ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Communication Technology Laboratory
Wireless Communication Group
Prof. Dr.-Ing. A. Wittneben

Impact of Imperfections on Cooperative Communication in Future Cellular Networks

Petra Tanaskovic
Department of Information Technology and Electrical Engineering

Supervisor: MSc. Raphael Rolny
Professors: Prof. Dr. Armin Wittneben, Prof. Dr. Irini Reljin

March 7, 2012

Abstract

Interference presents a significant challenge in the cooperative cellular networks. Zero forcing is the technique that is most often used for interference cancelation in practice. This technique requires complete knowledge of the involved channel matrices and each base station disseminates some of the information to the other base stations. Therefore, the channel matrix has to be estimated first and then quantized in order to allow the computation of the precoding matrices. In this thesis different channel matrix quantization techniques are studied and their influence on the overall system performance is studied. Scalar quantizers based on Cartesian and Polar representations of the complex numbers are considered. In addition, vector quantization techniques based on K-means and LBG algorithms are studied. Matlab simulations that are performed show that the Polar quantization is superior to the Cartesian quantization. In addition, vector quantization is shown to be superior to the scalar quantization. Based on Matlab simulations, optimal parameters for tuning different quantizers are derived.

Preface

This Master Thesis is part of the graduate study at the Department of Telecommunication of Electrical Engineering at the Belgrade University, Electrical Engineering Faculty (ETF) Belgrade.

The author certifies that this Master Thesis, and the research to which it refers, are the product of the author's own work, and that any ideas or quotations from the work of other people, published or otherwise, are fully acknowledged in accordance with the standard referring practices of the discipline.

Petra Tanaskovic

Acknowledgements

I would like to thank the Wireless Communications Group of the Communication Technology Laboratory at ETH Zurich for the opportunity to complete my Master Thesis on such an exciting topic. Special thanks go to Raphael Rolny who supported this work with his good inputs and with inspiring discussions. He was always present with help and good ideas. I would also like to thank Marc Kuhn for his helpful comments and professors Irini Reljin and Armin Wittneben.

Author:	Petra Tanaskovic	vucicp@ee.ethz.ch
Advisor:	Raphael Rolny	rolny@nari.ee.ethz.ch
Professor:	Irini Reljin	irinitms@gmail.com
Professor:	Prof. Dr. Armin Wittneben	wittneben@nari.ee.ethz.ch

Contents

List of Figures

Chapter 1

Introduction

1.1 Motivation

Wireless telecommunication has seen a fast development during the last decades. Mobile phone networks are being improved and refined rapidly. In this context, cellular networks have received a lot of research attention. In cellular networks several base stations send information to the users that are in a certain radius from them. A typical problem that is encountered in cellular networks is interference. It is especially strong for the users that are located at cell borders. The most popular way to limit interference in current network is to divide resources between neighboring cells. Thereby, the communication in adjacent cells use e.g. orthogonal frequency bands (frequency division multiple access – FDMA), time slots (time division multiple access – TDMA), or codes (code division multiple access – CDMA).

A newly emerged concept to overcome the impairments of interference is through cooperation of base station within a network. This concept gives rise to cooperative cellular networks. In cooperative cellular networks, base stations are communicating with each other and share certain information that is important for the functioning of the cellular network.

This concept can e.g. be used for interference cancellation by using the zero forcing technique. In this technique each base station calculates a precoding matrix that is used for sending signals that guaranty interference cancellation. In order for the zero forcing technique to be used, base stations need to have accurate channel state information. The channel matrix has to be estimated [8] and the estimated values have to be quantized so that it can be shared among other base stations and used for the calculation of the precoding matrices. However, the process of channel matrix estimation and quantization introduces errors in the calculation of precoding matrices.

The main focus of this thesis is the analysis of the channel matrix quantization on the network performance. We assume the error due to quantization is bigger than the error due to channel matrix estimation and therefore we neglect the estimation error and only focus on the quantization. We analyze scalar and vector quantizers. In scalar quantization each element of the channel matrix is separately quantized in order to quantize the whole matrix. We analyze both Cartesian and Polar quantization. In Cartesian quantization, a Cartesian representation of the complex number is quantized, while the Polar representation is used in Polar

quantization. The influence of different design variables for all quantization methods are studied and some tuning methods are proposed. Several ways of quantization are compared with respect to their performance in the zero forcing algorithm through Matlab simulations. In addition, we consider vector quantization which can be used for quantization of the whole channel matrix instead of its elements. We study two vector quantizers, namely K-means and LBG quantizers which are very similar, but differ in their training phase. The two algorithms are compared with respect to the computational complexity of their training phases as well as with respect to their performance in the zero forcing algorithm. Different aspects of the training processes are analyzed based on Matlb simulations. Results under vector quantization are compared to those of the scalar quantizers. It is shown that the K-means and LBG quantization of channel matrix elements lead to better performance than scalar quantization. However, for the quantization of the whole channel matrix the training phases of the K-means and LBG algorithms become extremely computationally expensive and can not be practically applied.

1.2 Outline

- Chapter 2: Describes the multiuser communication system and gives important definitions that are used throughout the thesis. These are the definitions of achievable rate and capacity. In addition, a detailed analysis of the transmitted and the received communication signal is given. Error probability as well as the multiplexing gain and diversity gain are defined.

- Chapter 3: Describes cooperative cellular networks and the problem of interference in these networks. The zero forcing algorithm for interference cancellation is explained in detail.

- Chapter 4: Gives the theoretical background of quantization techniques. Quantization error is defined and methods for its reduction are analyzed. A classification of different quantization techniques is given.

- Chapter 5: Deals with scalar quantization. Quantization of complex number representations in Cartesian and Polar coordinates are considered. Various Matlab simulations are performed for these two quantizations and based on them optimal tuning of the quantizers is proposed and the two quantization techniques are mutually compared.

- Chapter 6: Deals with vector quantization. Two algorithms for vector quantization training are considered. These are the K-means and LBG algorithms. Various Matlab simulations are done for both algorithms and optimal selection of constants that can be adjusted is proposed. In addition, the two methods are compared with the scalar quantizers described in the previous chapter.

- Chapter 7: Summarizes the conclusions of the thesis and gives some suggestions for future work.

1.3 Terminology and Notation

In this section we give the notation that will be used throughout the thesis. *Sets* are marked by capital italic letters, e.g., $\mathcal{A} = \{0, 1, 2, \ldots, 9\}$.

A *vector* of dimension $m \in \mathbb{R}$, where $\mathbb{R}$ is the set of all real numbers, is denoted by a bold lowercase variable, e.g., $\mathbf{h} = \begin{bmatrix} h_1 \, h_2 \, \dots \, h_m \end{bmatrix}^T$. Elements of a vector are denoted by Roman lowercase characters followed by a subscript index, e.g. h_i.

A *matrix* with $m \in \mathbb{R}$ rows and $n \in \mathbb{R}$ columns is denoted by a bold capital letter, elements of a matrix are denoted by Roman characters with indices for both the corresponding row and column, respectively.

The $m \times m$ *identity*–matrix is denoted by $\mathbf{I}_m$ and the $m \times n$ *all-zero*-matrix with all entries equal to zero by $\mathbf{0}_{m,n}$.

Many quantities used in this work are random variables. In most cases, the random variables are drawn from a circularly symmetric complex Gaussian distribution. This distribution is denoted as $\sim \mathcal{CN}(\mu, \sigma^2)$, where μ is the mean and σ^2 is the variance.

Chapter 2

Multiuser Communication Systems

Multiuser communication systems are communication systems that consist of multiple transmitters and receivers [9]. These systems have been extensivly developed and researched during the last couple of decades. In this thesis we are concerned with wireless multiuser communication systems. In general each transmitter in the system can send information to each receiver. We are looking at the configuration in which there are M transmitters and K receivers and where some of the M transmitters communicate with some of the K receivers. Figure 2.1 illustrates such a multiuser communication system. Each transmitter $T_j, j \in \{1, ..., M\}$ sends a signal $\mathbf{x}_j(t)$. In the equivalent baseband representation, these T_x signals can be written as:

$$\mathbf{x}_j = \sum_{i=1}^{M_j} \mathbf{G}_i^{(j)} \mathbf{s}_i^{(j)}, \tag{2.1}$$

where the time index t is dropped for the sake of simplicity of notation. M_j stands for the number of signals that are sent at the same time to different receivers, i and j denotes i^{th} receiver and j^{th} transmitter. Each message $s_i^{(j)} \in \mathbb{C}^{N_s}$ that is transmitted to a receiver i is given by a vector of complex symbols. The matrix $\mathbf{G}_i^{(j)} \in \mathbf{C}^{N_t \times N_s}$ is a linear precoding matrix for j^{th} transmitter and for i^{th} receiver. We will further consider the precoding matrix in the chapters that follow. The signal that arrives at the i^{th} receiver, denoted by $\mathbf{U}_i$, can be expressed as:

$$\mathbf{y}_i = \mathbf{H}_i \sum_{i=1}^{M} \mathbf{x}_j + \mathbf{w}_i, \tag{2.2}$$

where $\sum_{i=1}^{M} \mathbf{x}_j$ is the sum of all transmitted signals which can be divided into desired signal and interference

$$\mathbf{y}_i = \mathbf{y}_i^{(\text{des})} + \mathbf{y}_i^{(\text{int})} + \mathbf{w}_i. \tag{2.3}$$

This topic will also be treated more deeply in the following chapters. Noise that is superimposed to the signal received by $\mathbf{U}_i$ is denoted by $\mathbf{w}_i$.

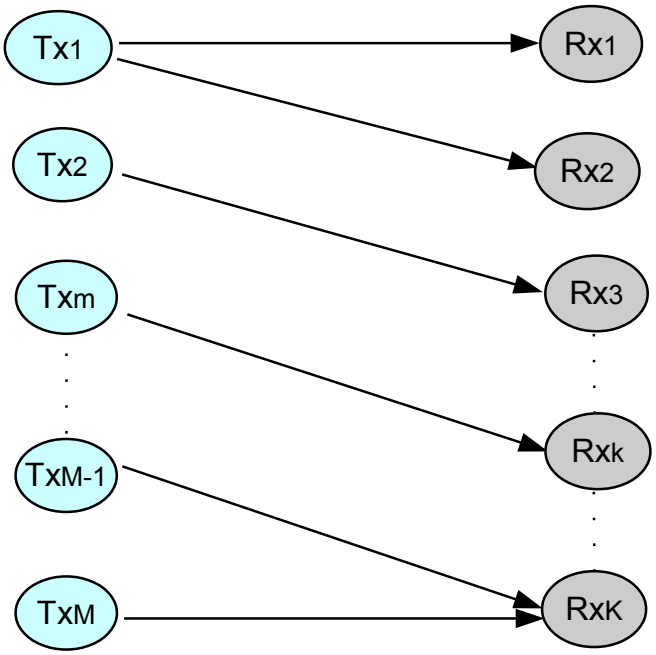

Figure 2.1: *Multiuser communication system model.*

2.1 Cellular Networks

In this thesis we focus our atention on cellular networks, which are a special kind of a multiuser communication systems. A Cellular network is divided into geographically separated cells. Each cell consists of a base station and a number of users. Figure 2.2 illustrates this. Each base station is sending signals to users that are not further away from it than a certain distance. If the user is close to the base station, the signal it receives will be stronger. As the distance from the base station gets larger, the received signal strength decreases [10]. The main problem that may occur in a cellular network is the interference of the signals sent by different base stations. Namely, in addition to the desired signal that the user receives, it may receive undesired signals from other base stations located in the neighbouring cells, and also signals from the own base station but intended for other users [4]. This effect is especially strong when the user is located close to the border of a cell.

A way to overcome this problem is the cooperation of base stations. This leads to cooperative cellular networks. In such a network a certain group of base stations is connected by a backhaul link and they have some information of the other base stations in the same cooperative group. In the narrow sense, channel

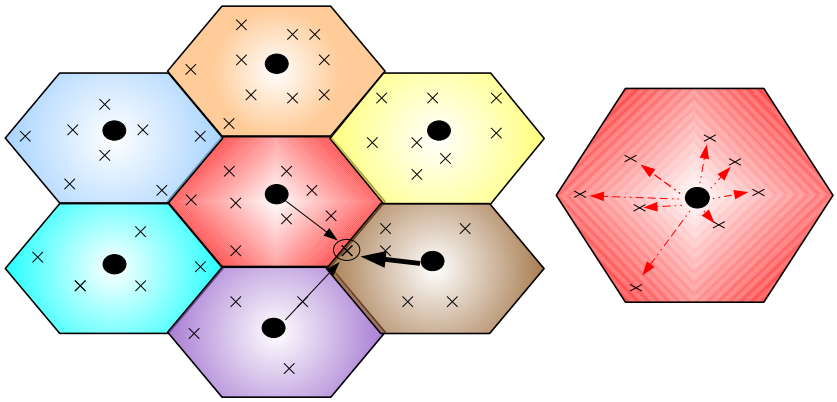

Figure 2.2: *Illustration of a cellular network (left) and a single cell (right).*

state information (CSI) and transmit symbols are exchanged among the base stations. The base station then jointly transmit signals to all mobiles they serve. This is also called coordinated multipoint transmission (CoMP) [4]. In a more broader sense, other information can be exchanged between the base stations. This can be information on scheduling, signal strength or other information. This information can be used to alarm the network. In addition, base stations can share information on how much data they are processing, which can be used to divide processing jobs more fairly between the base stations. Figure 2.3 illustrates an example of a cooperative cellular network that consists of one cooperative group of base stations. In the following chapter, we give details on different aspects of cellular networks and especially cooperative cellular networks.

2.1.1 Signal Model

We are considering a cooperative cellular network with multiuser communication. In Figure 2.3 each base station inside a cooperative group jointly sends signals to all mobiles they currently serve. The interference which is caused by other groups of base stations (which are not cooperating with the first group) is still present. However, this influence is not so significant due to the distance dependent path loss and is in the following considered as noise. The signal that is transmitted from the transmitter $T_j, (j = 1, ..., M)$ is represented as:

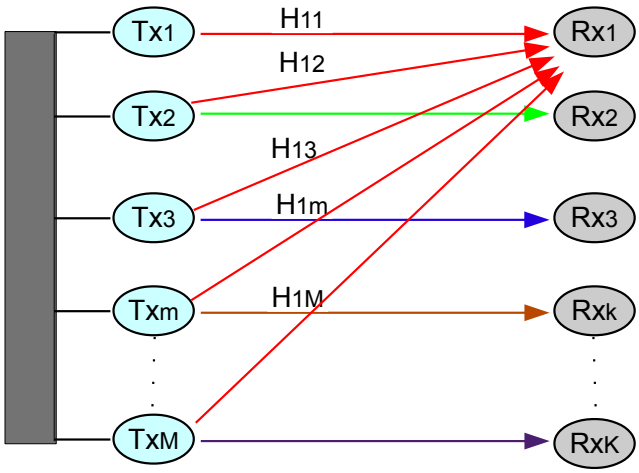

Figure 2.3: *Cooperative cellular network.*

$$\mathbf{x}_j = \sum_{i=1}^{K} \mathbf{G}_i^{(j)} \mathbf{s}_i, \tag{2.4}$$

where $\mathbf{s}_i$ is a message from the network for mobile user $\mathbf{U}_i$ and it is represented as a vector of complex numbers. This vector has dimension N_s, where $1 \leq N_s \leq MN_t$, and N_t is the number of antennas at each of the M transmitters. This vector of complex numbers that arrives from the network to the transmitter has to be coded first. In this process, the signals are optimized for the transmission and therefore we apply factors $\mathbf{G}_i^{(j)} \in \mathbb{C}^{N_t \times N_s}$.

For the system as shown in Figure 2.3, the wireless channel $h_{p,q}(\tau, t)$ can be modeled as a complex random process. We may assume that the channel coefficients are constant: $h_{p,q}(\tau, t) = h_{p,q} = \text{const.}, \forall p, q$ where $h_{p,q}$ is the coefficient of the p^{th} receiver antenna from the q^{th} transmit antenna. Such a channel model corresponds to a slow and frequency flat fading channel [1]. The channel matrix to the receiver $\mathbf{U}_i$ is denoted by $\mathbf{H}_i$ and the members of this matrix are also matrices $[\mathbf{H}_{i1}, \ldots \mathbf{H}_{iM}]$. The channel coefficient of the matrix $\mathbf{H}_{ij}$ can be represented in the following form:

$$\mathbf{H}_{ij} = \begin{bmatrix} h_{11} & \ldots & h_{1Nt} \\ \vdots & \vdots & \vdots \\ h_{Nr1} & \ldots & h_{NrNt} \end{bmatrix}, \tag{2.5}$$

where N_r and N_t are the numbers of antennas at the receiver and the transmitter, respectively. Through such a channel signals of different power are sent and therefore power optimization has to be done by multiplying

with the corresponding precoding matrix $\mathbf{G}_i^{(j)}$. The signal at each of the receivers *i* can be represented as:

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

In (2.6) the contributions of the desired signal and the interference can be separated. We will give an extended form of (2.6) in chapter 3 .

2.2 Capacity of MIMO

The main performance measure in this work is the achievable rate. In this section we define this notation and show how it can be calculated. The discussion that follows is mainly based on the textbooks [1] and [2].

2.2.1 Coding and Decoding

Each transmitter T_m from the set $\mathcal{T} = \{T_1, ..., T_M\}$ sends a message $S_i^{(m)}$ from the set of messages $\mathcal{S}_m$ to receiver U_i from the set $\mathcal{U} = \{U_1, ..., U_K\}$. The set $\mathcal{S}_m$ contains $|\mathcal{S}_m|$ different messages that are all assumed to be equally probable. Looking at the transmitter–receiver pair (T_m, U_i), the transmitter maps the messages for the receiver into a codeword $\mathbf{x}_m$ by using an *encoder*. Before it is transmitted, the signal can be regarded as a sequence of symbols. The receiver U_k receives the message by first receiving the signal $\mathbf{y}_m$ from which the sequence $\hat{S}_i^{(m)}$ is then estimated and *decoded*. Figure2.4 illustrates this concept.

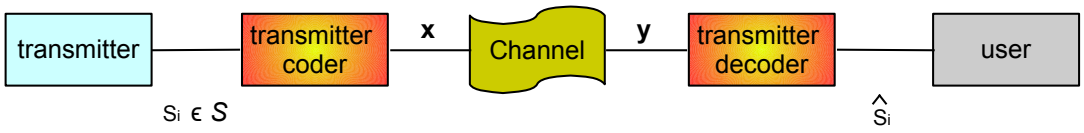

Figure 2.4: *Coding and decoding system*

A code $(\mathcal{S}_1^{(n)}, \ldots, \mathcal{S}_{n_\mathcal{S}}^{(n)}, n)$ consists of a set of $n_\mathcal{S}$ encoding functions $\{f_1^{(n)}, \ldots, f_{n_\mathcal{S}}^{(n)}, \}$ and a set of $n_\mathcal{S}$ deterministic rules $\{\Phi_1^{(n)}, \ldots, \Phi_{n_\mathcal{S}}^{(n)}\}$ called decoding functions. The encoding functions map each message to a sequence of symbols:

$$\begin{aligned} f_k^{(n)} : \mathcal{S}_m^{(n)} \to \mathcal{C}_m^{(n)}, \qquad & k \in \{1, \ldots, n_\mathcal{S}\}, \\ S_i^{(m)} \mapsto \mathbf{x}_m, \qquad & S_i^{(m)} \in \mathcal{S}_m^{(n)}, \end{aligned}$$

where $\mathcal{C}_m^{(n)} \subseteq \mathbb{C}^n$. The set $\mathcal{C}_m^{(n)}$ is also called a codebook. Decoding functions assign a guess to each receiver vector:

$$\Phi_m^{(n)} : \mathbb{C}^n \to \mathcal{S}_m^{(n)}, \qquad \mathbf{y}_m \mapsto \hat{S}_m \,, \qquad \forall m \in \{1, \ldots, n_{\mathcal{S}}\}.$$

The number n is called the block-lenght of the code $(\mathcal{S}_1^{(n)}, \ldots, \mathcal{S}_{n_{\mathcal{S}}}^{(n)}, n)$. Reliability of the link of a communication system is characterized by the probability that the decoder will make a wrong guess based on the signal it receives.

Achievable Rate

The error probabiliy $\Pr(\mathsf{e})$ is a key performance measure and is defined as the probability that the decoding function makes a wrong decision. It is given as:

$$\begin{aligned}\Pr(\mathsf{e}) &= \Pr\left[\hat{S}_1 \neq S_1 \cup \ldots \cup \hat{S}_{n_{\mathcal{S}}} \neq S_{n_{\mathcal{S}}}\right] \\ &= 1 - \Pr\left[\hat{S}_1 = S_1 \,,\, \ldots \,,\, \hat{S}_{n_{\mathcal{S}}} = S_{n_{\mathcal{S}}}\right].\end{aligned}$$

The rate r is called achievable if $\forall \varepsilon > 0 \; \exists \, n_0$, such that

$$\forall n \geq n_0 \quad \exists \text{ a } \left(\mathcal{S}^{(n)}, n\right) \text{ code with } \Pr(\mathsf{e}) \leq \varepsilon, \text{ with } \left|\mathcal{S}^{(n)}\right| = \lceil 2^{n\,r} \rceil.$$

Capacity of AWGN Channel

Capacity is defined as the maximum rate at which reliable communication is possible. Information is said to be communicated reliable if for every $\varepsilon > 0$ there exists a code of block length *n* so that $\Pr(\mathsf{e}) < \varepsilon$. Let us consider first a single-input-single-output (SISO) system with a single antenna on the transmitter and the receiver. We further assume that there is no interference in the channel, but we have an additive white Gaussian noise represented as:

$$w \sim \mathcal{CN}(0, \sigma_w^2), \tag{2.7}$$

where σ_w^2 is the noise variance. Therefore, we are considering a scenario of point–to–point communication with the additive white Gaussian noise channel (AWGN). We can define the *rate* for an $(\mathcal{S}^{(n)}, n)$ code as

$$r = \frac{\log_2\left(\left|\mathcal{S}^{(n)}\right|\right)}{n} \quad [\text{bit per channel use}].$$

The capacity C is then defined as the supremum over all achievable rates. In the AWGN channel, the capacity can be achieved by using a Gaussian codebook $\mathcal{C}^{(n)}$. In the Gaussian codebook all codewords are represented as $\mathcal{CN}(0, \sigma_x^2)$, where the value σ_x^2 is the transmit power. The transmit power is constraint to be:

$$\sigma_x^2 = \frac{1}{n}\mathbb{E}\left\{\sum_{t=1}^{n} |\mathbf{x}[t]|^2\right\} \leq P_s,$$

where Ps is the available power.

The capacity of the AWGN channel with power constraint P_s and noise variance σ_w^2 is given by:

$$C = \log_2 \left(1 + \frac{P_s}{\sigma_w^2}\right). \tag{2.8}$$

The proof for this theorem can be found in [2]. The term P_s/σ_w^2 is called signal to noise ratio (SNR).

2.2.2 Capacity of MIMO Channel

In the case of multiple–input multiple–output (MIMO) systems, both the receiver and the transmitter have multiple antennas. It has been shown in [15] that the capacity of a MIMO channel with N_t transmit and N_r receive antennas (Figure: 2.5) is given by:

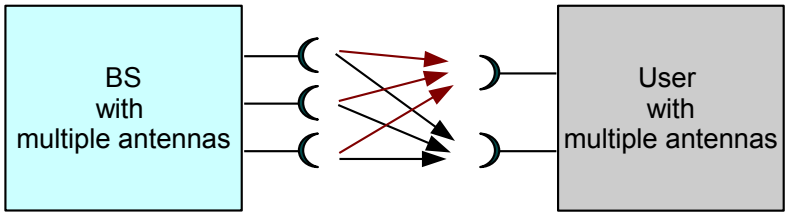

Figure 2.5: *Single-user with multiple antennas.*

$$C = \max_{\mathbf{Q_s}} \log_2 \det(\mathbf{I}\sigma_w^2 + \mathbf{H}\mathbf{Q_s}\mathbf{H}^H) - \log_2 \det(\mathbf{I}\sigma_w^2), \tag{2.9}$$

where $\mathbf{Q}_s = \mathbb{E}\{\mathbf{x}\mathbf{x}^H\}$ is the covariance matrix of the transmit signal, which is constrained to be Trace$(\mathbf{Q_s}) \leq P_s$. The transmitter and the receiver are communicating over a communication channel described by the matrix $\mathbf{H} \in \mathbb{C}^{N_r \times N_t}$. In the case of non–cooperative source (a source in which antennas are mutually independent), independent data streams of the power $\frac{1}{N_t}P_s$ are transmitted over different antennas, i.e $\mathbb{E}\{\mathbf{s}\mathbf{s}^H\} = \frac{1}{N_t}P_s \cdot \mathbf{I}_{N_t}$. This choice is optimal in the absence of channel information at the T_x. Therefore, the following rate is achievable for the considered MIMO system:

$$r = \log_2 \det\left(\mathbf{I}\sigma_w^2 + P_s \cdot \mathbf{H}\mathbf{H}^H\right) - \log_2 \det\left(\mathbf{I}\sigma_w^2\right). \tag{2.10}$$

2.2.3 Diversity and Spatial Multiplexing Gain

Definitions Of Diversity and Multiplexing Gain

Consider a family of codes C(SNR) of block length n, one at each SNR level. Let R(SNR) be the rate of the code C(SNR) and $\mathrm{P_0(SNR)} = \Pr(\log_2(1 + |h|^2\mathrm{SNR}) < \mathrm{R}_{target})$ [1] the outage probability of the code C(SNR). R_{target} is a specific rate which is desired in the system. If the channel described by coefficient h does not support the rate R_{target}, the corresponding transmission is lost. This event is called outage. If the following condition is valid:

$$\lim_{\mathrm{SNR}\to\infty} \frac{P_0\,(\mathrm{SNR})}{\log\,(\mathrm{SNR})} = -d, \tag{2.11}$$

then C(SNR) is said to achieve diversity gain d [12].

If the following condition is valid:

$$\lim_{\mathrm{SNR}\to\infty} \frac{R\,(\mathrm{SNR})}{\log\,(\mathrm{SNR})} = r, \tag{2.12}$$

then C(SNR) is said to achieve multiplexing gain r [12].

The following table shows the advantages and disadvantages of diversity and multiplexing [12]:

DIVERSITY	MULTIPLEXING
BENEFIT LOWER ERROR/OUTAGE PROBABILITY	*BENEFIT* HIGHER RATE
REQUIRE SEND SAME INFORMATION	*REQUIRE* SEND INDEPENDENT INFORMATION
CONFLICT: CAN NOT HAVE BOTH **MUST BE A TRADE OFF**	

Table 2.1: *Benefits and disadvantages of diversity and multiplexing.*

For each multiplexing gain r, $d^*(r)$ is defined as the supremum of the diversity advantage over all

schemes. Assume that condition $n \geq N_t + N_r - 1$ is fulfilled, then $d^*(r)$ can be given by the curve $(r, d^*(r))$, where $r = (0, 1, ..., \min(Nt, Nr))$, and the following equation is valid:

$$d^*(r) = (N_t - r)(N_r - r). \tag{2.13}$$

This connection (2.13) between diversity and multiplexing gain is given in the following graph:

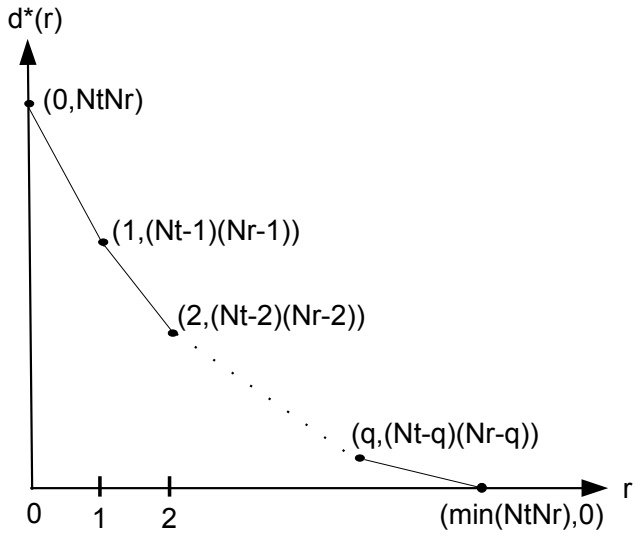

Figure 2.6: *Diversity–multiplexing trade off.*

2.2.4 Concrete Examples of Diversity and Multiplexing Gain

We consider a system made of one transmitter and one receiver with two antennas each.

Spatial Multiplexing Gain

Let h_{ij} be the Rayleigh distribution channel gain from transmit antenna to the receiver antenna. We further assume that the transmit and receive antenna are distant such that the fading gains h_{ij} can be assumed to be independent. So we have four independent paths for a signal that is transmitted from the transmitter to the receiver.

The spatial multiplexing gain on the concrete example can be noticed as the slope of the rate versus SNR curve at high SNR values. Systems with multiple transmit and receive antennas can achieve larger slopes due to spatial multiplexing of multiple transmit signals. In order to illustrate this concept, the transmitter sends two different symbols s_1 and s_2. In this case, the spatial multiplexing gain is two. Figure 2.7 illustrates this situation.

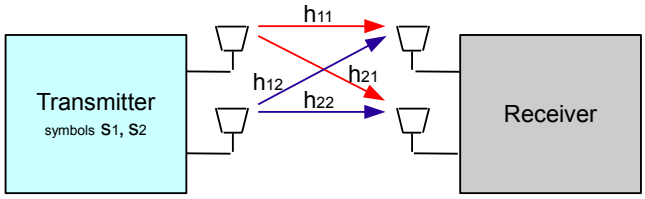

Figure 2.7: *Spatial multiplexing gain*

In the following, we derive the multiplexing gain for this example. The signal at the receiver is given by:

$$\mathbf{y} = \mathbf{HGs} + \mathbf{w}. \tag{2.14}$$

we decompose this signal by the singular value decomposition (SVD) [1]. This means that the received signal can be described as:

$$\begin{aligned} \mathbf{y} &= \mathbf{HGs} + \mathbf{w} && (2.15)\\ &= \mathbf{U\Lambda V}^H\mathbf{Gs} + \mathbf{w}, && (2.16)\\ & && (2.17) \end{aligned}$$

where $\mathbf{U}$, and $\mathbf{V}$ are unitary matrices, i.e. $\mathbf{UU}^H = \mathbf{U}^H\mathbf{U} = \mathbf{I}$ and $\mathbf{VV}^H = \mathbf{V}^H\mathbf{V} = \mathbf{I}$. Using the precoding matrix $\mathbf{G} = \mathbf{V}$ and multiplying the receive signal with $\mathbf{U}^H$, the receive signal can be written as:

$$\begin{aligned} \tilde{\mathbf{y}} &= \mathbf{U}^H\mathbf{y} && (2.18)\\ &= \mathbf{U}^H\mathbf{U\Lambda V}^H\mathbf{Vs} + \mathbf{w} && (2.19)\\ &= \mathbf{\Lambda s} + \mathbf{w} && (2.20)\\ &= \begin{bmatrix} \Lambda_{11} & 0 \\ 0 & \Lambda_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}. && (2.21) \end{aligned}$$

With this decomposition, we have turned the MIMO system into two parallel, non–interfering single antenna systems. Therefore, the achievable can be given as the sum of the rates of two SISO links as:

$$R(\text{SNR}) = \log_2(1 + |\Lambda_{11}|^2\text{SNR}) + \log_2(1 + |\Lambda_{22}|^2\text{SNR}). \tag{2.22}$$

Now we can evaluate the spatial multiplexing gain (2.12) and obtain:

$$\lim_{\text{SNR}\to\infty} \frac{R\,(\text{SNR})}{\log\,(\text{SNR})} = \frac{2\log(\text{SNR})}{\log\,(\text{SNR})} = 2. \tag{2.23}$$

Therefore, we can conclude that in the example of a MIMO system with two antennas at transmitter and receiver, the spatial multiplexing gain is two.

Diversity Gain

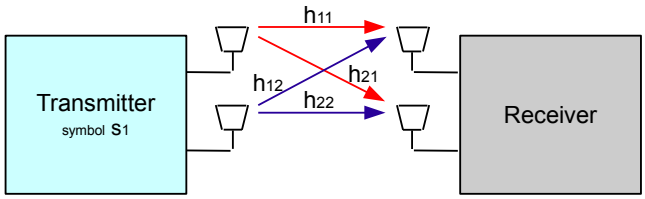

Figure 2.8: *Illustration for the diversity approach.*

We consider the same situation as the example in figure 2.7, but in this situation the transmitter transmits one symbol s_1. In this case maximal diversity gain is four. If the transmitted symbol is s_1 then the signal at the receiver has the form:

$$y_1 = h_{11}s_1 + h_{12}s_1 + w_1, \tag{2.24}$$

$$y_2 = h_{21}s_1 + h_{22}s_1 + w_2, \tag{2.25}$$

Effective channel with gain $\sum_{i=1}^{2}\sum_{j=1}^{2}(h_{ij})^{(2)}$ is created yielding a 4-fold diversity gain. If the links h_{11}, h_{12} and h_{21} fail, s_1 will be transmitted over the link h_{22}. The probability of symbol s_1 at the receiver is big, however only one symbol is send, so the receive probability is high, but the rate is moderate.

It can be proven that the diversity gain is four in this case. In the previous case for the prove of spatial multiplexing gain we used SVD, here we also use SVD and we can write the receive signal in the following form:

$$\tilde{\mathbf{y}} = \mathbf{U}^H\mathbf{y} = \mathbf{U}^H\mathbf{U}\mathbf{\Lambda}\mathbf{V}^H\mathbf{V}\mathbf{s} \tag{2.26}$$

$$= \mathbf{\Lambda}\mathbf{s} \tag{2.27}$$

$$= \begin{bmatrix} \Lambda_{11} & 0 \\ 0 & \Lambda_{22} \end{bmatrix} \begin{bmatrix} s_1 \\ s_1 \end{bmatrix} + \begin{bmatrix} w_1 \\ w_1 \end{bmatrix} \tag{2.28}$$

Now (2.11) is valid for this system only if the following condition for outage probability is given by $P_0 \approx SNR^{-N}$, where N is the total number of antennas and in this system $N = 4$. Thus, the diversity gain $d = 4$ can be achieved [13].

2.2.5 Achievable Rate for MIMO with Interference

If we are considering a single user MIMO system with N_t transmit and N_r receive antennas and only the noise is present, the formula for the rate is given by (2.10). But if we are including interference which is the consequence of other signals arriving to the $\mathbf{U}_k$ receiver, the rate given by (2.10) is extended by a new term $\mathbf{K}_\mathsf{I}$ which is the interference covariance matrix. Here, we are looking at one receiver $\mathbf{U}_k$ and one transmitter $\mathbf{T}_i$ and the signal at the receiver has the form:

$$\mathbf{y}_k = \mathbf{H}_k\mathbf{G}_k\mathbf{s}_k + \mathbf{H}_k\sum_{i\neq k}\mathbf{G}_i^{(j)}\mathbf{s}_i + \mathbf{w}_k, \tag{2.29}$$

where $\mathbf{y}^{(int)} = \mathbf{H}_k\sum_{i\neq k}\mathbf{G}_i^{(j)}\mathbf{s}_i$ is the interference.

In a multiuser situation, more than one interference source is present. The covariance of interference for a receiver $\mathbf{U}_k$ can be described by the following formula:

$$\mathbf{K}_\mathsf{I} = \mathbb{E}\left[\mathbf{x}_i\mathbf{x}_j^H\right] = \sum_{i\neq k}\sum_{j\neq k}\mathbf{H}_{ki}\mathbb{E}\left[\mathbf{s}_i\mathbf{s}_j^H\right]\mathbf{H}_{kj}^H = \sum_{i\neq k}\mathbf{H}_{ki}P_s\mathbf{I}_{nr}\mathbf{H}_{kj}^H, \tag{2.30}$$

where we assume:

$$E\left[\mathbf{s}_i\mathbf{s}_j^H\right] = \begin{cases}\mathbf{O}, & \text{if} \quad i \neq j \\ P_s\mathbf{I}, & \text{if} \quad i = j\end{cases} \tag{2.31}$$

So achievable *rate* is than given by the following equation:

$$r = \log_2\det\left(\mathbf{K}_\mathsf{W} + \mathbf{K}_\mathsf{I} + \mathbf{K}_\mathsf{S}\right) - \log_2\det\left(\mathbf{K}_\mathsf{W} + \mathbf{K}_\mathsf{I}\right). \tag{2.32}$$

2.2.6 Uplink/Downlink Superposition Coding Rate Regions

In case when every user treats the interference from the other users as noise, a successive interference cancellation (SIC) receiver is needed to achieve capacity [1].

Uplink Case

In this case we are looking at more than one transmitter and receiver as depicted in Figure 2.9.

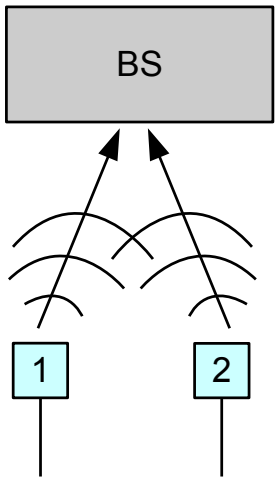

Figure 2.9: *Uplink case.*

.

The baseband discrete-time model for uplink AWGN channel with two users is given by:

$$y\left[m\right] = h_1 x_1\left[m\right] + h_2 x_2\left[m\right] + w\left[m\right], \tag{2.33}$$

where $w[m] \sim \mathcal{CN}(0, \sigma_w^2)$ is i.i.d complex Gaussian noise. User k has an average power constraint of P_k [J/symbol]. In order for the communication in the scenario of Figure 2.9 to be possible, the following set of inequalities has to be satisfied:

$$R_1 < \log\left(1 + \frac{|h_1|^2 P_1}{\sigma_w^2}\right), \tag{2.34}$$

$$R_2 < \log\left(1 + \frac{|h_2|^2 P_2}{\sigma_w^2}\right), \tag{2.35}$$

$$R_1 + R_2 < \log\left(1 + \frac{|h_1|^2 P_1 + |h_2|^2 P_2}{\sigma_w^2}\right). \tag{2.36}$$

User one can therefore achieve its single-user bound while at the same time can have a non-zero rate:

$$R_2 < \log\left(1+\frac{|h_1|^2P_1+|h_2|^2P_2}{\sigma_w^2}\right)-\log\left(1+\frac{|h_1|^2P_1}{\sigma_w^2}\right)=\log\left(1+\frac{|h_2|^2P_2}{|h_1|^2P_1+\sigma_w^2}\right). \tag{2.37}$$

The user information is decoded in two stages. In the first stage the data of the user two is decoded, while the signal from the user one is treated as Gaussian interference. The rate of the user two is denoted as R_2. Once the data of the user two is decoded, its signal can be reconstructed and subtracted from the aggregated receive signal. The data of the user one can then be easily decoded. The maximal rate at which user one can transmit is its single user bound $\log\left(1+\frac{|h_1|^2P_1}{\sigma_w^2}\right)$. The described receive strategy is characteristic for the successive cancellation decoder (SIC).

In general there are K users and the $K-user$ capacity region is described by 2^K-1 constraints, where each constraint is defined for each possible non empty subset S of users:

$$\sum_{k\in S} R_k < \log\left(1+\frac{\sum_{k\in S}|h_k|^2P_k}{\sigma_w^2}\right) \qquad \text{for all } S\subset(1,...,K). \tag{2.38}$$

The sum capacity in the general case is given by :

$$C_{sum}=\log\left(1+\frac{\sum_{k=1}^{K}|h_k|^2P_k}{\sigma_w^2}\right) \qquad \text{[bits per secound per Hertz]}. \tag{2.39}$$

In the case of equally received powers ($P_1=...=P_K=P$) the equations are simplified and the sum capacity is then:

$$C_{sum}=\log\left(1+\frac{K|h_k|^2P_k}{\sigma_w^2}\right). \tag{2.40}$$

In this case the symmetric capacity is given by:

$$C_{sym}=\frac{1}{K}\log\left(1+\frac{K|h_k|^2P_k}{\sigma_w^2}\right). \tag{2.41}$$

Downlink Case

We now consider an opposite situation in which one base station is sending separate (independent) information to multiple receivers. The case with two users is shown in Figure 2.10.

In this case the baseband downlink AWGN channel with two users can be described as:

$$\mathbf{y}_k[m]=h_k\mathbf{x}[m]+\mathbf{w}_k[m], \tag{2.42}$$

where $\mathbf{y}_k[m]$ is the received signal at the user *k* at the time moment m. Here h_k is the fixed (complex) channel gain corresponding to the user *k*. It is assumed that h_k is known to both the transmitter and the

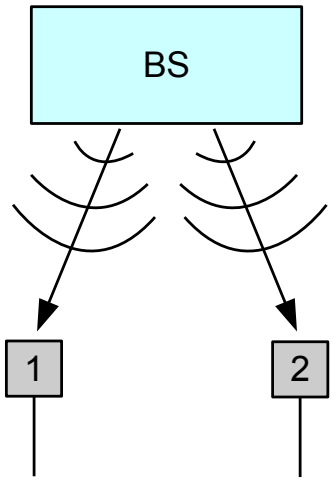

Figure 2.10: *Downlink case.*

.

user *k*. The transmitted signal $\mathbf{x}[m]$ has an average power constraint of P [J/symbol]. The users separately decode their data using the received signal. Similar as in the uplink case, we can define the capacity region C as the region of rates (R_1, R_2) at which the two users can simultaneously reliably communicate. The single user bounds are given by:

$$R_1 < \log\left(1 + \frac{P|\ h_1\ |^2}{\sigma_w^2}\right), \tag{2.43}$$

$$R_2 < \log\left(1 + \frac{P|\ h_2\ |^2}{\sigma_w^2}\right). \tag{2.44}$$

In the symetric case ($|\ h_1\ | = |\ h_2\ |$), the SNR is the same for both users. Therefore, the sum information rate must be bounded by the single user capacity:

$$R_1 + R_2 < \log\left(1 + \frac{P|\ h_1\ |^2}{\sigma_w^2}\right). \tag{2.45}$$

The main idea is that if the user one can decode its data from y_1, the user two, which has the same SNR

should be able to decode the data of user one from y_2. Then the user two can subtract the codeword of user one from its received signal in order to better decode its own data. In other words it can perform SIC.

The transmitter encodes the information for each user using an i.i.d Gaussian code spread at the entire bandwidth. The user one treats the signal for the user two as noise and can be communicated reliably at a rate:

$$R_1 = \log\left(1 + \frac{P_1 |\ h_1\ |^2}{P_2 |\ h_1\ |^2 + \sigma_w^2}\right) = \log\left(1 + \frac{(P_1 P_2)\ |\ h_1\ |^2}{\sigma_w^2}\right) - \log\left(1 + \frac{P_2 |\ h_1\ |^2}{\sigma_w^2}\right). \tag{2.46}$$

The user two performs the SIC and it can support a rate given by:

$$R_2 = \log\left(1 + \frac{P_2 |\ h_2\ |^2}{\sigma_w^2}\right). \tag{2.47}$$

In case when the channel coefficients are not the same (for example ($|\ h_1\ | < |\ h_2\ |$)), the user that has the better channel performs SIC. In this case rates are given by:

$$R_1 = \log\left(1 + \frac{P_1 |\ h_1\ |^2}{P_2 |\ h_1\ |^2 + \sigma_w^2}\right), \tag{2.48}$$

$$R_2 = \log\left(1 + \frac{P_2 |\ h_2\ |^2}{\sigma_w^2}\right). \tag{2.49}$$

In general case when there are K users in the network and the channels are symetric ($|\ h_1\ | = ... = |\ h_K\ | = |\ h\ |$) the rate is constrained by:

$$\sum_{k=1}^{K} R_k < \log\left(1 + \frac{P |\ h\ |^2}{\sigma_w^2}\right). \tag{2.50}$$

If there is no symmetry ($|\ h_1\ | \leq ... \leq |\ h_k\ |$) , the rate is constraint by:

$$\sum_{k=1}^{K} R_k = \log\left(1 + \frac{P_k |\ h_k\ |^2}{\left(\sum_{j=k+1}^{K} P_j\right) |\ h_k\ |^2 + \sigma_w^2}\right), \tag{2.51}$$

where $P = \sum_k^K P_k$ is the power split among the users.

Chapter 3

Cooperative Cellular Networks

In this chapter, zero-forcing for interference cancellation is described. It is an adaptive algorithm and a special case of the beam-forming technique which is often used for high capacity systems [9] like the one considered in this thesis. The error at the receiver side which is caused by traveling of the signal from the transmitter to the receiver through the wireless medium is being reduced. The zero-forcing technique does interference cancellation by forcing the interference signal at the receiver side to be equal to zero.

The cooperative network considered in the following is made of M transmitters each having N_t antennas and K receivers with N_r antennas each. Each receiver antenna receives signals from all the transmit antennas. However, for a particular receiver not all of the received signals are desired. Therefore, a solution which removes interference has to be found. In the following text one solution for eliminating interference will be described - the zero-forcing aproach. Figure 3.1 illustrates the described cooperative network.

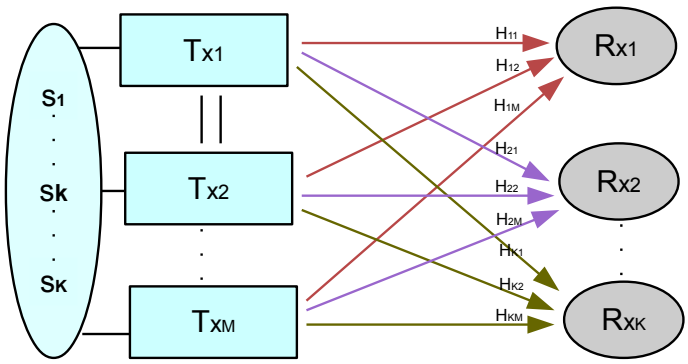

Figure 3.1: *Cooperative network: multiple base stations are connected together and jointly transmit information to multiple mobiles*

3.1 Description of Cooperative Cellular Network

In the model of Figure 3.1 each transmitter $T_j \in \{T_1, ..., T_M\}$ forms its transmit signal by multiplying each transmit symbol $\mathbf{s}_i$ with a precoding matrix $\mathbf{G}_i^{(j)}$. The resulting transmit signal can thus be written as:

$$\mathbf{x}_j = \sum_{i=1}^{K} \mathbf{G}_i^{(j)} \mathbf{s}_i = \begin{bmatrix} \mathbf{G}_1^{(j)} & \mathbf{G}_2^{(j)} & \dots & \mathbf{G}_K^{(j)} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_K \end{bmatrix}, \tag{3.1}$$

where $\mathbf{s}_i$ is the vector of symbols intended for user $U_i \in \{U_1, ..., U_K\}$, $\mathbf{G}_i^{(j)}$ is the precoding matrix for the i^{th} symbol and the j^{th} transmitter, $\mathbf{x}_j$ is the transmitted signal of the j^{th} transmitter. Each $\mathbf{s}_i$ is a vector with dimension N_s that is limited by the number of transmitter antennas: $1 \leq N_s \leq MN_t$. Each symbol is assumed to be a complex number ($\mathbf{s}_k \in \mathbb{C}^{N_s}$) with distribution $\mathcal{CN}(0, 1)$ having variance equal to one. All vectors $\mathbf{s}_k$ of symbols are independent and identical distributed (i.i.d.). The precoding matrix $\mathbf{G}_i^{(j)}$ has a size determined by the number of transmitter antennas and the dimension of the symbol vectors. Therefore, we can write: $\mathbf{G}_i^{(j)} \in \mathbb{C}^{N_t \times N_s}$.

As can be seen in Figure 3.1, channel response from the transmitter j to the receiver i is denoted by $\mathbf{H}_{ij}$. The size of this matrix is determined by the number of transmitter and receiver antennas: $\mathbf{H}_{ij} \in \mathbb{C}^{N_r \times N_t}$. Therefore, for each receiver, we can form the channel matrix as:

$$\mathbf{H}_i = \begin{bmatrix} \mathbf{H}_{i1} & \mathbf{H}_{i2} & \dots & \mathbf{H}_{iM} \end{bmatrix}. \tag{3.2}$$

The received signal at the receiver U_i can then be written as:

$$\mathbf{y}_i = \mathbf{H}_i \mathbf{x} + \mathbf{w}_i, \tag{3.3}$$

where $\mathbf{x}$ represents the vector of all transmitted signals:

$$\mathbf{x} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_K \end{bmatrix}, \tag{3.4}$$

and $\mathbf{w}_i$ represents the noise at the receiver U_i. Based on (3.1) and (3.3), we obtain the equation of the received signal as a function of network symbols as:

$$\mathbf{y}_i = \mathbf{H}_i\mathbf{x} + \mathbf{w}_i \tag{3.5}$$

$$= \mathbf{H}_i \begin{bmatrix} \mathbf{G}_1^{(1)} & \dots & \mathbf{G}_K^{(1)} \\ \mathbf{G}_1^{(2)} & \dots & \mathbf{G}_K^{(2)} \\ \vdots & \vdots & \vdots \\ \mathbf{G}_1^{(M)} & \dots & \mathbf{G}_K^{(M)} \end{bmatrix} \begin{bmatrix} \mathbf{s}_1 \\ \mathbf{s}_2 \\ \vdots \\ \mathbf{s}_K \end{bmatrix} + \mathbf{w}_i \tag{3.6}$$

$$= \sum_{j=1}^{M} \mathbf{H}_{ij}\mathbf{x}_j + \mathbf{w}_i \tag{3.7}$$

$$= \sum_{j=1}^{M}\sum_{k=1}^{K} \mathbf{H}_{ij}\mathbf{G}_k^{(j)}\mathbf{s}_k + \mathbf{w}_i \tag{3.8}$$

$$= \mathbf{H}_i \sum_{j=1}^{M} \mathbf{G}_i^{(j)}\mathbf{s}_i + \mathbf{H}_i \sum_{j=1}^{M}\sum_{k\neq i}^{K} \mathbf{G}_k^{(j)}\mathbf{s}_k + \mathbf{w}_i. \tag{3.9}$$

The received signal can be represented in the following way:

$$\mathbf{y}_i = \mathbf{y}^{(\text{des})} + \mathbf{y}^{(\text{int})} + \mathbf{w}_i, \tag{3.10}$$

where $\mathbf{y}^{(\text{des})}$ and $\mathbf{y}^{(\text{int})}$ stand for the desired signal and the interference and are given by:

$$\mathbf{y}^{(\text{des})} = \mathbf{H}_i \sum_{j=1}^{M} \mathbf{G}_i^{(j)}\mathbf{s}_i, \tag{3.11}$$

$$\mathbf{y}^{(\text{int})} = \mathbf{H}_i \sum_{j=1}^{M}\sum_{k\neq i}^{K} \mathbf{G}_k^{(j)}\mathbf{s}_k. \tag{3.12}$$

3.2 Power Constraint and Achievable Rate Calculation

In order for the further discussion to be realistic, we consider the power constraints. Since the power of the whole communication system is limited, each node has a limited power which might be different from node to node due to different modulation of each carrier. This fact has to be taken into account when calculating the precoding matrices $\mathbf{G}_k$ for each transmitter. If P_{tx} is the maximal power at each node, then the power of each transmitter has to be less then P_{tx}, which we can denote in the following form:

$$P_j \leq P_{tx}, \forall j \in (1, \dots, M), \tag{3.13}$$

where P_j is the total transmit power of the transmitter T_j, given by the following formula:

$$P_j = \text{Trace}\left\{E\left[\mathbf{x}_j\mathbf{x}_j^H\right]\right\} \tag{3.14}$$

$$= \text{Trace}\left\{E\left[\left(\sum_{i=1}^{K}\mathbf{G}_i^{(j)}\mathbf{s}_i\right)\left(\sum_{i=1}^{K}\mathbf{G}_i^{(j)}\mathbf{s}_i\right)^H\right]\right\} \tag{3.15}$$

$$= \text{Trace}\left\{E\left[\sum_{i=1}^{K}\sum_{i'=1}^{K}\mathbf{G}_i^{(j)}\mathbf{s}_i\mathbf{s}_{i'}^H\left(\mathbf{G}_{i'}^{(j)}\right)^H\right]\right\} \tag{3.16}$$

$$= \text{Trace}\left\{\sum_{i=1}^{K}\sum_{i'=1}^{K}\mathbf{G}_i^{(j)}E\left[\mathbf{s}_i\mathbf{s}_{i'}^H\right]\left(\mathbf{G}_{i'}^{(j)}\right)^H\right\} \tag{3.17}$$

$$= \text{Trace}\left(\sum_{i=1}^{K}\mathbf{G}_i^{(j)}\left(\mathbf{G}_i^{(j)}\right)^H\right), \tag{3.18}$$

where the last step follows because

$$E\left[\mathbf{s}_i\mathbf{s}_{i'}^H\right] = \begin{cases}\mathbf{O}, & \text{if} \quad i \neq i'\\ \mathbf{I}, & \text{if} \quad i = i',\end{cases} \tag{3.19}$$

since the elements of the symbol vectors $\mathbf{s}_i$ are assumed to be independent. In order to satisfy the power constraint in (3.13), the precoding matrices are normalized according to:

$$\tilde{\mathbf{G}}_i^{(j)} = \frac{\sqrt{P_{\text{tx}}}}{\sqrt{\max(P_j)}}\mathbf{G}_i^{(j)}. \tag{3.20}$$

Note that this adjustment for $\tilde{\mathbf{G}}_i^{(j)}$ is not optimal since we are dividing with the maximal output power. Therefore some of the base station may actually transmit with power less than P_{tx} [9]. A solution that is optimal for all the base stations would allow all base stations to transmit with full power P_{tx}. However, such a solution is hard to implement in practice.

Since the signal at the receiver consists of the desired signal, interference and the noise, the achievable rate can be calculated by considering the contributions of each signal part. In this section we will show how the covariance matrices of the desired signal, the interference and the noise are calculated.

The covariance matrix of the desired signal is given by:

$$\mathbf{K_S}^{(k)} = E\left[\mathbf{y}_k^{(\text{des})}\left(\mathbf{y}_k^{(\text{des})}\right)^H\right] \tag{3.21}$$

$$= E\left[\left(\sum_{j=1}^{M}\mathbf{H}_{kj}\tilde{\mathbf{G}}_k^{(j)}\mathbf{s}_k\right)\left(\sum_{j=1}^{M}\mathbf{H}_{kj}\tilde{\mathbf{G}}_k^{(j)}\mathbf{s}_k\right)^H\right] \tag{3.22}$$

$$= E\left[\sum_{j=1}^{M}\sum_{j'=1}^{M}\mathbf{H}_{kj}\tilde{\mathbf{G}}_k^{(j)}\mathbf{s}_k\mathbf{s}_k{}^H\left(\tilde{\mathbf{G}}_k^{(j')}\right)^H\left(\mathbf{H}_{kj'}\right)^H\right] \tag{3.23}$$

$$= \sum_{j=1}^{M}\sum_{j'=1}^{M}\mathbf{H}_{kj}\tilde{\mathbf{G}}_k^{(j)}E\left[\mathbf{s}_k\mathbf{s}_k{}^H\right]\left(\tilde{\mathbf{G}}_k^{(j')}\right)^H\left(\mathbf{H}_{kj'}\right)^H \tag{3.24}$$

$$= \mathbf{H}_k\tilde{\mathbf{G}}_k(\mathbf{H}_k\tilde{\mathbf{G}}_k)^H, \tag{3.25}$$

where $\mathbf{y}_k^{(\text{des})}$ stands for the desired signal, $\mathbf{H}_k$ is the channel response matrix for the receiver $\mathbf{U}_k$ and $\tilde{\mathbf{G}}_k$ is the normalized precoding matrix for the same receiver.

The covariance matrix of the interference signal for the receiver $\mathbf{U}_k$ is given as:

$$\mathbf{K_I}^{(k)} = E\left[\mathbf{y}_k^{(\text{int})}\left(\mathbf{y}_k^{(\text{int})}\right)^H\right] \tag{3.26}$$

$$= E\left[\left(\sum_{j=1}^{M}\sum_{\substack{i=1\\i\neq k}}^{K}\mathbf{H}_{kj}\tilde{\mathbf{G}}_i^{(j)}\mathbf{s}_i\right)\left(\sum_{j=1}^{M}\sum_{\substack{i=1\\i\neq k}}^{K}\mathbf{H}_{kj}\tilde{\mathbf{G}}_i^{(j)}\mathbf{s}_i\right)^H\right] \tag{3.27}$$

$$= \sum_{j=1}^{M}\sum_{\substack{i=1\\i\neq k}}^{K}\sum_{j'=1}^{M}\sum_{\substack{i'=1\\i'\neq k}}^{K}\mathbf{H}_{kj}\tilde{\mathbf{G}}_i^{(j)}E\left[\mathbf{s}_i\mathbf{s}_{i'}{}^H\right]\left(\tilde{\mathbf{G}}_{i'}^{(j')}\right)^H\left(\mathbf{H}_{kj'}\right)^H \tag{3.28}$$

$$= \sum_{j=1}^{M}\sum_{\substack{i=1\\i\neq k}}^{K}\sum_{j'=1}^{M}\mathbf{H}_{kj}\tilde{\mathbf{G}}_i^{(j)}\left(\tilde{\mathbf{G}}_i^{(j')}\right)^H\left(\mathbf{H}_{kj'}\right)^H \tag{3.29}$$

$$= \sum_{i\neq k}\mathbf{H}_k\tilde{\mathbf{G}}_i(\mathbf{H}_k\tilde{\mathbf{G}}_i)^H, \tag{3.30}$$

where $\mathbf{y}_k^{(\text{int})}$ is the interference signal.

The covariance matrix of the noise is given by the following form:

$$\mathbf{K_W}^{(k)} = E\left[\mathbf{w}_k\mathbf{w}_k^H\right] = \sigma_w^2\mathbf{I}. \tag{3.31}$$

where σ_w^2 is the variance of the white Gaussian noise due to assumption of $\mathbf{w}$ with elements that are i.i.d $\mathcal{CN}(0,\sigma_w^2)$.

According to (2.32) an achievable rate for the receiver k is given by the following equation [4]:

$$R_k = \log\left(\det\left(\mathbf{K_S}^{(k)} + \mathbf{K_I}^{(k)} + \mathbf{K_W}^{(k)}\right)\right) - \log\left(\det\left(\mathbf{K_I}^{(k)} + \mathbf{K_W}^{(k)}\right)\right). \tag{3.32}$$

3.3 Interference Cancellation - Zero Forcing

In this section we describe a way to cancel the interference, or reduce the interference signal. If the interference is known to the receiver, then the best method for interference cancellation is Dirty-Paper coding [11]. However, this method is very complex and can not be easily implemented in practice. Therefore, the linear beamforming technique is more often used since it achieves good interference cancellation and is much easier to implement. The beamforming technique is a signal processing technique used for directional signal transmission or reception. It combines the elements in the array in a way such that signals at particular angles experience constructive interference while others experience destructive interference. Zero-Forcing is one kind of a beamforming technique. It is a linear, adaptive algorithm that forces the interference at the receiver side to be zero as its name says [9]. In the following section we give a mathematical description of the zero-forcing technique by an appropriate choice of the precoding matrix. Each receiver receives both the desired signal and the interference signal which is the result of channel response to signals sent to other receivers. In order to cancel the interference according to equation (3.12) it has to hold:

$$\mathbf{y}_i^{(\text{int})} = \mathbf{H}_i \sum_{k \neq i}^{K} \mathbf{G}_k^{(j)} \mathbf{s}_k = \mathbf{O}. \tag{3.33}$$

A way to satisfy this condition is to have each member in the sum equal to zero, which leads to the following homogenous system of equations:

$$\mathbf{H}_{km}\mathbf{G}_k^{(m)} = 0, \qquad m = (1, ..., k-1, k+1, ..., M), \tag{3.34}$$

where T_m is m^{th} transmitter and U_k is k^{th} receiver.

Actually, we can write one condition for the system in the extended form:

$$\mathbf{H}_{km}\mathbf{G}_k^{(m)} = \begin{bmatrix} h_{11}^{(m)} & h_{12}^{(m)} & \dots & h_{1Nt}^{(m)} \\ h_{21}^{(m)} & h_{22}^{(m)} & \dots & h_{2Nt}^{(m)} \\ \vdots & & & \\ h_{Nr1}^{(m)} & h_{Nr2}^{(m)} & \dots & h_{NrNt}^{(m)} \end{bmatrix} \begin{bmatrix} g_{11}^{(k)} & g_{12}^{(k)} & \dots & g_{1Nt}^{(k)} \\ g_{21}^{(k)} & g_{22}^{(k)} & \dots & g_{2Nt}^{(k)} \\ \vdots & & & \\ g_{Nt1}^{(k)} & g_{Nt2}^{(k)} & \dots & g_{NtNt}^{(k)} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 0 \\ \vdots & & & \\ 0 & 0 & \dots & 0 \end{bmatrix}. \tag{3.35}$$

We can write this also in the following form:

$$\sum_{i=1,j=1}^{Nr,Nt} h_{ij}^{(km)} g_{ji}^{(k)} = 0. \tag{3.36}$$

$\mathbf{G}_k$ is the k^{th} column of the $\mathbf{G}$ matrix in (3.9):

$$\mathbf{G}_k = \begin{bmatrix} \mathbf{G}_k^{(1)} \\ \mathbf{G}_k^{(2)} \\ \vdots \\ \mathbf{G}_k^{(M)} \end{bmatrix}. \tag{3.37}$$

Clearly the solution of the equation system (3.34) is given by finding the corresponding null space, so (3.34) is solved by the following choice of the precoding matrix:

$$\mathbf{G}_k = \text{null} \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_{k-1} \\ \mathbf{H}_{k+1} \\ \vdots \\ \mathbf{H}_K \end{bmatrix}. \tag{3.38}$$

By this selection of precoding matrices at each transmitter, interference signals are canceled for all the receivers.

By analysing the system of equation (3.34) we can specify the number of necessary antennas for the system. In order for the system of equations (3.34) to be solvable, the number of unknown variables has to be greater or equal to the number of equations. We are dealing with a system of M transmitters and K receivers. The number of unknown variables is determined by the number of the transmit antennas since the dimension of each matrix $\mathbf{G}_k$ is N_tN_t and we have K receivers, the number of independent variables is equal to MKN_tN_t. On the other hand, the number of equations in the system is equal to $MK(K-1)N_rN_t$ because there are $(K-1)$ matrix equations and the dimension of $\mathbf{H}_{km}\mathbf{G}_k$ is N_rN_t. Therefore, the condition that the antenna numbers have to satisfy in order for the system (3.34) to be solvable is:

$$MKN_tN_t \geq KM\,(K-1)\,N_rN_t, \tag{3.39}$$

$$N_t \geq (K-1)\,N_r. \tag{3.40}$$

However, if condition (3.40) is fulfilled with equality, only the trivial solution exist, i.e. the resulting coefficient in the precoding matrices are all null. In order to find non–trivial solutions, the condition is adjusted to $N_t > (K-1)\,N_r$.

3.4 Computer Simulation

Achievable rates of the network were experimentally calculated in Matlab. In the experiment, channel responses $\mathbf{H}$ were simulated by Rayleigh fading with i.i.d. Rayleigh elements $\mathcal{CN}(0,1)$. The rate was cal-

culated at different values of transmit power. Each calculation was repeated 100 times and the achievable rate was taken as the mean of the calculated rate values. The simulation was done for a system with 3 transmitters and 3 receivers. Dependence of the rate on transmit power was compared for different combinations of the transmitter and the receiver antenna numbers. The obtained experimental results are shown in Figure 3.2.

In order to have any signal at the receiver side, certain limits on the numbers of transmitter and receiver antennas have to be satisfied. For example in case when there are three receivers with two antennas each, the total number of antennas at the transmitter side should be such that each receiver receives the signal from at least one transmit antenna. In this case the minimum number of the transmit antennas should be five. This is a direct consequence of (3.38). Therefore, in case when each transmitter has only one antenna and each receiver has two antennas, this condition is not satisfied and hence the rate is identically equal to zero as can be seen in Figure 3.2. In such a case the precoding matrix remains an empty matrix. Other plots in Figure 3.2 compare the rate as a function of transmitted power for different combinations of transmitter and receiver antenna numbers for which the condition is satisfied. As can be seen in case of two receiver antennas, having three or two transmitter antennas does not make a big difference in terms of achievable rate and therefore having two transmitter antennas would be preferable, because of the resource saving. However, in case two transmitter and one receiver antenna is used, the achievable rate is significantly reduced. On the other hand, increasing the number of transmitter and receiver antennas can significantly increase the achievable rate.

From the figure 3.2 we can see that multiplexing gain in the case of two transmit and one receiver antenna is equal to one. If we look at the rate for the transmitted signal of 30 dB, we see that it is equal to 28.7 bit/sec. While for a power of 10 dB more (40 dB) we read out the rate of 38.3 bit/sec which is 10 bit/sec more. Based on this, we can calculate the multiplexing gain as 1. This corresponds (as already explained in chapter 2) to $r = 1$. Similarly if we look at the cases where $r = 2$, we see that 10 dB transmit power increase leads to 20 bit/sec rate increase and therefore the calculate multiplexing gain is two. A similar calculation can be made for $r = 3$. Therefore, we can see that the multiplexing gain is determined by the smaller antenna count either on the transmit or on the receive side. Therefore in case when there are two receive antennas, there is no reason (if we only consider the multiplexing gain) to have more than two antennas on the transmit side. However, having more antennas at the transmit side leads to higher rate, which is a consequance of an additional diversity gain.

In addition, in Figure 3.2, we see the rate for the case in which the precoding matrix was not calculated according to (3.38), but the elements of the matrix were taken as random numbers. The rate corresponds, almost to a flat line around 5 bit/s. In this case there is no multiplexing gain (as the curve is flat). This means that the interference is very strong and the network saturates in the interference limited regime. No matter how much the transmitted power is increased or the number of transmit and receive antennas, the multiplexing gain will remain at 0 and the interference will be strong. This highlights the need for an effective interference cancellation for cellular networks.

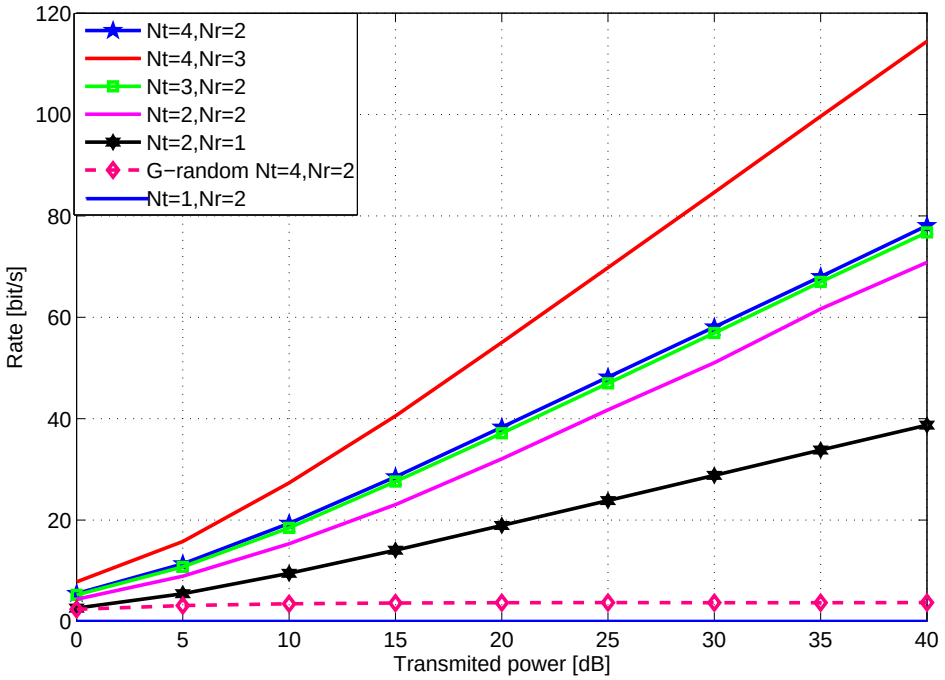

Figure 3.2: *Dependence of the rate on the transmit power for different numbers of transmitter and receiver antennas.*

Chapter 4

Chanel Estimation and Quantization

In order for the wireless communications system described in the previous chapters to work properly, it is necessary for the communication medium to be known. Moreover, in order to use the zero-forcing interference cancellation technique from the previous chapter, channel matrices have to be known exactly. In a real system, channel matrices are therefore estimated. In addition they are quantized, so that CSI can be disseminated to other base station. Both the estimation and the quantization introduce errors in the channel matrix representation, and this will lead to wrong $\mathbf{G}_i$ which can lead to performance loss. The quantization error is usually assumed to be much bigger than the estimation error and we will therefore focus on this type of error. In what follows, we assume that both the transmitters and the receivers have perfect knowledge of the local CSI, i.e. the communication channels can be directly measured.

4.1 Channel State Information

Known channel properties of a communication link are often referred as channel state information (CSI). This information shows how a signal propagates from the combined effect of scattering, fading and power decay with distance. The CSI is most usually estimated and quantized at the receiver and then sent back to the transmitter. The CSI can be different at the transmitter and the receiver. There are two levels of CSI, the instantaneous CSI and the statical CSI.

Instantaneous CSI (or short-term CSI) is the information on the current channel conditions. It can be regarded as a knowledge of the channel impulse response. It can be used to adapt the transmitted signal in order to optimize transmission.

Statistical CSI (or long-term CSI) is the information on the statistical characteristics of the communication channel. Such information may typically include the fading distribution, the average channel gain and the spatial correlation. This information can also be used for optimizing the transmit signal.

The base station sends pilot signals to the receiver. Upon receiving these pilot signals, the receiver estimates the communication channel and sends the information on the CSI back to the base station. Once the station has the knowledge of the channel characteristics, it can send the desired signal. In cooperative

networks, multiple transmitters send their data jointly with the goal to increase the overall performance of the network, e.g. to cancel interference if zero-forcing is applied. To this end, the base stations have to share their local CSI. To do so, the analog values (of infinite precision) of the channel coefficients need to be quantized, such that they can be sent digitally over the backhaul, which connect different base stations. The base stations use then this obtained CSI to calculate the precoding matrices and to apply zero-forcing in order to suppress the interference at the receivers.

4.2 Quantization

Quantization is the process of transforming an analog into a digital value. The device that performs quantization is called a quantizer. In this process part of the information contained in the original analog value is lost. This is because quantization is rounding the analog value to some unit of precision. The set of analog values with infinite precision is mapped to a discrete set of values which can be represented by certain number of bits. In order to reduce the quantization error, a greater number of bits should be used. However, the number of bits that can be used is often limited by the used hardware or other factors. Often the number of bits is a design factor which should be selected so that the minimal possible quantization error is achieved with reasonable quantization infrastructure cost.

4.3 Quantization of the Channel Matrix

In the previous chapter, all the calculations were done by assuming that the channel responses are perfectly well known and precise. However, this is not true in practice. Channel responses first need to be estimated [8], and since the estimated values cannot be represented in hardware with infinite precision, they need to be quantized. The actual process of calculating precoding matrices is illustrated in Figure 4.1.

Both the channel estimation and the quantization bring errors to the channel response representation. However, as sophisticated estimation algorithms of good performance are available [14], we assume that the quantization error is much bigger than the error introduced by the estimation. Therefore, in this chapter we will consider the estimation error to be equal to zero and we will only concentrate on the quantization error and its effects on interference cancellation. We can represent the quantized channel matrix in the following form:

$$\tilde{\mathbf{H}} = \mathbf{H} + \mathbf{E}_q, \tag{4.1}$$

where $\mathbf{E}_q$ is the quantization error, and $\tilde{\mathbf{H}}$ is the the channel matrix after quantization. This condition is true if we are following the assumption of zero estimation error:

$$\hat{\mathbf{H}} = \mathbf{H} + \mathbf{E}_{estim}, \tag{4.2}$$

where $\mathbf{E}_{estim}$ is the error due to channel estimation. If this error is considering too small and disregarded, we get:

$$\mathbf{E}_{estim} = 0, \tag{4.3}$$

$$\hat{\mathbf{H}} = \mathbf{H}, \tag{4.4}$$

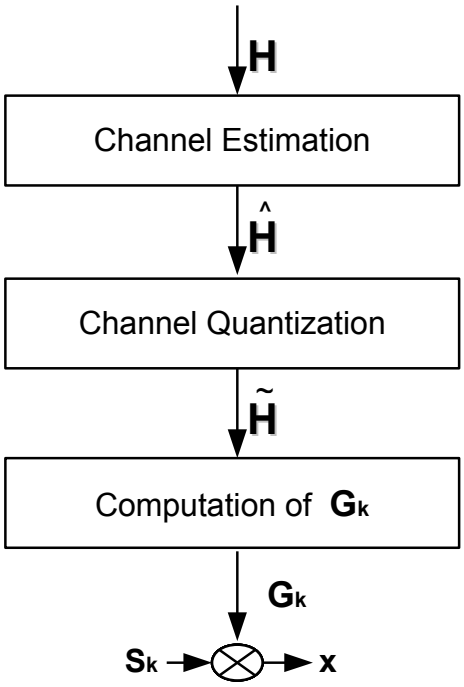

Figure 4.1: *Transmitted signal calculation procedure*

.

$$\tilde{\mathbf{H}} = \hat{\mathbf{H}} + \mathbf{E}_q, \tag{4.5}$$

where $\mathbf{H}$ represents the estimation of the analog channel response matrix with infinite precision and $\mathbf{E}_q$ is the error introduced by the quantization. In this notation the representation of the channel matrix for the transmitter j becomes: $\left(\tilde{\mathbf{H}}_{1j}, \tilde{\mathbf{H}}_{2j}, \ldots, \tilde{\mathbf{H}}_{kj}\right)$.

The quantization error is propagated to the calculation of the precoding matrix. This is because the calculation of the precoding matrix does not use actual channel response $\mathbf{H}$ with infinite precision like in (3.38), but rather the quantized channel matrix $\tilde{\mathbf{H}}$. The k$^{\text{th}}$ column of the precoding matrix $\mathbf{G}_k$ is calculated

as a null space of quantized channel responses:

$$\mathbf{G}_k = \text{null} \begin{bmatrix} \tilde{\mathbf{H}}_\mathbf{1} \\ \vdots \\ \tilde{\mathbf{H}}_\mathbf{k-1} \\ \tilde{\mathbf{H}}_\mathbf{k+1} \\ \vdots \\ \tilde{\mathbf{H}}_\mathbf{K} \end{bmatrix}, \tag{4.6}$$

In addition, the quantization error will give rise to the error in calculating the normalized precoding matrix (3.20). For the calculation of the desired signal (3.25) and interference covariance (3.30), the estimated channel response matrices with infinite precision are used as before.

Rounding of the Unit Values

There are different quantization methods which differ in their performance and level of complexity. Taking the performance and the complexity level we can divide the quantization techniques into:

- Scalar quantization,
- Vector quantization.

In scalar quantization each matrix element is quantized as a separate scalar value. We can further divide scalar quantization into:

- Uniform quantization in which the quantization intervals are of the same uniform size
- Non-uniform quantization in which the quantization intervals are optimized with respect to the distribution of the input data. If the data, in our case the channel coefficients, are of a Gaussian distribution, the quantization intervals around the mean are more typical than the intervals for away form the mean. If the input distribution is not uniform, such optimal quantizers lead to smaller errors, but their complexity is much higher than that of a uniform quantizer.

Vector quantization is much more complex than the scalar quantization, but it leads to much smaller quantization error, as we will see in the following chapters. In vector quantization, whole vectors (as sets of scalar values) are quantized together. In this thesis we considered two types of vector quantization algorithms:

- K-means,
- LBG.

Both algorithms are very similar, as we can see later. However the LBG is faster than the K-means.

Chapter 5

Scalar Quantization

Entries of the estimated channel response matrices are complex numbers. Therefore these matrices can be split into two matrices, with real entries representing the real and imaginary entries representing the imaginary part of the complex numbers. In scalar quantization each entry of these matrices is separately quantized. The numbers that need to be quantized in this process are random. According to our channel model of Rayleigh fading, they are normally distributed, with the probability density function given as:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \mathrm{e}^{-\frac{(x-\mu)^2}{2\sigma^2}}, \tag{5.1}$$

where μ stands for the mean value and σ^2 is the variance. The associated probability density function is "bell"-shaped, and is known as the Gaussian function shown in Figure 5.1. As can be seen, any value can be expected, although very big positive or negative values are not very probable. However, in quantization these values need to be rounded to a limited number of quantization levels. Therefore, a certain maximal limit ($\max_{\text{lim}}$) has to be determined first. Values that are outside the range determined by $\pm \max_{\text{lim}}$ first need to be rounded to $\pm \max_{\text{lim}}$. This way a probability density function with limited set of possible values is created artificially. In order to perform the rounding, uniform quantization can be used. This means that the interval $[+ \max_{\text{lim}} - \max_{\text{lim}}]$ is divided into segments of the same size. To each segment, the center of that segment is attributed as the quantization level to which all the values inside the segment are rounded. The number of the segments is related to the number of quantization bits according to:

$$\text{quant}_{\text{level}} = 2^{\text{n}_{\text{bit}}}, \tag{5.2}$$

where n_{bit} represents the number of quantization bits.

The given discussion corresponds to the quantization of complex numbers in the Cartesian coordinate system. The same principles hold when the polar coordinate system is used.

Thereby, different quantizers are applied to the amplitude and angle, respectively. If the values to be quantized are circularly symmetric complex Gaussian random variable as in our channel model, the angle is uniformly distributed in the interval $[-\pi, \pi]$ and the amplitude is Rayleigh distributed. In contrast to the Cartesian representation, where both the real and imaginary part are of the same distribution, the angle and amplitude differ in the Polar representation. We can therefor apply different quantizers to this two dimensions.

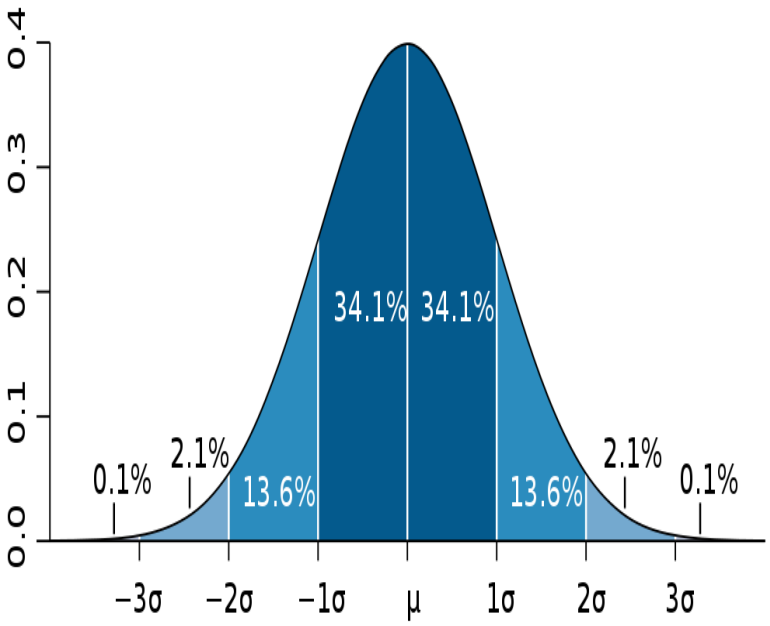

Figure 5.1: *Gaussian representation*

In section 5.3, we present some simulation results and based on them compare the quantization in Cartesian and Polar coordinates.

We also consider vector quantization in high dimensional spaces. Simulations in the vector case are done for the space dimension of up to 16. Chapter 6 gives details on vector quantization, explains simulation results and gives comparisons.

5.1 Cartesian System for Quantization

In this section we discuss scalar quantization of complex numbers in Cartesian coordinates. Each complex number can be represented as:

$$x = Re\{x\} + \mathrm{i}Im\{x\}, \tag{5.3}$$

where Re stands for the real part of the number, Im is its imaginary part, $\mathrm{i} = \sqrt{-1}$. Real and imaginary part are quantized separately by rounding each to the nearest quantization level. Figure 5.2 illustrats this concept of quantization. As can be seen both the real and imaginary axes are divided into intervals of the same size. The number of intervals is defined by the number of quantization bits used according to equation (5.2). Quantization is done by rounding a value inside an interval to the interval center. For example all

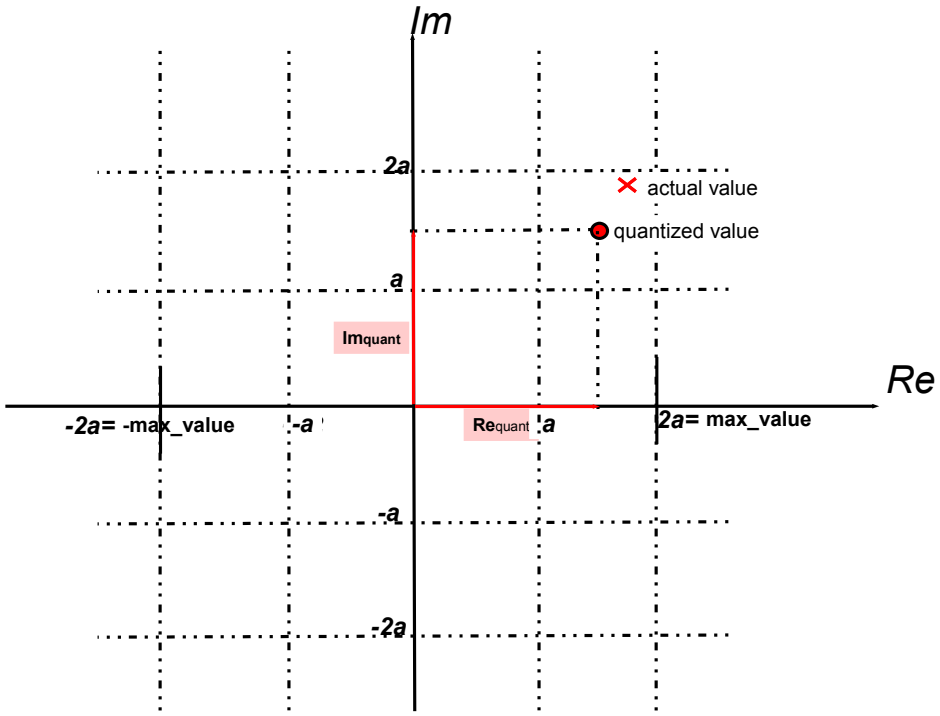

Figure 5.2: *Quantization for Cartesian coordinate system*

real or imaginary part values falling inside the interval $(0, a/2)$ will be rounded to $a/4$, all the values falling inside the interval $(a/2, a)$ will be rounded to $3a/4$ and so on. Since the number of quantization intervals is limited, upper and lower bounds have to be found that limit the segments of the imaginary and real axes to be uniformly divided. All the values that are outside these limiting values are first rounded to the limiting values and then the normal quantization procedure is done.

It is clear that by using more quantization bits, the number of quantization intervals increases, its width decreases and hence the quantization error is decreased. On the other hand, having more quantization bits can significantly increase the traffic burden on the backhole network. Therefore a good compromise between the quantization error (and subsequently achievable rate) and the quantization bit number is required. Our goal is to find the quantization algorithm that yealds small quantization errors for a reasonably small number of quantization bits. For Cartesian quantization one can optimize the upper and lower bounds such that the error is minimized. This optimization is further described in the section (5.3).

5.2 Polar System for Quantization

A complex number can be represented by its real and imaginary part but it can also be represented by its amplitude and phase by:

$$x = re^{i\phi}, \tag{5.4}$$

where $r \in \mathbb{R}^+$ stands for the amplitude and $\phi \in [0, 2\pi]$ is the phase of the complex number x. If the real and imaginary parts of x are normally distributed, the amplitude has Rayleigh distribution and the phase is uniformly distributed and therefore more suitable for uniform quantization described in previous section. Rayleigh distribution is given by the formula:

$$f(x;\sigma) = \frac{x}{\sigma^2}\mathrm{e}^{-\frac{x^2}{2\sigma^2}}, \tag{5.5}$$

where we see that the amplitude can take any value between zero and infinity. Therefore an upper limit has to be found. The interval between zero and the upper limit is divided into segments of uniform size. The phase interval $[0, 2\pi]$ is also divided into segments of the same size, where the number of segments is defined by equation (5.2). The principle for quantization of complex numbers in Polar coordinate is illustrated in Figure 5.3. As we can see, if the complex number has the amplitude in the interval $(0, b)$ it is rounded to $b/2$. Similarly, for the phase, if the phase of the complex number is in the segment $[0, \pi/6]$, then it is rounded to $\pi/12$. Hence the complex value to which the analogue value is rounded is $b/2e^{(i\pi/12)}$.

Similarly as for the quantization in Cartesian coordinates, increase in the quantization bit number reduces the quantization error. In case of Polar quantization we may divide quantization bits to those used for representing the amplitude and those that are used to represent the phase. We can therefor assign a different number of bits for the amplitude and phase, respectively. It is beneficial to use more bits for representing the phase, since this leads to smaller quantization error, as we will see in the following section.

5.3 Simulations for Cartesian and Polar Quantization

In order to better understand the choice of different parameters in the quantization algorithms and their influence on the rate of cooperative networks, several Matlab are performed. We are interested in investigating the achievable rate of the network for different values of quantization bits and interval borders of both the uniform and Polar quantizer. Also we are interested in finding the balance between bits used to represent the amplitude and phase that maximizes the rate.

In the first simulation, the signal to noise ratio of the transmitted signal was fixed to 20dB. Channel response matrices were simulated by Rayleigh fading with elements i.i.d. $\mathcal{CN}(0, 1)$ Rayleigh fading. Chanel response matrices were then quantized by quantizing the real and imaginary part of each matrix entry separately, as explained in the previous section. Based on the quantized channel responses, the precoding matrices were calculated. The achievable rate was calculated for different combinations of the number of quantization bits and maximal limit. The achievable rate was calculated according to (3.32). Figure 5.4 shows the dependence of the achievable rate on maximal level for different number of quantization bits. As

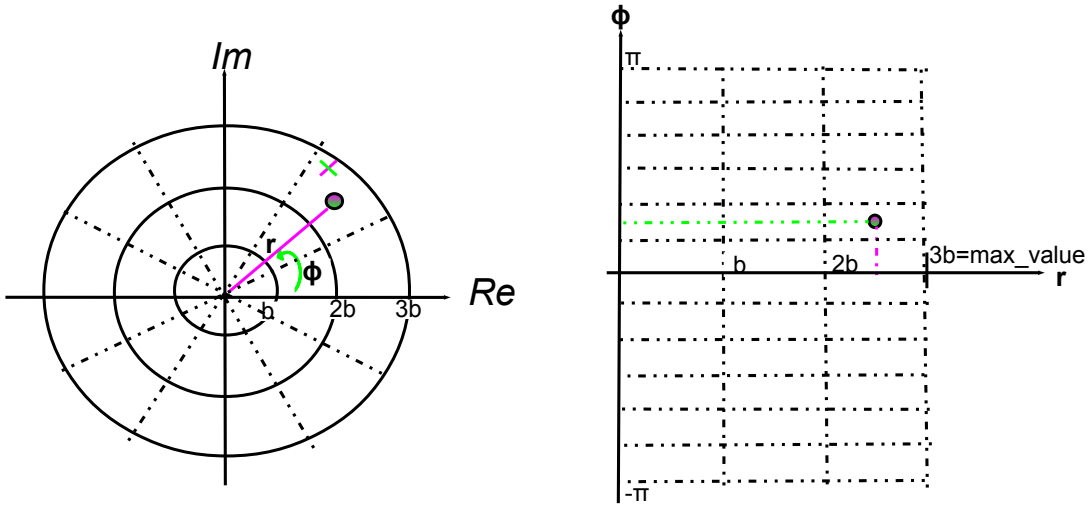

Figure 5.3: *Quantization for Polar coordinate system.*

.

can be seen, for each bit number there is a certain maximal level for which the rate attains its maximum. At the maximal point, the given combination of quantization bits number and maximal level achieves minimal quantization error. In addition, for each bit number, the optimal maximal level was determined. Optimal maximal level as the function of bit number is shown in Figure 5.6. We can see that for high quantization bit numbers, the maximal interval does not play an important role. This is because higher number of bits mean that the interval is divided in many intervals which decreases the quantization error. This is in accordance to (5.2).

In addition, optimal rate values for different bit numbers were compared to cases when there is no quantization and when the precoding matrix is randomly chosen. This is shown in Figure 5.5. As we can see, already for 16 quantization bits the achievable rate comes very close to the rate that can be achieved when channel matrices are taken with infinite precision. On the other hand, we see that even very coarse quantization (only 2 bits) gives a rate that is significant higher(almost double) than in the case when the precoding matrix is randomly selected and no zero-forcing is applied. According to Figure 5.4, the rate that can be achieved with twelve bits comes very close to the rate that can be achieved with sixteen bits. Therefore, in case resources are very limited, 12 bits quantization is more efficient than the 16 bit quantization.

In addition to the described figures, we show the optimal mean quantization error obtained in the experiments for the Cartesian quantization. This is shown in Figure 5.7.

The same simulations as for Cartesian quantization are repeated for Polar quantization. From these experiments we conclude that smaller quantization errors and better achievable rates can be achieved with Polar quantization than with the Cartesian quantization.

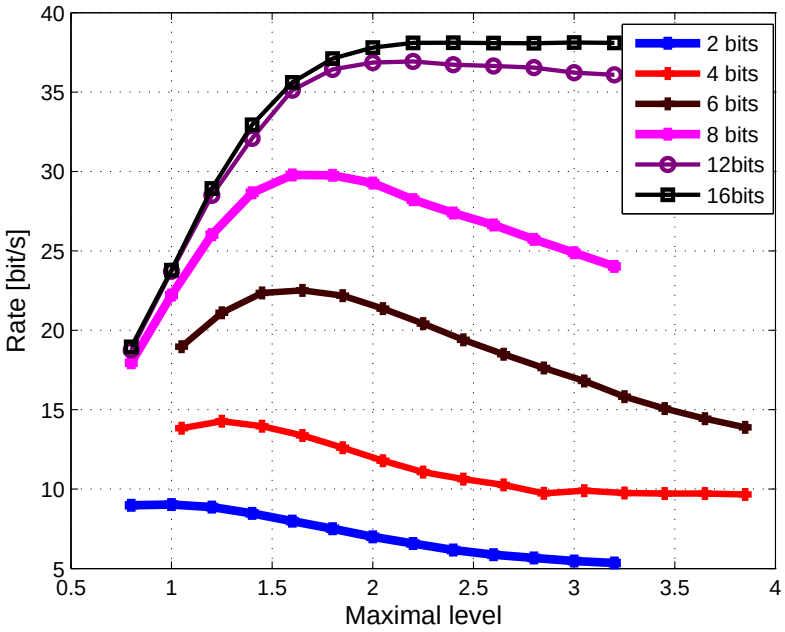

Figure 5.4: *Rate as a function of maximal level.*

.

For Polar quantization we are interested in investigating how the maximal amplitude interval for the amplitude and also uneven use of quantization bits for representing amplitude and phase influence the quantization error and the achievable rate. Comparison of different results we obtained for 8 bit quantization are shown in Figure 5.8. We can see the rate that can be achieved for using 5 bits for angle representation and 3 bits for amplitude representation. The experiments show that it is always better to use more bits for representing the phase then the amplitude. In Figure 5.9 for optimal bit combinations, the dependence of the achievable rate on the maximal amplitude interval is shown. Once again there is always an optimal value that can be selected so that the rate is maximized. Figure 5.11 shows the optimal achievable rates for different numbers of quantization bits in case of Polar quantization and again compares it with the case where channel matrices are used with infinite precision and when the precoding matrix is randomly selected. We can see that the rate is generally bigger than in the case of Cartesian quantization for the quantization. This is more prominent for bigger values of the quantization bit numbers. Figure 5.12 shows the mean quantization error of the experiments with optimal Polar quantization.

Figure 5.13 shows the curves of the achievable rate versus the transmit power for different quantization schemes. As we can see in the case when quantization is done with low bit number, the curves saturate at one point and the rate does not grow with the increase of the transmitted power. This is not the case for quantization with higher number of quantization bits only for P_{tx} up to 40dB, where we see that the achievable rate continues to grow with the increase of transmitted power. From these results we can conclude that the multiplexing gain can not be exploited in case bad quantizer. In addition we can notice the number of bits and the transmit power have a significant influence on the rate. If we choose to quantize with 16 bits, the multiplexing is $r = 2$ for every value of the transmit power values. On the other hand, in case of 2 bit

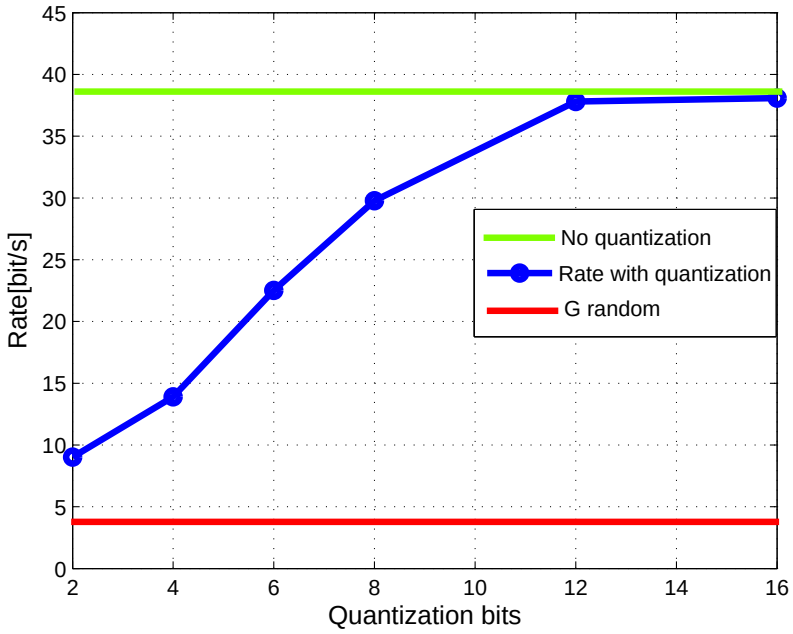

Figure 5.5: *Maximal achievable rate for different quantization bits number.*

.

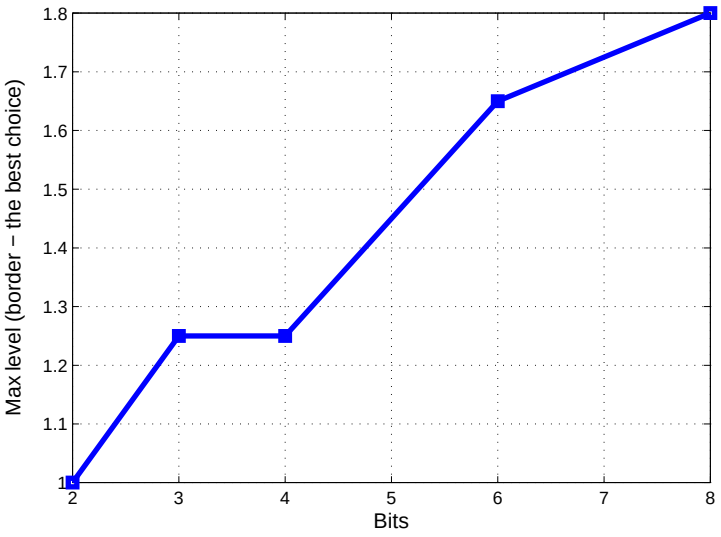

Figure 5.6: *Optimal maximal level for different quantization bits number.*

.

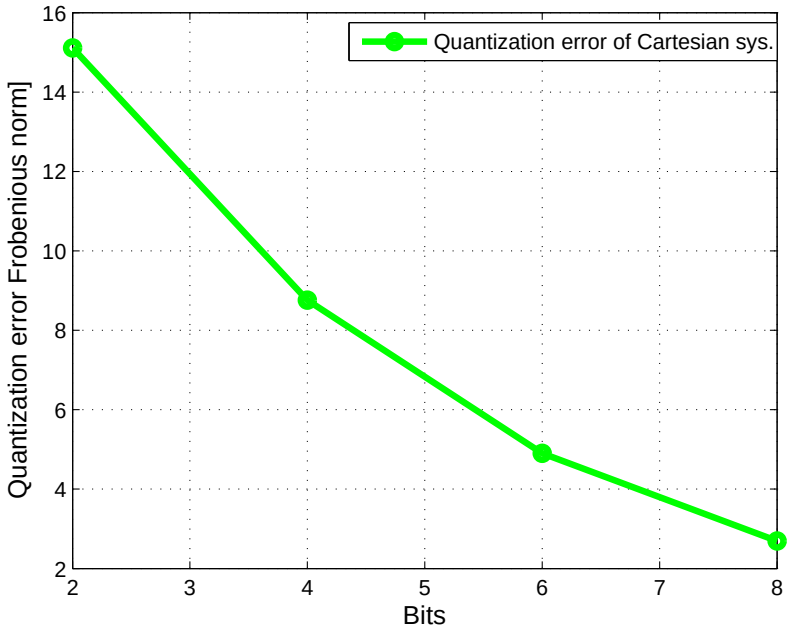

Figure 5.7: *Quantization error for Cartesian quantization.*

.

quantization there is no multiplexing gain, the curve is almost flat for all the transmit power values. Looking at the rate for 12 bits curve, we see that rate of 50bit/s corresponds to the transmit power gives 50 dB, where is the 10 dB higher transmit power gives 55 bit/s, which means that the multiplexing gain is equal to zero in this interval. On the same curve we can see that the multiplexing gain is not zero for the transmit power lower than 30 dB. In fact, the multiplexing gain is $r = 2$ for the transmit power in the range from 10 dB to 20 dB and $r = 1$ for the range from 20 dB to 30 dB.

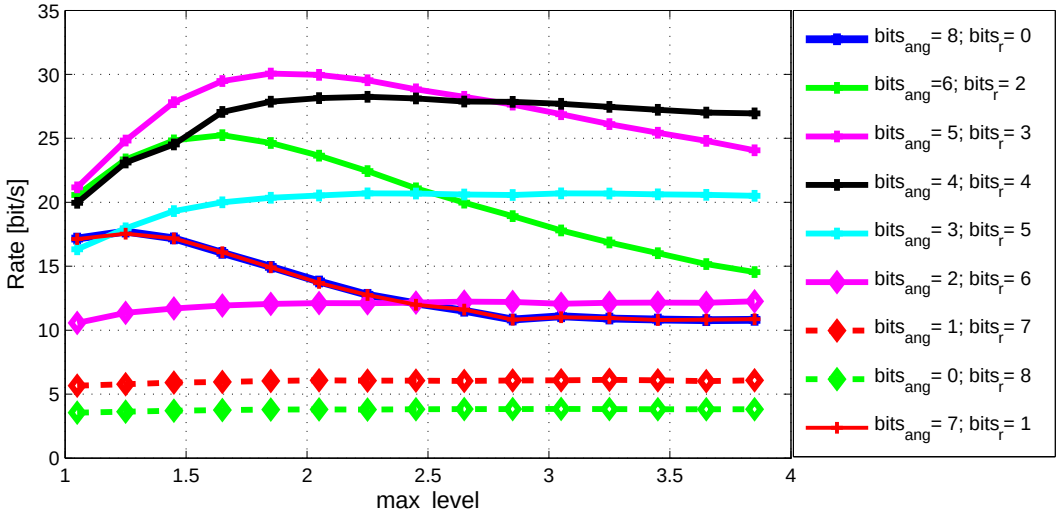

Figure 5.8: *Rate as a function of maximal level in Polar quantization.*

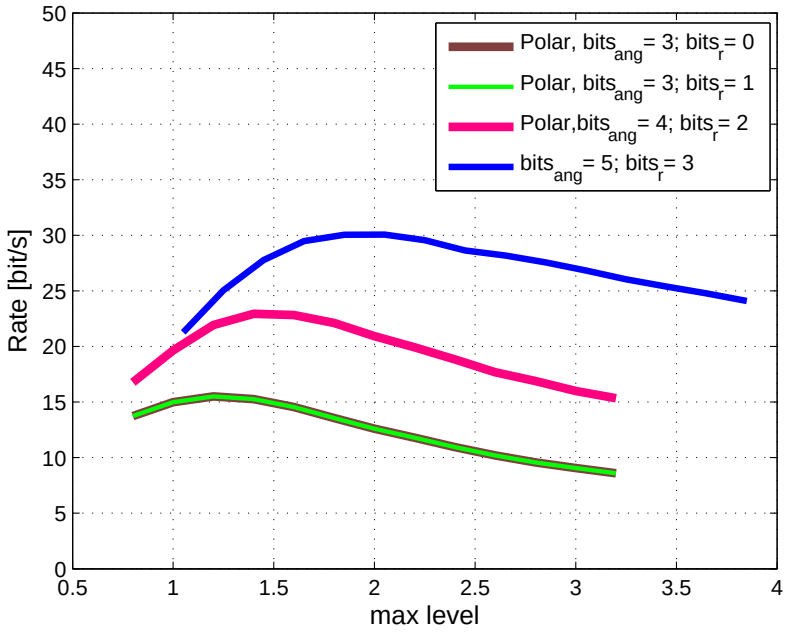

Figure 5.9: *Maximal achievable rate for different quantization bit numbers in Polar quantization.*

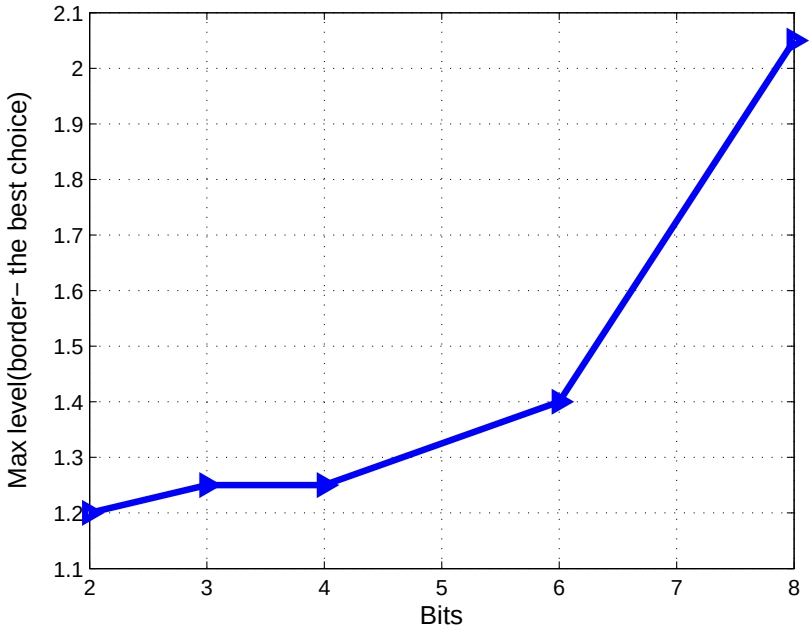

Figure 5.10: *Optimal maximal level for different quantization bit numbers in Polar quantization.*

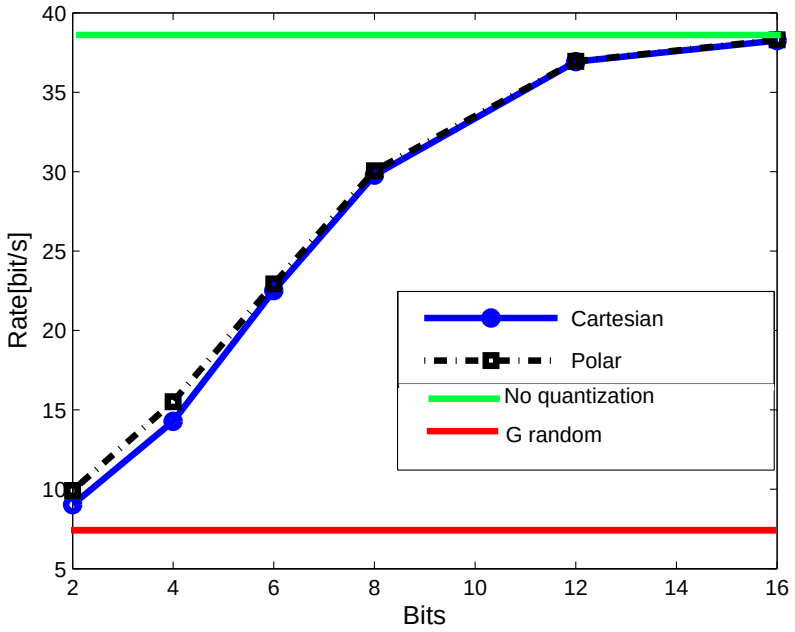

Figure 5.11: *Maximal achievable rate for different quantization bit numbers.*

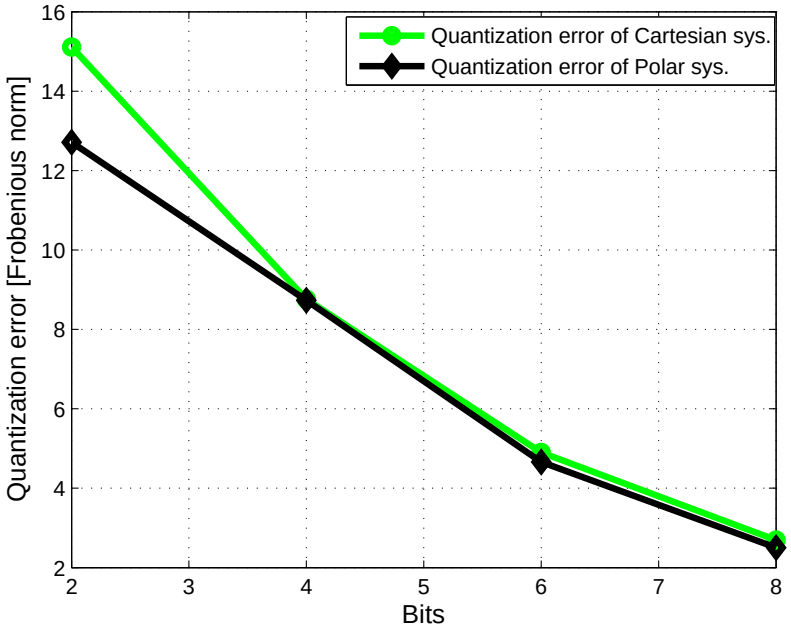

Figure 5.12: *Quantization error for Polar quantization.*

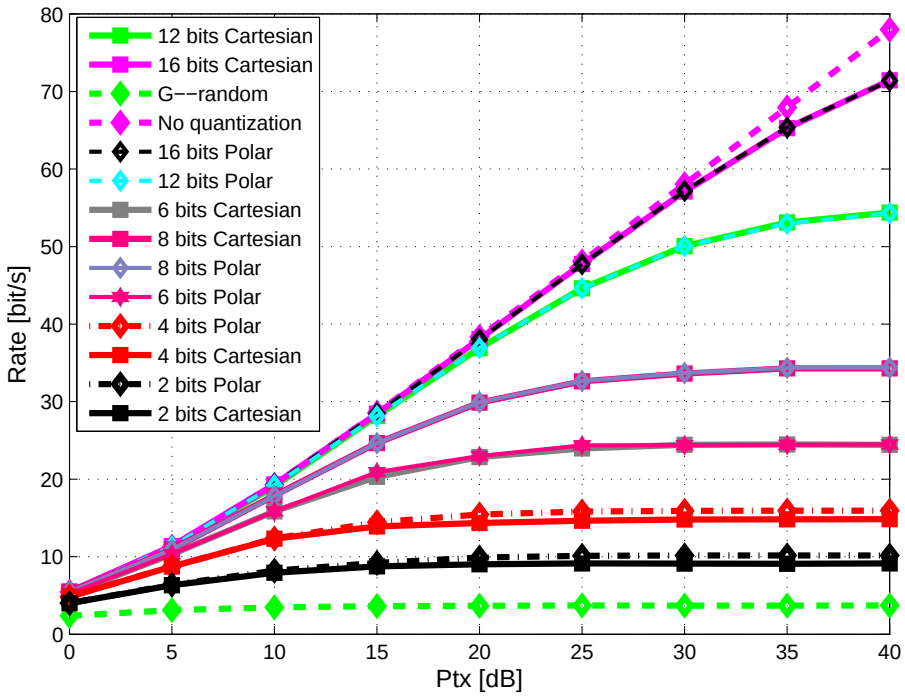

Figure 5.13: *Rate as a function of transmitt power.*

Chapter 6

Vector Quantization

Vector quantization is used to quantize whole vectors [5] instead of quantizing scalar values as discussed in the previous chapter. Vector quantization is more complex and computationally expensive than scalar quantization. However, it leads to smaller quantization errors. Vector quantization is a method of representing some values on the specific way. If there is a multi dimensional space (which is defined by the borders) with multi dimensional values, then we are choosing some multi dimensional value to represent all multi dimensional values in this space. It is not important that specific value is placed in the middle of the space. In this thesis we considered two algorithms which were helping us to find a borders and a specific value which represented all values in a huge space. These are the K-means and the LBG algorithms. The algorithms are quite similar and they only differ in the training phase. In the training phase, the quantizer is designed and in the actual quantization phase analog values are assigned to their corresponding quantization values. In the following sections we describe these two algorithms in detail.

6.1 K – means Algorithm

The K-means clustering is a method that is used in machine learning. The idea is to classify samples into K-clusters. Each cluster is represented by its centroid [5]. These centroids often correspond to the center of gravity and can be calculated according to the following equation:

$$\mathbf{c}_j = \frac{1}{N} \sum_{i=1}^{N} \mathbf{a}_{ij}, \tag{6.1}$$

where $\mathbf{c}_j \in \mathbb{C}^{(N\times 1)}$ represents the centroid of the j^{th} cluster and $\mathbf{a}_{ij} \in \mathbb{C}^{(N\times 1)}$ $(i = 1 \dots N)$ represents the i^{th} sample of the j^{th} cluster. The value N is the dimension of the space in which the classification is done and it represents the size of the vectors that are being classified.

When the samples are classified, they are sorted to the cluster with the centroid closest to that sample. In case of quantization, samples are represented as vectors. Any vector that has to be quantized is replaced by the centroid closest to it. In this way, all analog vectors are represented by one of K discrete vectors.

In order to be able to use the K-means algorithm, the quantizer has to be trained first. In the training

process, centroids of the quantizer are determined. The number of centroids (clusters) directly depends on the number of bits used for the quantization and is given by:

$$K = 2^{\mathrm{n_{bit}}}, \tag{6.2}$$

where $\mathrm{n_{bit}}$ represents the number of used quantization bits. The K-means quantizer training consists of three phases: initialization, classification and centroid recalculation. A principal schematic of the K-means training algorithm is shown in Figure 6.1.

In order to train the K-means algorithm, a training set is needed. The training set consists of vectors that are randomly generated with the same probability density function that is expected to be generating the actual values to be quantized later. Alternatively, a sample set of measured data can be used. In the training phase centroids are found based on the vectors in the training set. The exact number of vectors in the training set N_s is a design variable. In order for the training to be possible, the number of samples should satisfy the following condition: $N_s > 12K$. The training procedure is initialized by assigning the K-clusters with centroids. One option for initialization is to initially generate centroids as random vectors. However, as the probability distribution of the sample set is usually unknown, the problem of what kind of probability distribution for generating the centroids should be used arises. This problem can be easily overcome by taking centroids as a random subset of the sample set. In this case, centroids are initialized as particular vectors in the sample set. Therefore, the centroid initialization is done by randomly taking K samples out of the set of N_s samples. This concludes the initialization phase.

Once each cluster is assigned to its centroid, samples can be classified. Classification is done by sorting the samples to the corresponding clusters based on their distance to the centroids. Distance of a samples to centroid $\mathbf{c}$ is calculated as Euclidian distance :

$$d\left(\mathbf{a}, \mathbf{c}\right) = \sqrt{\sum_{i=1}^{N} \left(a_i - c_i\right)}, \tag{6.3}$$

where, $\mathbf{a}$ and $\mathbf{c}$ are N-space vectors representing sample and centroid respectivly:

$$\mathbf{a} = \left(a_1, ..., a_N\right) \tag{6.4}$$

$$\mathbf{c} = \left(c_1, ..., c_N\right). \tag{6.5}$$

Each sample is classified to the cluster having the centroid with the smallest Euclidian distance to that sample. In the next step, the cluster centroids are recalculated as the mean values of all the samples classified to the given cluster. In iteration step k this is done according to:

$$c^{[k]} = \frac{1}{N_j} \sum_{a_{ij} \in K_j} a_{ij}, \tag{6.6}$$

where $c^{[k]}$ stands for the new recalculated centroid, N_j is the number of samples classified into cluster j. The steps of classification and centroid recalculation are repeated until the algorithm converges. Convergence can be determined by looking at the movement of recalculated centroids over algorithm iterations. the centroid movement can be calculated by using the equation for Euclidian distance:

$$\Delta {c_j}^{[k]} = d\left(c_j[k], c_j[k-1]\right). \tag{6.7}$$

Therefore, the K-means training procedure can be stopped once the following condition is fulfilled:

$$\Delta c_j \leq \epsilon, \quad \forall j = 1 \dots K, \tag{6.8}$$

where the stopping criterium ϵ is a design parameter.

An alternative to this stopping criteria is to look at the sum of all centroid movements and use the following criterium:

$$\sum_{j=1}^{K} ||c_j|| \leq K\epsilon. \tag{6.9}$$

In our realizations this criterion has been used. Simulations show that there is no big difference between the two stopping criteria.

Figure 6.2 illustrates the training procedure for vectors in $\mathbb{R}^2$ and for $K = 4$. The upper plot shows the sample set and initial centroids randomly selected from the set. Subsequent plots show how the centroids move until they are placed into right positions and 4 clusters are formed. These final positions define separation lines that are borders between different clusters. Once the training phase is over and the K centroids have been determined, the quantizer can be used for channel matrix quantization. We can quantize complex values $h_{lr} = Re\{h_{lr}\} + Im\{h_{lr}\}$, where $i = (1, ..., N_r)$ and $r = (1, ..., N_t)$ in the matrix $\mathbf{H}_{mk}$. These complex values are quantized by calculating the distance from the actual value to each of the K centroids:

$$d_j = ||h_{lr} - c_j|| . \tag{6.10}$$

Then, the analog value is replaced by the centroid with the smallest distance d_k, where:

$$d_k = \text{argmin}(d_j). \tag{6.11}$$

An alternative to quantizing each element of the matrix $\mathbf{H}_{mk}$ is to use the K-means quantization algorithm to quantize the whole matrix $\mathbf{H}_{mk}$. This would certainly result in quantization with smaller error. However, a large number of centroids and a very large quantization space would be necessary for a quantization that is accurate enough. This becomes computationally too expensive. If we want to quantize a 2×4 matrix with 8 bits, then there is altogether 16 real values to be quantized and every real value should be quantized with half of bit. With this number of bits it is possible to calculate the centroids (256) in resonable time (the training phase lasts for one day). But if we quantize with 32 bits, which means for the same matrix 2 bits per real value, then the number of centroids is too large: 2^{32}, and it is practicaly not possible to compute the centroids. Memory requirements in the training phase are extremly big and the training could last for months. Table 6.3 gives the dependance of the centroid number on the number of quantization bits for the whole matrix and for a single matrix value.

As in the previous chapter, we focus on the achievable rate for different values of the design parameters that can be adjusted in the training procedure. The results are also compared to the result of scalar quantization. Simulation results are given in section 6.3.

6.2 LBG – Algorithm

The Linde-Buzo-Gray (LGB) algorithm is a special version of the K-means algorithm [5]. Quantizers based on the LBG and K-means algorithm only differ in the training phase. The LBG algorithm uses a slightly different training procedure than the one described in the previous section. Its training phase is considered to be better than the one for K-means as it converges faster and reduces the training time for the quantizer.

The training procedure of the LBG quantizer partly consists of the training procedure of the K-means quantizer described in the previous section. Figure 6.4 illustrates the training procedure for the LBG quantizer.

The training process starts by calculating a single centroid c_0 of the whole training sample set:

$$c_0 = \frac{1}{N_s} \sum_{i=1}^{N_s} a_{ij}, \tag{6.12}$$

where again N_s stands for the number of samples in the training set which is a design variable. This means that initially, the whole training sample set is considered to be a single cluster, and the centroid of that cluster is calculated in order to represent the whole cluster.

In the centroid splitting phase, this single centroid gives rise to two new centroids by adding random values:

$$c_1 = c_0 + \epsilon_1, \tag{6.13}$$

$$c_2 = c_0 + \epsilon_2, \tag{6.14}$$

where ϵ_1 and ϵ_2 represent random variables and c_1 and c_2 are the newly created centroids. This means that the sample set should be split into two clusters. Each of the clusters will be represented by a centroid. However, the exact position of the cluster centroids is not known. Therefore, their values are initialized by random values, according to (6.14) and then the real values are determined in an iterative procedure.

The LBG training procedure continues in the same way as for the K-means described before. Centroids c_1 and c_2 are fitted to the available sample set through the iterative process of classification and centroid recalculation. This procedure is stopped once the movement of the centroids is smaller than a given threshold (same as in the case of K-means training). Once this procedure terminates, each centroid is again splitted by adding two random numbers. This means that in each splitting step, the number of centroids (and thus the number of clusters) is doubled. The whole algorithm is repeated until the number of centroids becomes equal to the desired number.

The LBG quantizer has three design parameters that need to be determined during its design. Like in the case of K-means, these are the size of the training set and the minimal centroid movement. However, this quantizer has one additional design variable. This is the size of the random variables that are added in the centroid splitting phase ϵ. It is clear that the size should not be too big or too small as it could result in clusters with no members from the training set and certain numerical problems during training. Therefore, this value should be carefully chosen.

Figure 6.5 illustrates the LBG training procedure in $\mathbb{R}^2$ and for $K = 4$. The upper plot shows the training initialization in which a single centroid for the whole sample set is found. In the second plot, this centroid is splitted into two new centroids that move in order to split the sample set into two clusters. In the

next step each of the two centroids splits again and the new centroids are then moved to their right places. The calculated centroids now divide the space in $\mathbb{R}^2$ into 4 regions (clusters). Once the training phase is over, the quantization of the channel matrix can be done in the same way as for the K-means quantizer. The same discussion on quantizing the whole matrix or some of its elements also holds here.

6.3 Matlab Simulations for K–means and LBG Scalar Quantizations

Quantizers based on both the K-means and LBG algorithms are implemented and tested in Matlab. We first analyze and compare the training part of the algorithm for both cases as this is their main difference. A set of training experiments for both K-means and LBG quantizer were made. All the experiments were done with an identical sample set that consists of 10000 two dimensional vectors that were randomly generated. Eight bit quantizers were considered. Trainings were performed for different sizes of the basic sample sets. In cases when sample sets with less then 10000 samples were used, the original sample set was truncated in order to obtain a smaller set. In order to make an estimate of the computational load of the training phases of both algorithms, the number of Frobenius norm calculations during the training procedure had been recorded. This number corresponds to the number of iterations in the training procedure.

Figure 6.6 shows the recorded calculation number for different sizes of the training set and different values of the stopping criterium. From this figure, we can see that the LBG algorithm is faster than K-means. In case of the sample set with 1000 samples, the LBG training phase has almost 3 times less iterations. This illustrates the main advantage of the LBG which is its faster convergence that results in a training procedure that is less computationally expensive.

In order to give quantitative guidance for optimal choice of the stopping criterium and the size of the training set, one has to look at both the computational burden of the training phase and the quantization error of the quantizer. Choosing these two parameters has to be guided by a trade off between the accuracy of the quantizer and reasonable training time. To this end, we analyzed the mean quantization error of both K-means and LBG, algorithms with 8 bits obtained for different values of the stopping criterium and training set sizes. In these simulations, two–dimensional vectors and training space were considered. In the experiment complex matrices of size $(4\text{x}2)$ were quantized. In each experiment, 1000 such random matrices were generated and the mean quantization error was calculated by using the Frobenius norm. Figure 6.7 shows the result for the K-means and LBG algorithms, respectively. We can notice that in average the LBG quantizer leads to slightly smaller mean quantization errors. From figures 6.6 to 6.9 we can calculate that the reasonable choice for the stopping criterium would be in the range of $[10^{-3}, 10^{-2}]$, the optimal choice for the size of the training sample set is in the range $[4000, 8000]$. Design values should be chosen from these sets depending on the need to have short training time. In case the computational burden of the training stage is not critical , the stopping criterium should be chosen as 10^{-3} or less and the size of the training sample set of 8000 or more.

Another design decision that has to be made in the case of LBG algorithm is the size of the random variables used in the centroid spitting phase of the LBG training algorithm. This value has to be selected carefully in order to have the convergence of the algorithm. In our simulations we used the value between 0.001 and 0.01 depending on the number of samples in the training set and the number of centroids to be calculated. If the sample set is relatively small and the number of centroids to be calculated is high (typical example $N_s = 4000, K = 256$), then this value should be small (typical 0.001) as it is reasonable to expect

that the centroids will be close to each other. On the other hand, for cases when the sample set is relatively large compared to the number of centroids (typical $N_s = 10000, K = 256$) bigger values should be used (typical 0.1 or 0.01) in order to have fast convergence.

In order to compare the K-means and the LBG algorithms to the Cartesian and Polar quantizers considered before, several experiments were made. In these experiments, a system with 3 transmitters, 3 receivers, and having four antennas at each transmitter and two antennas at each receiver was considered. The signal to noise ratio was fixed to 20 dB. In the experiment, 1000 randomly generated channel matrices were used to calculate the average rate for K-means, LBG algorithms, Cartesian and Polar quantizers. Both the K-means and LBG quantizer were used for quantizing each element in the channel matrix instead of quantizing the whole matrix. Results for quantizers from two to eight bits were obtained and plotted. The K-means and LBG algorithm were trained by using the stopping criterium of 10^{-3} and the training sample set of 10 000. In the training procedure for the LBG, the amplitude of a random variable used for centroid splitting was set 0.001. Figure 6.10 shows the obtained results. We can see that except for the case of two bits for which the rate obtained with the K-means, LBG and the Cartesian quantizer give the same rate, K-means and LBG lead to higher rates. At higher bits we can see that the performance of the LBG quantizer determine. Another interesting aspect to note is a very good performance of the Cartesian quantizer for low bit numbers. However, as expected this quantizer has the poorest performance for higher bit quantization.

In addition to the rate, we look at the quantization error. Figure 6.11 shows the obtained results. As can be seen, the quantization error remains smaller for the K-means and the LBG quantizers for all bit numbers. It is interesting to note that the polar quantizer exhibits big errors for the two bits case which results in its poor performance in case low bit quantization is used. Figure 6.12 shows the dependance of the mean rate from quantization for all four quantizers. As expected, high rates correspond to small quantization errors and vice versa.

In addition, we investigated the performance of K-means and LBG quantizers in cases when the whole channel matrix is quantized. The biggest challenge in this quantization is that the training phase is extremly computationally expensive and time consuming. We consider quantization with 8, 12 and 16 bits. For the training phase sample sets of 10000, 100000, and 400000 were used. A stopping criterium of 0.01 was used and the random variable amplitude used for centroid splitting in LBG was 0.01. Again the scenario with two transmit and two receive antennas was used and the transmitted signal to noise ratio was fixed to 20 dB. Averaging over 10000 randomly generated channel matrices were used to calculate the achievable rate and quantization error. Obtained results are shown in Figure 6.13 and 6.17. This demands huge sampling sets and makes the training phase, but also the quantization process, extremely computationally complex and lengthy.

6.4 Traffic Calculation in Backhaul Network

It would be interesting to calculate the total number of bits per secound for the system that we considered in the previous section, and based on this calculation to suggest a compromise solution. If we consider a system with 4 transmitters and 4 receivers with 4 and 2 antennas respectively as we had in the previous chapters, then the channel matrix has the dimension (2×4) and it consists of 8 complex numbers and therefore 16 real values. If each of these real numbers is quantized with 4 bits, it means that we need 64 bits/channel matrix. Since the signal transmitted from the transmitter goes through multiple paths (multipath) due to

wireless transmission, at the receiver side there may be multiple taps. Let us assume that there are 10 taps at the receiver side. This means that there will be 10×64 bits/channel matrix, which means having 640 bits/physical channel matrix. The calculated number is valid for a single transmitter and a single receiver. Since in our example we have 4 transmitters and 4 receivers, the calculated number has to be multiplied by 16 and this means that in the cooperative group there will be 16×640bits/channel matrix=10240 bits/coop. group. This is valid for a single communication pass and the same procedure has to be constantly repeated in order to have normal communication. Therefore the measurements have to be repeated very often. If we take a millisecond updating, we get that the bit number that is going through the backhaul network is 10240bits/coop. group×1000=10.2 Mbits/s. If the transmit power of 28 dB is chosen, as we can see from Figure 6.17, for LBG and Polar quantization, the achievable rate is 50 bit/s/Hz. This means that there will be 1Gbit through the backhaul network in the maximum bandwidth LTE case.

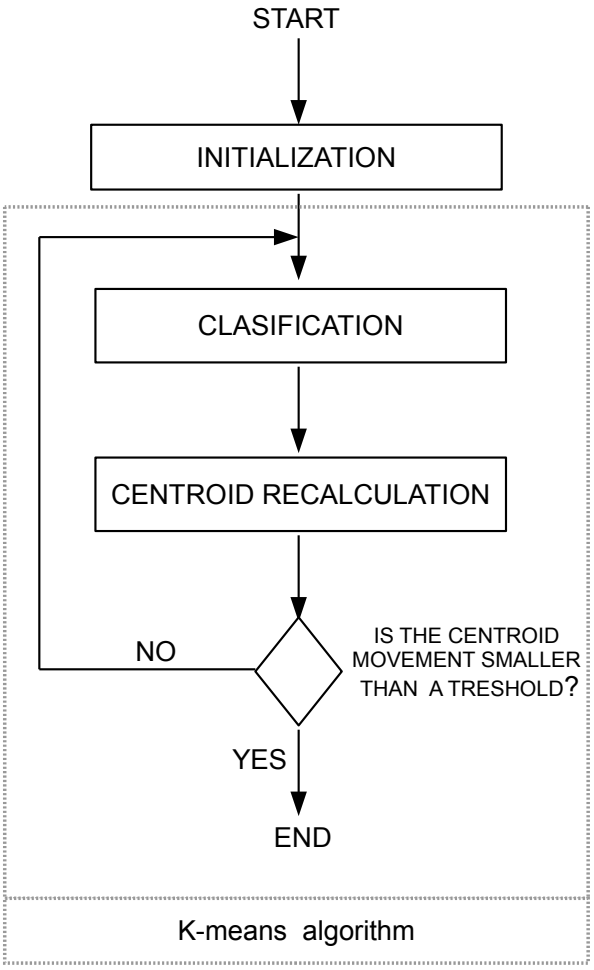

Figure 6.1: *Principal schematic of the K-means training procedure.*

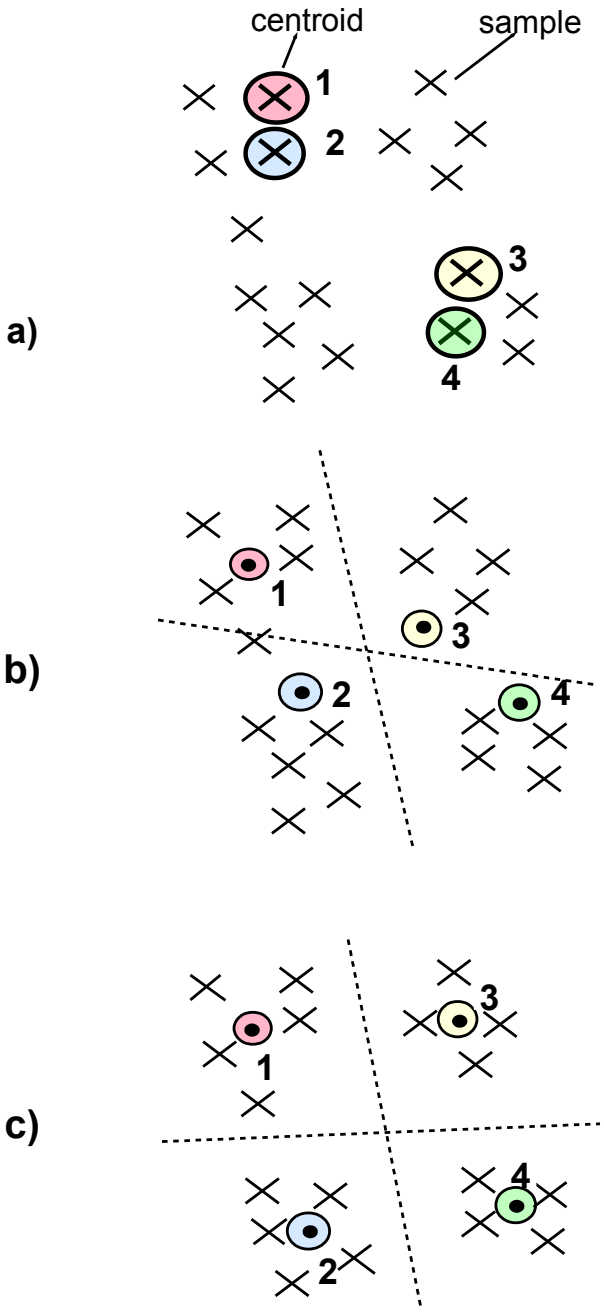

Figure 6.2: *Graphical schematic of the K-means training procedure, initialization phase (a), classification phase (b), recalculation phase (c).*

Number of bits	Bits/real value	Number of centroids
4	0.25	16
8	0.5	256
12	0.75	4096
16	1	65536
32	2	4.2950e+009
64	4	1.8447e+019
128	8	3.4028e+038

Figure 6.3: *Dependance of the centroid number on the number of quantization bits for 16 dimensions.*

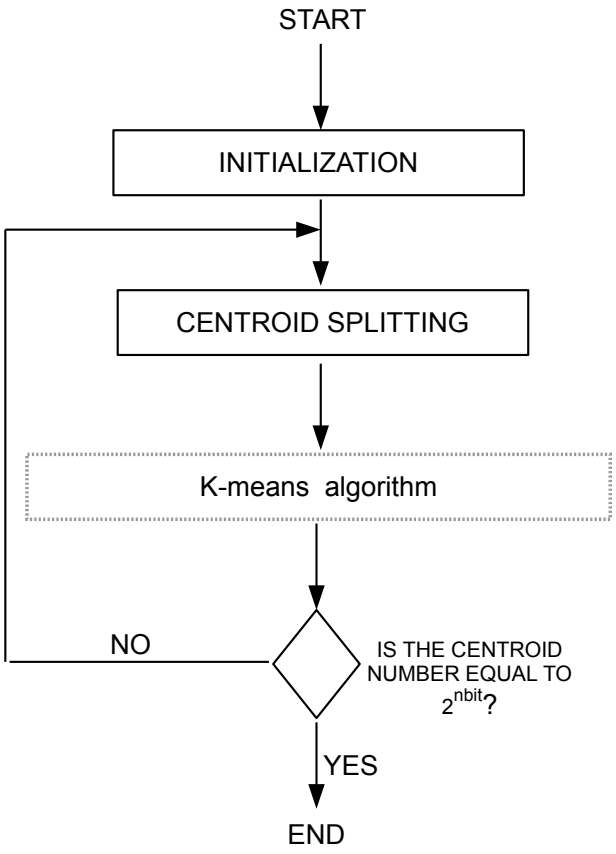

Figure 6.4: *Principal schematic of the LBG training procedure.*

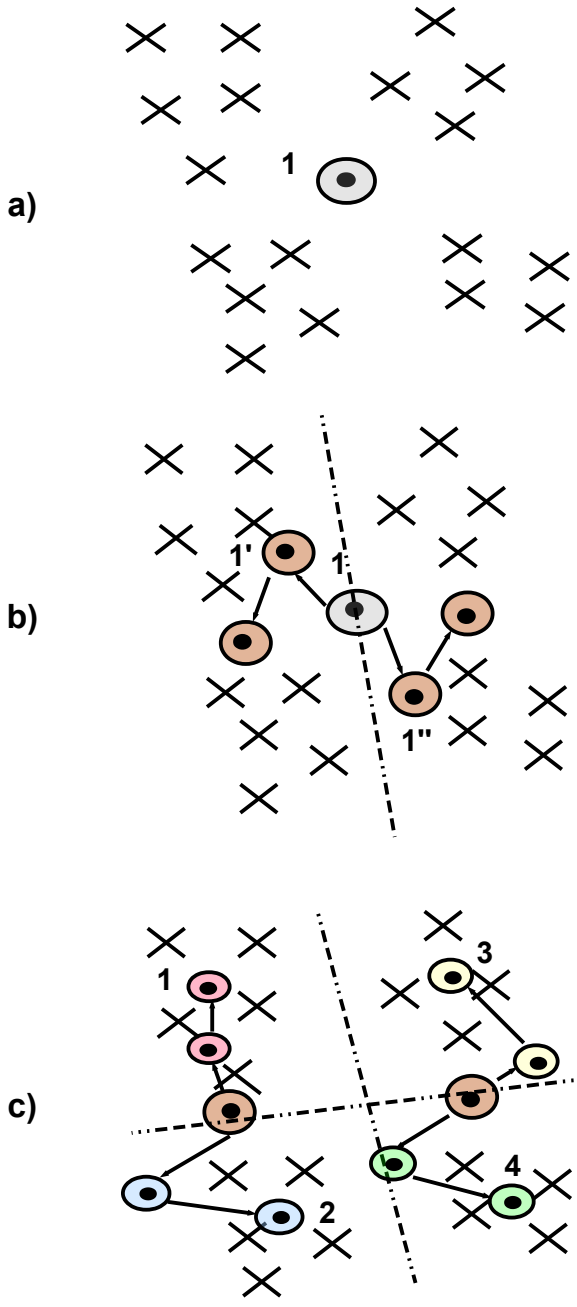

Figure 6.5: *Graphic expression of LBG algorithm, initialization phase (a), classification phase (b), recalculation phase (c).*

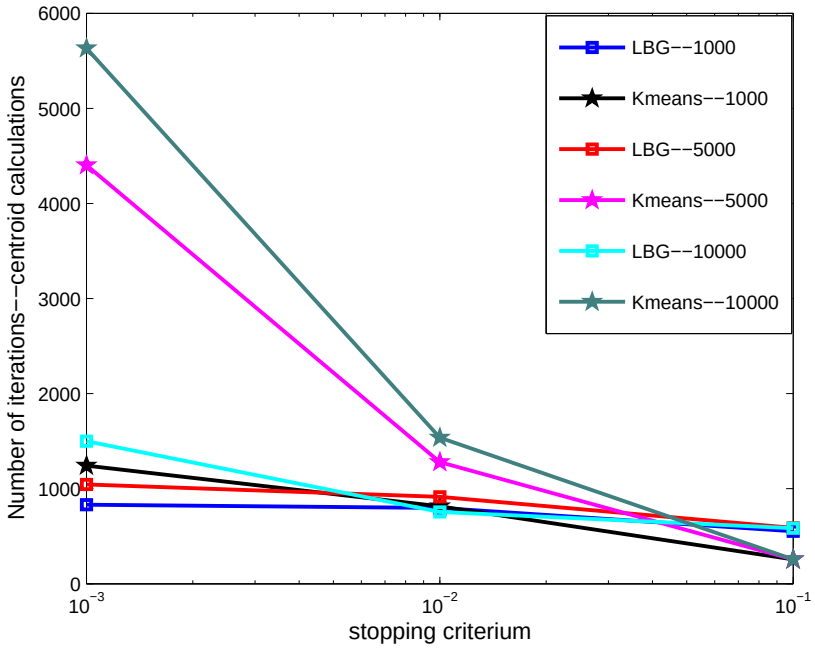

Figure 6.6: *Number of centroid calculations for K-means and LBG quantizers.*

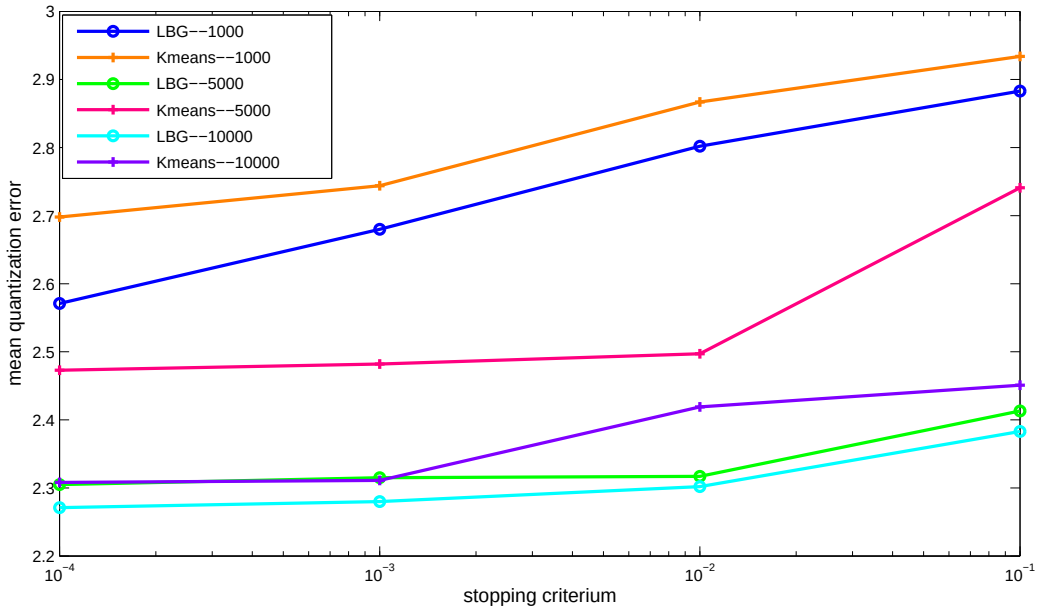

Figure 6.7: *Mean quantization error for the LBG and K-means quantizers for different values of stopping criterium and different sizes of the training sample set with eight bits.*

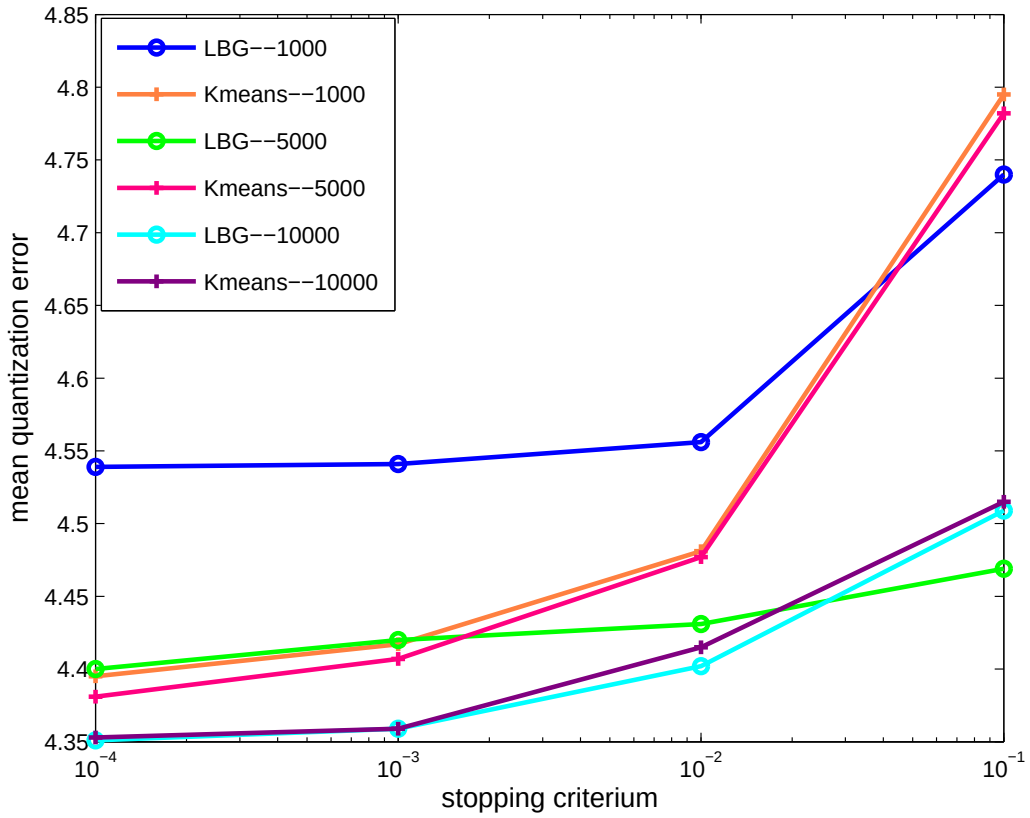

Figure 6.8: *Mean quantization error for the LBG and K-means quantizers for different values of stopping criterium and different sizes of the training sample set with six bits.*

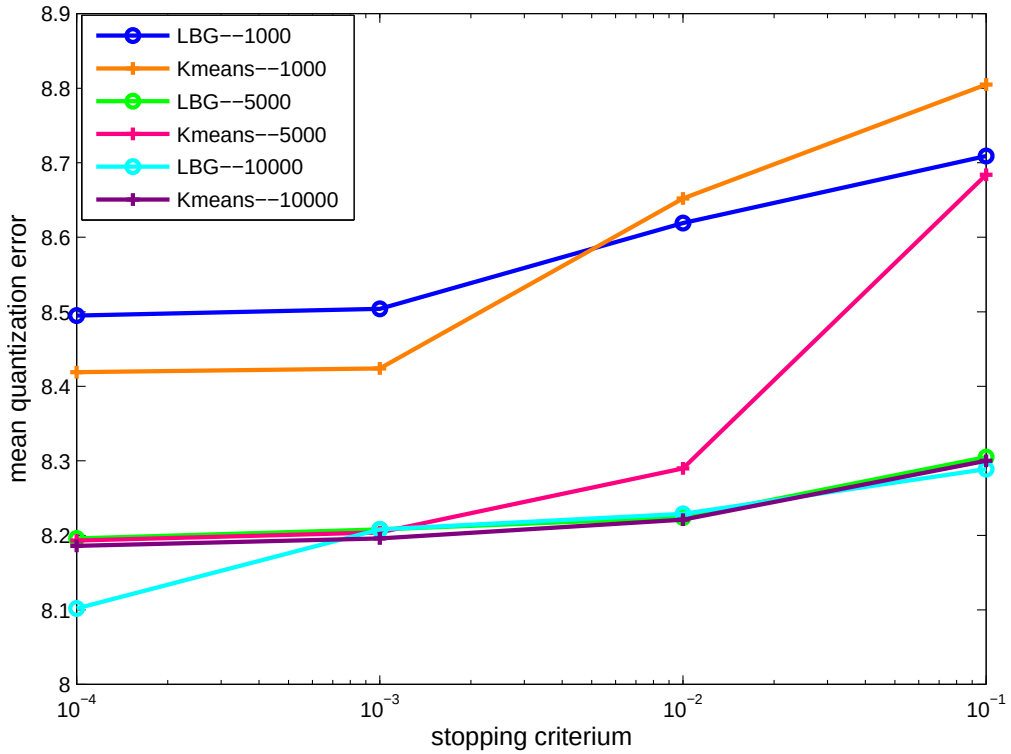

Figure 6.9: *Mean quantization error for the LBG and K-means quantizers for different values of stopping criterium and different sizes of the training sample set with four bits.*

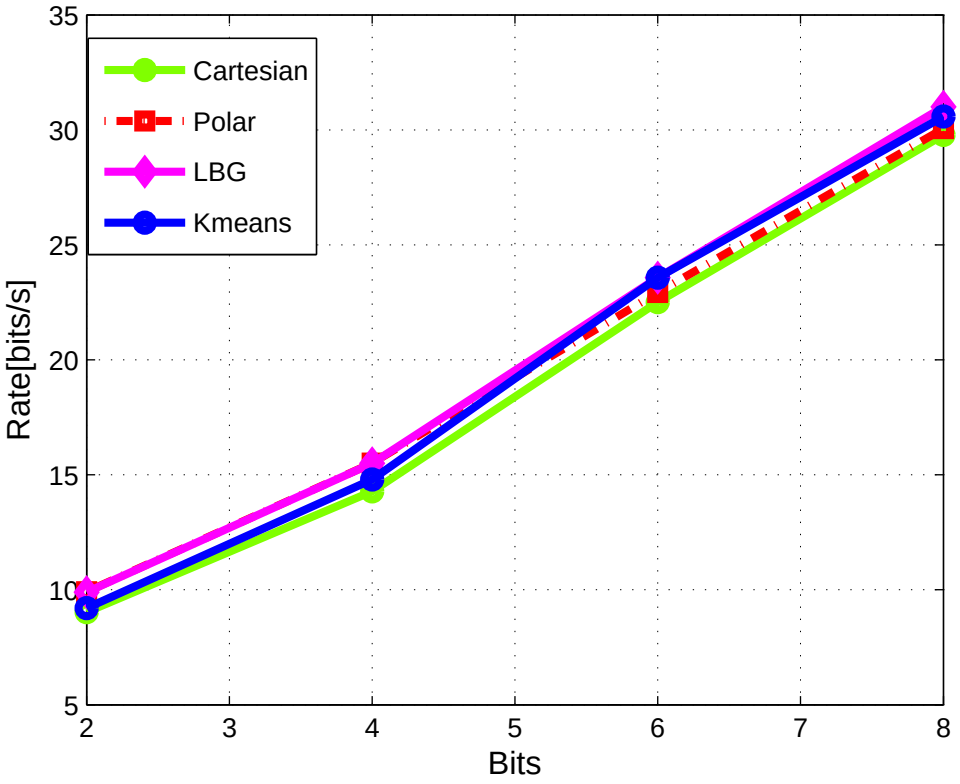

Figure 6.10: *Mean rate different quantizers for different values of quantization bits.*

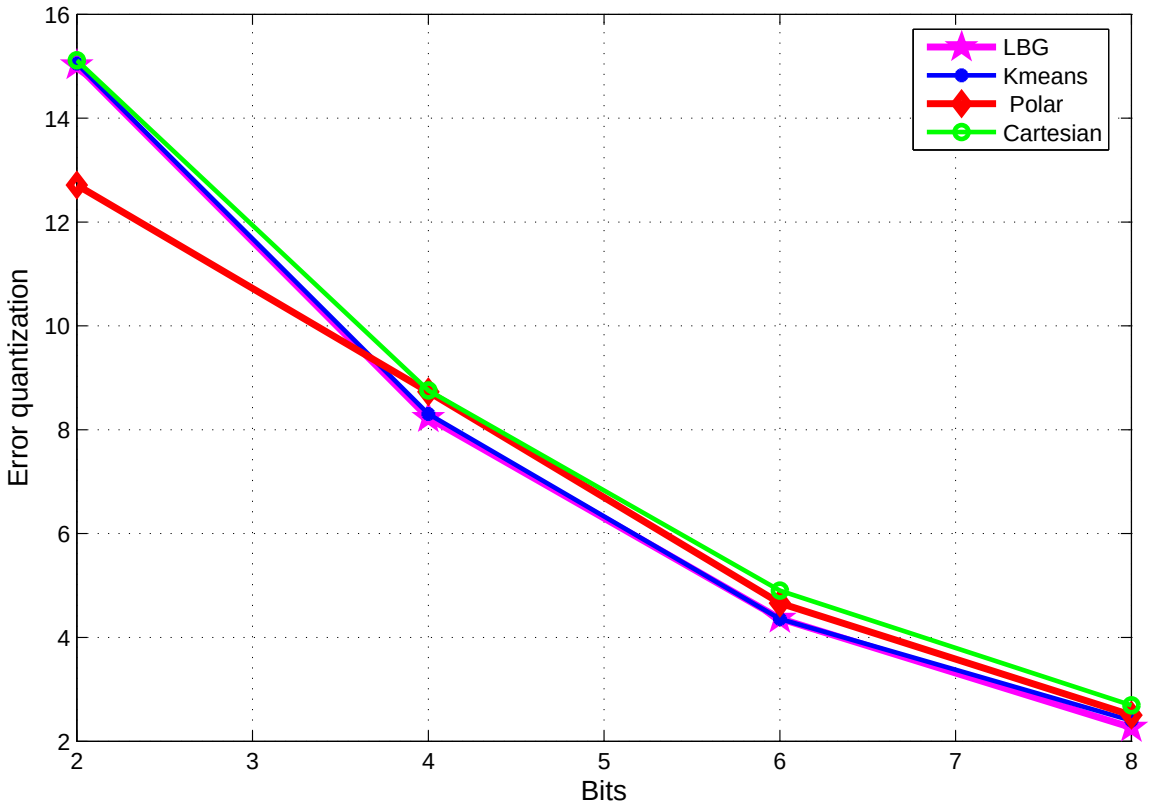

Figure 6.11: *Quantization error of different quantizers for different values of quantization bit numbers.*

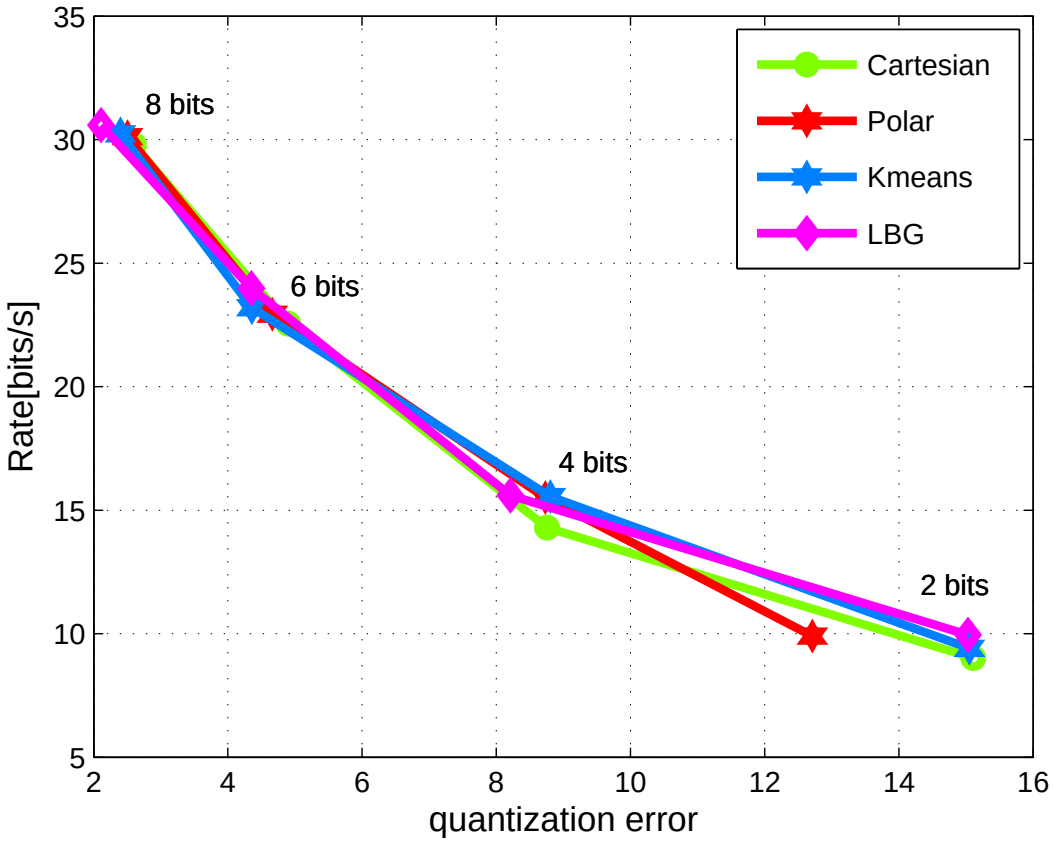

Figure 6.12: *Relation between the quantization error and rate for different quantizers.*

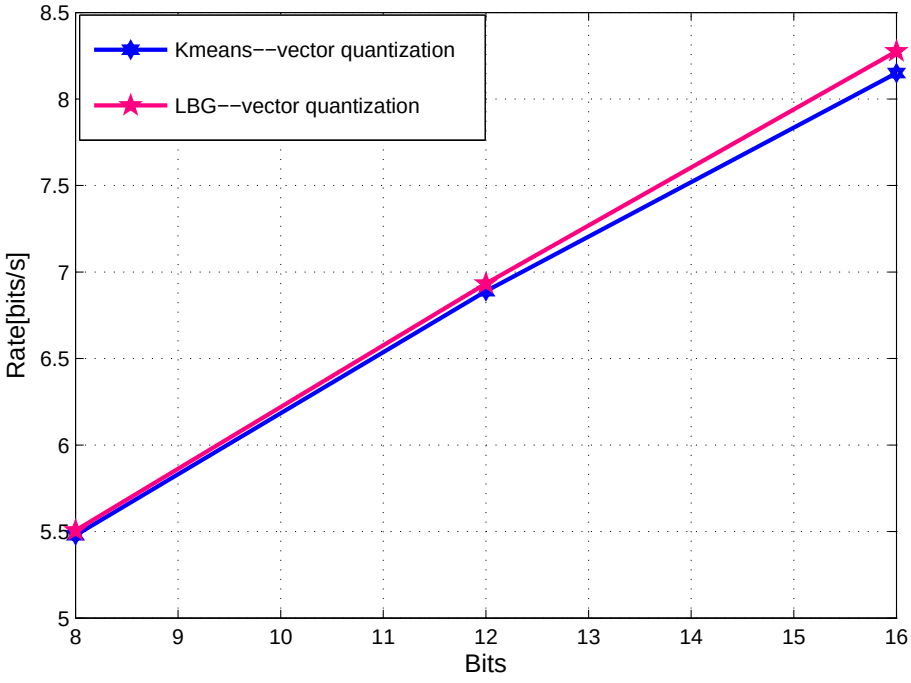

Figure 6.13: *Rate for whole vector.*

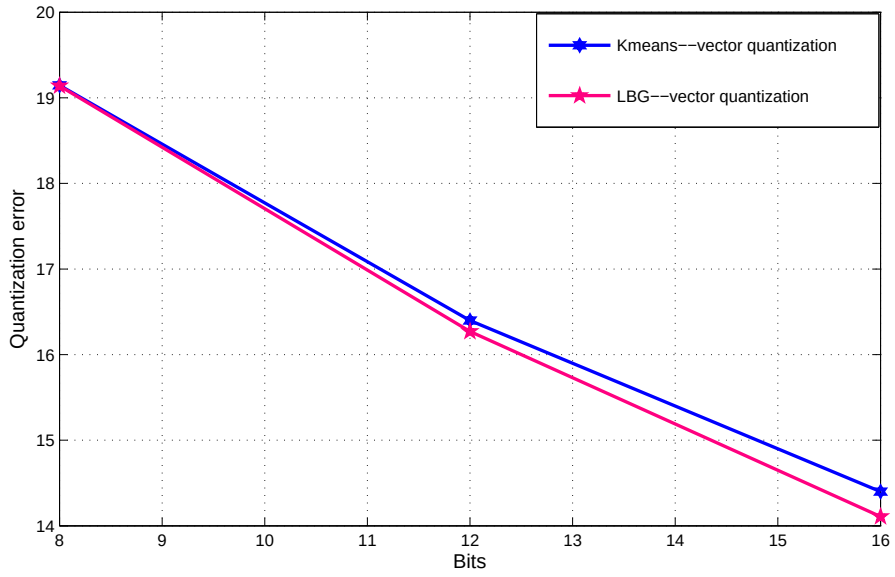

Figure 6.14: *Error quantization for whole vector.*

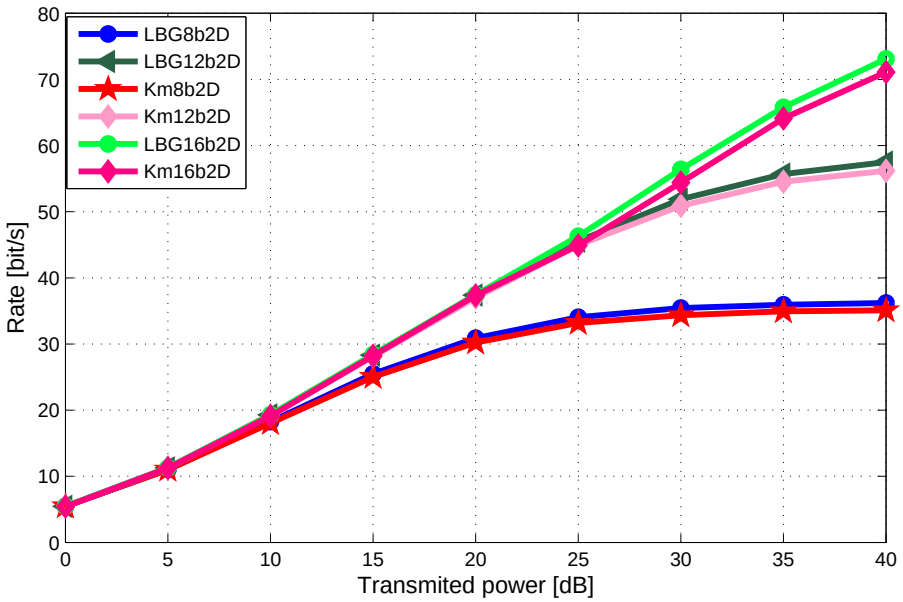

Figure 6.15: *Rate as a function of transmit power.*

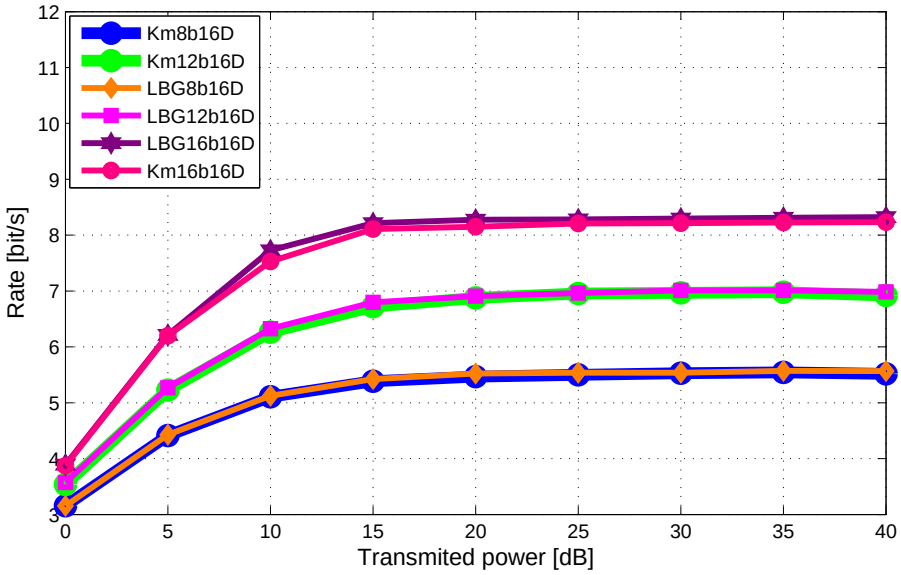

Figure 6.16: *Rate as a function of transmitted power - whole matrix.*

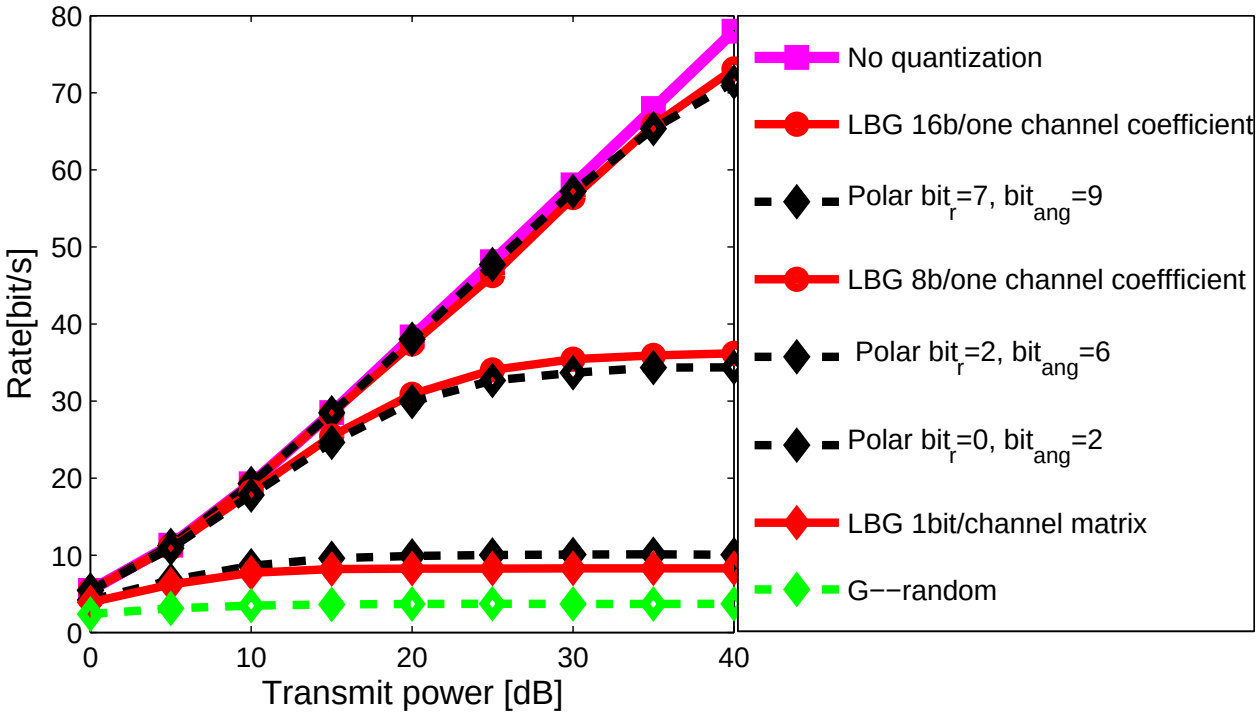

Figure 6.17: *Rate as a function of transmitted power - whole matrix.*

Chapter 7

Conclusion and Outlook

The goal of this thesis project was to consider different ways of channel matrix quantization and its impact on the performance of the zero-forcing interference cancellation method in cooperative cellular networks. In our analysis we disregarded the error due to channel matrix estimation and only focused on the quantization and its influence on the network performance. Our analysis based on simulations showed that if the quantization is properly designed, network performance is very similar to the case when the channel matrix is taken with infinite precision. On the other hand, even very coarse quantization with big quantization error gives better results compared to the case when the elements of the precoding matrix are taken as random variables. This finding illustrates that studying the quantization and its effects on the interference cancellation algorithm performance can be useful in further improving the overall performance of cellular networks.

We studied both scalar and vector quantization of the channel matrix. In the case of scalar quantization each matrix element is separately quantized. We considered the cases of Cartesian and Polar quantization. The difference between these two ways of quantizing complex numbers is that Cartesian and Polar coordinate representations are quantized. For both cases uniform distribution of the values is assumed. Based on Matlab simulations for normally distributed channel coefficients, we calculated optimal values for tuning these two quantiziers. Matlab simulations show that the quantization based on Polar representation leads to smaller quantization error and subsequently to higher achievable rate. This result can be explained by the fact that both the real and imaginary parts of the entries in the channel response matrix are Rayleigh distributed and therefore an error is made when these values are quantized by using uniformly distributed quantization levels. On the other hand the phase of the entries is uniformly distributed and there is no error when quantizing the phase. In addition, we studied vector quantization. In vector quantization, the whole matrices and vectors, rather than their single elements are quantized. Vector quantization consists of two phases. These are the training phase and the actual quantization of the channel matrix. We studied two vector algorithms that only differ in their training phases. These are the K-means and the LBG algorithm. The latter is supposed to have shorter training phase as it converges faster. This was confirmed by Matlab simulations. For both algorithms, reasonable ranges for selecting decision constants in the training phase that give a good balance between the computational complexity of the training phase and the quantization error are determined. This was done based on extensive Matlab simulations. Vector quantizers were compared with scalar quantizers in the case when each complex number in the channel matrix is considered as a vector. The simulations show much lower quantization error and higher achievable rate for the vector

quantization than for the scalar quantization. In addition, several simulations were done for the case when the whole channel matrix is quantized. This quantization leads to a big computational complexity in the training phase if reasonably small quantization error is desired.

The multiplexing gain, that is very important for the high capacity systems, does not exist if the channel matrix is badly quantized. In all of the cases that we analized in this thesis the LBG algorithm for vector quantization and Polar quantization algorithm in 2D showed the best results. Therefore, in further developments the LBG algorithm and Polar should be used as it results in smaller quantization errors which becomes especially evident for quantization with a lot of bits. However, this quantizer requires much more memory and computational effort and time then the Polar quantizer. Therefore, in order to make the use of the LBG algorithm easier, the issues of high memory requirements and high computationally complexity should be solved in future. In general further improvements of vector quantization algorithms will result in more accurate quantization if the problems of high computational complexity of this algorithms can be resolved. In addition, our analysis and simulations were based on the scenario in which channel matrices are static which corresponds to the case that the users in the network are static. Therefore, it would be interesting to see how the different quantization techniques behave in the dynamic case.

Bibliography

[1] D. Tse and P. Viswanath, "Fundamentals of wireless communication", *Cambridge University Press*, 2005

[2] T. M. Cover and J. A. Thomas, "Elements of information theory", 1st edition, *Wiley interscience*, 1991

[3] S. Arya, D. M. Mount, "Algorithms for fast vector quantization", *Computer and Information Science, vol. 93, no 1.*, 1993, pp. 381-390

[4] M. Khun, R. Rolny, A. Wittneben, "The potential of restricted PHY cooperation for the downlink of LTE-advanced", *IEEE Vehicular Technology Conference, VTC Fall, San Francisco*, Sept. 2011

[5] L. Gersho and R. M. Gray, "Vector quantization and signal compression", *The Springer International Series in Engineering and Computer Science*, 1992

[6] L. Mirsky, "An introduction to linear algebra", *Dover Books on Mathematics*, 1990

[7] P. Croy, "LTE backhaul requirements: a reality check", *Aviat networks technical report*, available at: *www.aviatnetworks.com*

[8] Miroslav L. Dukic, "Principi telekomunikacija", *Akademska misao*, 2008

[9] Natasa Neskovic, "Radio tehnika", *Lecture notes, University of Belgrade*, 2010

[10] Aleksandar Neskovic, "Radio komunikacije", *Lecture notes, University of Belgrade*, 2009

[11] M Costa, "Writting on dirty paper", *IEEE Trans. Information Theory*, 1983

[12] Yuzhe Jin, "Capacity, Diversity and Multiplexing Gain for MIMO channels", June 2007

[13] David N.C.Tse, Pramod Viswanath and Lizhong Zheng, "Diversity–Multiplexing Tradeoff in Multiple Access Channels", March 2004

[14] Vladik Kreinovich and Gang Xiang, "Estimating Information Amount under Uncertainty: Algorithmic Solvability and Computational Complexity", *International Journal of General Systems*, February 2010

[15] E. Telatar, "Capacity of multi-antenna Gaussian channels", *European Transactions on Telecommunications*,1999