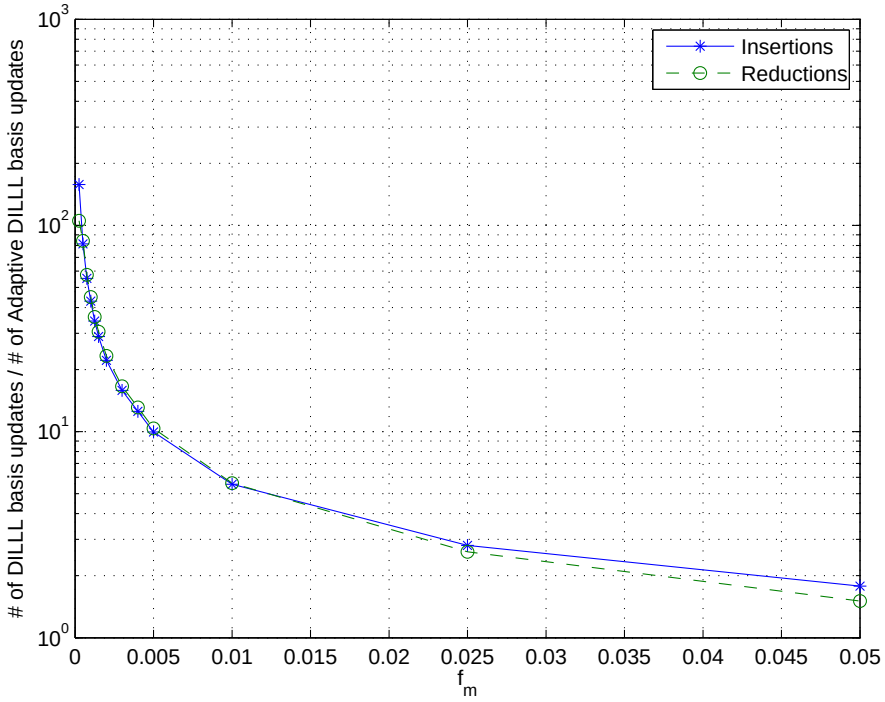


Figure 3.11: Performance of Adaptive DILLL for different f_m(chosen based on GSM model)

Chapter 4

Broadcast System Model

Quite recently the multiuser broadcast problem received considerable attention in comparison with the point to point scenario. Gains of using multiple antennas in a single user scenario is now apparent to every one. Caire and Shamai [32] showed that many of these advantages translate to their counter part in the multiuser scenario. Authors of [1] and [2] showed that the capacity of MIMO point-point channel increases linearly with the minimum number of the receive and transmit antennas. Many different techniques have been proposed by different authors that exploit these advantages. Equivalently, it is also shown in [1] that the sum capacity defined as the total of transmission rates of all the users grows linearly with the minimum of the number of users and number of transmit antennas. Achieving the sum rate capacity is shown to be possible via using some information theoretic schemes, based on dirty paper coding. Practical solutions are investigated by researchers in recent years. Nested lattices is a method proposed by the authors of [33] as a practical method for interference channel and general multiuser channel. These methods are still of high complexity for implementation.

Pre-coding techniques are lately used as practical and less complex methods, in multiuser MIMO systems. Channel inversion is one of the most obvious methods proposed in [32] as a solution. In this technique the data for different users can effectively separated at the receiver side. Opposed to its ease for implementation, channel inversion method is vulnerable to poor-channel conditions. The near singularity of the channel results in poor symbol error rate for this method.

Vector perturbation method is another method proposed in [34] that overcomes this problem. However this technique requires a lattice decoder which is NP-hard in general.

In this work a multiple antenna broadcast system with N_T transmit antennas and $K \leqslant N_T$ users with single antennas are considered. Considering $\mathbf{x}^c = [x_1^c, ..., x_{N_T}^c]^T$, $\mathbf{y}^c = [y_1^c, ..., y_K^c]^T$, $\mathbf{n}^c = [n_1^c, ..., n_K^c]^T$ and the $N_T \times K$ matrix $\mathbf{H}^c$, respectively as the transmitted signal, the received signal, the noise vector and the channel matrix, this system model can be expressed as follows

$$\mathbf{y}^c = \mathbf{H}^c\mathbf{x}^c + \mathbf{n}^c \tag{4.1}$$

Moreover the transmitted signal energy is constrained with $E(||\mathbf{x}^c||^2) = 1$ and elements of the noise are assumed to be i.i.d complex Gaussian random variables, i.e., $n_k \sim \mathcal{CN}(0, \sigma_n^2)$. A requirement for all the pre-equalization techniques is knowing the channel state. In this work it is assumed that the channel state is completely known at the transmitter. Using the same relations as the ones in chapter 3, the system can be converted to the real counterpart which results in the following real valued model,

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{4.2}$$

The data to be transmitted to the K users, will be denoted as $\mathbf{a}^c = [a_1^c, \ldots, a_K^c]^T$. Each a_i belongs to a M-QAM constellation. This means for example for $M = 4$, $a_i^c \in \{\pm(1/2) \pm j(1/2)\}$, $i = 1, \ldots, K$. For the real equivalent of the model this would be $a_i \in \{\pm(1/2)\}$, for $i = 1, \ldots, 2K$.

In a broadcast system the users can not cooperate with each other, and each one has to decode its data separately. The main idea to overcome this constraint is to use pre-coding techniques at the transmitter side, which make it possible for the users to decode their data separately. The easiest method in the category of pre-coding techniques is the channel inversion.

4.1 Channel Inversion

In this method the complex model of the system is used. At the transmitter side the data vector is multiplied by right pseudo inverse of the channel and then sent over the channel. In this way each user at the receiver side only receives its data plus noise. This method can be written explicitly as follows,

$$\mathbf{x}^c = \mathbf{H}^{c\dagger}\mathbf{a}^{\mathbf{c}} \tag{4.3}$$

where $\mathbf{H}^{c\dagger} = \mathbf{H}^{cH}(\mathbf{H}^{c}\mathbf{H}^{cH})^{-1}$ is the right pseudo inverse of $\mathbf{H}^{c}$. In this method each user can decode its data by simply quantizing its received signal to the QAM constellation.

There exist a problem that makes this system inefficient in terms of symbol error rate. When the channel is poorly conditioned i.e. having a determinant of near zero, this method results in infinite power meaning that $||\mathbf{x}^{c}||$ will be quite large. This can happen when one of the singular values of $\mathbf{H}^{c}$ is very small , which results in vectors with large norm as the columns of $\mathbf{H}^{cH}$.

In [35] a better method in terms of power efficiency, called Tomlinson-Harashima, was proposed by the authors. This method requires a modulo operation at the transmitter side, and uses another modulo operation at the receiver side before quantizing the received signal to QAM constellation.

Both these methods perform linear pre-equalization for at least one of the sub channels, and therefore they both suffer from the imposed diversity order of $N_T - K + 1$.

4.2 Search based broadcast precoding

In this method which is proposed in [34], the authors use the real equivalent of the system and they employ a modulo operation at the receiver side, which can be written as follows,

$$\tilde{y}_k = y_k \mod A = y_k - A\left\lfloor \left(y_k + \frac{A}{2}\right)/A \right\rfloor \tag{4.4}$$

The scalar A should be chosen in a way that all the constellation points used in the transmitter side can be uniquely recovered at the receiver side. For the case of 4-QAM constellation $A = 2$ can be used. In this method the transmitted signal is based upon the formula that follows,

$$\mathbf{x} = \mathbf{H}^{\dagger}(\mathbf{a} + \mathbf{p}) \tag{4.5}$$

$$\mathbf{p} = \arg \min_{\mathbf{p}' \in A\mathbb{Z}^{2K}} ||\mathbf{H}^{\dagger}(\mathbf{a} + \mathbf{p}')||^2 \tag{4.6}$$

Taking a closer look at (4.6) one can see that the minimization problem in (4.6) is a closest lattice point search. As it was mentioned in chapter 3 the solution to this

problem is in general NP-hard. For this system the receiver observes

$$\mathbf{y} = \mathbf{H}\mathbf{H}^{\dagger}(\mathbf{a} + \mathbf{p}) + \mathbf{n} \tag{4.7}$$

$$= \mathbf{a} + \mathbf{p} + \mathbf{n} \tag{4.8}$$

Therefore each user in the receiver side can decode its data by first using the modulo operation and then rounding to the QAM constellation.

$$\tilde{y}_k = (a_k + p_k + n_k) \mod A = (a_k + n_k) \mod A \tag{4.9}$$

4.3 Lattice-Reduction-Aided Broadcast Pre-coding

Comparing the search based broadcast precoding with the channel inversion and Tommlinson-Harashima linear pre-equalization methods, one may find it quite complex. In order to reduce complexity, authors of [10] offer to solve the minimization in (4.6) with an approximate method similar to the one used in [36] and [37]. The LLL lattice reduction can be used to solve equation (4.6) for an approximate value. Having the real equivalent data $\mathbf{a}$ and the real equivalent channel $\mathbf{H}$ the LLL algorithm can be performed to get,

$$\mathbf{W} = \mathbf{H}^{\dagger}\mathbf{R} \tag{4.10}$$

In which $\mathbf{R}$ is a uni-modular transformation matrix and $\mathbf{W}$ is the reduced lattice. Using the procedure *rounding off* in [38] the approximate solution can be written as,

$$\mathbf{p}_{approx.} = -\mathbf{R}Q_{A\mathbb{Z}^{2K}}\{\mathbf{R}^{-1}\mathbf{a}\} \tag{4.11}$$

in which $Q_{A\mathbb{Z}^{2K}}\{\cdot\}$ is used to denote componentwise rounding of a $2K$-dimensional vector to the scaled integer lattice. The authors of [10] claimed and showed by simulations that their proposed method achieves a diversity gain, equal to the one for the vector perturbation method in [34].

In figure 4.1 you can see the simulation results by [10],

4.4 Proposed Method

The idea is the same as the one in section 3.5. As it was mentioned before, in Lattice Aided reduction method the goal is to find the approximate solution to

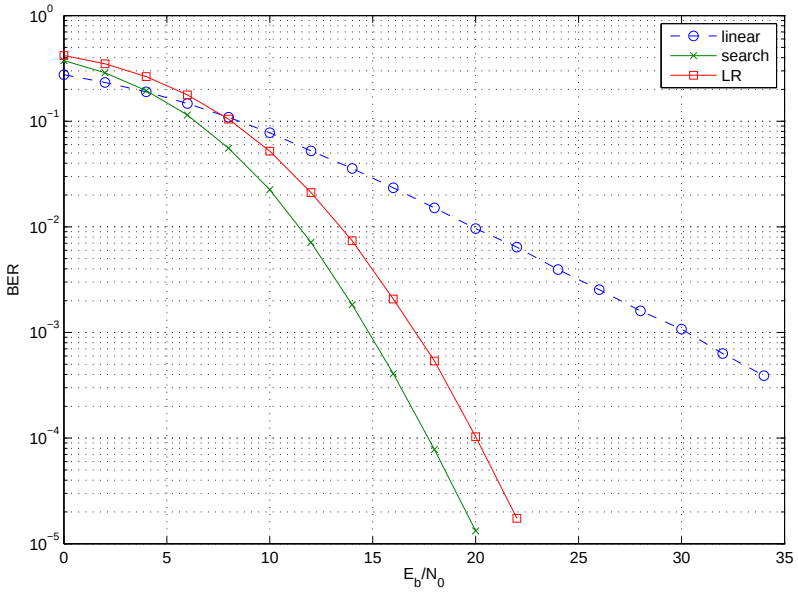


Figure 4.1: Simulation Results for different precoding schemes $N_T = K = 4$, 4-QAM in channel without temporal correlation

4.11. Therefore, when the channel realizations are correlated in time, as the one we are using as our channel model, it is possible to take advantage of this correlation to reduce the complexity of precoding stage in a broadcast system. The Adaptive Lattice Aided Reduction which can be defined as performing the adaptive method explained in 3.5 to approximate 4.11 is a way to achieve this reduction in complexity.

4.5 Simulation Results

In this section, the results of using the proposed method, Adaptive LLL is represented. The results are compared with the results of the none adaptive case. As the simulation results for the MIMO detection case, error performance and the complexity of the proposed method is considered. We consider a Rayleigh fading channel in which each 10000 are correlated in time.

For error performance analysis, a system with $N_T = 4$ transmitter antennas, and $K = 4$ users is considered. The constellation used for this system is a 4-QAM constellation. In figure 4.2, bit error rate is plotted against E_b/N_0. As it was expected, the proposed method has the same performance as the Lattice Reduction Aided Precoding method that uses the non-adaptive LLL. It can also be seen that

the error performance of the system is exactly the same as the case without any temporal correlation between channel realizations.

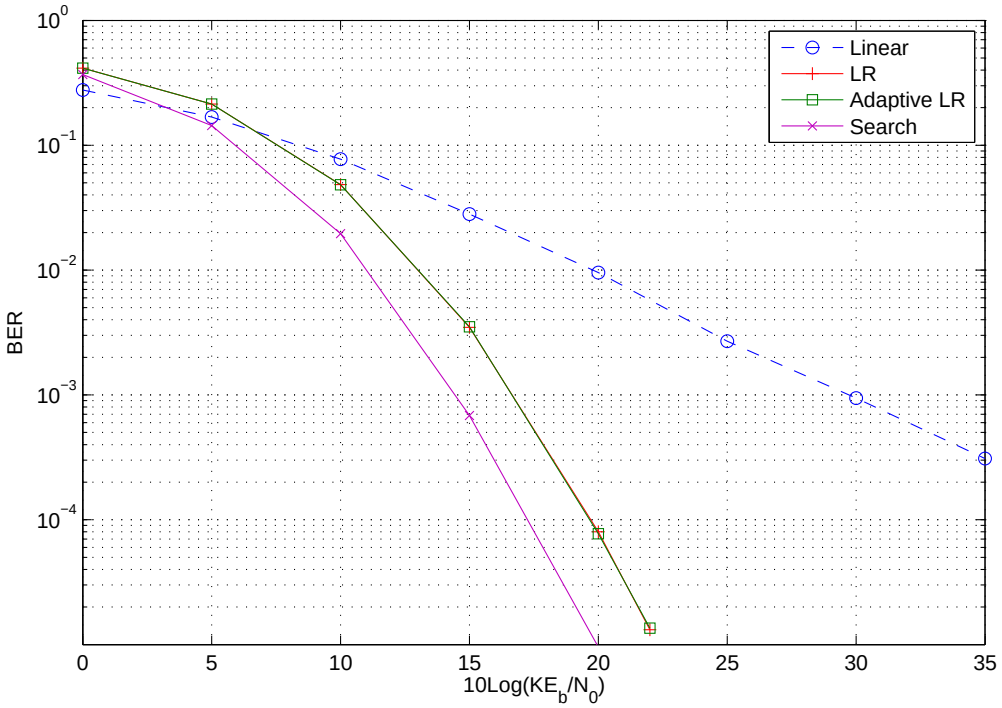


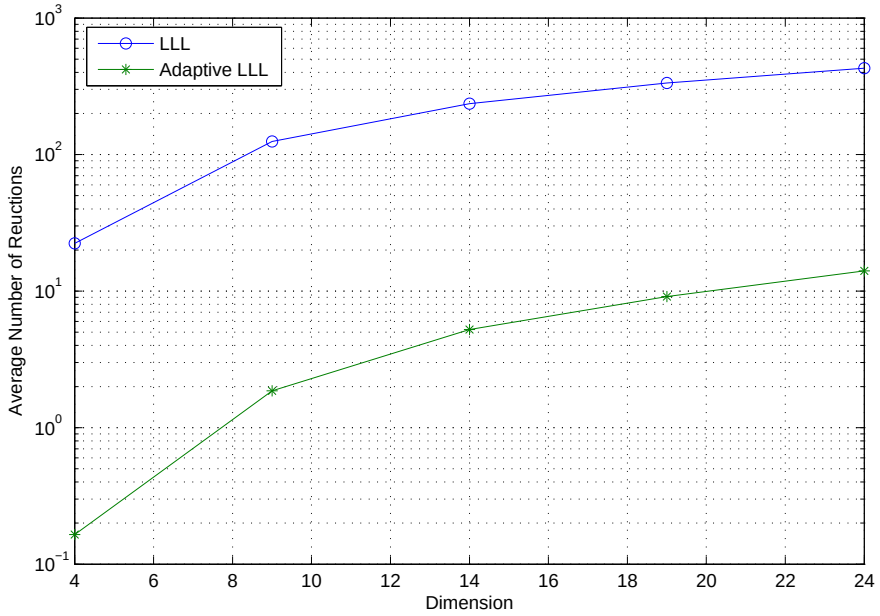
Figure 4.2: Error performance of the proposed method in temporal correlated channel

In figure 4.3 the number of basis updates are considered. Simulations show that using the adaptive LLL method instead of the LLL, gives an almost constant gain in terms of average number of basis updates over the the range of antennas considered.

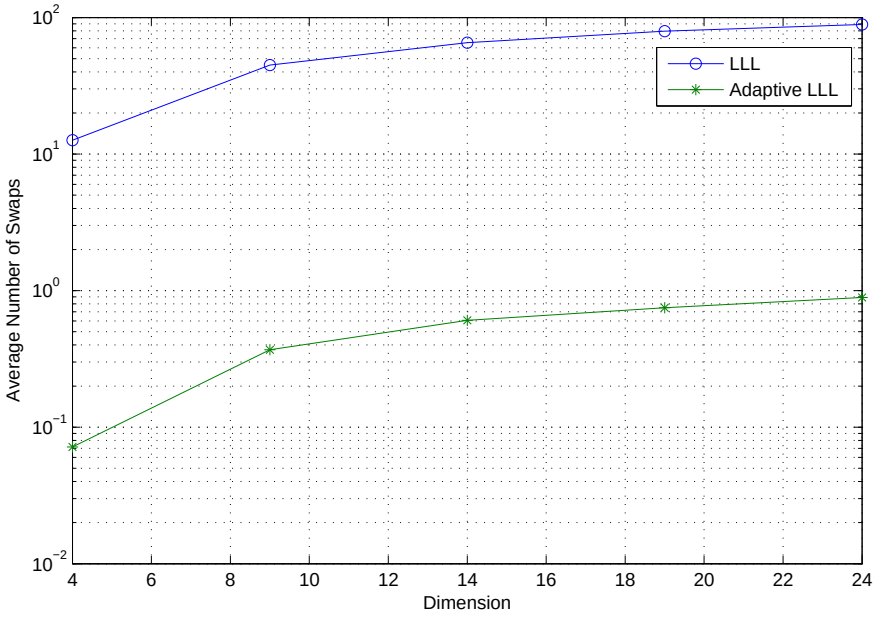
Figure 4.4 compares the average number of flops required for the reduction, for the LLL and the Adaptive LLL method. For the range of number of antennas considered, there has always been a gain in using the Adaptive method. However the gain seems to be lessened as the number of antennas is increased.

In figure 4.5 number of flops is considered as a measure for comparing the complexity of the Lattice Reduction method using LLL, and Adaptive LLL. As it can be seen from the figure, for all the dimensions considered, the adaptive LLL method was able to reduce the complexity of the reduction method. Comparing the three plots in figure 4.5, one can see that the increase in number of antennas results in an increase in number of flops required in the precoding stage.

Figure 4.3: Average number of basis updates for different number of antennas



(a) Average Number of Reductions



(b) Average Number of Swaps

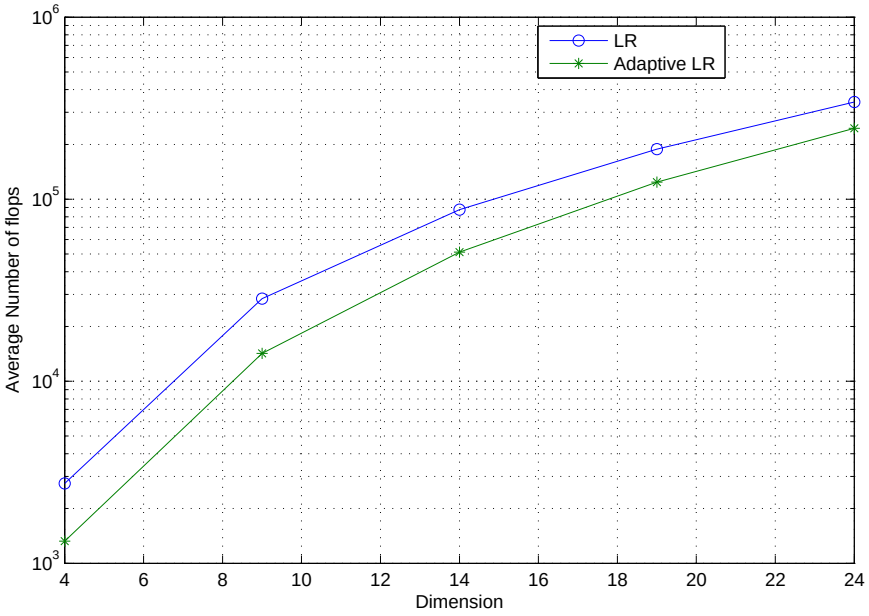
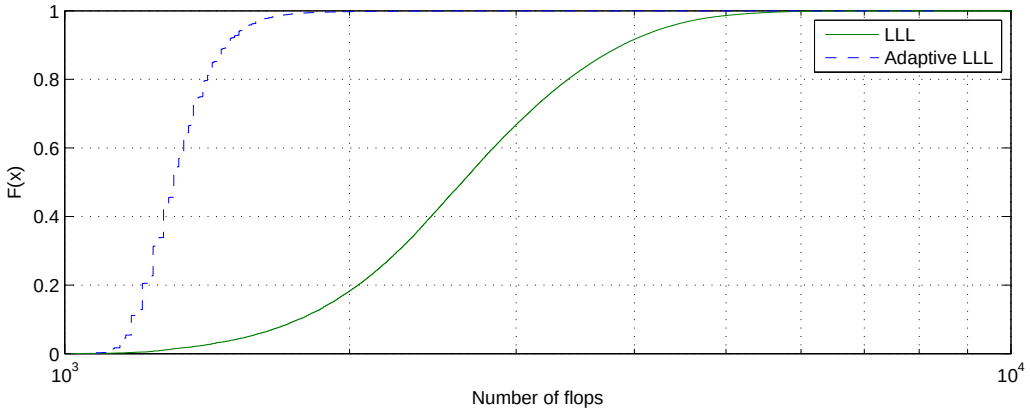


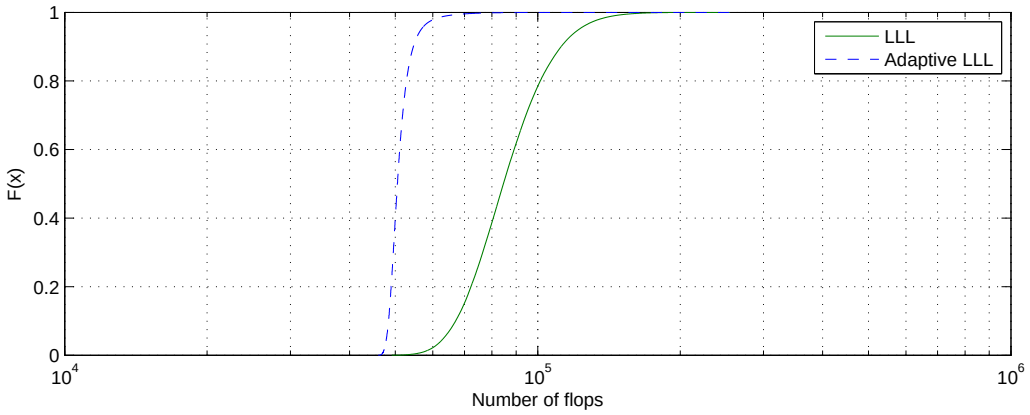
Figure 4.4: Average Number of flops for Lattice Aided Reduction Precoding

To summarize, the curves show that, for the range of number of antennas considered there has always been a gain in complexity by using the Adaptive method, without any loss in error performance.

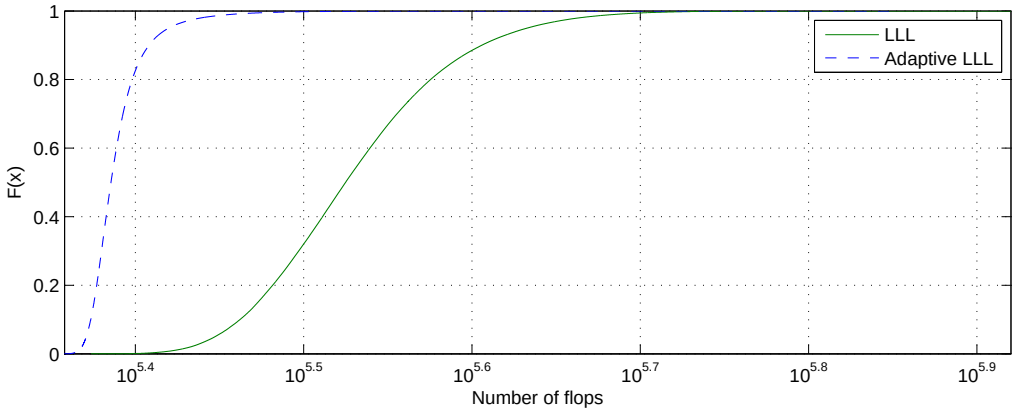
Figure 4.5: CDF of number of flops for different number of antennas



(a) 4×4 4-QAM



(b) 14×14 4-QAM



(c) 24×24 4-QAM

Chapter 5

Conclusions and Future Work

In this work different methods used in MIMO detection scenario were studied. Most of the efficient decoding techniques require a preprocessing stage that involves lattice reduction. In a Rayleigh fading channel that the channel realizations are correlated in time, it was shown that it is possible to take advantage of this temporal correlation, and reduce the complexity of the lattice reduction, and therefore the preprocessing stage. This helps the decoding systems to be practical to be used in today's communication devices that have a constraint on energy and processing power.

As it was explained throughout this work, using the adaptive method improves the complexity of the lattice reduction algorithms in a Rayleigh fading channel MIMO system, without any loss in error performance. This makes the proposed algorithm to be quite practical and appealing to be used in any MIMO scenario that needs lattice reduction.

Furthermore, the use of the proposed algorithm was tested in the Broadcast scenario. In Broadcast scenario the lattice reduction was used to reduce the pseudo inverse of the channel. The proposed algorithm gave us the same gains in complexity, again without any loss in power efficiency of the system.

As a future work the performance of the proposed algorithm can be studied in MAC protocol. It is assumed that it will again improve the complexity without any loss in error performance.

Also, it is possible to perform the Gram-Schmidt orthogonaliztion adaptively. In matrix algebra there are methods to update the Gram-Schmidt orthogonal vectors for a rank one and rank two updates [39]. This method can be added to our proposed

method, and it will further reduce the complexity of the lattice reduction, and it will make lattice reduction more practical to be use in MIMO systems.

References

[1] I. E. Teletar, "Capacity of multi-antanna gaussian channel,", *Europ. Trans. Telecommun.*, pp. 585–595, Nov 1999. 1, 35

[2] G. J. Foschini, "Layerd space-time architecture for wireless communication in a fading environment when using multi-element antennas", *Bell labs. Tech. J*, vol. 1, no. 2, pp. 41–59, Jul 1996. 1, 35

[3] W. H. Mow, "Maximum likelihood sequence estimation from the lattice viewpoint", *IEEE Trans. Infor. Theory*, vol. 40, pp. 1591–1600, Sep 1994. 1, 8

[4] E. Viterbo and J. Boutros, "A universal lattice code decoder for fading channels", *IEEE Trans. Infor. Theory*, vol. 45, pp. 1369–1642, Jul 1999. 1, 8

[5] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices", *IEEE Trans. Infor. Theory*, vol. 48, pp. 2201–2214, Jul 2002. 1, 8

[6] A. K. Lenstra, H. W. Lenstra, and L. Lovász, "Factoring polynomials with rational coefficients", *Math. Ann*, vol. 261, pp. 515–534, 1982. 1, 15, 18, 25

[7] A. D. Murugan, H. E. Gamal, M. O. Damen, and G. Caire, "A unified framework for tree search decoding: redisovering the sequential decoder", *IEEE Trans. Infor. Theory*, vol. 52 no.3, pp. 933–953, 2006. 1, 12

[8] W. Yu and J. Cioffi, "Sum capacity of a gaussian vector broadcast channel", *in Proc. IEEE International Symposium on Information Theory*, p. 496, 2002. 2

[9] P. Viswanath and D. Tse, "Sum capacity of the vector gaussian broadcast channel and uplink-downlink duality", *IEEE Trans. on Information Theory*, pp. 1912–1921, Aug 2003. 2

[10] C. Windpassinger, R. F. H. Fischer, and J. B. Huber, "Lattice-reduction-aided broadcast precoding", *IEEE Trans. on Communications*, pp. 2057–2060, Dec 2004. 2, 38

[11] D. J. Young and N. C. Beaulieu, "The generation of correlated rayleigh random variates by inverse fourier transform", *IEEE Trans. Commun.*, vol. 48, pp. 1114–1127, July 2000. 3

[12] C. Loo and N. Second, "Computer models for fading channels with applications to digital transmission", *IEEE Trans. Veh. Technol.*, vol. 40, pp. 700–707, Nov 1991. 3

[13] P. Hoher, "A statistical discrete-time model for wssus multipath channel", *IEEE Trans. Veh. Technol.*, vol. 41, pp. 461–468, July 1992. 3

[14] R. H. Clarke, "A statistical theory of mobile-radio reception", *Bell Syst. Tech. J.*, vol. 47 no. 6, pp. 957–1000, Jul-Aug 1968. 3

[15] R. F. Pop and N. C. Beaulieu, "Limitations of sum of sinusoids fading channel simulators", *IEEE Trans. Commun.*, vol. 49 no. 4, pp. 699–407, Apr 2001. 3

[16] K. E. Baddour and N. C. Beaulieu, "Autoregressive modeling for fading channel simulations", *IEEE Trans. on Wireless Commun.*, vol. 4 no. 4, pp. 1650–1662, Jul 2005. 3, 5

[17] P. A. Bello, "Characterization of randomly time-variant linear channels", *IEEE Trans. on Commun. Sys.*, vol. CS. 11 no. 4, pp. 360–393, Dec 1963. 4

[18] W. C. Jakes, *Microwave mobile communications*, New York: Wiley, 1974. 4

[19] S. M. Kayt, *Modern Spectral Estimation*, Englewood Cliffs, NJ: Prentice-Hall, 1988. 4, 5

[20] M. O. Damen, A. Chkeif, and J. C. Belfiore, "Lattice codes decoder for space-time codes", *IEEE Commun. Lett.*, vol. 4 no. 5, pp. 161–163, May 2000. 8

[21] M. Pohst, "On the computation of lattice vectors of minimal length, successive minima and reduced basis with applications", *ACM SIGSAM*, vol. 15, pp. 37–44, 1981. 8

[22] C. P. Schnorr and M. Euchner, "Lattice basis reduction: Improved practical algorithms and solving subset sum problems", *Math. Programming*, vol. 66, pp. 181–191, 1994. 10, 15, 18

[23] L. Babai, "On lovász' lattice reduction and the nearest lattice point problem", *Combinatorica*, vol. 6 no. 1, pp. 1–13, 1986. 12

[24] J. Foschini, G. Golden, R. Valenzuela, and P. Woliansky, "Simplified preprocessing for high spectral efficiency wireless communication employing multi-element arrays", *IEEE J. Sel. Areas Commun.*, vol. 17 no. 11, pp. 1841–1852, Nov 1999. 13, 14

[25] J. M. Cioffi, G. P. Dudevoir, M. V. Eyuboglu, and G. D. Forney Jr., "MMSE decision-feedback equalizers and coding. I. equalization results", *IEEE Trans. Commun.*, vol. 43 no. 10, pp. 2582–2594, Oct 1995. 13

[26] H. Cohen, *A course in computational algebraic number theory*, Springer, 2000. 16, 18

[27] C. Ling and N. Howgrave-Graham, "Effective LLL reduction for lattice decoding", *In the Proc. ISIT2007, Nice, France*, pp. 196–200, Jun 2007. 18, 20

[28] C. Ling and N. Howgrave-Graham, "Low complexity data detection using seysen's lattice reduction algorithm", *In the Proc. ICASSP*, pp. III–53–III–56, 2007. 21

[29] M. Seysen, "Simultaneous reduction of a lattice basis and its reciprocal basis", *Combinatorica*, vol. 13, pp. 363–376, 1993. 22

[30] M. O. Damen, H. El. Gamal, and G. Caire, "On maximum likelihood detection and the search for the closest lattice point", *IEEE Trans. on Information Theory*, vol. 49, No 2, pp. 2389–2402, Oct 2003. 28

[31] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis", *Math. Computation*, vol. 44, pp. 463–471, Apr 1985. 28

[32] G. Caire and S. Shamai, "On the achievable throughput of a multi-antanna gaussian broadcast channel", *IEEE Trans. Inf. Theory*, vol. 43, pp. 1691–1706, Jul 2003. 35

[33] R. Zamir, S. Shamai, and U. Erez, "Nested linear/lattice codes for structured multiterminal binning", *IEEE Trans. Inf. Theory*, vol. 48, pp. 1250–1276, Jun 2002. 35

[34] C. B. Peel, B. M. Hochwald, and A. L. Swindlehurst, "A vector-perturbation technique for near capacity multiantanna multiuser communication-part ii", *IEEE Trans. Inf. Theory*, pp. 537–544, Mar 2005. 35, 37, 38

[35] R. F. H. Fischer, *Precoding and signal shaping for digital transmission*, New York: Wiley, 2002. 37

[36] H. Yao and G. W. Wornell, "Lattice-reduction-aided detectors for MIMO communication systems", *Proc. IEEE Globecom, Taipei, Taiwan*, pp. 424–428, Nov 2002. 38

[37] C. Windpassinger and R. F. H. Fischer, "Low-complexity near-maximum-liklihood detection and precoding for MIMO system using lattice reduction", *in Proc. IEEE information theory workshop, Paris, France*, pp. 345–348, Mar 2003. 38

[38] R. F. H. Fischer, C. Windpassinger, A. Lampe, and J. B. Huber, "MIMO precoding for decentralized receivers", *in Proc. IEEE Int. Symp. Information Theory, Lousanne, Switzerland*, p. 496, Jun 2002. 38

[39] G. W. Stewart, *Matrix algorithms*, vol. 1: Basic Decompositions, SIAM, 1998. 44