Diplomarbeit

MIMO Receivers Using Soft Information

ausgeführt zum Zwecke der Erlangung des akademischen Grades eines Diplom–Ingenieurs
unter der Leitung von

Ao.Univ.Prof. Dipl.-Ing. Dr.techn. Gerald Matz
Dipl.-Ing. Johannes Maurer

Institut für Nachrichtentechnik und Hochfrequenztechnik (E 389)

eingereicht an der Technischen Universität Wien
Fakultät für Elektrotechnik und Informationstechnik

von

Günter Reise
0026095
Hirschengasse 21/14, 1060 Wien

Wien, Oktober 2007

Kurzfassung

Drahtlose MIMO (Multiple input multiple output) Systeme verwenden mehrere Sende- und Empfangsantennen und erlauben eine Verbesserung der Zuverlässigkeit (diversity gain) und der Datenrate (multiplexing gain) der Übertragung. Diese Vorteile kommen jedoch Hand in Hand mit einem höheren Rechenaufwand für die Detektion, insbesondere bei Verwendung von Soft-Detektion, die im Vergleich zu Hard-Detektion ein besseres Detektionergebnis liefert und eine Grundvoraussetzung für iterative Empfängerstrukturen, so genannte Turbo-Empfänger, darstellt. In dieser Diplomarbeit gehen wir von einem Sphere-Decoder für die MIMO Soft-Detektion aus und präsentieren Modifikationen, die einen flexiblen Abtausch von Rechenaufwand auf der einen Seite und Detektionsleistung bzw. Diversität auf der anderen Seite erlauben. Der flexible Abtausch zwischen Rechenaufwand und Detektionsleistung wird dabei durch Beschneiden der LLRs erreicht, das in den Sphere-Decoder einbezogen wird, und durch Vorgabe des Maximalwertes für die LLRs gesteuert werden kann. Für den flexiblen Abtausch zwischen Rechenaufwand und Diversität wird ein partieller Entzerrer zusammen mit einem fehlangepassten ML-Detektor verwendet, wobei ein einziger Parameter, der den Grad der Vorentzerrung beschreibt, den Abtausch bestimmt. Zusätzlich zu diesen beiden Modifikationen wird eine untere Schranke für Teilmetriken im Spere-Decoder präsentiert, die eine weitere Komplexitätsreduktion ohne Verlust an Detektionsleistung erlaubt. Zum Abschluss dieser Diplomarbeit werden Simulationsergebnisse für die vorgestellten Detektionsalgorithmen präsentiert, die eine Beurteilung ihrer Leistungsfähigkeit erlauben.

Abstract

Multiple input multiple output (MIMO) wireless systems that use multiple transmit and receive antennas allow a gain in reliability (diversity gain) and capacity (multiplexing gain). Unfortunately, these advantages come at the cost of higher computational complexity, particularly when using soft detection (SD). SD features improved performance compared to hard detection and is a prerequisite for iterative receiver structures (turbo receivers). In this diploma thesis we consider the sphere decoder for soft-output MIMO detection and present modifications that allow flexible trade-offs between computational complexity on the one hand and performance and diversity on the other hand. For the trade-off between complexity and performance we incorporate LLR clipping into the sphere decoding algorithm that allows to adjust the trade-off via the clipping level. For the trade-off between complexity and diversity we use a partial equalizer followed by a mismatched ML detector where the trade-off is adjusted using a single parameter governing the amount of pre-equalization. Additionally to these two modifications we present a lower bound on partial sphere decoder metrics that allows to reduce the complexity of the detection using the sphere decoder without performance degradation. The thesis is rounded up by simulations that evaluate the performance of the presented detection schemes.

Danksagungen

Mein Dank gilt meinen Betreuern Gerald Matz und Johannes Maurer, die mich bei der Erstellung dieser Diplomarbeit ausgezeichnet unterstützten und bei der Erarbeitung des Themas und auch bei allen anderen Fragen zu Matlab und LaTeX immer ein offenes Ohr für mich hatten.

Darüber hinaus danke ich meinen Kollegen, Freunden und meiner Familie für die Unterstützung in jeglicher denkbaren Form. Besonderer Dank gilt meinen Eltern Günter und Hermine Reise für ihre mir Zeit meines Lebens entgegengebrachte Geduld, ihr Verständnis und ihr Vertrauen.

Contents

Chapter 1

Introduction

1.1 Motivation

1.1.1 Spectrally Efficient Wireless Systems

The conversion of the modern society into an information society, in which the creation and distribution of information pose a significant economic, political, and cultural activity, leads to a high demand for information and for powerful communication systems for the exchange of this information. The information transmitted is gained from a multitude of sources, such as audio, video as well as data and text. In addition, increasing mobility of the users requests mobile communication systems that allow to access information everywhere and anytime. The strongly increasing amount of data transmitted using wireless communication technologies leads to a shortage of communication resources. This is especially true for the limited resource of frequency bandwidth. The increased demand for high transmission capacity has to be coped by increasing the spectral efficiency, in other words by more advantageous utilisation of existing frequency bands and channel conditions.

1.1.2 Fading and Diversity

The transmission of data using wireless systems is impaired by signal fading and interference. Fading is a consequence of multipath propagation that occurs due to the presence of reflectors and scatterers (e.g. houses, mountains, . . .) in the environment surrounding the transmitter and the receiver. The environment provides multiple paths over which a transmitted signal can travel. This results in a superposition of multiple copies of the transmitted signal at the receiver, each different in attenuation, delay and phase shift. The different characteristics of the paths can result in either constructive or destructive interference that lead to an amplification or an attenuation of the signal power at the receiver. Strong destructive interference and therefore a strong attenuation of the signal, called deep fade, may result in a temporary failure of communication due to a decrease in the signal-to-noise ration (SNR). In mobile communications the channel is permanently changing and deep fades can occur anytime. While in additive white Gaussian noise (AWGN) channels the bit error rate (BER) decreases exponentially with increasing signal-to-noise ratio the probability of deep fades dominates the BER in Rayleigh fading channels which causes the BER to decrease only linearly with increasing SNR [1].

Diversity techniques can be applied to reduce the probability of deep fades and therefore enhance the link reliability. These techniques are based on the concept of providing replicas of the same information signal through independently fading channels to the receiver. The probability all these signal components will fade simultaneously is reduced considerably. This reduction is called *diversity gain*.

There are several ways in which the receiver can be provided with multiple replicas of the same signal [2]. Frequency diversity can be used in case of frequency-selective fading. Then the signal can be transferred utilizing several carriers (e.g. OFDM), where the separation between successive carriers has to be larger or equal than the coherence bandwidth, or spread over a larger frequency band (e.g. spread spectrum).

In the case of time-selective fading temporal diversity can be used, where the signal is transferred at different points in time. The separation between successive time slots has to be at least the coherence time of the channel. Temporal diversity is usually exploited via interleaving, forward error correction coding (FEC), and automatic repeat request (ARQ). Other rarely used ways of exploiting diversity are angle diversity and polarisation diversity.

A more recent method for achieving diversity that will be dealt with throughout this thesis, is spatial (or antenna) diversity. It is employed based on transmitting or receiving information signals from multiple antennas. The antennas have to be spaced by more than the coherence distance, which is the minimum spatial separation of antennas for independent fading and depends on the wavelength and angular spread of the multiple paths arriving at or departing from an antenna array. We distinguish between *receive diversity*, i.e. the use of multiple antennas at the receiver, and *transmit diversity*, i.e. the use of multiple antennas at the transmitter [3].

For transmit diversity two different cases are possible. Transmit diversity in the case where the transmitter has knowledge of the channel state information (CSI) involves transmission such that the signals sent from the individual antennas arrive in phase at the receive antenna. If the CSI is not known at the transmitter, transmit diversity requires more sophisticated methods such as space-time coding which uses coding across antennas and time. The idea behind these methods is to send the information with different preprocessing from different antennas such that the receiver can combine these signals to obtain diversity. Transmit and receive diversity are both similar and different in many ways. While receive diversity, assuming an appropriate receiver, needs solely multiple independently fading antennas and is independent of the used coding and modulation scheme, transmit diversity needs special coding and modulation in order to be effective. Also, receive diversity provides array gain, whereas transmit diversity does not provide array gain when the channel is unknown at the transmitter [1].

1.1.3 MIMO and Spatial Multiplexing

Employing multiple antennas at the transmit and the receive side allows to obtain spatial multiplexing [3]. The idea behind spatial multiplexing is that the use of multiple antennas at both sides of the link together with rich scattering in the propagation environment opens up multiple transmission links within the same frequency band to yield an increase in capacity, called *multiplexing gain*. It grows linear in the number of antennas and is very attractive since it comes at no extra bandwidth or power consumption.

Multiple input multiple output (MIMO) wireless systems use multiple antennas at both link ends and are therefore able to provide improved link reliability (diversity gain) and increased spectral efficiency (multiplexing gain) [3]. Unfortunately the gain in link reliability and transmission rate come at the cost of higher computational complexity. This complexity is particularly high when using soft detection for MIMO, which features improved performance compared to hard detection and is a prerequisite for iterative receiver structures (turbo receivers). It is a major issue of current investigation and this diploma thesis in particular to reduce this complexity while still retaining full diversity and multiplexing gain.

1.2 Issues Addressed in this Diploma Thesis

This thesis deals with detection algorithms for MIMO soft detection and starts in Chapter 2 with a short introduction to the basics of MIMO systems followed by an explanation of the MIMO system model used throughout this thesis and hard and soft-detection for MIMO. A sphere detector for soft output detection is reviewed in Chapter 3 followed by refinements and modifications to reduce the computational complexity of this detector in Chapter 4. Chapter 5 provides numerical simulations of a 4×4 MIMO system, presents their results and closes this thesis with an interpretation of the results.

Chapter 2

MIMO Basics

Multiple input multiple output (MIMO) wireless systems are systems that use both multiple transmit and receive antennas. This can lead to a combination of transmit and receive diversity. Assuming M_T transmit and M_R receive antennas, a maximum of $M_\mathrm{T}M_\mathrm{R}$ links is available and therefore, if all of these links fade independently and appropriate diversity combining is employed, the diversity order is $M_\mathrm{T}M_\mathrm{R}$th [3]. The multiple transmit and receive antennas additionally allow to exploit spatial multiplexing which leads to an increase in capacity, such that the data rate grows linearly in the number of receive antennas [3].

MIMO systems provide diversity gain and multiplexing gain in rich scattering environments. These gains in reliability and transmission rate do not need additional bandwidth or power but come at the cost of higher computational complexity and at the cost of increased hardware effort due to the use of multiple antennas.

2.1 MIMO System Model

In this diploma theses we use a MIMO transmission system for single-user communication and consider a point-to-point link using spatial multiplexing. The transmission system that consists of several functional blocks is depicted in Figure 2.1 and described in the following.

The source data constitute the input of the transmission system. We assume that these source data have already been source coded to reduce redundancy and to represent the data in an efficient way. The source data is channel coded by the *channel encoder* that introduces redundancy in a controlled manner to allow error-detection and error-correction. In a next step, the *interleaver* (Π), the coded data is reordered to protect the transmission against burst errors. In the mapper the coded data is mapped to transmit symbols. The sequence of transmit symbols is divided in the *demultiplexer* into sub-sequences for the transmission from multiple transmit antennas. In the *signal generators* the sub-sequences are converted into transmit signals that are transmitted over the *channel* using multiple transmit signals. The channel corrupts the transmit signals, adds noise and produces the receive signals that are demodulated to pseudosymbols in the *demodulators*. The soft detector jointly detects the bits contained in the transmit symbol stream and as a result delivers soft information that is reordered to the original order in the *deinterleaver* (Π^{-1}). The *channel decoder* decodes the stream of soft-information into estimates for the source data.

Most of the blocks in the MIMO transmission system are similar to the single input single output (SISO) case and therefore well known. Hence we don't mention them in detail but take a closer look on the MIMO specific parts of the transmission system. Under the assumption of flat fading, the blocks encircled by the dashed line in Figure 2.1 can be represented by the simplified MIMO system illustrated in Figure 2.2. It consists of one transmitter equipped with M_T antennas and one receiver that has $M_\mathrm{R} \geq M_\mathrm{T}$ antennas and uses the receive signals of all antennas to jointly detect the transmit symbols. The transmitter simultaneously sends the symbols $s_1, \ldots, s_{M_\mathrm{T}}$ that are obtained by mapping blocks of length M_C of the bits b_i from the output of the interleaver to the elements of

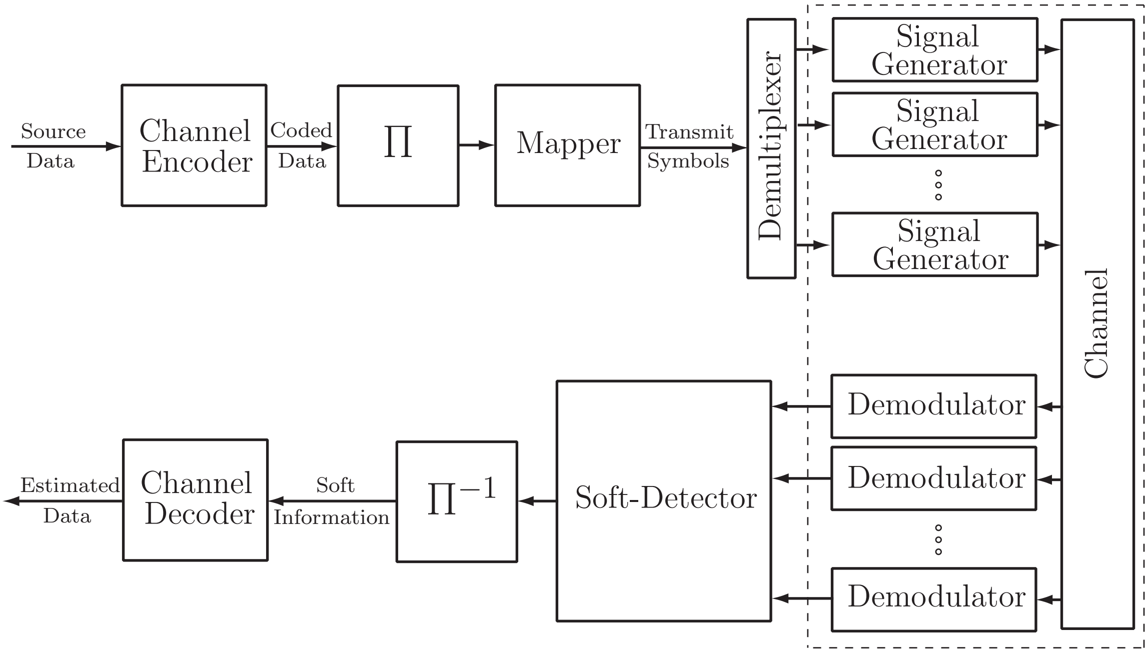


Figure 2.1: *Block diagram of a MIMO transmission system from source to estimated data.*

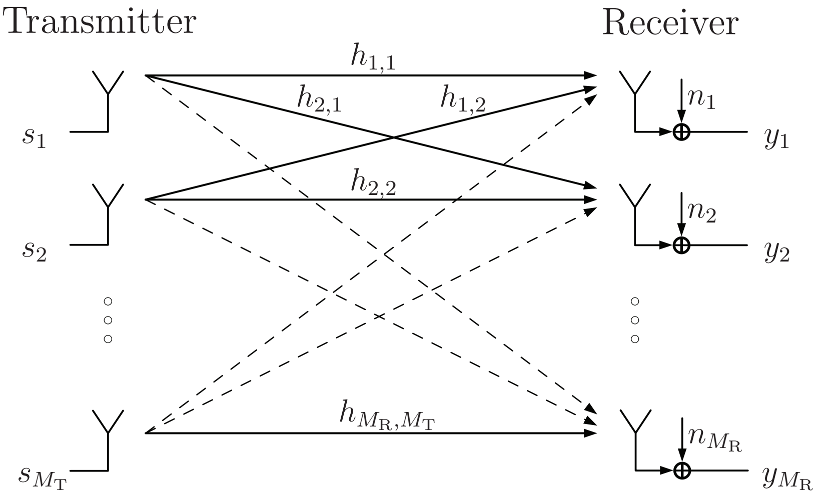


Figure 2.2: *MIMO system with M_T transmit and M_R receive antennas.*

the symbol alphabet $\mathcal{A}$ using the mapping function

$$s_m = s\left(b_{(m-1)M_\mathrm{C}+1}, \ldots, b_{mM_\mathrm{C}}\right), \qquad m = 1, \ldots, M_\mathrm{T}. \tag{2.1}$$

Here M_C is the number of bits per alphabet symbol and therefore the symbol alphabet $\mathcal{A}$ has cardinality $|\mathcal{A}| = 2^{M_\mathrm{C}}$. The symbols $s_1, \ldots, s_{M_\mathrm{T}}$ are sent within the same frequency band from the M_T antennas over the channel that is assumed to be flat-fading. It affects the sent symbols which is described by the complex fading channel coefficients $h_{i,j}$ with $i = 1, \ldots, M_\mathrm{R}$ and $j = 1, \ldots, M_\mathrm{T}$, where $h_{i,j}$ describes the link between the jth transmit antenna and the ith receive antenna. At the ith receive antenna we obtain a superposition of the transmitted symbols weighted by the channel coefficients $h_{i,j}$ plus Gaussian noise n_i,

$$y_i = \sum_{j=1}^{M_\mathrm{T}} h_{i,j} s_j + n_i.$$

After arranging $s_1, \ldots, s_{M_\mathrm{T}}$, $y_1, \ldots, y_{M_\mathrm{T}}$, and $n_1, \ldots, n_{M_\mathrm{T}}$ into the vectors $\mathbf{s}$, $\mathbf{y}$, and $\mathbf{n}$ respectively, and arranging the channel coefficients $h_{i,j}$ into a matrix $\mathbf{H}$, we obtain the vector-matrix description for the MIMO system,

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{M_\mathrm{R}} \end{bmatrix} = \begin{bmatrix} h_{1,1} & h_{1,2} & \ldots & h_{1,M_\mathrm{T}} \\ h_{2,1} & h_{2,2} & \ldots & h_{2,M_\mathrm{T}} \\ \vdots & \vdots & \ddots & \vdots \\ h_{M_\mathrm{R},1} & h_{M_\mathrm{R},2} & \ldots & h_{M_\mathrm{R},M_\mathrm{T}} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{M_\mathrm{T}} \end{bmatrix} + \begin{bmatrix} n_1 \\ n_2 \\ \vdots \\ n_{M_\mathrm{R}} \end{bmatrix},$$

or, in a more compact form,

$$\mathbf{y} = \mathbf{H}\mathbf{s} + \mathbf{n}. \tag{2.2}$$

The receive vector $\mathbf{y}$ is an $M_\mathrm{R} \times 1$ vector containing the received signals. The transmit vector $\mathbf{s}$ is a $M_\mathrm{T} \times 1$ vector of symbols whose entries are chosen from the symbol alphabet $\mathcal{A}$ and are independent and identically distributed (i.i.d.), i.e. $\mathrm{E}\{\mathbf{s}\mathbf{s}^H\} = \mathbf{I}$. The $M_\mathrm{R} \times M_\mathrm{T}$ channel matrix $\mathbf{H}$ contains the channel coefficients that are complex Gaussian with zero mean and variance σ_h^2 (Rayleigh fading), i.e. $h_{i,j} \sim \mathcal{CN}(0, \sigma_h^2)$ and $\mathrm{E}\{h_{i,j} h_{k,l}\} = \sigma_h^2 \delta_{i,k} \delta_{j,l}$, where $\delta_{i,k}$ and $\delta_{j,l}$ denote the Kronecker delta. In case of a large number of scatterers in the channel contributing to the signal at the receiver, application of the central limit theorem leads to this Gaussian model for the channel matrix coefficients [2]. We assume fast fading, which means that the channel matrix is constant during one use of the channel, i.e. for the duration of the transmission of one $\mathbf{s}$, and changes in an independent fashion from one use of the channel to another. Furthermore we assume that the channel matrix $\mathbf{H}$ is perfectly known at the receiver. The noise vector $\mathbf{n}$ is an $M_\mathrm{R} \times 1$ vector of i.i.d. complex Gaussian noise entries with zero mean and variance σ_n^2, i.e. $n_i \sim \mathcal{CN}(0, \sigma_n^2)$ and $\mathrm{E}\{\mathbf{n}\mathbf{n}^H\} = \sigma_n^2 \mathbf{I}$.

2.2 MIMO Hard-Detection

In the MIMO case the hard-detection problem is given by the calculation of an estimation $\hat{\mathbf{s}}$ for the transmitted symbol vector $\mathbf{s}$ as a function $\mathcal{D}(\mathbf{y})$ of the observed receive vector $\mathbf{y}$, i.e.

$$\hat{\mathbf{s}} = \mathcal{D}(\mathbf{y}). \tag{2.3}$$

The estimation for the bits $\hat{b}_j$ that have been mapped to the symbols in the symbol vector $\mathbf{s}$ can be obtained by demapping the estimate $\hat{\mathbf{s}}$. The optimum decision function is the one that maximizes the probability $P\{\hat{\mathbf{s}} = \mathbf{s}\}$ of a correct decision, [4]

$$\mathcal{D}_\mathrm{opt}(\mathbf{y}) = \arg\max_{\mathcal{D}(\mathbf{y})} P\{\hat{\mathbf{s}} = \mathbf{s}\} = \arg\max_{\mathcal{D}(\mathbf{y})} P\{\mathcal{D}(\mathbf{y}) = \mathbf{s}\}. \tag{2.4}$$

The probability of a correct decision (i.e. $\hat{\mathbf{s}} = \mathbf{s}$) is

$$P\{\hat{\mathbf{s}} = \mathbf{s}\} = \int_{\mathbb{C}^{M_\mathrm{R}}} P\{\hat{\mathbf{s}} = \mathbf{s}|\mathbf{y}\} f(\mathbf{y}) d\mathbf{y},$$

where $P\{\hat{\mathbf{s}} = \mathbf{s}|\mathbf{y}\}$ is the conditional probability that it has been decided for the correct MIMO symbol given the receive vector $\mathbf{y}$ and $f(\mathbf{y})$ is the probability distribution function (pdf) of the receive vector $\mathbf{y}$. Since $f(\mathbf{y})$ does not depend on the decision function $\mathcal{D}_\mathrm{opt}(\mathbf{y})$ and since moreover $f(\mathbf{y}) \geq 0$, $P\{\hat{\mathbf{s}} = \mathbf{s}\}$ is maximized if and only if the conditional probability $P\{\hat{\mathbf{s}} = \mathbf{s}|\mathbf{y}\}$ is maximized for each $\mathbf{y} \in \mathbb{C}^{M_\mathrm{R}}$. This conditional probability can be developed as

$$P\{\hat{\mathbf{s}} = \mathbf{s}|\mathbf{y}\} = P\{\mathbf{s} = \mathcal{D}(\mathbf{y})|\mathbf{y}\} = p\left(\mathcal{D}(\mathbf{y})|\mathbf{y}\right)$$

where $p\left(\mathcal{D}(\mathbf{y})|\mathbf{y}\right)$ is the conditional probability mass function (pmf) of the estimated MIMO transmit symbol $\hat{\mathbf{s}}$ given the received vector $\mathbf{y}$. Thus the optimum decision function $\mathcal{D}_\mathrm{opt}(\mathbf{y})$ in (2.4) is obtained by maximizing $p\left(\mathcal{D}(\mathbf{y})|\mathbf{y}\right)$ for fixed $\mathbf{y}$ with respect to the variable $\mathcal{D}_\mathrm{opt}(\mathbf{y})$. Writing $\mathcal{D}_\mathrm{opt}(\mathbf{y}) = \mathbf{s}$ and recalling that $\mathcal{D}_\mathrm{opt}(\mathbf{y}) \in \mathcal{A}^{M_\mathrm{T}}$, we obtain

$$\mathcal{D}_\mathrm{opt}(\mathbf{y}) = \arg\max_{\mathbf{s}\in\mathcal{A}^{M_\mathrm{T}}} p\left(\mathbf{s}|\mathbf{y}\right),$$

which means that the conditional probability of $\mathbf{s}$ given $\mathbf{y}$ has to be maximized. This conditional probability is know as the posterior probability of $\mathbf{s}$ and the decision rule is accordingly called the maximum a-posteriori (MAP) decision rule. To omit the decision function $\mathcal{D}_\mathrm{opt}(\mathbf{y})$ we rewrite the MAP decision rule as

$$\hat{\mathbf{s}}_\mathrm{MAP}(\mathbf{y}) = \arg\max_{\mathbf{s}\in\mathcal{A}^{M_\mathrm{T}}} p\left(\mathbf{s}|\mathbf{y}\right), \tag{2.5}$$

i.e. the MAP decision detects the symbol vector $\hat{\mathbf{s}}$ that is most probable for the given $\mathbf{y}$. The formulation of the MAP decision rule in (2.5) is inconvenient since the posterior probability $p\left(\mathbf{s}|\mathbf{y}\right)$ in not directly known. Using Bayes' rule we express the MAP decision rule in terms of known quantities

$$\begin{aligned}\hat{\mathbf{s}}_\mathrm{MAP}(\mathbf{y}) &= \arg\max_{\mathbf{s}\in\mathcal{A}^{M_\mathrm{T}}} p\left(\mathbf{s}|\mathbf{y}\right) \\ &= \arg\max_{\mathbf{s}\in\mathcal{A}^{M_\mathrm{T}}} \left\{ f\left(\mathbf{y}|\mathbf{s}\right) \frac{p(\mathbf{s})}{f(\mathbf{y})} \right\} \\ &= \arg\max_{\mathbf{s}\in\mathcal{A}^{M_\mathrm{T}}} \left\{ f\left(\mathbf{y}|\mathbf{s}\right) p(\mathbf{s}) \right\}, \end{aligned} \tag{2.6}$$

where $f\left(\mathbf{y}|\mathbf{s}\right)$ is the conditional pdf of the receive vector $\mathbf{y}$ given the transmitted signal vector $\mathbf{s}$ and $p(\mathbf{s})$ is the pmf of the transmit symbol vector $\mathbf{s}$. In case that all transmit vectors $\mathbf{s}$ are equally likely or this is assumed because of the distribution beeing unknown the pmf $p(\mathbf{s})$ is equal and constant for all $\mathbf{s}$, i.e. $p(\mathbf{s}) = \frac{1}{2^{M_\mathrm{C} M_\mathrm{T}}}$, and can therefore be omitted in (2.6). We obtain

$$\hat{\mathbf{s}}_\mathrm{ML}(\mathbf{y}) = \underset{\mathbf{s} \in \mathcal{A}^{M_\mathrm{T}}}{\arg\max} \left\{ f\left(\mathbf{y}|\mathbf{s}\right) \frac{1}{2^{M_\mathrm{C} M_\mathrm{T}}} \right\} = \underset{\mathbf{s} \in \mathcal{A}^{M_\mathrm{T}}}{\arg\max} \, f\left(\mathbf{y}|\mathbf{s}\right), \tag{2.7}$$

with the conditional pdf $f\left(\mathbf{y}|\mathbf{s}\right)$ that is called a likelihood function if it is viewed as a function of $\mathbf{s}$ with $\mathbf{y}$ fixed. Hence the decision rule is called the maximum likelihood (ML) decision rule. The detector using the MAP decision rules (2.5) is optimum in the sense that it yields the minimum MIMO symbol error probability. In case of equally likely symbols the ML decision rule (2.7) equals the MAP decision rule and is therefore optimal as well in that case. For the system model with additive white Gaussian noise (AWGN) and known channel (see Section 2.1) we obtain the likelihood function

$$f(\mathbf{y}|\mathbf{s}) = \frac{1}{(\pi \sigma_n^2)^{M_\mathrm{R}}} \, e^{-\frac{1}{\sigma_n^2} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2},$$

which leads to the ML decision rule

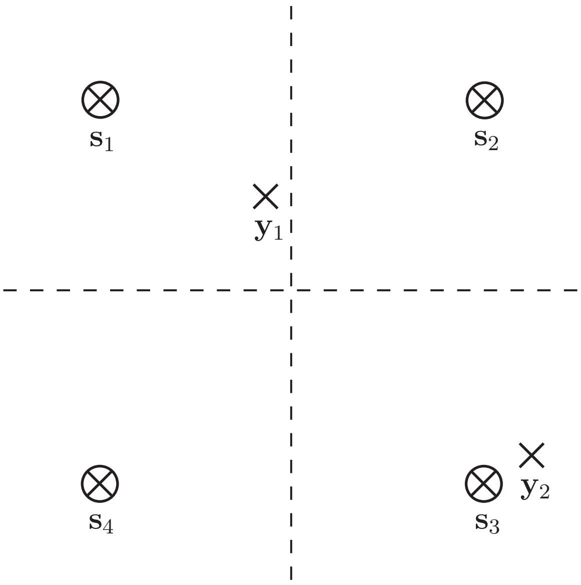
$$\hat{\mathbf{s}}_\mathrm{ML} = \underset{\mathbf{s} \in \mathcal{A}^{M_\mathrm{T}}}{\arg\min} \, \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 . \tag{2.8}$$

It can be seen that the hard-detection in case of additive white Gaussian noise and equally likely symbols simplifies to a minimum distance decision.

The hard-detection delivers hard decisions for the estimate $\hat{\mathbf{s}}$ of the symbol vector $\mathbf{s}$ and, after demapping $\hat{\mathbf{s}}$, for the estimate $\hat{b}_j$ of the corresponding bits. It does not deliver any reliability information that could be exploited by the channel decoder. Doing so, hard-detection omits a large part of the information contained in the receive vector $\mathbf{y}$.

2.3 MIMO Soft-Detection

Soft-detecion delivers, like the hard-detection covered in Section 2.2, estimates $\hat{b}_j$ for the bits b_j that are mapped to the transmit vector $\mathbf{s}$. Additionally it delivers information on



$$\mathbf{s}_1 = [s_1\ s_1]^T \leftrightarrow \mathbf{b}_1 = [0\ 0]^T$$
$$\mathbf{s}_2 = [s_2\ s_1]^T \leftrightarrow \mathbf{b}_2 = [1\ 0]^T$$
$$\mathbf{s}_3 = [s_1\ s_2]^T \leftrightarrow \mathbf{b}_3 = [0\ 1]^T$$
$$\mathbf{s}_4 = [s_2\ s_2]^T \leftrightarrow \mathbf{b}_4 = [1\ 1]^T$$

Figure 2.3: *Detection example for a 2×2 MIMO system using BPSK modulation.*

how reliable these estimates are. This supplementary information, termed soft information, can be exploited by an appropriate channel decoder to achieve a better decoding performance. Moreover the performance can be further improved using iterative receiver structures, so called turbo-receivers (e.g. in [5]).

We want to compare hard and soft-detecion by means of an example using a simple 2×2 MIMO system with BPSK modulation. Figure 2.3 shows the possible transmit symbols $\mathbf{s}_1, \ldots, \mathbf{s}_4$ (with the bit vectors $\mathbf{b}_1, \ldots, \mathbf{b}_4$ mapped to them, which will be denoted using the symbol "$\leftrightarrow$" in the following) and two possible receive vectors $\mathbf{y}_1, \mathbf{y}_2$ in the signal space. In the case of an AWGN channel with equally likely symbols s_1 and s_2 the optimal MAP decision rule (2.6) simplifies to the minimum distance decision (2.8),

$$\hat{\mathbf{s}}_{\text{MAP}} = \hat{\mathbf{s}}_{\text{ML}} = \underset{\mathbf{s} \in \mathcal{A}^{M_\text{T}}}{\arg\min} \|\mathbf{y} - \mathbf{H}\mathbf{s}\|^2 ,$$

i.e. the estimate $\hat{\mathbf{s}}$ is the symbol that is closest to the receive vector $\mathbf{y}$ in the sense of Euclidean distance. For that case the dashed lines in the Figure 2.3 mark the boundaries of the decision regions for the hard-decision case. Under these circumstances it is obvious that for the receive vectors $\mathbf{y}_1$ and $\mathbf{y}_2$ a detector using hard-detection would decide for

$\hat{\mathbf{s}}(\mathbf{y}_1) = \mathbf{s}_1 \leftrightarrow \mathbf{b}_1$ and $\hat{\mathbf{s}}(\mathbf{y}_2) = \mathbf{s}_3 \leftrightarrow \mathbf{b}_3$. This decision, however, omits the reliability information contained in the receive vectors and ignores the code structure. Assuming that low noise values are more likely than larger ones (e.g. for zero-mean Gaussian noise), the information on how reliable the decision is can be taken out of the distance between the receive vector $\mathbf{y}$ and the corresponding transmit symbol estimate $\hat{\mathbf{s}}$. In Figure 2.3 the decision for $\hat{\mathbf{s}}(\mathbf{y}_2) = \mathbf{s}_3$, for instance, is much more reliable than the decision for $\hat{\mathbf{s}}(\mathbf{y}_1) = \mathbf{s}_1$ because $\mathbf{y}_2$ is closer to $\mathbf{s}_3$ than $\mathbf{y}_1$ is to $\mathbf{s}_1$. Exploiting the reliability information together with the channel code structure underlying the bits mapped to the transmit symbols leads to a performance gain.

In the soft-detection case the information about the decision and its reliability are usually delivered jointly for every bit using log-likelihood ratios (LLRs). These are defined as

$$L\left(b_j|\mathbf{y}\right) = \ln \frac{P(b_j = 1|\mathbf{y})}{P(b_j = 0|\mathbf{y})}, \tag{2.9}$$

where $P(b_j = 1|\mathbf{y})$ and $P(b_j = 0|\mathbf{y})$ are the conditional probabilities for $b_j = 1$ and $b_j = 0$ respectively, conditioned on $\mathbf{y}$. The definition of the LLRs in (2.9) is quite clear. By putting the posterior probabilities for $b_j = 1$ and $b_j = 0$ into relation and taking the logarithm we get values in the interval $(-\infty, \infty)$ that allow to obtain the hard decision for a bit by calculating

$$\hat{b}_j(\mathbf{y}) = \begin{cases} 0 & \text{if} \quad L(b_j|\mathbf{y}) < 0 \\ 1 & \text{if} \quad L(b_j|\mathbf{y}) \geq 0. \end{cases} \tag{2.10}$$

The reliability information, the soft information, can be obtained by exploiting the modulus $|L(b_{j)}|$. The higher the modulus the higher the reliability of the bit decision.

For the example in Figure 2.3 the soft detector could deliver the LLRs $L(b_1|\mathbf{y}_1) = -5$ and $L(b_2|\mathbf{y}_1) = -15$. These values are given as an example and are therefore only descriptive and not exact but they represent the information contained in the receive vector $\mathbf{y}_1$. The hard decision for the bits $\mathbf{b}(\mathbf{y}_1) = [b_1(\mathbf{y}_1)\ b_2(\mathbf{y}_1)]^T$ can be derived from the sign according to (2.10) and results in $b_1(\mathbf{y}_1) = 0$ and $b_2(\mathbf{y}_1) = 0$. Looking at the moduli of the LLRs we see that the higher value $|L(b_2|\mathbf{y}_1)| = 15$ states a quite high reliability of the decision for the bit b_2 which reflects the fact that the receive vector $\mathbf{y}_1$ is quite close to the transmit symbol $\mathbf{s}_1$, where $b_1 = 0$, and far from the transmit symbols $\mathbf{s}_3$ and $\mathbf{s}_4$, where $b_1 = 1$, i.e where the bit is complementary. The smaller value of $|L(b_1|\mathbf{y}_1)| = 5$ on the other side reflects the fact that the receive vector $\mathbf{y}_1$ is close to the transmit symbol $\mathbf{s}_1$, where $b_1 = 0$, and far from the transmit symbols $\mathbf{s}_2$ and $\mathbf{s}_4$, where $b_1 = 1$, i.e where the bit is complementary. The difference in the LLR values for the two bits arises out of the fact that the distance between $\mathbf{y}_1$ and $\mathbf{s}_2$ is smaller than the distance between $\mathbf{y}_1$ and $\mathbf{s}_3$ and the decision is therefore less reliable.

The detector passes the LLRs on to the channel decoder that is capable of using the soft information for the decoding, e.g. channel decoders using the BCJR (Bahl, Cocke, Jelinek and Raviv) algorithm [6]. This allows to reach a performance gain compared to the hard-detection case.

Chapter 3

Soft-Output Sphere Detection

3.1 Log-Likelihood Ratios

Minimizing the probability of making an error detecting a bit means maximizing the a posteriori probability. Therefore the maximum a posteriori (MAP) detection is optimal in the sense of error probability. The a posteriori probability is usually expressed using log-likelihood ratios (LLRs).

Based on the MIMO system model presented in Section 2.1 the a-posteriori LLR of a bit b_j for $j = 1, \ldots, M_\mathrm{T} M_\mathrm{C}$ conditioned on the receive vector $\mathbf{y}$, is [5]

$$L\left(b_j|\mathbf{y}\right) = \ln \frac{P(b_j = 1|\mathbf{y})}{P(b_j = 0|\mathbf{y})}, \tag{3.1}$$

where $P(b_j = 1|\mathbf{y})$ and $P(b_j = 0|\mathbf{y})$ are the conditional probabilities for $b_j = 1$ and $b_j = 0$ respectively, conditioned on $\mathbf{y}$.

Using Bayes' theorem, we can write the soft output value (3.1) as

$$L\left(b_j|\mathbf{y}\right) = \ln \frac{p(\mathbf{y}|b_j = 1)\frac{P(b_j=1)}{p(\mathbf{y})}}{p(\mathbf{y}|b_j = 0)\frac{P(b_j=0)}{p(\mathbf{y})}} = \ln \frac{p(\mathbf{y}|b_j = 1)}{p(\mathbf{y}|b_j = 0)} + \ln \frac{P(b_j = 1)}{P(b_j = 0)}. \tag{3.2}$$

Here, $p(\mathbf{y}|b_j)$ is the conditional probability density function (pdf) of $\mathbf{y}$ conditioned on b_j, $p(\mathbf{y})$ is the pdf of $\mathbf{y}$ and $P(b_j = 1)$ and $P(b_j = 0)$ are the probabilities for $b_j = 1$ and

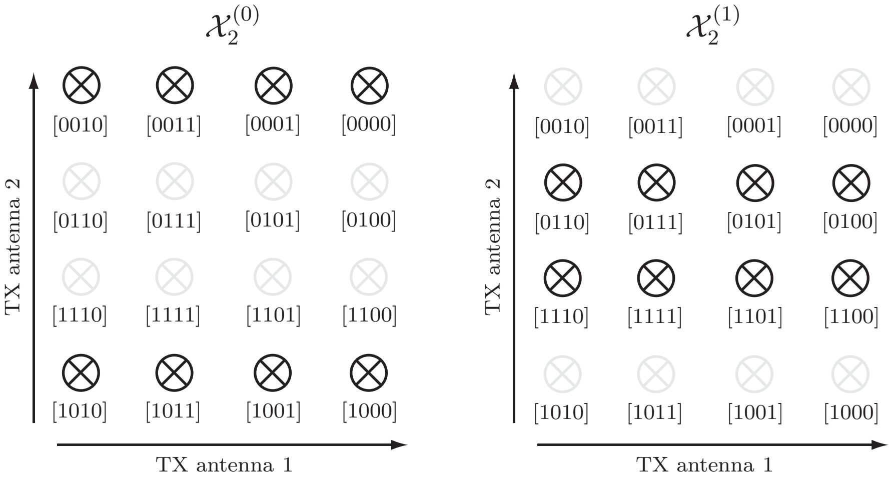


Figure 3.1: *Example for the sets $\mathcal{X}_2^{(0)}$ and $\mathcal{X}_2^{(1)}$ in a 2×2 MIMO system using 4-PSK modulation.*

$b_j = 0$ respectively. Transforming the conditional probability distributions $p(\mathbf{y}|b_j = 1)$ and $p(\mathbf{y}|b_j = 0)$ in (3.1) and assuming that both cases, $b_j = 0$ and $b_j = 1$, are equally likely, we obtain

$$L\left(b_j|\mathbf{y}\right) = \ln \frac{\sum\limits_{\mathbf{b}\in\mathcal{X}_j^{(1)}} p(\mathbf{y}|\mathbf{b})P(\mathbf{b})}{\sum\limits_{\mathbf{b}\in\mathcal{X}_j^{(0)}} p(\mathbf{y}|\mathbf{b})P(\mathbf{b})} = \ln \frac{\sum\limits_{\mathbf{b}\in\mathcal{X}_j^{(1)}} p(\mathbf{y}|\mathbf{b})}{\sum\limits_{\mathbf{b}\in\mathcal{X}_j^{(0)}} p(\mathbf{y}|\mathbf{b})}, \tag{3.3}$$

where $\mathcal{X}_j^{(0)}$ and $\mathcal{X}_j^{(1)}$ are disjoint sets of vector symbols that have the jth bit b_j of the transmit symbol $\mathbf{s}$ equal to 0 and 1, respectively, and $\mathbf{b}$ is the vector containing the bits b_j for $j = 1, \ldots, M_\mathrm{T}M_\mathrm{C}$. Figure 3.1 gives an example for such sets for the second bit b_2 in case of a 2×2 MIMO system using 4-PSK modulation. We furthermore assume that $\mathbf{y}$ is complex Gaussian distributed, i.e. [4]

$$p(\mathbf{y}|\mathbf{b}) = p\left(\mathbf{y}|\mathbf{s} = \mathbf{s}\left(\mathbf{b}\right)\right) = \frac{1}{(\pi\sigma_n^2)^{M_\mathrm{R}}}\, e^{-\frac{1}{\sigma_n^2}\|\mathbf{y}-\mathbf{H}\,\mathbf{s}(\mathbf{b})\|^2}. \tag{3.4}$$

As a next step we define the Jacobian logarithm [5]

$$\begin{aligned} \mathrm{jacln}(a_1, a_2) &\triangleq \ln(e^{a_1} + e^{a_2}) = \max(a_1, a_2) + \ln(1 + e^{-|a_1-a_2|}) \\ &= \max(a_1, a_2) + r(|a_1 - a_2|), \end{aligned} \tag{3.5}$$

where $r(|a_1 - a_2|)$ can be viewed as a refinement of the coarse approximation $\max(a_1, a_2)$. Simplifying the calculation of the logarithm by entirely omitting $r(|a_1 - a_2|)$ leads to the max-log approximation,

$$\ln(e^{a_1} + \ldots + e^{a_N}) = \ln\left(\sum_{n=1}^{N} e^{a_n}\right) \approx \max(a_1, \ldots, a_N), \tag{3.6}$$

which only retains the dominating term in (3.6) and allows the approximate calculation of the logarithm by means of a maximum search. Inserting (3.4) in (3.3) and using the max-log approximation (3.6) for the Jacobian logarithm (3.5), we obtain

$$\begin{aligned}
L\left(b_j|\mathbf{y}\right) &= \ln \frac{\sum\limits_{\mathbf{b}\in\mathcal{X}_j^{(1)}} p(\mathbf{y}|\mathbf{b})}{\sum\limits_{\mathbf{b}\in\mathcal{X}_j^{(0)}} p(\mathbf{y}|\mathbf{b})} = \ln \sum_{\mathbf{b}\in\mathcal{X}_j^{(1)}} p(\mathbf{y}|\mathbf{b}) - \ln \sum_{\mathbf{b}\in\mathcal{X}_j^{(0)}} p(\mathbf{y}|\mathbf{b}) \\
&= \ln \sum_{\mathbf{b}\in\mathcal{X}_j^{(1)}} \mathrm{e}^{-\frac{1}{\sigma_n^2}\|\mathbf{y}-\mathbf{Hs}\|^2} - \ln \sum_{\mathbf{b}\in\mathcal{X}_j^{(0)}} \mathrm{e}^{-\frac{1}{\sigma_n^2}\|\mathbf{y}-\mathbf{Hs}\|^2} \\
&= \max_{\mathbf{b}\in\mathcal{X}_j^{(1)}} \left\{-\frac{1}{\sigma_n^2}\left\|\mathbf{y}-\mathbf{Hs}\right\|^2\right\} - \max_{\mathbf{b}\in\mathcal{X}_j^{(0)}} \left\{-\frac{1}{\sigma_n^2}\left\|\mathbf{y}-\mathbf{Hs}\right\|^2\right\}.
\end{aligned}$$

Finally, after omitting the noise variance σ_n^2, we attain the max-log approximation for the calculation of the LLRs

$$L\left(b_j|\mathbf{y}\right) \approx \min_{\mathbf{b}\in\mathcal{X}_j^{(1)}} \left\|\mathbf{y}-\mathbf{Hs}\right\|^2 - \min_{\mathbf{b}\in\mathcal{X}_j^{(0)}} \left\|\mathbf{y}-\mathbf{Hs}\right\|^2. \tag{3.7}$$

The LLR $L\left(b_j|\mathbf{y}\right)$ has to be computed for all coded bits b_j. For each bit, one of the two minima in (3.7) is the value corresponding to the maximum likelihood (ML) solution

$$\lambda^{\mathrm{ML}} = \min_{\mathbf{s}\in\mathcal{A}^{M_\mathrm{T}}} \left\|\mathbf{y}-\mathbf{Hs}\right\|^2 = \left\|\mathbf{y}-\mathbf{H}\hat{\mathbf{s}}_{\mathrm{ML}}\right\|^2, \tag{3.8}$$

where

$$\mathcal{A}^{M_\mathrm{T}} = \mathcal{X}_j^{(0)} \bigcup \mathcal{X}_j^{(1)}.$$

Here, with a slight abuse of notation, we use $\mathcal{X}_j^{(0)}$ and $\mathcal{X}_j^{(1)}$ to denote sets of both the bit vectors $\mathbf{b}$ and the symbol vectors $\mathbf{s}$. The other minimum in (3.7) is given by

$$\lambda_j^{\overline{\mathrm{ML}}} = \min_{\mathbf{s}\in\mathcal{X}_j^{(\overline{b_j^{\mathrm{ML}}})}} \left\|\mathbf{y}-\mathbf{Hs}\right\|^2 \tag{3.9}$$

where $\overline{b_j^{\mathrm{ML}}}$ denotes the binary complement of the jth bit in the binary label of the ML solution $\hat{\mathbf{s}}_{\mathrm{ML}}$ and $\mathcal{X}_j^{(\overline{b_j^{\mathrm{ML}}})}$ is the set of vector symbols for which $b_j = \overline{b_j^{\mathrm{ML}}}$. Using (3.8) and (3.9) allows to write the max-log approximation (3.7) for the LLRs as

$$L\left(b_j|\mathbf{y}\right) \approx \begin{cases} \lambda^{\mathrm{ML}} - \lambda_j^{\overline{\mathrm{ML}}} & \text{if} \quad b_j^{\mathrm{ML}} = 0 \\ \lambda_j^{\overline{\mathrm{ML}}} - \lambda^{\mathrm{ML}} & \text{if} \quad b_j^{\mathrm{ML}} = 1. \end{cases} \tag{3.10}$$

This shows that the calculation of the LLRs narrows down to the calculation of λ^{ML} and the $\lambda_j^{\overline{\mathrm{ML}}}$ for $j = 1, \ldots, M_{\mathrm{C}}M_{\mathrm{T}}$. However, the calculation itself is computational very expensive as there have to be calculated the ML metric and $M_{\mathrm{C}}M_{\mathrm{T}}$ metrics of the counter hypothesis. The ML metric itself is the result of a minimum search over $|\mathcal{A}|^{M_{\mathrm{T}}} = 2^{M_{\mathrm{C}}M_{\mathrm{T}}}$ and every metric of the counter hypothesis itself is the result of the minimum search over $|\mathcal{X}_j^{(\overline{b_j^{\mathrm{ML}}})}| = 2^{M_{\mathrm{C}}M_{\mathrm{T}}-1}$ different symbol vectors $\mathbf{s}$.

3.2 Tree Search Problem

To reduce the computational complexity the sphere decoding algorithm [7] is used after transforming the search for (3.8) and (3.9) into tree-search problems.

We introduce the QR-decomposition of a general complex $n \times m$ matrix $\mathbf{A}$ [8]. If $n \geq m$ there is an $n \times m$ matrix $\mathbf{Q}$ with orthonormal columns ($\mathbf{Q}^H\mathbf{Q} = \mathbf{I}$) and an upper triangular $m \times m$ matrix $\mathbf{R}$ such that $\mathbf{A} = \mathbf{QR}$. If $m = n$, $\mathbf{Q}$ is unitary, i.e. $\mathbf{QQ}^{\mathrm{H}} = \mathbf{Q}^{\mathrm{H}}\mathbf{Q} = \mathbf{I}$. If in addition $\mathbf{A}$ is nonsingular, i.e. $\mathbf{Ax} = \mathbf{0}$ only for $\mathbf{x} = \mathbf{0}$, then $\mathbf{R}$ may be chosen so that all its diagonal entries are real-valued and positive, and in this case, the factors $\mathbf{Q}$ and $\mathbf{R}$ are both unique.

We use the QR-decomposition $\mathbf{H} = \mathbf{QR}$ of the channel matrix $\mathbf{H}$ and transform the input-output relation (2.2) into the modified input-output relation

$$\tilde{\mathbf{y}} \triangleq \mathbf{Q}^{\mathrm{H}}\mathbf{y} = \mathbf{Rs} + \mathbf{Q}^{\mathrm{H}}\mathbf{n} \tag{3.11}$$

by left-multiplying by $\mathbf{Q}^{\mathrm{H}}$. Note that $\mathbf{Q}^{\mathrm{H}}\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2\mathbf{I})$ since $\mathbf{Q}^{\mathrm{H}}\mathbf{Q} = \mathbf{I}$. This leads to the following equivalent formulations for (3.8) and (3.9):

$$\lambda^{\mathrm{ML}} = \min_{\mathbf{s}\in\mathcal{A}^{M_\mathrm{T}}} \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2 , \tag{3.12}$$

$$\lambda_j^{\overline{\mathrm{ML}}} = \min_{\mathbf{s}\in\mathcal{X}_j^{(\overline{b_j^{\mathrm{ML}}})}} \|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2 . \tag{3.13}$$

We define the partial symbol vectors (PSVs) $\mathbf{s}^{(i)} = [s_i, s_{i+1}, \ldots, s_{M_\mathrm{T}}]^T$ and arrange them in a tree. Figure 3.2 shows the corresponding tree structure with M_T levels, where every level corresponds to one transmit antenna, and nodes that are represented by the PSVs $\mathbf{s}^{(i)}$. The tree's root lies above level $i = M_\mathrm{T}$ and each path down via several nodes on the levels $i = M_\mathrm{T} - 1, \ldots, 2$ to a leaf on level $i = 1$ corresponds to a symbol vector $\mathbf{s} \in \mathcal{A}^{M_\mathrm{T}}$. The path drawn with bold lines, for instance, corresponds to the symbol vector $\mathbf{s} = [s^{(3)}, s^{(3)}, \ldots, s^{(2)}, s^{(1)}]$. The leaf with the smallest metric in $\mathcal{A}^{M_\mathrm{T}}$ and $\mathcal{X}_j^{(\overline{b_j^{\mathrm{ML}}})}$ corresponds to the solution of (3.12) and (3.13), respectively. Taking into account the upper triangular structure of $\mathbf{R}$ the calculation of the term $\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2$ simplifies to

$$\begin{aligned}
\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2 = &\underbrace{\left|\tilde{y}_{M_\mathrm{T}} - R_{M_\mathrm{T},M_\mathrm{T}} s_{M_\mathrm{T}}\right|^2}_{\left|e_{M_\mathrm{T}}\right|^2} \\
&+ \underbrace{\left|\tilde{y}_{M_\mathrm{T}-1} - R_{M_\mathrm{T}-1,M_\mathrm{T}-1} s_{M_\mathrm{T}-1} - R_{M_\mathrm{T},M_\mathrm{T}} s_{M_\mathrm{T}}\right|^2}_{\left|e_{M_\mathrm{T}-1}\right|^2} + \ldots \\
&\ldots + \underbrace{\left|\tilde{y}_1 - R_{1,1}s_1 - R_{1,2}s_2 - \ldots - R_{1,M_\mathrm{T}} s_{M_\mathrm{T}}\right|^2}_{|e_1|^2} = \sum_{i=1}^{M_\mathrm{T}} |e_i|^2 ,
\end{aligned}$$

where the metrics can be calculated recursively by summing up the distance increments (DIs)

$$|e_i|^2 = \left|\tilde{y}_i - \sum_{k=i}^{M_\mathrm{T}} R_{i,k} s_k\right|^2 \tag{3.14}$$

of all branches from the root ($i = M_\mathrm{T}$) to a leaf ($i = 1$) of the tree. The partial sums

$$d_i = \sum_{k=i}^{M_\mathrm{T}} |e_k|^2 = d_{i+1} + |e_i|^2 \quad \text{with} \quad d_{M_\mathrm{T}+1} = 0, \tag{3.15}$$

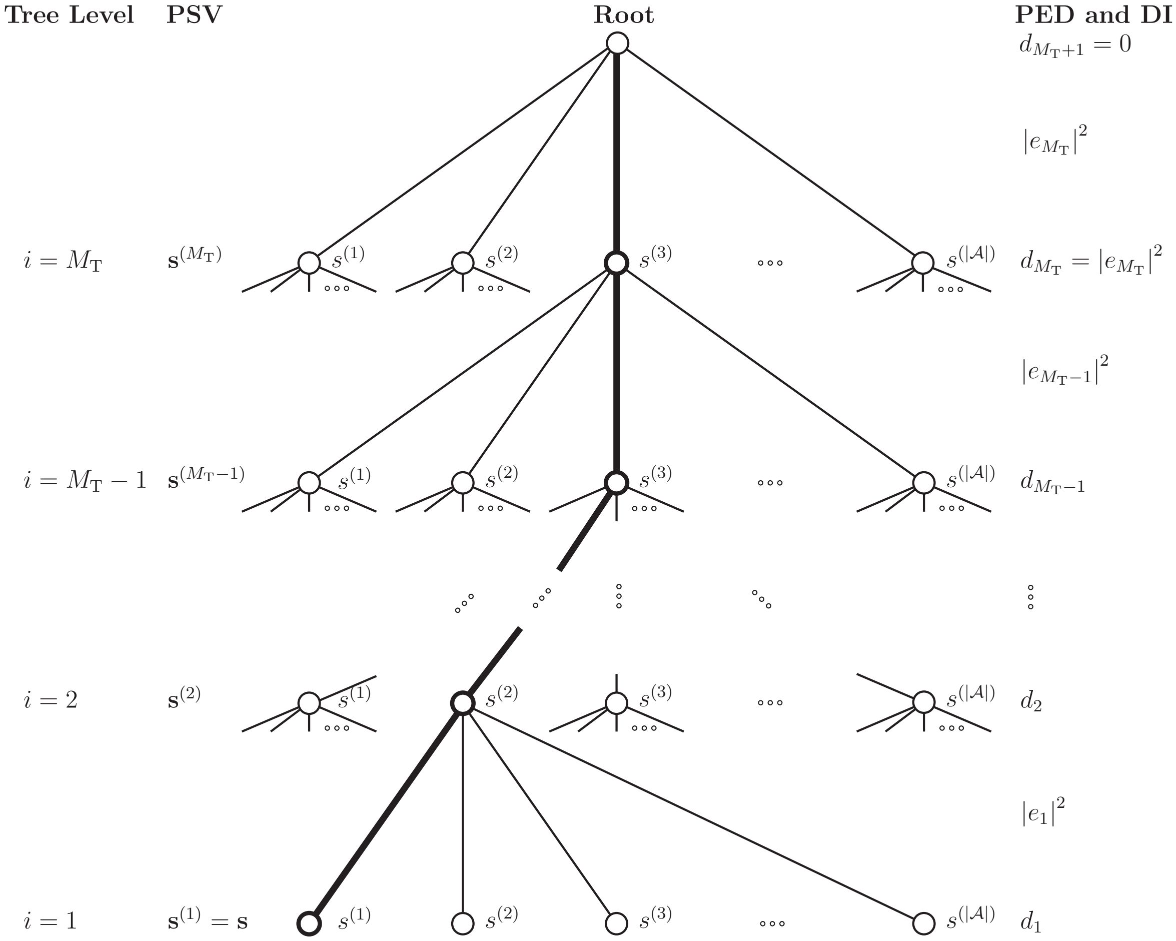


Figure 3.2: *Sketch of the tree structure of an $M_T \times M_R$ MIMO system.*

named partial Euclidean distances (PEDs), correspond to a node on level i. Every node in the tree is associated with a PED d_i, representing the DIs $|e_k|^2$ $(k = i, \ldots, M_{\mathrm{T}})$ summed up traversing the branches from the tree's root to the node.

3.3 Schnorr-Euchner Sphere Decoder

The transformation of the max-log LLR calculation into a tree structure allows the use of a sphere decoding algorithm, like the one of Schnorr-Euchner [9] (SESD). The algorithm constrains the search to nodes that lie within a hypersphere of radius R around $\tilde{\mathbf{y}}$ by comparing the PEDs (3.15) of every node with the current radius. If the PED d_i of a node already exceeds the radius R it is not necessary to traverse its sub-tree since $d_i \geq d_{i+1}$. Therefore the tree is pruned and the complexity of the minimum search is reduced efficiently. The algorithm additionally performs a radius reduction starting with an initial radius of $R = \infty$ and updating the radius according to $R^2 = d(\mathbf{s})$ whenever a leaf $\mathbf{s}$ has been reached. To reduce the radius more quickly the sub-trees of nodes are traversed in ascending order of their PEDs.

3.3.1 Repeated Tree Search

To solve (3.12) and (3.13) it is necessary to search for the ML metric and $M_{\mathrm{C}} M_{\mathrm{T}}$ metrics of the counter hypothesis. The most straightforward way to do this is a repeated tree-search (RTS) described in [5]. First (3.12) is solved using the SESD to find the ML hypothesis b_j^{ML} and the corresponding ML metric λ^{ML}. The SESD is reran to determine the metrics of the counter hypothesis $\lambda_j^{\overline{\mathrm{ML}}}$ after prepruning the search tree by forcing the decoder to exclude from the search all nodes (and their corresponding subtrees) that do not have leaves in $\mathcal{X}_j^{(\overline{b_j^{\mathrm{ML}}})}$ (i.e. for which $b_j = b_j^{\mathrm{ML}}$). The main disadvantage of the RTS strategy lies in the fact that large parts of the tree are traversed repeatedly.

3.3.2 Single Tree Search

The key to a more efficient tree-search than RTS is to ensure that every node in the tree is visited at most once. This can be accomplished by solving (3.12) and (3.13) simultaneously, i.e., by searching the ML metric λ^{ML} and all metrics of the counter hypothesis $\lambda_j^{\overline{\mathrm{ML}}}$, $j = 1, \ldots, M_{\mathrm{C}} M_{\mathrm{T}}$, in one tree traversal. This single tree-search (STS) strategy, presented in [10], is based on a list containing the ML metric λ^{ML}, its corresponding bit sequence $\mathbf{b}^{\mathrm{ML}}$ and the metrics of all counter hypothesis $\lambda_j^{\overline{\mathrm{ML}}}$. This list has to be administrated whenever a leaf of the tree is reached. During the search the list contains only preliminary values and the final values are obtained after traversing the complete tree. In the following, with a slight abuse of notation, we use the same symbols for the preliminary and the final values.

After initializing the algorithm by setting all metrics to infinity, i.e. $\lambda^{\mathrm{ML}} = \lambda_j^{\overline{\mathrm{ML}}} = \infty \, (\forall j)$ the list is administrated according to the following steps when reaching a leaf.

1. If a new ML hypothesis is found, i.e. $d(\mathbf{s}) < \lambda^{\mathrm{ML}}$, the metric of the former ML hypothesis λ^{ML} becomes the metric of the new counter-hypotheses $\lambda_j^{\overline{\mathrm{ML}}}$ for each bit in the ML hypothesis that is changed in the update process, i.e. $\lambda_j^{\overline{\mathrm{ML}}} = \lambda^{\mathrm{ML}}$ for all j for which $b_j = \overline{b_j^{\mathrm{ML}}}$. This is followed by updating the ML metric $\lambda^{\mathrm{ML}} = d(\mathbf{s})$ and the ML hypothesis $\mathbf{b}^{\mathrm{ML}} = \mathbf{b}$. This procedure ensures that all $\lambda_j^{\overline{\mathrm{ML}}}$ always contain the metric associated with a valid counter-hypotheses to the current ML hypothesis.

2. If no new ML hypothesis is found, i.e. $d(\mathbf{s}) \geq \lambda^{\mathrm{ML}}$, only the counter-hypotheses have to be checked. For all j for which $d(\mathbf{s}) < \lambda_j^{\overline{\mathrm{ML}}}$ and $b_j = \overline{b_j^{\mathrm{ML}}}$, the metric of the counter-hypotheses is updated, $\lambda_j^{\overline{\mathrm{ML}}} = d(\mathbf{s})$.

Since the STS algorithm traverses the tree only the implementation of the pruning criterion mentioned above is more complicated. A given node $\mathbf{s}^{(i)}$ on level i and the subtree originating from that node have the partial binary label $\mathbf{b}^{(i)}$ consisting of the bits

b_j, $j = iM_\text{C}, iM_\text{C}{+}1, \ldots, iM_\text{C}{+}M_\text{T}$. The remaining bits b_j $(j = 1, 2, \ldots, (i{-}1)M_\text{C}{+}M_\text{T})$ corresponding to the subtree are unknown at this point. The pruning criterion for $\mathbf{s}^{(i)}$ along with its subtree is compiled from two conditions. First, the bits in the partial binary label $\mathbf{b}^{(i)}$ are compared with the corresponding bits in the binary label of the current ML hypothesis. In this comparison, for all j with $b_j = \overline{b_j^{\text{ML}}}$, the corresponding metric of the counter-hypotheses $\lambda_j^{\overline{\text{ML}}}$ might be affected when further searching the node's subtree. Second, all counter-hypotheses corresponding to the subtree of $\mathbf{s}^{(i)}$ with the associated metrics $\lambda_j^{\overline{\text{ML}}}$, $j = 1, 2, \ldots, (i-1)M_\text{C} + M_\text{T}$, may also be updated since the corresponding bits are not yet known. In summary, the metrics which may be affected during further search in the subtree of a node $\mathbf{s}^{(i)}$ are given by the set

$$\mathcal{N} = \{n_l\} = \left\{ \lambda_j^{\overline{\text{ML}}} | b_j = \overline{b_j^{\text{ML}}}, j \geq iM_\text{C} \right\} \cup \left\{ \lambda_j^{\overline{\text{ML}}} | j < iM_\text{C} \right\}.$$

The node $\mathbf{s}^{(i)}$ and its subtree are pruned if its PED $d(\mathbf{s}^{(i)})$ satisfies

$$d(\mathbf{s}^{(i)}) > \max_{n_l \in \mathcal{N}} \{n_l\}, \tag{3.16}$$

that is, if the PED in a node already exceeds all metrics relevant for the subtree.

This pruning criterion ensures that the subtree of a given node is explored only if it can lead to an update of either the ML hypothesis or of at least one of the counter-hypotheses.

We will revise the function of the SESD using STS by means of a short example. We use the simple case of a 2 × 2 MIMO system using BPSK modulation, and assume that the symbols s_1 and s_2 represent the bit values 0 and 1 respectively. The tree structure corresponding to the system is depicted in Figure 3.3. We start with the QR-decomposition of the channel matrix $\mathbf{H}$ and set the metrics $\lambda^{\text{ML}} = \lambda_1^{\overline{\text{ML}}} = \lambda_2^{\overline{\text{ML}}} = \infty$. We start traversing the tree from its root and calculate the PEDs on tree level 2, i.e. d_2^A and d_2^B, by setting $d_{M_\text{T}+1} = d_3 = 0$ and calculating the DIs $|e_2|^2$ for every symbol on tree level 2, i.e. $|e_2|_A^2$ and $|e_2|_B^2$, according to (3.14) using the PSV $\mathbf{s}_A^{(2)}$ and $\mathbf{s}_B^{(2)}$. So far no leaf of the tree has been reached and the comparison of the calculated PEDs

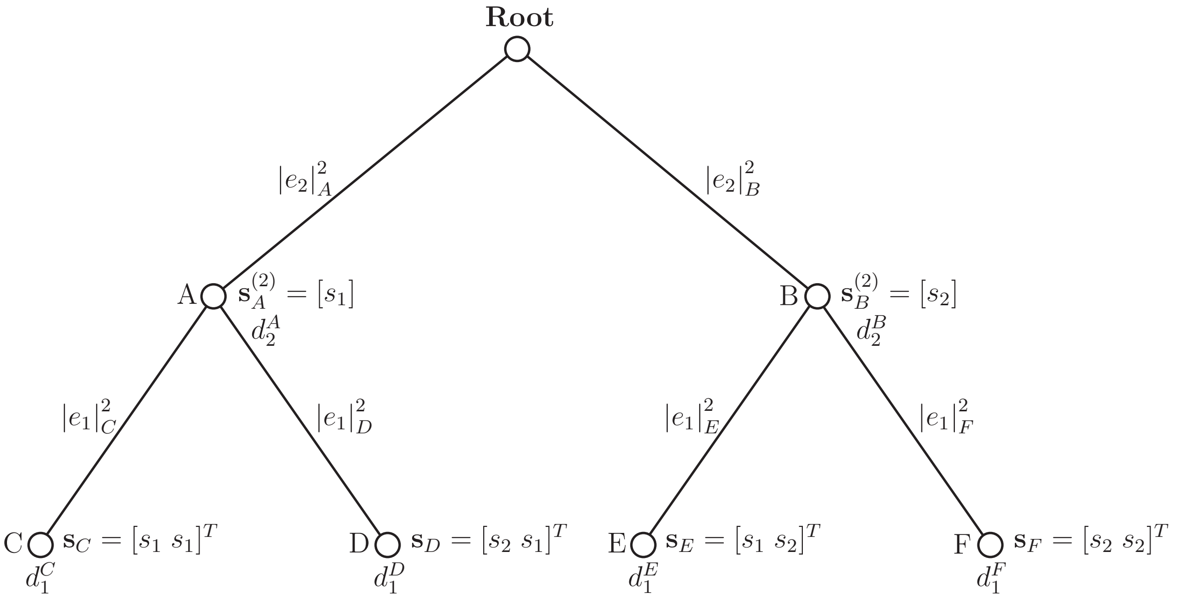

Root
$|e_2|^2_A$
$|e_2|^2_B$
A
$\mathbf{s}_A^{(2)} = [s_1]$
d_2^A
B
$\mathbf{s}_B^{(2)} = [s_2]$
d_2^B
$|e_1|^2_C$
$|e_1|^2_D$
$|e_1|^2_E$
$|e_1|^2_F$
C
$\mathbf{s}_C = [s_1\ s_1]^T$
d_1^C
D
$\mathbf{s}_D = [s_2\ s_1]^T$
d_1^D
E
$\mathbf{s}_E = [s_1\ s_2]^T$
d_1^E
F
$\mathbf{s}_F = [s_2\ s_2]^T$
d_1^F

Figure 3.3: *Tree structure for a 2 × 2 MIMO system using BPSK modulation.*

with the metrics according to (3.16) will not lead to a pruning of the tree since still $\lambda^{\mathrm{ML}} = \lambda_1^{\overline{\mathrm{ML}}} = \lambda_2^{\overline{\mathrm{ML}}} = \infty$.

The subtrees of the nodes on level 2 are traversed in order of the size of their PEDs, starting with the smallest value. For this example we assume that $d_2^A < d_2^B$ so that we continue with the subtree of node A. We traverse to tree level 1 and calculate its PEDs, i.e. d_1^C and d_1^D, by calculating the DIs $|e_1|^2$ for every symbol on tree level 1, i.e. $|e_1|^2_C$ and $|e_1|^2_D$ and adding them to d_2^A. The PEDs are sorted by size again and the smallest value, which we assume is d_1^C, is used to update the ML metric λ^{ML}, because $d_2^A < \lambda^{\mathrm{ML}} = \infty$, and the ML hypothesis is set to $\mathbf{b}^{\mathrm{ML}} = [0\ 0]$ which is the bit label corresponding to $\mathbf{s}_C$. For the symbol vector $\mathbf{s}_D$ the first bit is complementary to the ML hypothesis and it is therefore used to update the metric $\lambda_1^{\overline{\mathrm{ML}}}$ for the counter-hypothesis for this bit, because $d_1^D < \lambda_1^{\overline{\mathrm{ML}}} = \infty$.

The left side of the tree has been completely traversed now and the algorithm continues on the right branch because the comparison of d_2^B with the metrics according to (3.16) will not lead to a pruning of the tree since still $\lambda_2^{\overline{\mathrm{ML}}} = \infty$. We continue with calculating the PEDs on tree level 1, i.e. d_1^E and d_1^F, by calculating the DIs $|e_1|^2$ for every symbol on tree level 1, i.e. $|e_1|_E^2$ and $|e_1|_F^2$ and adding them to d_2^B. The PEDs are sorted by size again and we assume that the smaller value is d_1^E. We assume that $d_1^E > \lambda^{\mathrm{ML}}$, i.e. the ML hypothesis and its metric are not updated, but d_1^E is used to update the Metric $\lambda_2^{\overline{\mathrm{ML}}}$ for the counter-hypothesis for this bit, because $d_1^E < \lambda_2^{\overline{\mathrm{ML}}} = \infty$ and the second bit for this node is complementary to the ML hypothesis. Finally, assuming that $|e_1|_F^2$ has a large value, we obtain a large value for d_1^F as well, which leads to a pruning of node F according to (3.16).

Efficient tree pruning can start after finding preliminary values for the ML hypothesis and its metric as well for the metrics of the counter-hypothesis. In our example we had to traverse the tree until reaching node E before we found the first values for all these variables. The reason for that is the size of the tree. For larger trees the pruning would cut off larger parts of the tree and therefore reduce the computational complexity to a greater extent.

3.4 LLR Clipping

The dynamic range of LLRs is a priori not bounded. Practical systems need to constrain the maximum LLR value to enable implementation which evidently leads to a performance degradation. A straightforward way to ensure that LLR values are bounded is to clip them after the detection stage so that

$$|L(b_j)| \leq L_{\max}, \tag{3.17}$$

i.e. they do not exceed a certain level $L_{\max}$.

An enhanced and very common way to apply the LLR clipping is to incorporate it into the single tree-search algorithm such that it leads to a reduced search complexity [11]. Whenever a leaf has been reached and a new ML hypothesis has been found, after carrying out the steps described in Section 3.3.2, the metrics of the counter-hypotheses have to be updated according to

$$\lambda_j^{\overline{\mathrm{ML}}} \longleftarrow \min\left\{\lambda_j^{\overline{\mathrm{ML}}}, \lambda^{\mathrm{ML}} + L_{\max}\right\} \quad \forall j. \tag{3.18}$$

This causes the counter-hypotheses to be reduced more quickly and leads to enhanced pruning. For $L_{\max} \longrightarrow \infty$, we obviously approach the exact max-log solution, which has a high complexity. For $L_{\max} \longrightarrow 0$, the decoder performance approaches that of a hard-output ML detector for which the complexity is significantly lower than for $L_{\max} \longrightarrow \infty$. The parameter $L_{\max}$ can therefore be used to adjust the complexity-performances trade-off. Regarding complexity, smaller $L_{\max}$ leads to a reduction of complexity, as a more aggressive pruning criterion is used. Instead of the original criterion (3.16) the criterion

$$d(\mathbf{s}^{(i)}) > \min\left\{\max_{n_l \in \mathcal{N}}\left\{n_l\right\}, \lambda^{\mathrm{ML}} + L_{\max}\right\} \tag{3.19}$$

is used to decide if a node $\mathbf{s}^{(i)}$ and its subtree are pruned. It additionally prunes nodes and their subtrees that have PEDs $d(\mathbf{s}^{(i)}) \in [\lambda^{\mathrm{ML}} + L_{\max}, \lambda_j^{\overline{\mathrm{ML}}}]$. For lower clipping levels $L_{\max}$ this interval is larger and therefore provokes more nodes to be pruned.

Note that the receiver using LLR clipping is not optimal in the sense of the max-log approximation (3.7) any more.

Chapter 4

Modifications for the Sphere Decoder

The soft-output sphere detector presented in Chapter 3 represents a basic detector concept that can be further developed to allow a more sophisticated search for the solution of (3.8) and (3.9) in order to reduce the computational complexity of the soft-detection. In this chapter, modifications of the original detector are presented for the sake of further complexity reduction. Furthermore we present a detection scheme that allows for a complexity-diversity tradeoff.

The schemes are mainly based on the fact, that in mobile communication scenarios, where the channel remains constant for several symbol durations, it is much more preferable to spend most of the computational effort only once, at the beginning of each frame. They add additional tasks (and complexity) to the overall detection algorithm, e.g. a QR-decomposition or a singular value decomposition (SVD), to reduce the complexity of the actual detection. The complexity added by the additional tasks is compensated by the reduction in complexity in the actual detection. This represents a shift of complexity from the detection to the additional measures. Since the additional tasks are performed only once but serve for the detection of several symbols, they lead to even more reduction in the overall computational complexity of the detection algorithm.

4.1 Partial Equalizer

The tree-search algorithm in Section 3.2 is optimal in the sense of the max-log approximation (see (3.7)) and consists of the two ML detections (3.8) and (3.9). The computational complexity of the ML detection

$$\hat{\mathbf{s}}_{\mathrm{ML}} = \arg\min_{\mathbf{s}\in\mathcal{A}^{M_{\mathrm{T}}}} \left\|\mathbf{y}-\mathbf{H}\mathbf{s}\right\|^2 . \tag{4.1}$$

in general grows exponentially with M_{T}, even if the detection is implemented using the sphere-detector [12]. We define the diversity order as [13, 14]

$$d = \lim_{\frac{1}{\sigma_n^2}\to\infty} \frac{\log P(\mathbf{s}_1 \to \mathbf{s}_2)}{\log \sigma_n^2}, \tag{4.2}$$

where $P(\mathbf{s}_1 \to \mathbf{s}_2)$ denotes the pairwise error probability (PEP). For ML detection the PEP for the event that $\mathbf{s}_1$ was transmitted and $\mathbf{s}_2$ is detected conditioned on the channel matrix $\mathbf{H}$, $P_{\mathrm{ML}}(\mathbf{s}_1 \to \mathbf{s}_2|\mathbf{H})$, equals [15]

$$P_{\mathrm{ML}}(\mathbf{s}_1 \to \mathbf{s}_2|\mathbf{H}) = Q\left(\frac{\|\mathbf{H}\boldsymbol{\delta}\|}{\sqrt{2\sigma_n^2}}\right),$$

where $\boldsymbol{\delta} = \mathbf{s}_1 - \mathbf{s}_2$ is the error vector and $Q(\cdot)$ denotes the Q-function. For i.i.d. Rayleigh fading, the mean PEP for ML detection is upper bounded as [15]

$$P_{\mathrm{ML}}(\mathbf{s}_1 \to \mathbf{s}_2) = E_{\mathbf{H}}\left\{P_{\mathrm{ML}}(\mathbf{s}_1 \to \mathbf{s}_2|\mathbf{H})\right\} \leq C_{\mathrm{ML}}\left(1+\frac{\|\boldsymbol{\delta}\|^2}{4\sigma_n^2}\right)^{-M_{\mathrm{R}}}, \tag{4.3}$$

where $C_{\mathrm{ML}} > 0$ is a constant. Using (4.2) and (4.3) it follows, that the diversity order that can be reached using ML detection is $d = M_{\mathrm{R}}$, which is the full diversity.

ZF detection, on the other hand, is based on equalizing the receive vector $\mathbf{y}$ with the pseudoinverse [16]

$$\mathbf{H}^{+} = (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H,$$

i.e. $\mathbf{r} = \mathbf{H}^{+}\mathbf{y}$, and component-wise quantisation of the equalized receive vector $\mathbf{r} = \mathbf{H}^{+}\mathbf{y}$ to the symbol alphabet $\mathcal{A}$, i.e.

$$\hat{\mathbf{s}}_{\mathrm{ZF}} = Q_{\mathcal{A}}\left\{\mathbf{r}\right\} = Q_{\mathcal{A}}\left\{\mathbf{H}^{+}\mathbf{y}\right\} = Q_{\mathcal{A}}\left\{\mathbf{s} + \mathbf{H}^{+}\mathbf{n}\right\}, \tag{4.4}$$

where $Q_{\mathcal{A}}\left\{\cdot\right\}$ denotes the quantisation function.

For ZF detection the PEP conditioned on the channel matrix $\mathbf{H}$, $P_{\mathrm{ZF}}(\mathbf{s}_1 \rightarrow \mathbf{s}_2|\mathbf{H})$, equals [17]

$$P_{\mathrm{ZF}}(\mathbf{s}_1 \rightarrow \mathbf{s}_2|\mathbf{H}) = Q\left(\frac{\|\boldsymbol{\delta}\|^2}{\sqrt{2\sigma_n^2\boldsymbol{\delta}^H(\mathbf{H}^H\mathbf{H})^{-1})\boldsymbol{\delta}}}\right),$$

and for i.i.d. Rayleigh fading, the mean PEP for ZF detection is upper bounded as [18]

$$P_{\mathrm{ZF}}(\mathbf{s}_1 \rightarrow \mathbf{s}_2) = E_{\mathbf{H}}\left\{P_{\mathrm{ZF}}(\mathbf{s}_1 \rightarrow \mathbf{s}_2|\mathbf{H})\right\} \leq C_{\mathrm{ZF}}\left(1 + \frac{\|\boldsymbol{\delta}\|^2}{4\sigma_n^2}\right)^{-(M_{\mathrm{R}}-M_{\mathrm{T}}+1)}, \tag{4.5}$$

where $C_{\mathrm{ZF}} > 0$ is a constant. Using (4.2) and (4.5) it follows that the ZF detector yields diversity order $d_{\mathrm{ZF}} = M_{\mathrm{R}} - M_{\mathrm{T}} + 1$, but its computational complexity is low compared to the ML detector.

ML and ZF detection are opposite extremes in terms of diversity and complexity. In [13,14] a scheme for hard-detection is devised that allows a continuous trade-off between these two extremes to be able to adapt the detection to the available computational power and/or the necessary diversity gain. The adaption itself can be done by adjusting a single design parameter α whose meaning will be explained presently. In the following the scheme presented in [13,14] is described and finally slightly modified to be used for soft-detection.

The scheme is based on a unifying framework for the ML and ZF detector. As introduced in Section 2.1, $\mathbf{n}$ is spatially white complex Gaussian noise. From the ZF domain relation (4.4) we see that the noise in the ZF domain is distributed as $\mathbf{H}^{+}\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2\mathbf{G}^{-1})$, where the positive definite matrix $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ denotes the Gramian matrix of $\mathbf{H}$. Therefore the distribution of the equalized receive vector $\mathbf{r}$ conditioned on $\mathbf{s}$ is equal but centered to the symbol vector $\mathbf{s}$, $\mathbf{r} \sim \mathcal{CN}(\mathbf{s}, \sigma_n^2 \mathbf{G}^{-1})$. This allows to rewrite the ML detection (4.1) as

$$\hat{\mathbf{s}}_{\mathrm{ML}} = \arg\min_{\mathbf{s} \in \mathcal{A}^{M_{\mathrm{T}}}} (\mathbf{r} - \mathbf{s})^H \mathbf{G} (\mathbf{r} - \mathbf{s}). \tag{4.6}$$

Similarly, the component-wise detection (4.4) performed by the ZF detector can be written as

$$\hat{\mathbf{s}}_{\mathrm{ZF}} = \arg\min_{\mathbf{s} \in \mathcal{A}^{M_{\mathrm{T}}}} (\mathbf{r} - \mathbf{s})^H (\mathbf{r} - \mathbf{s}). \tag{4.7}$$

Both the ML and the ZF detection can thus be written in terms of a positive definite quadratic form in the ZF domain with the difference that ML takes the full correlation matrix into account (see (4.6)) by using $\mathbf{G}$ in the quadratic form whereas ZF detection completely ignores the noise correlation by using a quadratic form induced by $\mathbf{I} = \mathbf{G}^0$ (see (4.7)).

Based on this unified framework [13, 14] proposes an intermediate receiver between the ML and the ZF extremes by taking part of the noise correlation matrix into account, i.e.

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{s} \in \mathcal{A}^{M_{\mathrm{T}}}} (\mathbf{r} - \mathbf{s})^H \mathbf{G}^{1-\alpha} (\mathbf{r} - \mathbf{s}). \tag{4.8}$$

where $0 \leq \alpha \leq 1$. Choosing $\alpha = 0$ or $\alpha = 1$ yields the ML and ZF detector, respectively. For $0 < \alpha < 1$, we expect that (4.8) leads to some tradeoff between ML and ZF.

We introduce the singular value decomposition (SVD) of $\mathbf{H}$ [8],

$$\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^H,$$

where the ordered singular values $\sigma_{M_{\mathrm{T}}} \geq \ldots \geq \sigma_1 \geq 0$ of $\mathbf{H}$ are collected in $\mathbf{D} = \operatorname{diag}(\sigma_{M_{\mathrm{T}}}, \ldots, \sigma_1)$ and the $M_{\mathrm{R}} \times M_{\mathrm{T}}$ matrix $\mathbf{U}$ and the $M_{\mathrm{T}} \times M_{\mathrm{T}}$ matrix $\mathbf{V}$ have the properties $\mathbf{U}^H\mathbf{U} = \mathbf{V}^H\mathbf{V} = \mathbf{V}\mathbf{V}^H = \mathbf{I}$ and $\mathbf{U}\mathbf{U}^H \neq \mathbf{I}$. Note that the eigenvalues λ_i of the Gramian matrix $\mathbf{G} = \mathbf{H}^H\mathbf{H}$ are related to the singular values of $\mathbf{H}$ via $\lambda_i = \sigma_i^2 \geq 0$. We here also introduce $\mathbf{\Lambda} = \operatorname{diag}(\lambda_{M_{\mathrm{T}}}, \ldots, \lambda_1) = \mathbf{D}^2$.

In particular, the Gramian matrix $\mathbf{G}$ can now be written as

$$\mathbf{G} = \mathbf{H}^H\mathbf{H} = (\mathbf{U}\mathbf{D}\mathbf{V}^H)^H\mathbf{U}\mathbf{D}\mathbf{V}^H$$

$$= \mathbf{V}\mathbf{D}^H\mathbf{U}^H\mathbf{U}\mathbf{D}\mathbf{V}^H = \mathbf{V}\mathbf{D}^2\mathbf{V}^H = \mathbf{V}\mathbf{\Lambda}\mathbf{V}^H$$

and $\mathbf{G}^{1-\alpha}$ can be developed as

$$\mathbf{G}^{\frac{1-\alpha}{2}} = (\mathbf{V}\mathbf{\Lambda}\mathbf{V}^H)^{\frac{1-\alpha}{2}} = \mathbf{V}\mathbf{\Lambda}^{\frac{1-\alpha}{2}}\mathbf{V}^H.$$

$\mathbf{G}^{1-\alpha}$ has condition number

$$\kappa(\mathbf{G}^{1-\alpha}) = \kappa(\mathbf{G})^{1-\alpha}$$

that decays exponentially with α. The condition number κ is defined as the ratio of the largest and the smallest singular value, i.e. $\kappa = \sigma_{\max}/\sigma_{\min} = \sigma_{M_\mathrm{T}}/\sigma_1 \geq 1$. $\mathbf{G}^{1-\alpha}$ in general is better conditioned than $\mathbf{G}$, i.e. its condition number is smaller

$$\kappa(\mathbf{G}) \geq \kappa(\mathbf{G}^{1-\alpha}) = (\kappa(\mathbf{G}))^{1-\alpha},$$

from which we expect a lower computational complexity [19, 20].

Instead of solving (4.8) directly a more convenient formulation and interpretation is derived. To this end we rewrite the quadratic form in (4.8) using the square-root of $\mathbf{G}^{1-\alpha}$ as

$$(\mathbf{r}-\mathbf{s})^H\mathbf{G}^{1-\alpha}(\mathbf{r}-\mathbf{s}) = (\mathbf{r}-\mathbf{s})^H\mathbf{G}^{\frac{1-\alpha}{2}}\mathbf{G}^{\frac{1-\alpha}{2}}(\mathbf{r}-\mathbf{s}) = \left\|\mathbf{G}^{\frac{1-\alpha}{2}}\mathbf{r} - \mathbf{G}^{\frac{1-\alpha}{2}}\mathbf{s}\right\|^2. \tag{4.9}$$

Using the SVD $\mathbf{H} = \mathbf{U}\mathbf{D}\mathbf{V}^H$ of the channel matrix $\mathbf{H}$ we obtain

$$\begin{aligned}\mathbf{H}^+ &= (\mathbf{H}^H\mathbf{H})^{-1}\mathbf{H}^H = \mathbf{G}^{-1}\mathbf{H}^H = \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^H(\mathbf{U}\mathbf{D}\mathbf{V}^H)^H \\ &= \mathbf{V}\mathbf{\Lambda}^{-1}\mathbf{V}^H\mathbf{V}\mathbf{D}\mathbf{U}^H = \mathbf{V}\mathbf{D}^{-1}\mathbf{U}^H\end{aligned}$$

and

$$\begin{aligned}\mathbf{G}^{\frac{1-\alpha}{2}} &= (\mathbf{H}^H\mathbf{H})^{\frac{1-\alpha}{2}} = ((\mathbf{U}\mathbf{D}\mathbf{V}^H)^H\mathbf{U}\mathbf{D}\mathbf{V}^H)^{\frac{1-\alpha}{2}} \\ &= (\mathbf{V}\mathbf{D}^H\mathbf{U}^H\mathbf{U}\mathbf{D}\mathbf{V}^H)^{\frac{1-\alpha}{2}} = (\mathbf{V}\mathbf{D}^2\mathbf{V}^H)^{\frac{1-\alpha}{2}} \\ &= (\mathbf{V}\mathbf{\Lambda}\mathbf{V}^H)^{\frac{1-\alpha}{2}} = \mathbf{V}\mathbf{\Lambda}^{\frac{1-\alpha}{2}}\mathbf{V}^H.\end{aligned}$$

The first term on the right-hand side in (4.9) can then be developed as

$$\tilde{\mathbf{r}} = \mathbf{G}^{\frac{1-\alpha}{2}}\mathbf{r} = \mathbf{G}^{\frac{1-\alpha}{2}}\mathbf{H}^{+}\mathbf{y} = \mathbf{V}\mathbf{\Lambda}^{\frac{1-\alpha}{2}}\mathbf{V}^{H}\mathbf{V}\mathbf{D}^{-1}\mathbf{U}^{H}\mathbf{y} = \mathbf{V}\mathbf{D}^{-\alpha}\mathbf{U}^{H}\mathbf{y} = \mathbf{W}_{\alpha}^{-1}\mathbf{y}. \tag{4.10}$$

Here, the unitarity of $\mathbf{V}$ and the fact that $\mathbf{\Lambda} = \mathbf{D}^2$ have been used. It can be seen that the expression $\mathbf{G}^{\frac{1-\alpha}{2}}\mathbf{r}$ is equivalent to applying a partial equalizer $\mathbf{W}_{\alpha}^{-1} = \mathbf{V}\mathbf{D}^{-\alpha}\mathbf{U}^{H}$ to the receive vector $\mathbf{y}$. To further understand the effect of the partial equalizer, we insert (2.2) in (4.10), which yields

$$\tilde{\mathbf{r}} = \mathbf{W}_{\alpha}^{-1}(\mathbf{H}\mathbf{s} + \mathbf{n}) = \mathbf{C}_{\alpha}\mathbf{s} + \tilde{\mathbf{n}}. \tag{4.11}$$

The matrix $\mathbf{C}_{\alpha} = \mathbf{W}_{\alpha}^{-1}\mathbf{H} = \mathbf{V}\mathbf{D}^{-\alpha}\mathbf{U}^{H}\mathbf{U}\mathbf{D}\mathbf{V}^{H} = \mathbf{V}\mathbf{D}^{1-\alpha}\mathbf{V}^{H} = \mathbf{G}^{\frac{1-\alpha}{2}}$ is what remains of the channel after applying the partial equalizer and $\tilde{\mathbf{n}} = \mathbf{W}_{\alpha}^{-1}\mathbf{n} \sim \mathcal{CN}(\mathbf{0}, \sigma_n^2\mathbf{G}^{-\alpha})$ denotes the spatially correlated noise. Hence, using (4.11), (4.8) can finally be written as

$$\hat{\mathbf{s}}_{\text{MML}} = \arg\min_{\mathbf{s}\in\mathcal{A}^{M_{\text{T}}}} \left\|\tilde{\mathbf{r}} - \mathbf{C}_{\alpha}\mathbf{s}\right\|^2. \tag{4.12}$$

This looks like the ML detector for the linear model (4.11), but the correlation $\sigma_n^2\mathbf{G}^{-\alpha}$ of the noise $\tilde{\mathbf{n}}$ is ignored. Therefore (4.12) is called mismatched ML detection (MML). Note that for $\alpha = 0$ (4.12) reduces to the ML detector (4.1):

$$\begin{aligned}\hat{\mathbf{s}}_{\text{MML}} &= \arg\min_{\mathbf{s}\in\mathcal{A}^{M_{\text{T}}}} \left\|\tilde{\mathbf{r}} - \mathbf{C}_0\mathbf{s}\right\|^2 = \arg\min_{\mathbf{s}\in\mathcal{A}^{M_{\text{T}}}} \left\|\mathbf{G}^{\frac{1}{2}}\mathbf{r} - \mathbf{G}^{\frac{1}{2}}\mathbf{s}\right\|^2 = \\ &= \arg\min_{\mathbf{s}\in\mathcal{A}^{M_{\text{T}}}} (\mathbf{r}-\mathbf{s})^H\mathbf{G}(\mathbf{r}-\mathbf{s}) = \arg\min_{\mathbf{s}\in\mathcal{A}^{M_{\text{T}}}} (\mathbf{r}-\mathbf{s})^H\mathbf{H}^H\mathbf{H}(\mathbf{r}-\mathbf{s}) = \\ &= \arg\min_{\mathbf{s}\in\mathcal{A}^{M_{\text{T}}}} [\mathbf{H}(\mathbf{H}^{+}\mathbf{y}-\mathbf{s})]^H[\mathbf{H}(\mathbf{H}^{+}\mathbf{y}-\mathbf{s})] = \arg\min_{\mathbf{s}\in\mathcal{A}^{M_{\text{T}}}} \left\|\mathbf{y} - \mathbf{H}\mathbf{s}\right\|^2 = \hat{\mathbf{s}}_{\text{ML}}.\end{aligned}$$

Similarly for $\alpha = 1$ (4.12) equals

$$\hat{\mathbf{s}}_{\text{MML}} = \arg\min_{\mathbf{s}\in\mathcal{A}^{M_{\text{T}}}} \left\|\mathbf{H}^{+}\mathbf{y} - \mathbf{s}\right\|^2 = Q_{\mathcal{A}}\left\{\mathbf{H}^{+}\mathbf{y}\right\} = \hat{\mathbf{s}}_{\text{ZF}},$$

which is the ZF detector (4.4).

The scheme (4.8) proposed in [13, 14] can be implemented in terms of two stages, illustrated in Figure 4.1. The first stage, described by (4.11), partially equalizes the channel

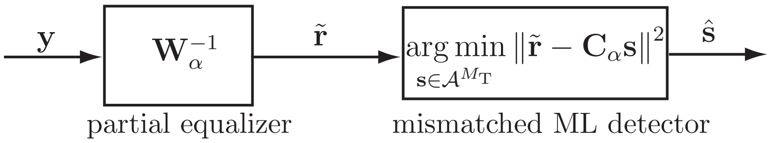

y
$\mathbf{W}_\alpha^{-1}$
$\tilde{\mathbf{r}}$
$\arg\min_{\mathbf{s}\in\mathcal{A}^{M_\mathrm{T}}} \|\tilde{\mathbf{r}} - \mathbf{C}_\alpha \mathbf{s}\|^2$
$\hat{\mathbf{s}}$
partial equalizer
mismatched ML detector

Figure 4.1: *Proposed receiver for hard-detection consisting of a partial equalizer and mismatched ML detection.*

and the second stage performs mismatched ML detection using the equalizer output according to (4.12), ignoring correlations between the components of the equalized noise vector $\tilde{\mathbf{n}}$. The amount of equalization and correspondingly the amount of noise correlation being ignored depend on the value of α. For α close to 0 there is very little equalization and low noise correlation that is ignored. For α close to 1 the mismatched ML detector performs poorly since it ignores strong noise correlations. However, the channel is almost fully equalized in this case, which results in strongly reduced complexity since $\mathbf{C}_\alpha$ almost equals the identity matrix.

Some modifications are necessary to be able to use the scheme for soft-detection. First (4.12) is modified to allow the calculation of the approximations

$$\widetilde{\lambda^{\mathrm{ML}}} = \min_{\mathbf{s}\in\mathcal{A}^{M_\mathrm{T}}} \|\tilde{\mathbf{r}} - \mathbf{C}_\alpha \mathbf{s}\|^2 \tag{4.13}$$

and

$$\widetilde{\lambda_j^{\overline{\mathrm{ML}}}} = \min_{\mathbf{s}\in\mathcal{X}_j^{(\overline{b_j^{\mathrm{ML}}})}} \|\tilde{\mathbf{r}} - \mathbf{C}_\alpha \mathbf{s}\|^2 \tag{4.14}$$

for (3.8) and (3.9), respectively. Then we use (4.13) and (4.14) for the calculation of the LLRs $\tilde{L}(b_j|\mathbf{y})$ similar to (3.10) to obtain

$$\tilde{L}\left(b_j|\mathbf{y}\right) = \begin{cases} \widetilde{\lambda^{\mathrm{ML}}} - \widetilde{\lambda_j^{\overline{\mathrm{ML}}}} & \text{if} \quad b_j^{\mathrm{ML}} = 0 \\ \widetilde{\lambda_j^{\overline{\mathrm{ML}}}} - \widetilde{\lambda^{\mathrm{ML}}} & \text{if} \quad b_j^{\mathrm{ML}} = 1. \end{cases} \tag{4.15}$$

The scheme modified for the application in soft-detection can be implemented in terms of two stages, illustrated in Figure 4.2. As in the hard-detection case the first stage, described by (4.11), partially equalizes the channel. The second stage, described by

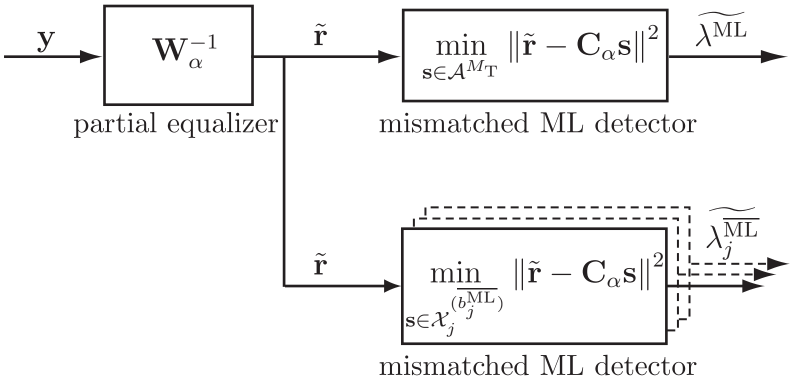

$\mathbf{y}$
$\mathbf{W}_\alpha^{-1}$
partial equalizer
$\tilde{\mathbf{r}}$
$\min_{\mathbf{s}\in\mathcal{A}^{M_\mathrm{T}}} \|\tilde{\mathbf{r}}-\mathbf{C}_\alpha\mathbf{s}\|^2$
$\widetilde{\lambda^{\mathrm{ML}}}$
mismatched ML detector
$\tilde{\mathbf{r}}$
$\min_{\mathbf{s}\in\mathcal{X}_j^{(\overline{b_j^{\mathrm{ML}}})}} \|\tilde{\mathbf{r}}-\mathbf{C}_\alpha\mathbf{s}\|^2$
$\widetilde{\lambda_j^{\mathrm{ML}}}$
mismatched ML detector

Figure 4.2: *Proposed receiver for soft-detection consisting of a partial equalizer and $|\mathcal{A}|$ mismatched ML detectors.*

(4.13) and (4.14) respectively, performs $1 + |\mathcal{A}|$ mismatched ML detections, ignoring correlations between the components of the equalized noise vector $\tilde{\mathbf{n}}$. The only difference between the mismatched ML detectors is that the minimum search is performed over different sets $\mathcal{A}^{M_\mathrm{T}}$ and $\mathcal{X}_j^{(\overline{b_j^{\mathrm{ML}}})}$. This search can be performed simultaneously using the STS algorithm presented in Section 3.3.2. The only modification necessary is to substitute (3.8) with (4.13) and (3.9) with (4.14) before performing the QR-decomposition. Considerable complexity is added by the SVD which is necessary for every realisation of the channel matrix $\mathbf{H}$ (i.e. in the worst case for every MIMO symbol $\mathbf{s}$ sent) but it is negligible compared to the complexity of the minimum search, especially in the case of soft-detection.

In the case of hard-detection the presented scheme allows a continuous trade-off in terms of diversity between ML and ZF detection. This can be done by adjusting a single parameter α. Assuming $M_\mathrm{R} \geq M_\mathrm{T}$, setting $\alpha = 0$ yields the ML detector that has high computational complexity but allows to reach the full diversity order $d = M_\mathrm{R}$. Setting $\alpha = 1$ yields the ZF detector that has low computational complexity but only allows a diversity order of $d = M_\mathrm{R} - M_\mathrm{T} + 1$. For $0 < \alpha < 1$ we get a detector that poses a compromise between the ML and ZF detectors. Defining

$$\alpha_0 = \min\left(1, \frac{M_\mathrm{R}}{M_\mathrm{T}-1} - \frac{1}{2}\right) > \frac{1}{2},$$

the diversity order of the mismatched ML detector in the hard-decision case is [14]

$$d_\mathrm{MML} = \begin{cases} (M_\mathrm{R} - M_\mathrm{T} + 1)\alpha + (1-\alpha)M_\mathrm{R} & \text{for} \quad \alpha \in [0, \alpha_0] \\ \frac{M_\mathrm{R}-M_\mathrm{T}+1}{2\alpha-1} & \text{for} \quad \alpha \in [\alpha_0, 1]. \end{cases}$$

For soft-detection we expect a similar behaviour which motivates us to perform extensive simulations. The results of these simulations and a discussion of the behaviour of the scheme in the case of soft-detection will be presented and in Chapter 5.

4.2 Lower Bound On Partial Metrics

The scheme presented in Section 4.1 showed to work properly only for small α. For larger α the partial equalizer causes strong noise enhancement and the sphere decoder spends a lot of computational complexity on unnecessarily traversing the tree. Therefore a variation of the sphere decoder proposed in [21] was used in combination with the partial equalizer presented in [13, 14]. The variation calculates a lower bound for the metric of the remaining part of the tree to further reduce the radius R and to allow more effective tree pruning which leads to a complexity reduction. In the following, the main idea of this sphere decoder variant is reviewed. Throughout this thesis it is used as in [13,14], i.e. in combination with the partial equalizer, but can be used without the partial equalizer as well. Therefore it is presented based on the original system model (2.2) instead of (4.11).

We start with the modified input-output relation (3.11) that allows to solve the decoding problem using a tree search. We split the corresponding metric into two terms including the first $k-1$ and the last $M_\mathrm{T} - k + 1$ layers respectively,

$$\|\tilde{\mathbf{y}} - \mathbf{R}\mathbf{s}\|^2 = \left\| \begin{bmatrix} \tilde{\mathbf{y}}_k \\ \tilde{\mathbf{y}}^k \end{bmatrix} - \begin{bmatrix} \mathbf{R}_k^1 & \mathbf{R}_k^2 \\ \mathbf{0} & \mathbf{R}^k \end{bmatrix} \begin{bmatrix} \mathbf{s}_k \\ \mathbf{s}^k \end{bmatrix} \right\|^2$$

$$= \left\| \begin{bmatrix} \tilde{\mathbf{y}}_k - \mathbf{R}_k^1 \mathbf{s}_k - \mathbf{R}_k^2 \mathbf{s}^k \\ \tilde{\mathbf{y}}^k - \mathbf{0} \mathbf{s}_k - \mathbf{R}^k \mathbf{s}^k \end{bmatrix} \right\|^2$$

$$= \left\| \tilde{\mathbf{y}}_k - \mathbf{R}_k^1 \mathbf{s}_k - \mathbf{R}_k^2 \mathbf{s}^k \right\|^2 + \left\| \tilde{\mathbf{y}}^k - \mathbf{R}^k \mathbf{s}^k \right\|^2 \tag{4.16}$$

where $\tilde{\mathbf{y}}_k = [\tilde{y}_1, \ldots, \tilde{y}_{k-1}]$, $\tilde{\mathbf{y}}^k = [\tilde{y}_k, \ldots, \tilde{y}_{M_\mathrm{R}}]$, $\mathbf{s}_k = [s_1, \ldots, s_{k-1}]$, $\mathbf{s}^k = [s_k, \ldots, s_{M_\mathrm{T}}]$ and $\mathbf{R}_k^1 = [\mathbf{R}]_{ij}$ for $i, j \in \{1, \ldots, k-1\}$, $\mathbf{R}_k^2 = [\mathbf{R}]_{ij}$ for $i \in \{1, \ldots, k-1\}$ and $j \in \{k, \ldots, M_\mathrm{T}\}$, $\mathbf{R}^k = [\mathbf{R}]_{ij}$ for $i, j \in \{k, \ldots, M_\mathrm{T}\}$.

At the kth layer, the sphere bound $\left\| \tilde{\mathbf{y}} - \mathbf{R}\mathbf{s} \right\|^2 \leq R^2$ amounts to

$$\left\| \tilde{\mathbf{y}}^k - \mathbf{R}^k \mathbf{s}^k \right\|^2 \leq R^2 - \left\| \tilde{\mathbf{y}}_k - \left[\mathbf{R}_k^1 \mathbf{R}_k^2 \right] \mathbf{s} \right\|^2 .$$

Using the Rayleigh-Ritz theorem [8] we obtain a lower bound ρ_k^2 for $\left\| \tilde{\mathbf{y}}_k - \left[\mathbf{R}_k^1 \mathbf{R}_k^2 \right] \mathbf{s} \right\|^2$ as

$$\rho_k^2 = \sigma_{\min}^2(\mathbf{R}_k^1) \left\| \mathcal{Q}\{\mathbf{a}_k\} - \mathbf{a}_k \right\|^2 ,$$

with $\mathbf{a}_k = (\mathbf{R}_k^1)^{-1}(\tilde{\mathbf{y}}_k - \mathbf{R}_k^2 \mathbf{s}^k)$ and $\sigma_{\min}(\mathbf{R}_k^1)$ denoting the minimum singular value of $\mathbf{R}_k^1$ and $\mathcal{Q}\{\mathbf{a}_k\}$ denoting the component-wise quantization with respect to the symbol alphabet $\mathcal{A}$. For the calculation of the singular value the Power Method [16] can be used, since $\sigma_{\min}^2(\mathbf{R}_k^1) = \lambda_{\min}((\mathbf{R}_k^1)^H \mathbf{R}_k^1)$ and the inversion of $\mathbf{R}_k^1$ and the matrix-matrix multiplication $(\mathbf{R}_k^1)^H \mathbf{R}_k^1$ can be iteratively computed.

The lower bound ρ_k^2 gives a foresight on the partial metric of the subtree (that has not been traversed yet) of the current node and can therefore be seen as a future metric. It allows at each layer to use a modified sphere bound

$$\left\| \tilde{\mathbf{y}}^k - \mathbf{R}^k \mathbf{s}^k \right\|^2 \leq \tilde{R}^2$$

with the reduced sphere radius

$$\tilde{R}^2 = R^2 - \rho_k^2 .$$

The reduced sphere radius leads to a more efficient pruning and hence to a smaller number of visited nodes.

To be able to use this variation of the sphere decoder with the SESD presented in Section 3.3 we have to apply a simple modification. Using the SESD we don't have a single radius but $1 + M_{\mathrm{C}} M_{\mathrm{T}}$ radii. Using the lower bound ρ_k^2 the pruning criterion for the SESD (3.16) modifies to

$$d(\mathbf{s}^{(i)}) > \max_{n_l \in \mathcal{N}} \{n_l\} - \rho_k^2, \tag{4.17}$$

which means that a node $\mathbf{s}^{(i)}$ and its subtree are pruned if (4.17) is satisfied. Note that, because ρ_k^2 is a lower bound, the sphere decoder still performs optimal in the sense of the max-log approximation (3.7).

If the lower bound is used together with LLR clipping, as described in Section 3.4, the pruning criterion (4.17) modifies to

$$d(\mathbf{s}^{(i)}) > \min \left\{ \max_{n_l \in \mathcal{N}} \{n_l\}, \lambda^{\mathrm{ML}} + L_{\max} \right\} - \rho_k^2. \tag{4.18}$$

Note that in this case the sphere decoder does not perform optimal anymore, which is caused by the LLR clipping.

Chapter 5

Simulation Results

For simulation purposes the algorithms described in the previous chapters were implemented in MATLAB. In the following the simulated transmission system is described in detail including a derivation of the model used for the channel impulse response and of the SNR definition used. Finally the chapter is closed by the a presentation of the conducted simulations.

5.1 Simulated Transmission System

For the simulations the MIMO-OFDM transmission system with M_T transmit antennas and M_R receive antennas shown in 5.1 was implemented. It consists of several blocks that will be described in the following.

A random bit sequence d_i with independent identically distributed bits is used as source data. The bit sequence d_i is channel coded by the *channel encoder* using a rate $R = \frac{1}{2}$ convolutional code, leading to the coded data sequence b'_j. In a next step, the *interleaver* (Π), the coded data sequence b'_j is reordered to protect the transmission against burst errors. This leads to the interleaved coded data sequence b_j. In the mapper the interleaved coded data sequence b_j is mapped to transmit symbols s_l of a 16-QAM alphabet $\mathcal{A}$. The

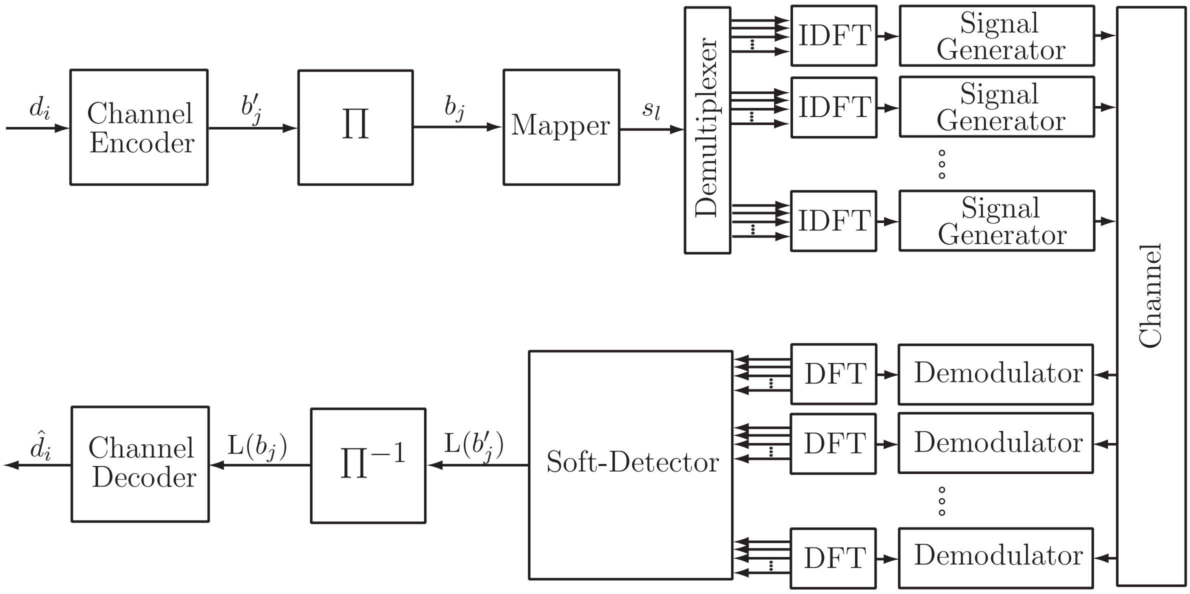


Figure 5.1: *Block diagram of the simulated MIMO transmission system implementing OFDM modulation from source to estimation data.*

transmit symbol sequence s_l is divided into M_{T} sub-sequences by the *demultiplexer* for the transmission from M_{T} transmit antennas. In a next step an OFDM modulation is performed for every sub-sequence. This is done by grouping K symbols to one OFDM symbol and performing a inverse discrete Fourier transform (IDFT). After adding the cyclic prefix (CP) of length L_{CP} the time domain signal of length $N = K + L_{\text{CP}}$ is converted into transmit signals in the *signal generators* and sent over the *channel* using multiple transmit antennas. The channel, whose model will be described presently, corrupts the transmit signals and produces the receive signals that are demodulated to pseudosymbols in the *demodulators*. The discrete Fourier transform (DFT) is performed to get back the sub-sequences of transmit symbols. The soft-detector jointly detects the bits contained in the transmit symbol sequences and as a result delivers log-likelihood ratios $L(b'_j)$ that are reordered to the original order in the *deinterleaver* (Π^{-1}). For our simulations we used several different soft-detectors that are all based on the SESD described in Section 3.3. The different versions are mentioned in detail later when the

Parameter	**Value**	**Description**
K	128	Number of OFDM subcarriers
M_{T}	4	Number of transmit antennas
M_{R}	4	Number of receive antennas
R	$\frac{1}{2}$	Channel code rate
L_{CP}	$K/4 = 32$	Cyclic prefix length
M_{C}	4	Number of bits per symbol

Table 5.1: *Parameters used for the simulations.*

conducted simulations are explained. The soft-detector delivers log-likelihood ratios $L(b_j)$ that are decoded by the *channel decoder*, in our case a soft-in Viterbi decoder. Finally we obtain the estimates $\hat{d}_i$ for the source data bits d_i. Table 5.1 summarises the parameters used for the simulations.

5.1.1 Channel Impulse Response

The channel coefficients $h_{i,j}[k]$ in the frequency domain, as described in 2.1, can be obtained by calculating the DFT of the impulse response $g_{i,j}[l]$. Since this procedure is the same for all antenna indices i and j we omit these in the following for the sake of convenience and write $h[k]$ and $g[l]$ instead of $h_{i,j}[k]$ and $g_{i,j}[l]$ respectively. The channel impules response $g[l]$ is modeled as a tapped delay line with uncorrelated taps and exponential delay profile. This is explained in more detail in the following.

Due to multipath propagation in the channel a superposition of multiple copies of the transmitted signal is obtained at the receiver. These copies are delayed and attenuated differently. To represent this fact, the channel's impulse response for the link from a transmit antenna to a receive antenna is a sum of taps, i.e. independently delayed unit impulses weighted by independent coefficients. We therefore model the channel's impulse response $g[l]$ using the following function

$$g[l] = \begin{cases} x[l]\,\mathrm{e}^{-l/2\tau} & \text{for} \quad 0 \le l \le L-1 \\ 0 & \text{for} \quad l < 0 \text{ and } l > L-1. \end{cases} \tag{5.1}$$

Here $x[l]$ is a strictly stationary, strictly white zero-mean complex Gaussian random process with mean function $\mathrm{E}\{x[l]\} = 0$ and autocorrelation function $\mathrm{E}\{x[l]x^*[l']\} = \sigma_x^2\,\delta[l-l']$, where $\mathrm{E}\{\cdot\}$ denotes the expected value with respect to the time index l. The impulse response (5.1) is therefore a nonstationary, strictly white zero-mean complex Gaussian random process with mean function

$$\mathrm{E}\{g[l]\} = \mathrm{E}\{x[l]\}\,\mathrm{e}^{-l/2\tau} = 0$$

and autocorrelation function

$$\begin{aligned} r_g[l,l'] &= \mathrm{E}\{g[l]g^*[l']\} = \mathrm{E}\{x[l]\,\mathrm{e}^{-l/2\tau}x^*[l']\mathrm{e}^{-l'/2\tau}\} = \\ &= \sigma_x^2\,\delta[l-l']\,\mathrm{e}^{-(l+l')/2\tau} = \sigma_x^2\,\delta[l-l']\,\mathrm{e}^{-l/\tau} \quad \text{for} \quad 0 \le l, l' \le L-1. \end{aligned} \tag{5.2}$$

The channel's impuls response therefore consists of L taps $l = 0, \ldots, L-1$, where the lth tap has l samples delay and variance $\mathrm{E}\{|g[l]|^2\} = r_g[l,l] = \sigma_x^2\,\mathrm{e}^{-l/\tau}$ for $0 \le l \le L-1$. Using the DFT of $g[l]$, defined as

$$h[k] = \sum_{n=0}^{K-1} g[l]\mathrm{e}^{-j\frac{2\pi kn}{K}},$$

we obtain the variance of $h[k]$ as

$$\begin{aligned} \sigma_h^2 &= E\left\{|h[k]|^2\right\} = E\left\{h[k]^*h[k]\right\} \\ &= E\left\{\left(\sum_{l=0}^{K-1} g[l]\mathrm{e}^{-j\frac{2\pi kl}{K}}\right)^* \left(\sum_{l=0}^{K-1} g[l]\mathrm{e}^{-j\frac{2\pi kl}{K}}\right)\right\} \\ &= E\left\{\sum_{l=0}^{K-1}\sum_{l'=0}^{K-1} \left(g[l]\mathrm{e}^{-j\frac{2\pi kl}{K}}\right)^* \left(g[l']\mathrm{e}^{-j\frac{2\pi kl'}{K}}\right)\right\} \\ &= E\left\{\sum_{l=0}^{K-1} g[l]^*g[l] + \sum_{l=0}^{K-1}\sum_{l'=0,l'\neq l}^{K-1} g[l]^*g[l']\mathrm{e}^{j\frac{2\pi k(l-l')}{K}}\right\} \end{aligned}$$

$$
\begin{aligned}
&= \sum_{l=0}^{K-1} E\left\{g[l]^* g[l]\right\} + \sum_{l=0}^{K-1} \sum_{l'=0, l' \neq l}^{K-1} E\left\{g[l]^* g[l']\right\} \mathrm{e}^{j\frac{2\pi k(l-l')}{K}} \\
&= \sum_{l=0}^{K-1} E\left\{g[l]^* g[l]\right\} = \sum_{l=0}^{L-1} \sigma_x^2 \, \mathrm{e}^{-l/\tau},
\end{aligned}
$$

where on the last line $L \leq K$ has been assumed and (5.2) has been used.

5.1.2 SNR Definition

Generally the SNR (in dB) is defined as

$$
\mathrm{SNR} = 10 \log \left(\frac{P_\mathrm{s}}{P_\mathrm{n}} \right)
$$

where P_s and P_n denote the signal power and the noise power in the received signal respectively.

In the following we will derive both the signal power and the noise power separately to be able to properly calculate the SNR for the simulations. At the ith receive antenna we obtain a superposition of the transmitted symbols weighted by the channel coefficients $h_{i,j}$ and additionally superposed by the noise term n_i,

$$
y_i = \sum_{j=1}^{M_\mathrm{T}} h_{i,j} s_j + n_i.
$$

We derive the powers P_s and P_n from y_i by taking into account only the signal part of the receive signal and only the noise part of the receive signal respectively. We obtain

$$
\begin{aligned}
P_\mathrm{s} &= E\left\{ \left(\sum_{j=1}^{M_\mathrm{T}} h_{i,j} s_j \right) \left(\sum_{j=1}^{M_\mathrm{T}} h_{i,j} s_j \right)^* \right\} = E\left\{ \sum_{j=1}^{M_\mathrm{T}} \sum_{j'=1}^{M_\mathrm{T}} \left(h_{i,j} s_j\right) \left(h_{i,j'} s_{j'}\right)^* \right\} \\
&= E\left\{ \sum_{j=1}^{M_\mathrm{T}} \left(|h_{i,j}|^2 |s_j|^2 \right) + \sum_{j=1}^{M_\mathrm{T}} \sum_{j'=1, j' \neq j}^{M_\mathrm{T}} \left(h_{i,j} s_j\right) \left(h_{i,j'} s_{j'}\right)^* \right\} \\
&= \sum_{j=1}^{M_\mathrm{T}} E\left\{|h_{i,j}|^2\right\} E\left\{|s_j|^2\right\} + \sum_{j=1}^{M_\mathrm{T}} \sum_{j'=1, j' \neq j}^{M_\mathrm{T}} E\left\{h_{i,j}\right\} E\left\{s_j\right\} E\left\{h_{i,j'}^*\right\} E\left\{s_{j'}^*\right\} \\
&= \sum_{j=1}^{M_\mathrm{T}} \sigma_h^2 \sigma_s^2 = M_\mathrm{T} \sigma_h^2 \sigma_s^2
\end{aligned}
$$

and

$$P_{\mathrm{n}} = E\left\{n_i n_i^*\right\} = E\left\{|n_i|^2\right\} = \sigma_n^2.$$

The powers P_{s} and P_{n} represent the signal and the noise power at a single receive antenna. Since we use M_{R} antennas, the total signal power at the receiver results to $M_{\mathrm{R}}P_{\mathrm{s}}$ and the total noise power at the receiver results to $M_{\mathrm{R}}P_{\mathrm{n}}$. The SNR at the receiver (in dB) can therefore be calculated as

$$\mathrm{SNR} = 10\log\left(\frac{M_{\mathrm{R}}P_{\mathrm{s}}}{M_{\mathrm{R}}P_{\mathrm{n}}}\right) = 10\log\left(\frac{M_{\mathrm{R}}M_{\mathrm{T}}\sigma_h^2\sigma_s^2}{M_{\mathrm{R}}\sigma_n^2}\right) = 10\log\left(\frac{M_{\mathrm{T}}\sum\limits_{l=0}^{L-1}\sigma_x^2\,\mathrm{e}^{-l/\tau}\sigma_s^2}{\sigma_n^2}\right)$$

5.2 Simulations and Results

The MIMO-OFDM transmission system presented in Section 5.1 was used for the simulations. For the different simulations the transmission system was changed only in one of the blocks, the soft-detector. In all cases a sphere-decoder was used as soft-detector, but the used algorithm and its parameters have been varied. The variations will be described in detail presently.

It was observed how different sphere-detection algorithms and their parameters influence the performance and the complexity of the soft-detection. The performance was measured counting the number of bit errors in the estimated data bits $\hat{d}_i$ and calculating a bit error rate (BER). The complexity was measured using the number of nodes that are visited by the particular sphere decoding algorithm. A node was considered visited if it is not pruned, i.e. its partial Euclidean distance (PED) d_i (cf. (3.15)) does not satisfy the pruning criterion (cf. (3.16),(3.19) or (4.17)). In the case of our transmission system using $M_{\mathrm{T}} = 4$ transmit antennas and a 16-QAM alphabet $\mathcal{A}$ the detector has to search the minima over a set $\mathcal{A}^{M_{\mathrm{T}}}$ with cardinality $|\mathcal{A}^{M_{\mathrm{T}}}| = |\mathcal{A}|^{M_{\mathrm{T}}} = 16^4 = 65536$ to calculate the LLRs according to (3.7). Therefore the maximum number of nodes visited is 65536.

5.2.1 Schnorr-Euchner Sphere Decoder with LLR Clipping

For the first simulation we used the basic Schnorr-Euchner Sphere Decoder (SESD) as described in Section 3.3. We additionally implemented the LLR clipping as described in Section 3.4 and simulated the influence of the clipping level L_{max} on the bit error rate (BER) and the number of nodes visited.

Figure 5.2 shows the resulting BER over the SNR for the clipping levels $L_{\text{max}} = \{\infty, 8, 4, 1, 0.2\}$. The corresponding complexity of the detection represented by the mean number of nodes visited over the SNR for the same clipping levels is shown in Figures 5.3 and 5.4. Figure 5.5 shows the mean number of nodes visited until the ML metric λ^{ML} has been found, which corresponds to the search complexity of the hard detection. Figures 5.6 and 5.7 show the cumulative distribution function (cdf) for the number of nodes visited and the number of nodes visited until the ML metric λ^{ML} has been found, respectively.

The clipping level $L_{\text{max}} = \infty$ corresponds to the SESD that performs optimal in the sense of a maximum likelihood (ML) detection. This optimal performance comes at the cost of the highest complexity represented by the high mean number of nodes visited in the area of some thousand nodes. Compared to the total number of nodes ($|\mathcal{A}^{M_\text{T}}| = 65536$) this means that only about a fifth of all nodes is visited. Setting the clipping level L_{max} to lower values causes the SESD to loose its optimality in the detection performance, but at the same time the complexity of the detection is reduces tremendously. For the case with $L_{\text{max}} = 8$ the mean complexity reduces to about five percent (for higher SNRs) which comes at the cost of a performance loss of about 2 dB. Further reducing the clipping level L_{max} increases this trend. The cdfs show the same dependency on the clipping level L_{max} and are shifted to the left when reducing L_{max}. Note that the value of L_{max} does not influence the slopes of the BER over SNR curves.

The simulation results show that, using the SESD with LLR clipping, the parameter L_{max} can be used for a performance-complexity trade-off. Applying moderate LLR clipping can help to significantly reduce complexity while only slightly degrading BER.

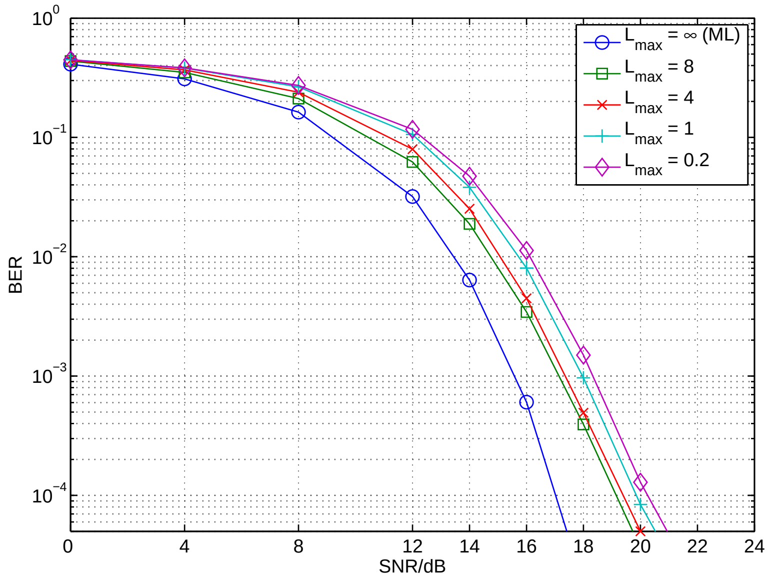


Figure 5.2: *Schnorr-Euchner sphere decoder with LLR clipping: bit eror rate (BER) versus signal to noise ratio (SNR) for different clipping levels $L_{\max}$.*

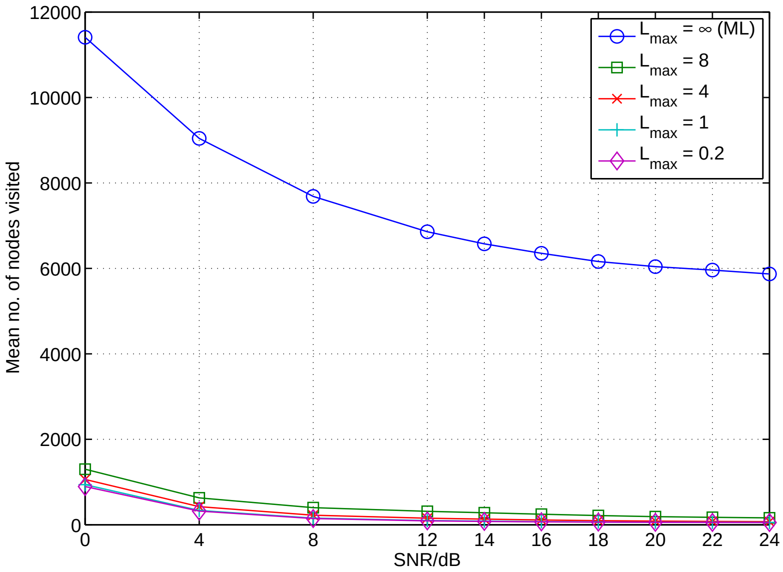


Figure 5.3: *Schnorr-Euchner sphere decoder with LLR clipping: mean number of nodes visited versus SNR for different clipping levels $L_{\max}$.*

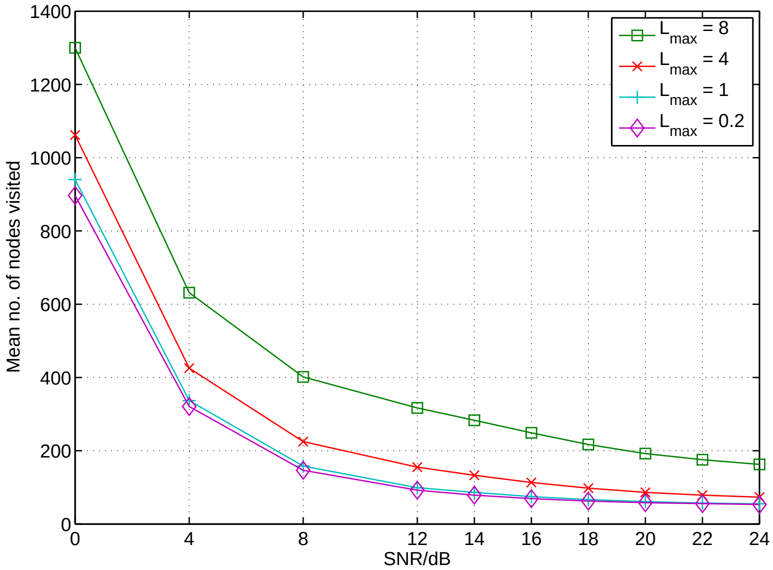


Figure 5.4: *Schnorr-Euchner sphere decoder with LLR clipping: mean number of nodes visited versus SNR for different clipping levels $L_{\max}$ (except $L_{\max} = \infty$).*

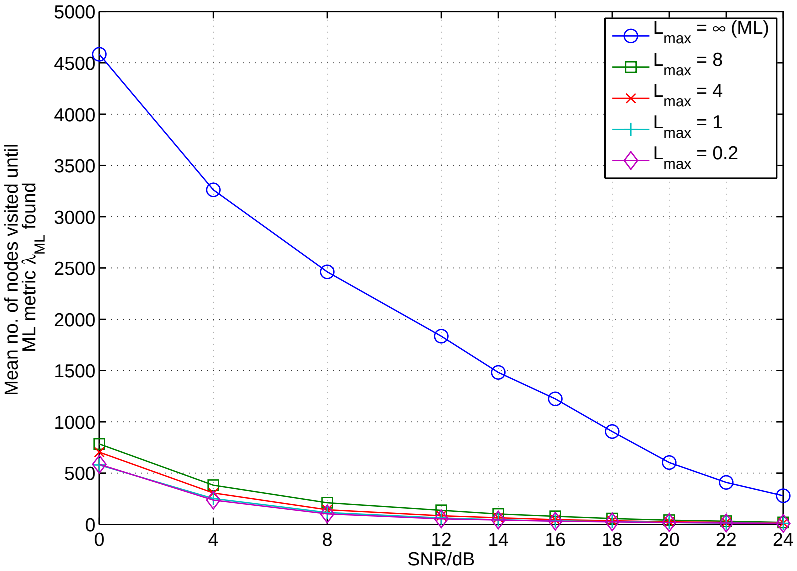


Figure 5.5: *Schnorr-Euchner sphere decoder with LLR clipping: mean number of nodes visited until ML metric λ^{ML} has been found versus SNR for different clipping levels $L_{\max}$ (except $L_{\max} = \infty$).*

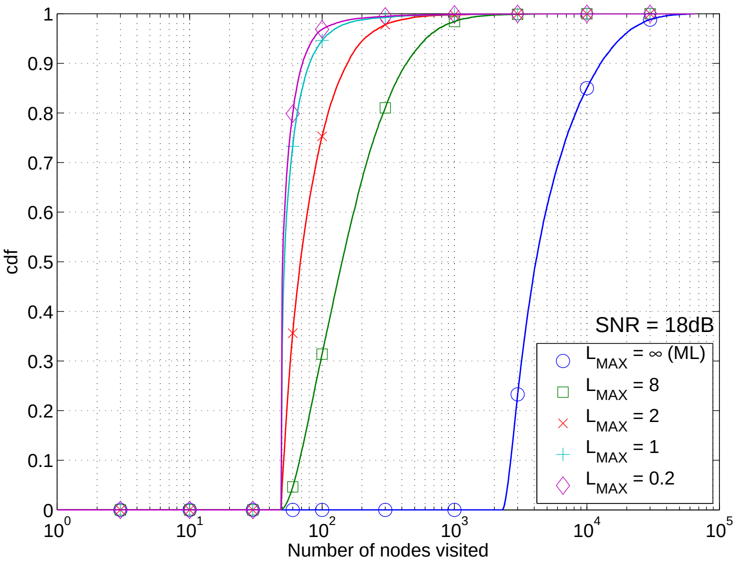


Figure 5.6: *Schnorr-Euchner sphere decoder with LLR clipping: cdf of the mean number of nodes visited for different clipping levels* $L_{\max}$ *and an SNR of 18dB.*

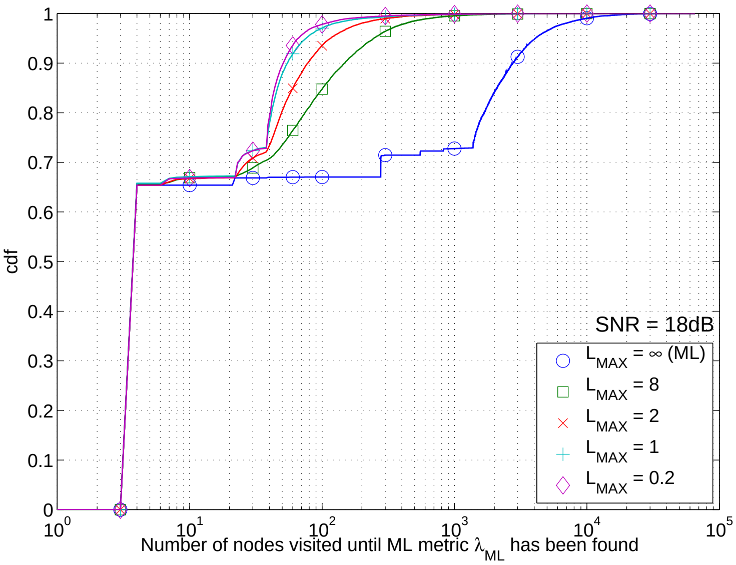


Figure 5.7: *Schnorr-Euchner sphere decoder with LLR clipping: cdf of the mean number of nodes visited until ML metric* λ^{ML} *has been found for different clipping levels* $L_{\max}$ *and an SNR of 18dB.*

5.2.2 Schnorr-Euchner Sphere Decoder with LLR Clipping and Lower Bound on Partial Metrics

We extended the SESD with LLR clipping used in the previous simulation with the lower bound on partial metrics presented in Section 4.2. Again, we monitored the influence of the clipping level $L_{\max}$ on the detection performance and the complexity of the detection.

Figure 5.8 shows the BER over the SNR for the clipping levels $L_{\max} = \{\infty, 8, 4, 1, 0.2\}$. No difference can be found in the performance between the previous case without the lower bound on partial metrics and the present one implementing this lower bound. This is what we expected since the lower bound does not affect the optimality of the detection. Figures 5.9 and 5.10 show the corresponding mean complexity. Figure 5.11 shows the complexity for finding the ML metric λ^{ML}. Figures 5.12 and 5.13 show the cumulative distribution function (cdf) for the number of nodes visited and the number of nodes visited until the ML metric λ^{ML} has been found, respectively.

It can be seen that, in case of ML detection $L_{\max} = \infty$, the use of the lower bound on partial metrics has only little effect on the cdf of the number of nodes visited and its mean. For lower SNR values the complexity is reduced noticeable but for larger SNR values the reduction in complexity is quite small.

For lower values of the clipping level $L_{\max}$ the saving in complexity is higher. The mean number of nodes visited is reduced to less than the half. For the ML detection case as well as for the lower values of $L_{\max}$ it can be seen that the mean number of nodes does not depend as intense on the SNR as without the lower bound.

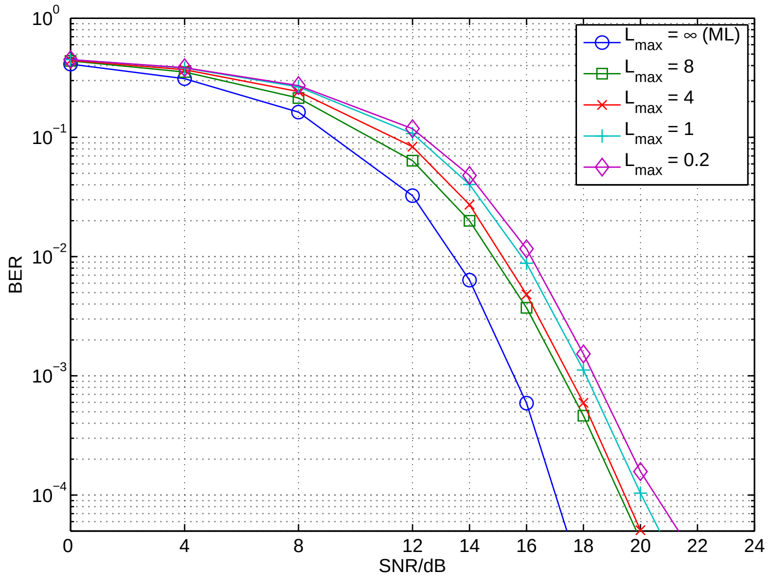


Figure 5.8: *Schnorr-Euchner sphere decoder with LLR clipping: bit eror rate (BER) versus SNR for different $L_{\max}$ with lower bound on partial metrics.*

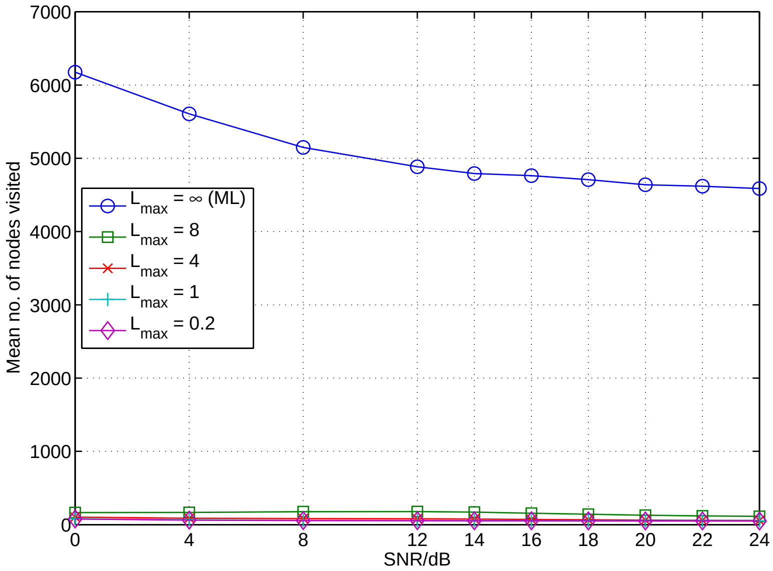


Figure 5.9: *Schnorr-Euchner sphere decoder with LLR clipping: mean number of nodes visited versus SNR for different clipping levels $L_{\max}$ with lower bound on partial metrics.*

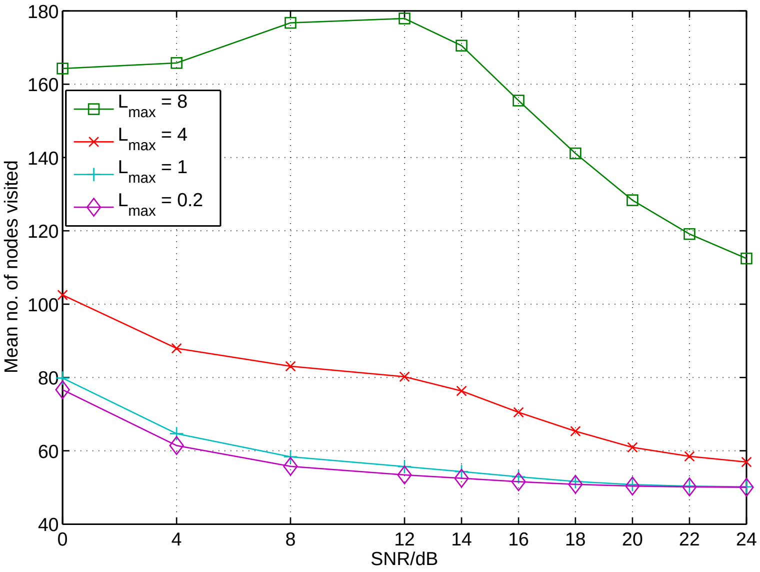


Figure 5.10: *Schnorr-Euchner sphere decoder with LLR Clipping: mean number of nodes visited versus SNR for different clipping levels $L_{\max}$ (except $L_{\max} = \infty$) with lower bound on partial metrics.*

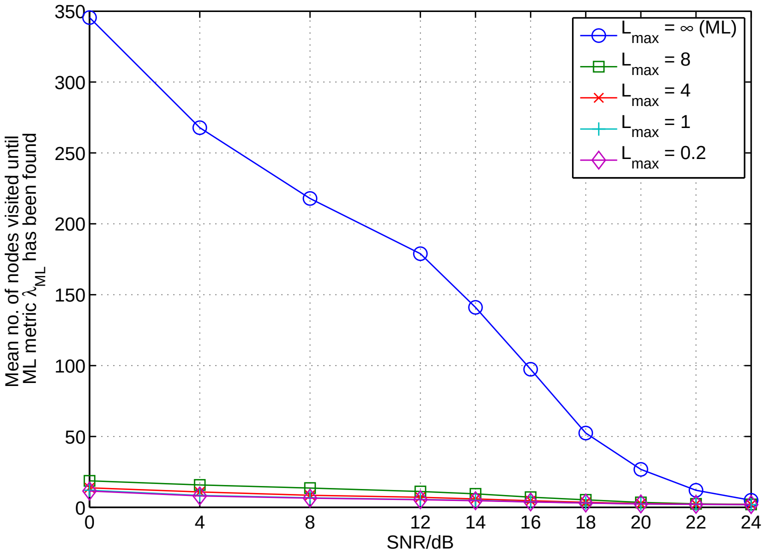


Figure 5.11: *Schnorr-Euchner sphere decoder with LLR clipping: mean number of nodes visited until ML metric λ^{ML} has been found over signal to noise ratio (SNR) for different clipping levels $L_{\max}$ (except $L_{\max} = \infty$) with lower bound on partial metrics.*

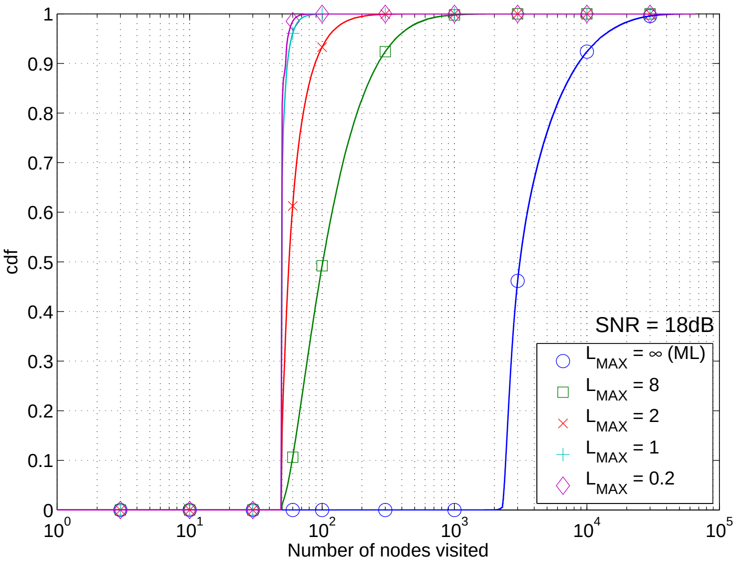


Figure 5.12: *Schnorr-Euchner sphere decoder with LLR clipping: cdf of the mean number of nodes visited for different clipping levels $L_{\max}$ with lower bound on partial metrics and an SNR of 18dB.*

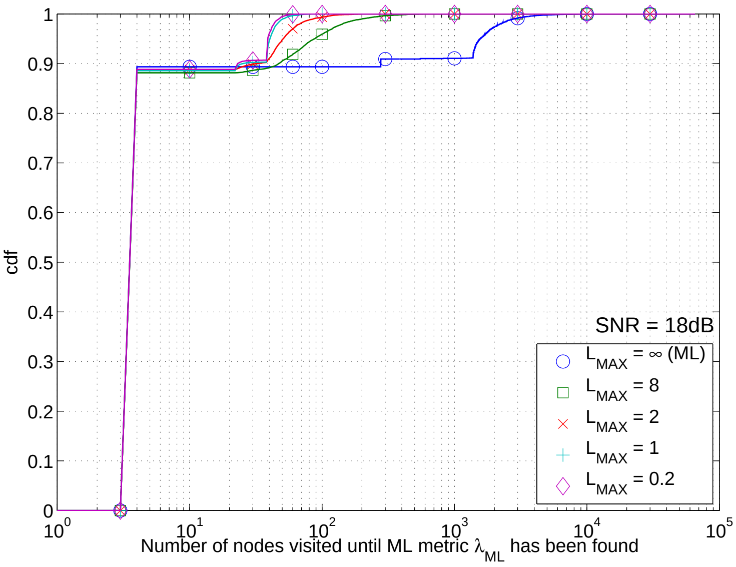


Figure 5.13: *Schnorr-Euchner sphere decoder with LLR clipping: cdf of the mean number of nodes visited until ML metric λ^{ML} has been found for different clipping levels $L_{\max}$ with lower bound on partial metrics and an SNR of 18dB.*

5.2.3 Schnorr-Euchner Sphere Decoder with Partial Equalizer

We used the basic Schnorr-Euchner Sphere Decoder (SESD) as described in Section 3.3 and additionally implemented the partial equalizer described in Section 4.1. We simulated the influence of the amount of partial equalization, described by the parameter α, on the detection performance and the complexity of the detection.

Figure 5.14 shows the BER over the SNR for different values of the parameter α. It can be seen, that for $\alpha = 0$ we obtain the same performance as for the basic SESD (cf. Figure 5.2), i.e. in this case the detector is ML optimal. For $\alpha = 1$ we obtain the performance for the zero forcing (ZF) detection, i.e. detection after full equalization of the channel's influence. For the values $0 \leq \alpha \leq 1$ we get performances that lie between the ML and the ZF case. It can be seen that with increasing α the performance degrades and the slopes of the curves get flatter too. This can be interpreted as a loss of diversity order. Figure 5.15 shows the mean detection complexity for different α and Figure 5.16 shows the mean number of nodes visited until the ML metric λ^{ML} has been found for different α, which corresponds to the search complexity of the hard detection. Figures 5.17 and 5.18 show the cdf of the nodes visited and the nodes visited until the ML metric λ^{ML} has been found, respectively. For $\alpha = 0$, the ML detection case, the complexity is not reduced at all. For higher values of α partial equalization is performed with higher intensity which leads to strong noise enhancement and therefore higher complexity for low SNRs. For higher SNRs however we observe a small reduction in complexity that rises with increasing α. From the cdfs it can be seen that the influence of α on the complexity of the search for the ML metric λ^{ML} is stronger than on overall complexity. All these observations are in accordance with our expectations.

The simulation results show that, using the SESD partial equalization, the parameter α can be used for a diversity-complexity trade-off.

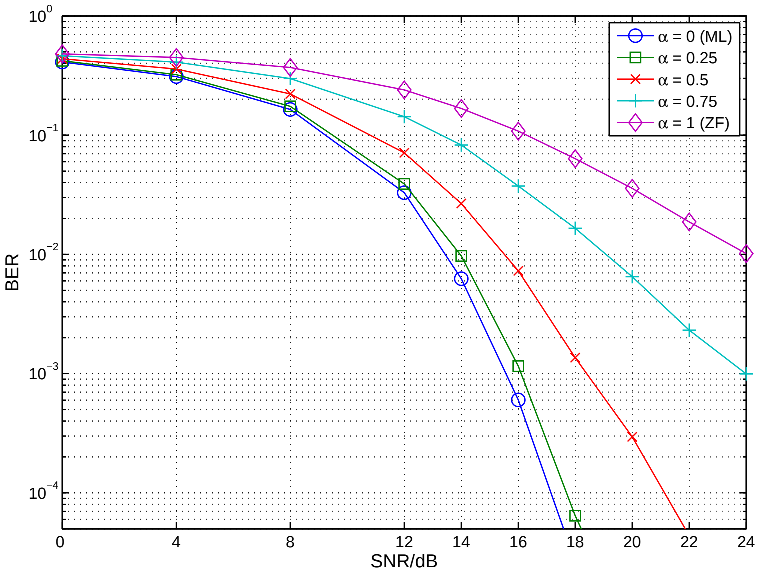


Figure 5.14: *Schnorr-Euchner sphere decoder with partial equalizer: bit eror rate (BER) versus SNR for different* α*.*

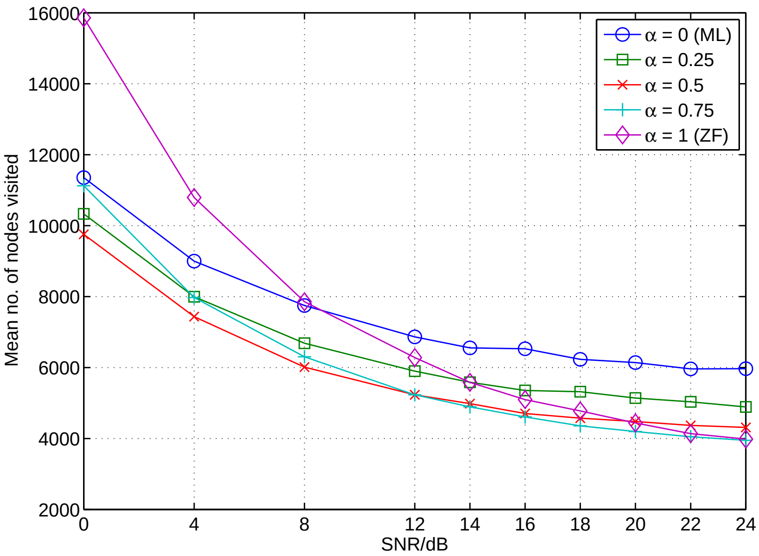


Figure 5.15: *Schnorr-Euchner sphere decoder with partial equalizer: mean number of nodes visited versus SNR for different* α*.*

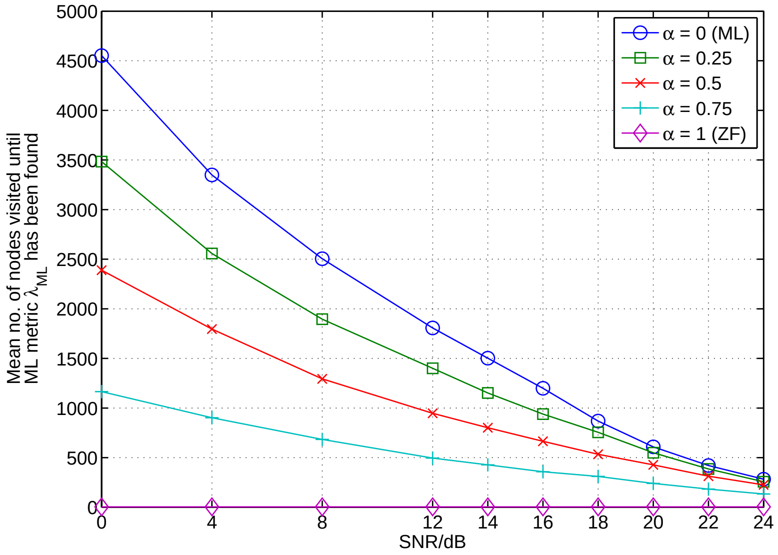


Figure 5.16: *Schnorr-Euchner sphere decoder with partial equalizer: mean number of nodes visited until ML metric λ^{ML} has been found versus SNR for different α.*

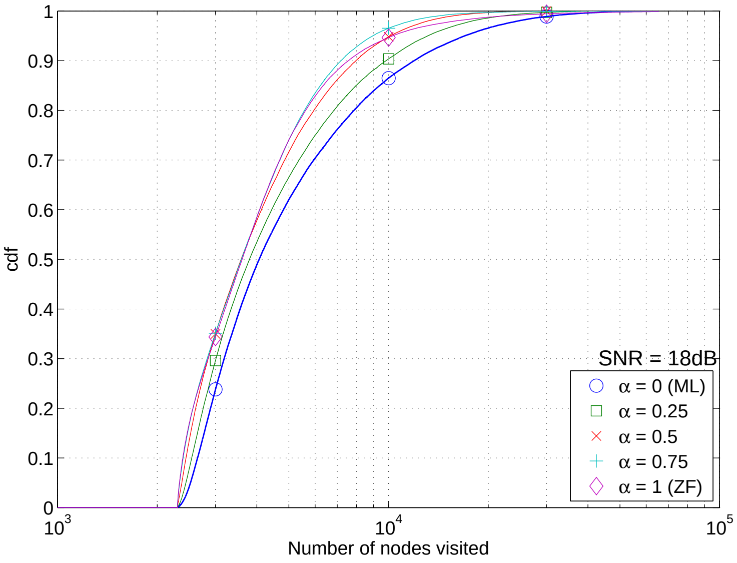


Figure 5.17: *Schnorr-Euchner sphere decoder with partial equalizer: cdf of the mean number of nodes visited for different α and an SNR of 18dB.*

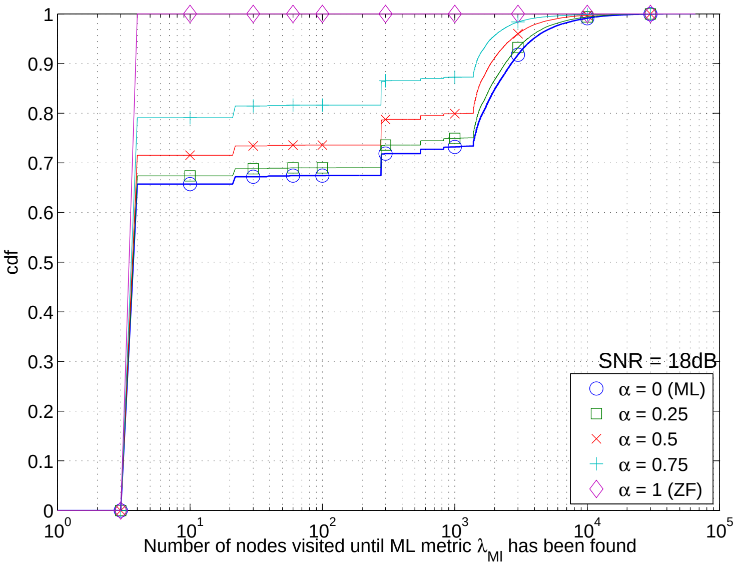


Figure 5.18: *Schnorr-Euchner sphere decoder with LLR clipping: cdf of the mean number of nodes visited until ML metric λ^{ML} has been found for different α and an SNR of 18dB.*

5.2.4 Schnorr-Euchner Sphere Decoder with Partial Equalizer and Lower Bound on Partial Metrics

The SESD using the partial equalizer was extended with the lower bound on partial metrics presented in Section 4.2. The influence of the amount of partial equalization, described by the parameter α, on the detection performance and the complexity of the detection was monitored.

Figure 5.19 shows the BER over the SNR for different values of the parameter α. We obtain the same performance as in the previous case without the lower bound on partial metrics. This is what we expected since the lower bound does not affect the detection performance. Figure 5.20 shows the mean detection complexity for different α and Figure 5.21 shows the mean number of nodes visited until the ML metric λ^{ML} has been found for different α, which corresponds to the search complexity of the hard detection. Figures 5.22 and 5.23 show the cdf of the nodes visited and the nodes visited until the ML metric λ^{ML} has been found, respectively. For $\alpha = 0$ the use of the lower bound on partial metrics has only little effect on the complexity. This is similar to the observations for $L_{\max} = \infty$ in Section 5.2.2. For all simulated values of α the complexity is reduced noticeable for lower SNR but for larger SNR values the reduction in complexity is quite small. It seams that the lower bound on partial metrics combats the influence of strong noise enhancement in the lower SNR region.

As in the previous simulation the parameter α can be used for a diversity-complexity trade-off, but with even lower detection complexity due to the lower bound on partial metrics.

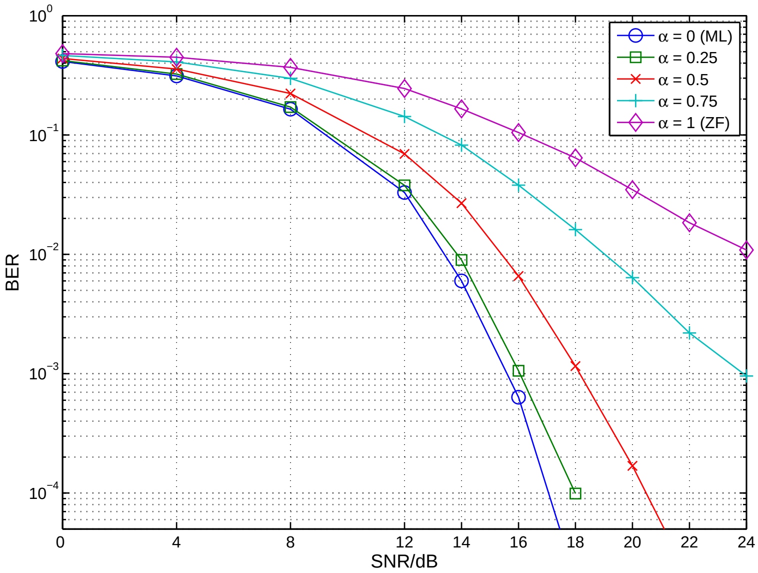


Figure 5.19: *Schnorr-Euchner sphere decoder with partial equalizer: bit eror rate (BER) versus SNR for different α with lower bound on partial metrics.*

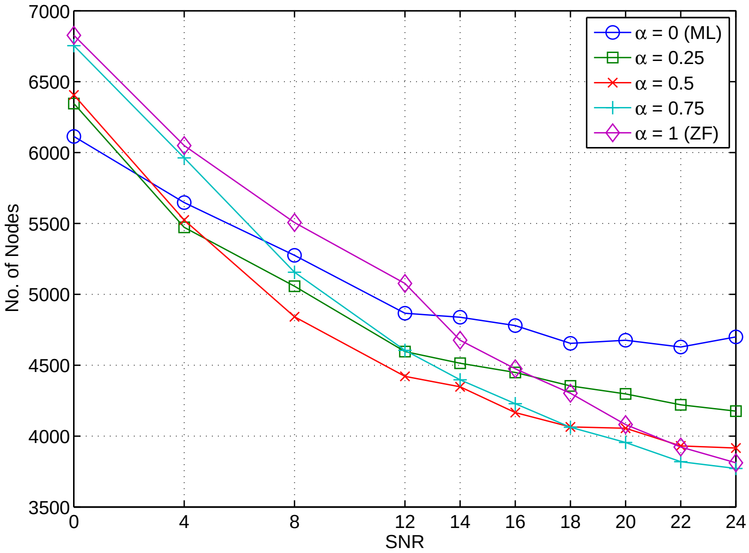


Figure 5.20: *Schnorr-Euchner sphere decoder with partial equalizer: mean number of nodes visited versus SNR for different α with lower bound on partial metrics.*

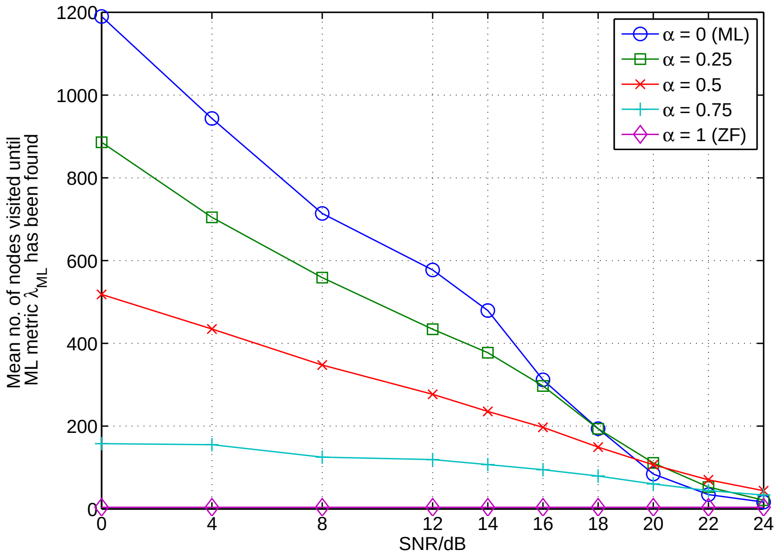


Figure 5.21: *Schnorr-Euchner sphere decoder with partial equalizer: mean number of nodes visited until ML metric λ^{ML} has been found versus SNR for different α with lower bound on partial metrics.*

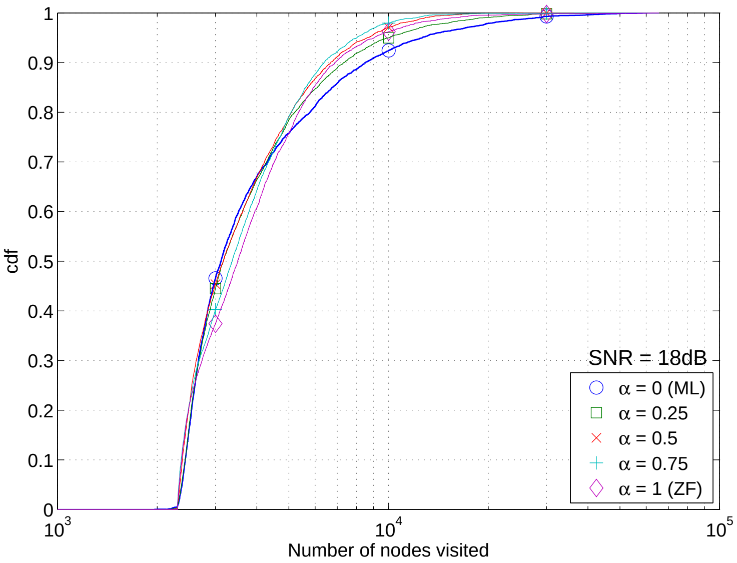


Figure 5.22: *Schnorr-Euchner sphere decoder with partial equalizer: cdf of the mean number of nodes visited for different α with lower bound on partial metrics and an SNR of 18dB.*

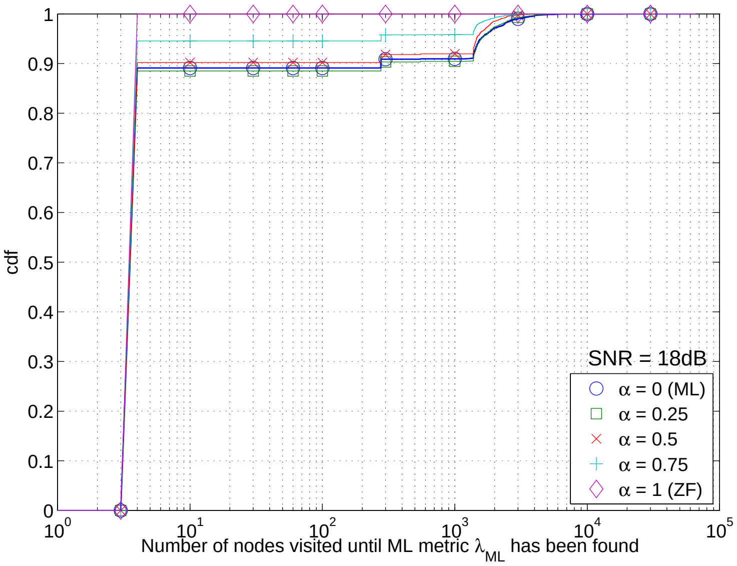


Figure 5.23: *Schnorr-Euchner sphere decoder with LLR clipping: cdf of the mean number of nodes visited until ML metric λ^{ML} has been found for different α with lower bound on partial metrics and an SNR of 18dB.*

5.2.5 Schnorr-Euchner Sphere Decoder with LLR Clipping, Partial Equalizer and Lower Bound on Partial Metrics

The SESD was implemented including both LLR clipping and the partial equalizer. The lower bound on partial metrics was implemented as well because its contribution to the reduction came at no performance costs in the previous simulations. The influence of the two parameters α and $L_{\max}$ on the detection performance an the complexity of the detection was monitored.

Figures 5.24, 5.25, and 5.26 show the BER, the mean number of nodes visited, and the mean number of nodes visited until the ML metric λ^{ML} has been found, respectively, versus the SNR for different values of the parameter α and the clipping level $L_{\max}$. The Figures 5.27 and 5.28 show the cdf of the number of nodes visited and the cdf of the number of nodes visited until the ML metric λ^{ML} has been found, respectively. Figures 5.29, 5.30, 5.31, 5.32, 5.33 and Figures 5.34, 5.35, 5.36, 5.37, 5.38 show the same values for the clipping levels $L_{\max} = 2$ and $L_{\max} = 0.2$, respectively.

It can be seen, that the partial equalizer together with LLR clipping does not perform as expected. Clipping the LLRs to a level of $L_{\max} = 8$ for instance (cf. Figures 5.24 to 5.28) does not change much in the detection performance but influences its complexity, which is lower than without LLR clipping but gets larger with increasing α. However, the complexity for finding the ML metric λ^{ML} is reduced as well by the LLR clipping and still gets lower with increasing α. It can therefore be stated that the LLR clipping interferes the function of the partial equalizer which does not offer a complexity-diversity trade-off anymore. Lowering the clipping level to $L_{\max} = 2$ (cf. Figures 5.29 to 5.33) and even further to $L_{\max} = 0.2$ (cf. Figures 5.34 to 5.38) worsens the detection performance for $\alpha = 0$ (ML) but slightly improves the detection performance for $\alpha = 1$ (ZF). At the same time the complexity is reduced and, as for $L_{\max} = 8$, it rises with increasing α.

Figure 5.39 compares the necessary SNR for $BER = 10^{-3}$ for different values of α and

$L_{\max}$. It can be seen that for constant $L_{\max}$ the performance degrades with increasing α, but for higher values of α low clipping levels allow a better detection performance. Figure 5.40 compares the corresponding complexities. Without any LLR clipping the complexity decreases with increasing α but when applying LLR clipping increasing α degrades the detection complexity. All these facts are summarised in Figure 5.41 that shows the necessary mean number of nodes versus the necessary SNR for $BER = 10^{-3}$. Without LLR clipping increasing α leads to a reduction in complexity and a degradation in detection performance, i.e. a rise in necessary SNR for the desired BER. When applying LLR clipping increasing α leads to an increase in both the complexity and the necessary SNR.

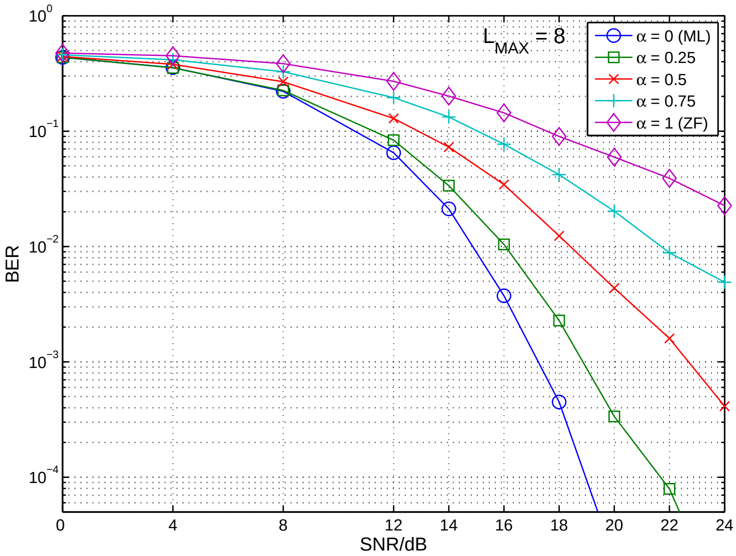


Figure 5.24: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: bit eror rate (BER) versus SNR for different α with lower bound on partial metrics and $L_{\max} = 8$.*

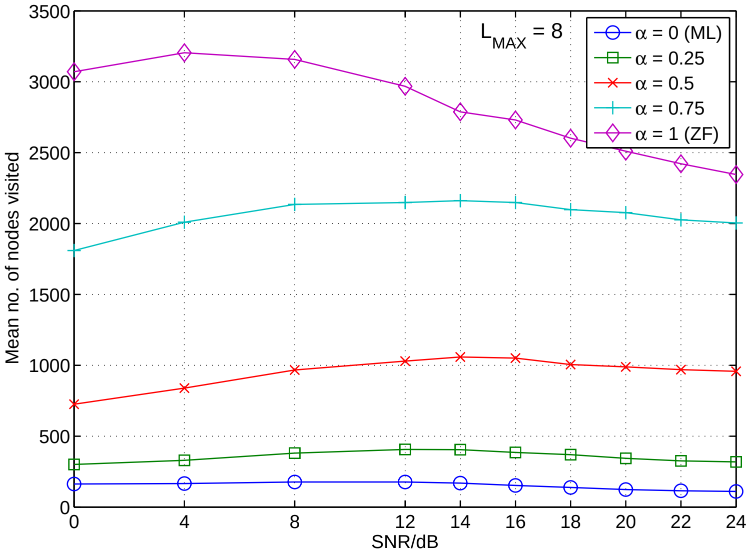


Figure 5.25: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: mean number of nodes visited versus SNR for different α with lower bound on partial metrics and $L_{\max} = 8$.*

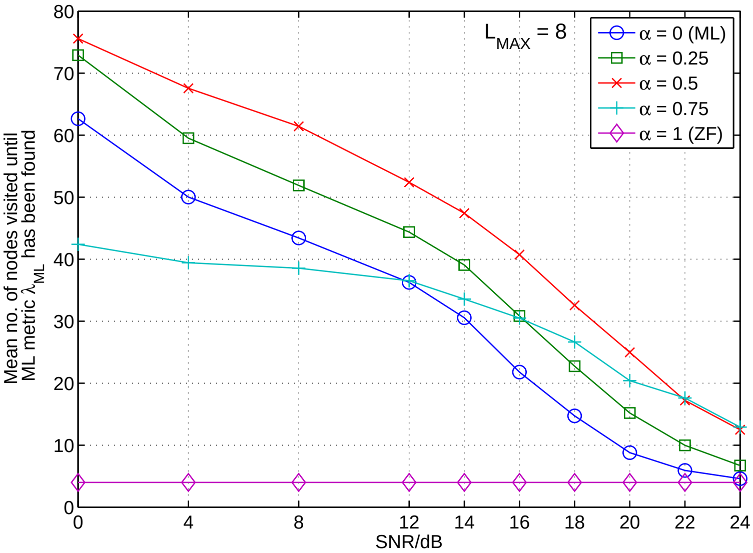


Figure 5.26: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: mean number of nodes visited until ML metric λ^{ML} has been found versus SNR for different α with lower bound on partial metrics and $L_{\max} = 8$.*

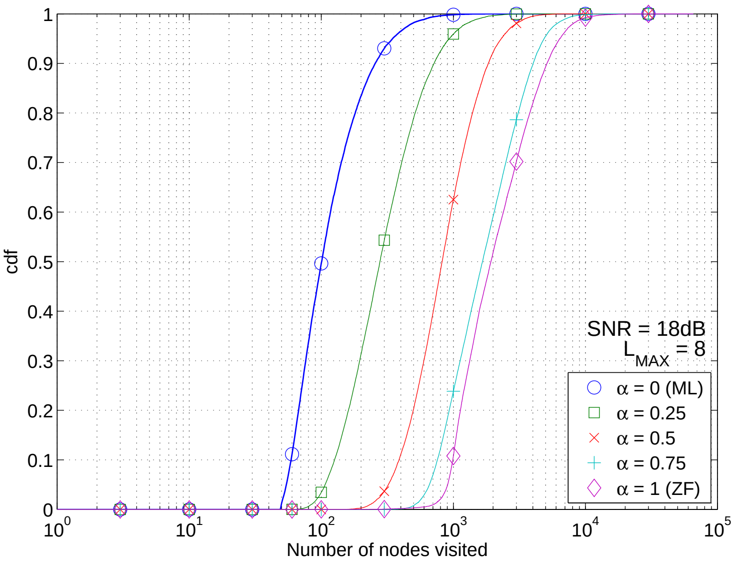


Figure 5.27: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: cdf of the mean number of nodes visited for different α with lower bound on partial metrics, an SNR of 18dB, and a clipping level of $L_{\max} = 8$.*

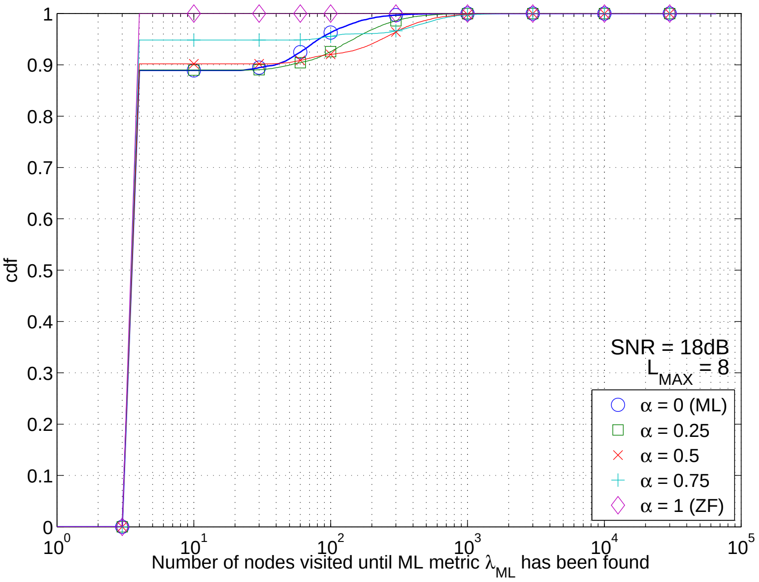


Figure 5.28: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: cdf of the mean number of nodes visited until ML metric λ^{ML} has been found for different α with lower bound on partial metrics, an SNR of 18dB and a clipping level of $L_{\max} = 8$.*

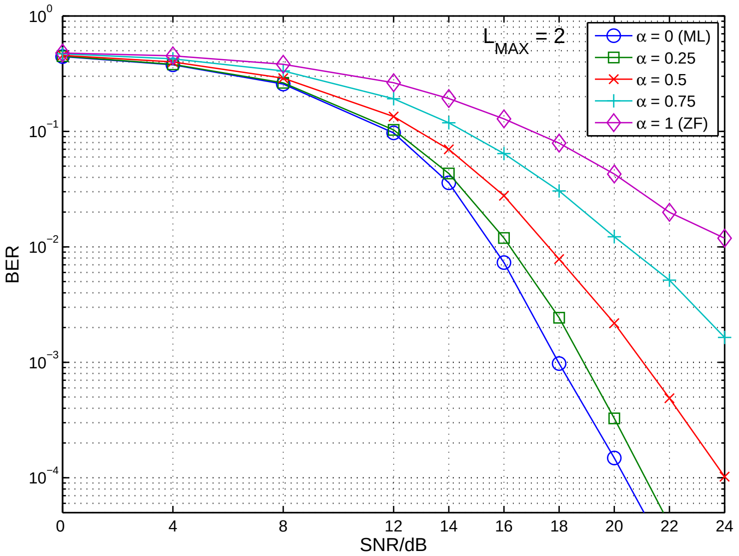


Figure 5.29: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: bit eror rate (BER) versus SNR for different α with lower bound on partial metrics and $L_{\max} = 2$.*

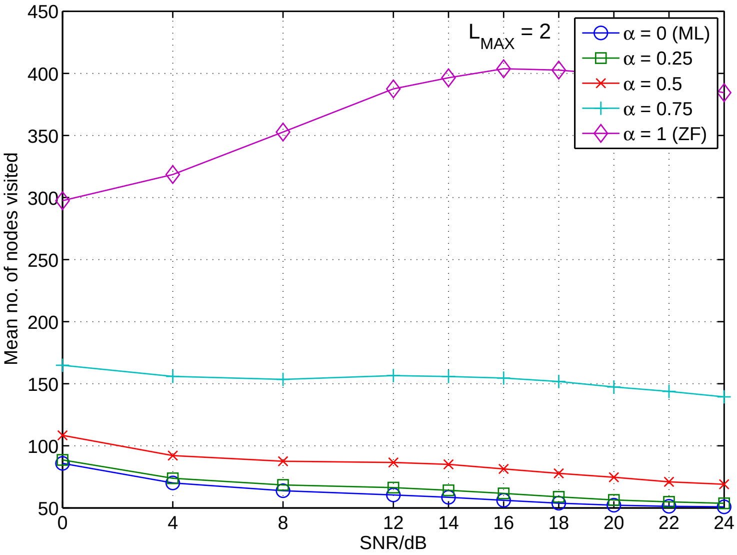


Figure 5.30: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: mean number of nodes visited versus SNR for different α with lower bound on partial metrics and $L_{\max} = 2$.*

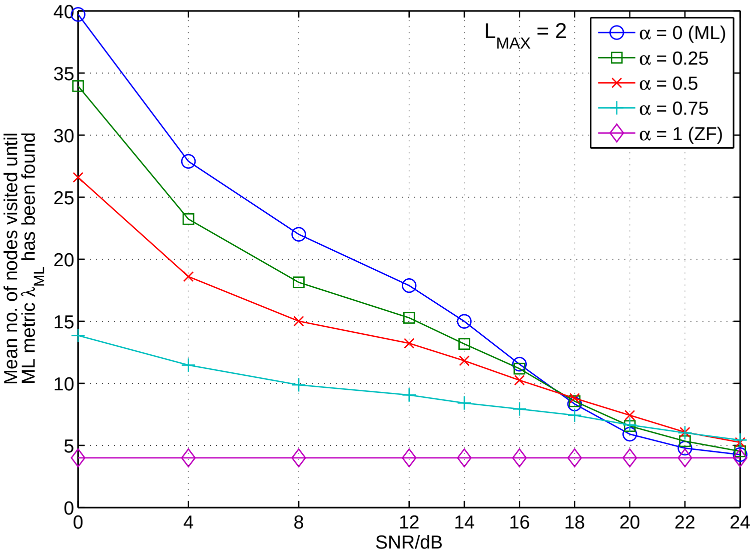


Figure 5.31: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: mean number of nodes visited until ML metric λ^{ML} has been found versus SNR for different α with lower bound on partial metrics and $L_{\max} = 2$.*

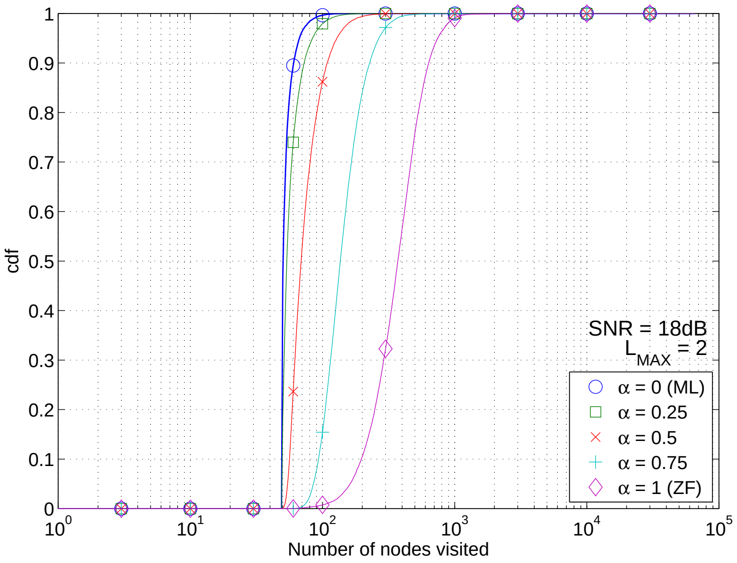


Figure 5.32: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: cdf of the mean number of nodes visited for different α with lower bound on partial metrics, an SNR of 18dB, and a clipping level of $L_{\max} = 2$.*

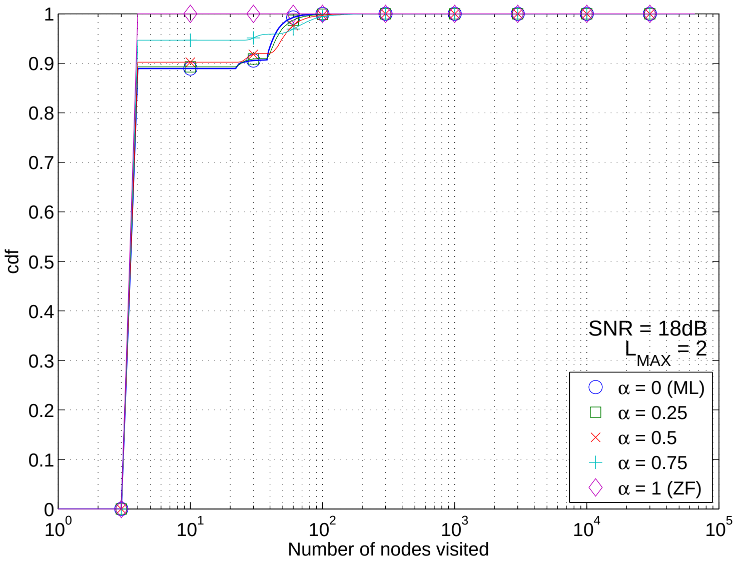


Figure 5.33: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: cdf of the mean number of nodes visited until ML metric λ^{ML} has been found for different α with lower bound on partial metrics, an SNR of 18dB and a clipping level of $L_{\max} = 2$.*

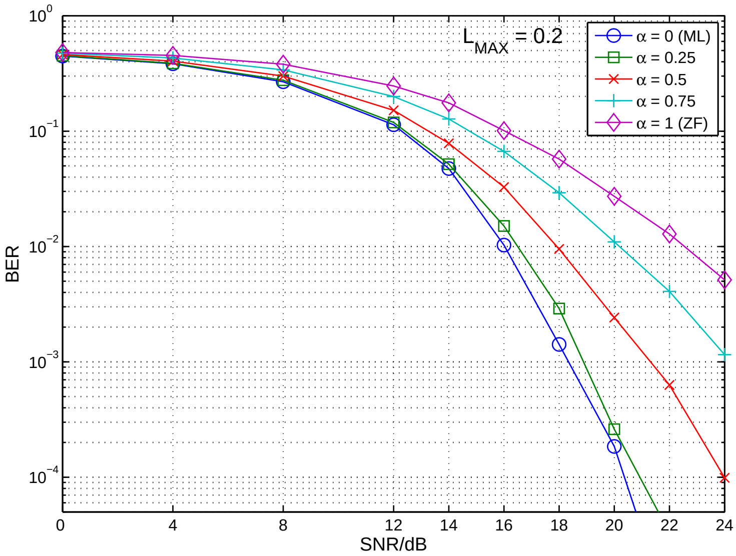


Figure 5.34: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: bit eror rate (BER) versus SNR for different α with lower bound on partial metrics and $L_{\max} = 0.2$.*

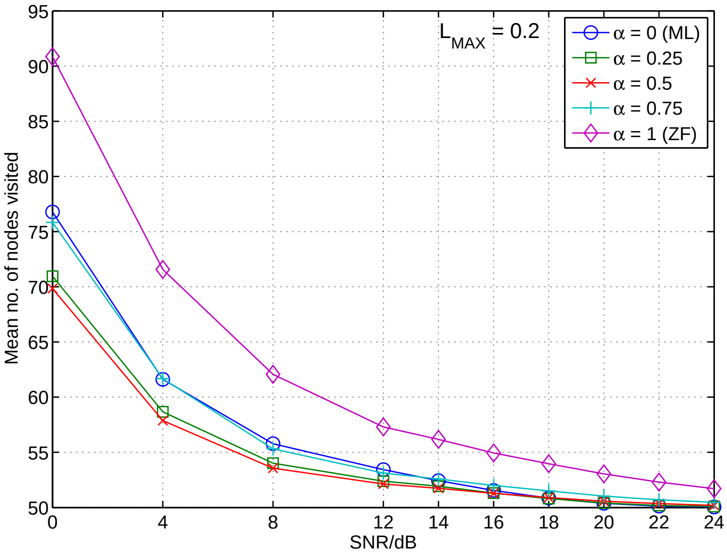


Figure 5.35: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: mean number of nodes visited versus SNR for different α with lower bound on partial metrics and $L_{\max} = 0.2$.*

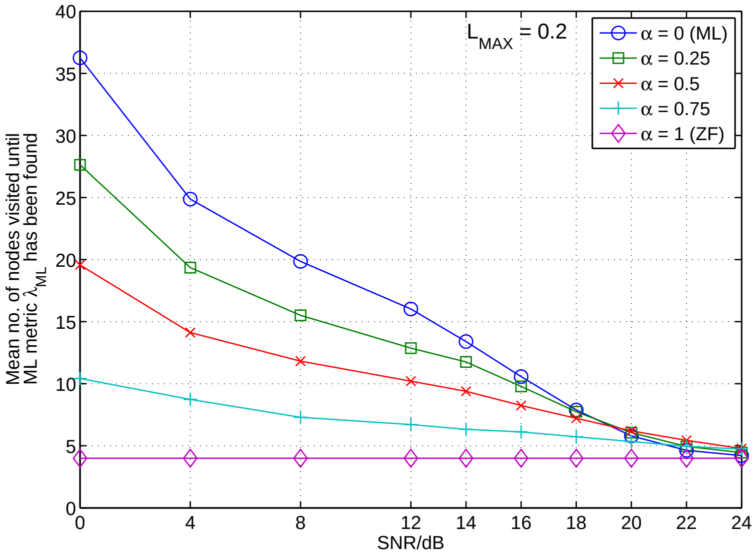


Figure 5.36: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: mean number of nodes visited until ML metric λ^{ML} has been found versus SNR for different α with lower bound on partial metrics and $L_{\max} = 0.2$.*

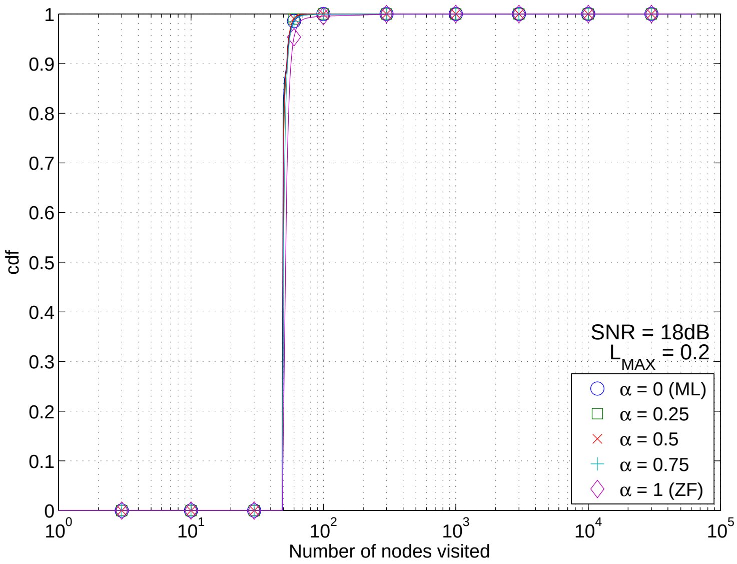


Figure 5.37: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: cdf of the mean number of nodes visited for different α with lower bound on partial metrics, an SNR of 18dB, and a clipping level of $L_{\max} = 0.2$.*

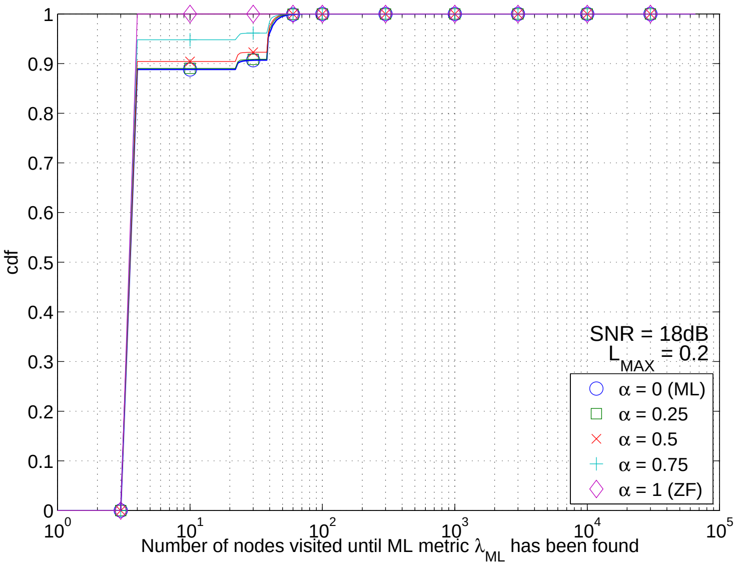


Figure 5.38: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: cdf of the mean number of nodes visited until ML metric λ^{ML} has been found for different α with lower bound on partial metrics, an SNR of 18dB and a clipping level of $L_{\max} = 2$.*

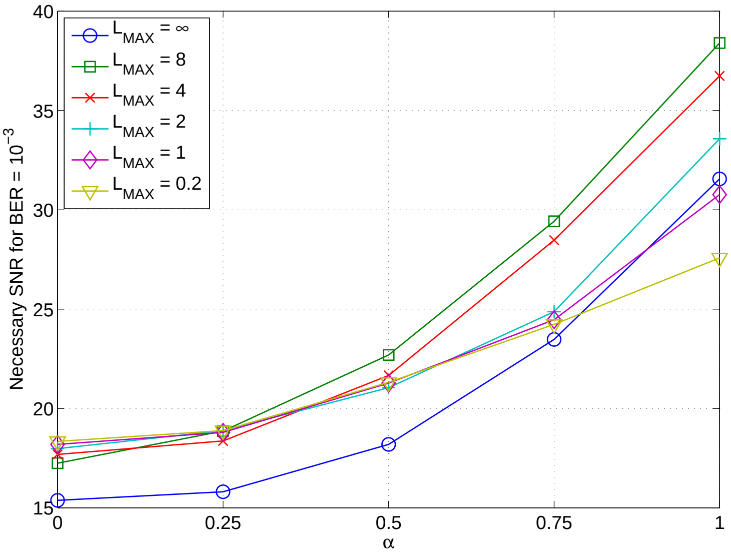


Figure 5.39: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: necessary SNR for BER $= 10^{-3}$ versus α for different $L_{\max}$ with lower bound on partial metrics.*

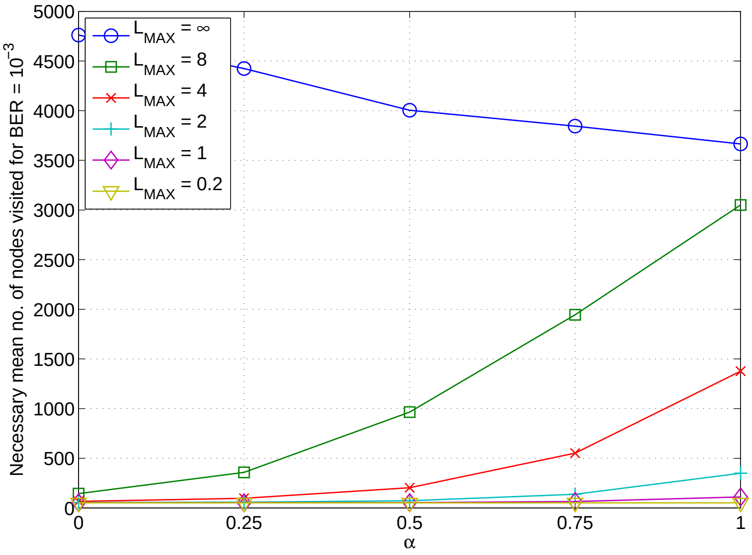


Figure 5.40: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: necessary mean number of nodes for BER* $= 10^{-3}$ *versus* α *for different* $L_{\max}$ *with lower bound on partial metrics.*

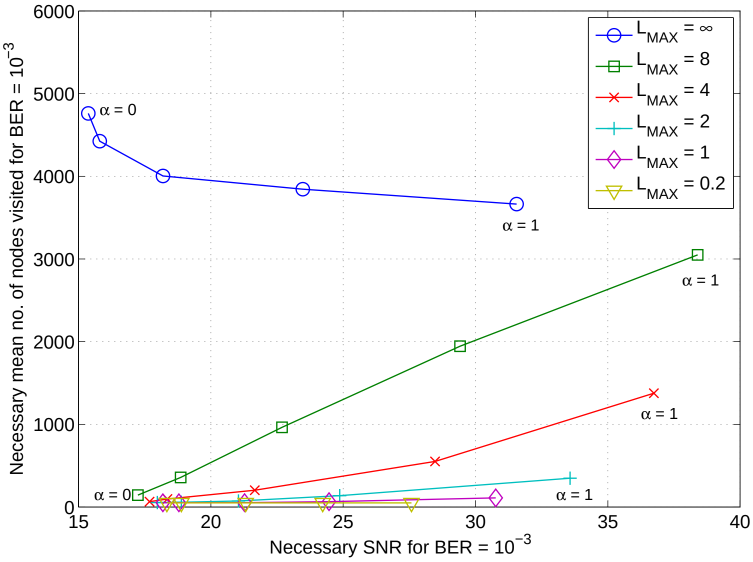


Figure 5.41: *Schnorr-Euchner sphere decoder with partial equalizer and LLR clipping: necessary mean number of nodes versus necessary SNR for BER* $= 10^{-3}$ *for different* α *and* $L_{\max}$ *with lower bound on partial metrics.*

Chapter 6

Summary and Outlook

The thesis dealt with soft detection for MIMO systems and the sphere decoder as the most common algorithm used for detection. To combat the high computational complexity of the original algorithm modifications were presented and their performance was evaluated by simulations. The first modification, LLR clipping, allows a flexible trade-off between performance and complexity by adjustment of the clipping level. The second modification, a partial equalizer together preceding the SD, allows a flexible trade-off between diversity and complexity by adjustment of the amount of equalization. The third modification, a lower bound on partial metrics, allows a complexity reduction without affecting the detection performance. The simulation results show that the modifications work as desired if they are applied independently of each other. However, a combination of the partial equalizer and LLR clipping does not result in the expected complexity reduction. It is likely that this is a result of too aggressive clipping. To approve this assumption and find out an appropriate clipping level for the combination of the partial equalizer with LLR clipping (potentially depending on the SNR) further investigation and simulations are necessary.

Bibliography

[1] A. F. Molisch, *Wireless Communications.* Wiley & Sons, 2005.

[2] J. G. Proakis, *Digital Communications*, 3rd ed. New York: McGraw-Hill, 1995.

[3] H. Bölcskei and A. J. Paulraj, "Multiple-input multiple-output (MIMO) wireless systems," in *The Communications Handbook*, 2nd ed., J. D. Gibson, Ed. Boca Raton, FL: CRC Press, 2002.

[4] F. Hlawatsch, "Modulation and detection techniques," Lecture Notes, March 2006.

[5] B. M. Hochwald and S. ten Brink, "Achieving near-capacity on a multiple-antenna channel," *IEEE Trans. Inf. Theory*, vol. 51, no. 3, pp. 389–399, March 2003.

[6] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Trans. Inf. Theory*, vol. 20, pp. 284–287, March 1974.

[7] U. Fincke and M. Pohst, "Improved methods for calculating vectors of short length in a lattice, including a complexity analysis," *Mathematics of Computation*, vol. 44, no. 170, pp. 463–471, Apr. 1985.

[8] R. A. Horn and C. R. Johnson, *Matrix Analysis.* Cambridge (UK): Cambridge Univ. Press, 1999.

[9] E. Agrell, T. Eriksson, A. Vardy, and K. Zeger, "Closest point search in lattices," *IEEE Trans. Inf. Theory*, vol. 48, no. 8, pp. 2201–2214, Aug. 2002.

[10] C. Studer, M. Wenk, A. Burg, and H. Bölscskei, "Soft-output sphere decoding: Performance and implementation aspects," *ACSSC '06 Fortieth Asilomar Conference on Signals, Systems and Computers*, 2006.

[11] M. S. Yee, "Max-log-map sphere decoder," *Proc. IEEE ICASSP 2005*, vol. 3, pp. 1013–1016, Mar. 2005.

[12] J. Jaldén and B. Ottersten, "On the complexity of sphere decoding in digital communications," *IEEE Trans. Signal Processing*, vol. 53, no. 4, pp. 1474–1484, Apr. 2005.

[13] J. Maurer, G. Matz, and D. Seethaler, "On the diversity-complexity tradeoff in MIMO spatial multiplexing systems," *ACSSC '06 Fortieth Asilomar Conference on Signals, Systems and Computers*, pp. 2077–2081, Oct.–Nov. 2006.

[14] J. Maurer, J. Jaldén, D. Seethaler, and G. Matz, "Achieving a continuous diversity complexity tradeoff using mismatched maximum likelihood detection," *in preparation*, 2007.

[15] A. Paulraj, R. U. Nabar, and D. Gore, *Introduction to space-time wireless communications.* Cambridge (UK): Cambridge Univ. Press, 2003.

[16] G. H. Golub and C. F. Van Loan, *Matrix Computations*, 3rd ed. Baltimore: Johns Hopkins University Press, 1996.

[17] E. Biglieri, G. Taricco, and A. Tulino, "Performance of space-time codes for a large number of antennas," *IEEEIT*, vol. 48, no. 9, pp. 1794–1803, 2002.

[18] D.A. Gore, J. R. W. Heath, and A.J. Paulraj, "Transmit selection in spatial multiplexing systems," *IEEE Comm. Letters*, vol. 6, no. 11, pp. 491–493, November 2002.

[19] U. Fincke and M. Phost, “Improved methods for calculating vectors of short length in a lattice, including a complexity analysis,” *Math. Comp.*, vol. 44, pp. 463–471, April 1985.

[20] M. Damen, A. Chkeif, and J. Belfiore, “Lattice code decoder for space-time codes,” *IEEE Comm. Letters*, vol. 4, no. 5, pp. 161–163, May 2000.

[21] M. Stojnic, H. Vikalo, and B. Hassibi, “A branch and bound approach to speed up the sphere decoder,” *Proc. IEEE ICASSP 2005*, pp. 429–433, May 2005.