Comparison of Optimal and Near-Optimal Detection in GSM/EDGE

M.Sc.E.E. Thesis
by
Morten Hansen

Informatics and Mathematical Modelling
Technical University of Denmark

Abstract

This thesis deals with optimal and near-optimal detection methods in GSM and EDGE. It is shown that the Maximum A Posteriori (MAP) receiver, which is the optimal receiver from a Bit Error Rate (BER) point of view, is capable of achieving a considerable gain in BER performance compared to the linear receivers. However, the complexity of the optimal receiver grows exponentially with the channel length and the number of transmitters, leading to an unrealistically high complexity in a Mobile Station (MS). When the Signal-to-Noise Ratio (SNR) is sufficiently high, a computationally efficient way of achieving Maximum Likelihood Sequence Estimation (MLSE), is by using Sphere Decoding (SD). It is described how the SD algorithm can be combined with the MAP receiver to obtain near-optimal Symbol-by-Symbol detection by forming approximate bit posteriors.
Various methods of choosing the radii for SD will be investigated and different search strategies through the trellis diagram are examined. A method to limit the complexity is also proposed, which will be of great interest in an actual implementation in a MS.
The simulation results concerning both the BER performance and the complexity of the implemented algorithms are presented. From the results it is seen that SD algorithms are capable of reducing the complexity significantly at high SNR, while still obtaining near-optimal BER performance.

Keywords: Wireless Communication, GSM, EDGE, MAP, MLSE, Multi User Detection, Sphere Decoding, Increasing Radii Search, Schnorr-Euchner search.

Resumé (Abstract in Danish)

Denne kandidatafhandling omhandler optimal- og nær-optimal detektion i GSM og EDGE.
Det vises, at Maximum A Posteriori (MAP) modtageren, hvilket er den optimale detektor set ud fra en Bit Error Rate (BER) betragtning, er i stand til at give en signifikant bedre BER performance sammenlignet med lineære modtagere. Desværre vil kompleksiteten for den optimale modtager vokse eksponentielt med kanallængden samt antallet af brugere, der ønskes detekteret. Dette bevirker at kompleksiteten bliver særdeles høj og dermed urealistisk at implementere i en mobil telefon. Når signal-støj-forholdet (SNR) er tilstrækkelig højt, vil Sphere Decoding være en beregningsmæssig effektiv måde at foretage Maximum Likelihood Sequence Estimation (MLSE) på. Det er beskrevet hvordan SD algoritmen kan kombineres med MAP modtageren for derved at opnå en nær-optimal Symbol-by-Symbol detektion. Dette gøres ved at approksimere posteriori sandsynligheden af symbolerne.
Adskillige metoder vil blive undersøgt til at bestemme radierne for den sfære som SD algoritmen benytter sig af, og forskellige søgnings-strategier gennem trellis diagrammet vil ligeledes blive undersøgt. Endvidere vil der blive foreslået en metode til at fastsætte en øvre grænse for kompleksiteten, hvilket vil være særdeles interessant i en egentlig implementation af SD algoritmen i en mobil telefon.
Der vil blive præsenteret simuleringsresultater vedrørende både BER performance og kompleksiten af de implementerede algoritmer. Ud fra resultaterne kan det ses, at SD algoritmerne er i stand til at reducere kompleksiteten signifikant ved højt SNR, samtidig med at der stadig opnås nær-optimal BER performance.

Nøgleord: Trådløs kommunikation, GSM, EDGE, MAP, MLSE, Multi User Detection, Sphere Decoding, Increasing Radii Search, Schnorr-Euchner søgning.

Preface

This thesis is written in order to achieve the degree Master of Science. The project has been carried out in a period of 9 month from September the 6th, 2005 to June 7th, 2006 in Modem System Design (MSD), Copenhagen, Nokia Technology Platforms. The workload corresponds to 45 ECTS points. The supervisors of the project are Lars P. B. Christensen, Zoltan Safar, Nokia and Ole Winther, Informatics and Mathematical Modelling, Technical University of Denmark (DTU).

First of all I would like to thank the supervisors, who have been helpful and given me guidance throughout the project. A special thank to Saad Akram and Morten Hagdrup for interesting discussions related to the subjects in this thesis. Likewise a special thank to Pedro Højen-Sørensen for taking his time to review this thesis and for the interesting discussions regarding communication in general. I would also like to thank engineering manager of MSD Niels Mørch, who gave me the opportunity to work on the project in his group.

Copenhagen, June the 7th, 2006

_____

Morten Hansen

Mail to: s001971@student.dtu.dk

Contents

List of Abbreviations

ACI	Adjacent Channel Interference
AMP	Amplitude Modulated Pulse
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Cocke, Jelinek, and Raviv
BER	Bit Error Rate
BLER	Block Error Rate
BPSK	Binary Phase Shift Keying
BS	Base Station
DDFSE	Delayed Decision-Feedback Sequence Estimation
DF	Degrees of Freedom
CCI	Co-Channel Interference
CDF	Cumulative Density Function
CIR	Carrier to Interference Ratio
CPM	Continuous Phase Modulation
EDGE	Enhanced Data rates for Global Evolution
FDMA	Frequency Division Multiple Access
GMSK	Gaussian Minimum Shift Keying
GSM	Group Spécial Mobile (a.k.a. Global System for Mobile communications)
HMM	Hidden Markov Model
HT	Hilly Terrain
IC	Interference Cancellation
IRS	Increasing Radii Search
ISI	Intersymbol Interference
LMMSE	Linear Minimum Mean Square Error
LS	Least Squares
MAP	Maximum A Posteriori
MIMO	Multiple-Input Multiple-Output
MLSE	Maximum Likelihood Sequence Estimation

MS	Mobile Station
MSE	Mean Square Error
MSK	Minimum Shift Keying
MUD	Multi User Detection
PDF	Probability Density Function
PSK	Phase Shift Keying
OQPSK	Offset Quadrature Phase Shift Keying
SD	Sphere Decoding
SE	Schnorr-Euchner
SE_S	Modified ES search strategy with a maximum constraint on the number of states
SbS	Symbol-by-Symbol
SNR	Signal to Noise Ratio
SUD	Single User Detection
TDMA	Time Division Multiple Access
TS	Training Sequence
TU	Typical Urban

1 Introduction

There is a constant evolution in wireless communication and for each day wireless technologies become more and more widespread. When “new” wireless technologies appear (such as wireless LANs for hot spot coverage), the demandings for the existing cellular systems increase, since they have to be competitive. Hence, the enhancement of capacity in the existing systems are of great importance.
One limiting factor in the GSM and EDGE systems is the interference from other users. Therefore, it is interesting to examine techniques which perform interference cancellation. Multi User Detection (MUD) is one way to make the mobile station (MS) more robust to interference. Another way of increasing the capacity is by improving the performance of the detectors.
The solutions of the optimal detectors are well known but unfortunately, they require a huge complexity, especially for higher-order modulation techniques and in the case of MUD. Thus, the implementation of the optimal detector in a MS is quite unrealistic. Therefore, an active research topic in mobile communication is to reduce the complexity of the receiver as much as possible, and still achieve near-optimal performance.

One promising method for performing Maximum Likelihood Sequence Estimation (MLSE) in a computational efficient way is the Sphere Decoding (SD) algorithm. When the requirement regarding optimal MLSE detection is relaxed in the SD algorithm, a considerably reduction in complexity can be expected. Therefore, it is interesting to examine the performance of the SD algorithm in the GSM and EDGE systems. The complexity reduction is especially needed for EDGE, since it has the largest alphabet and therefore will have the largest complexity, when the optimal detection is made.

1.1 Outline of the Project

As it has been described above the aim of this project is to investigate near-optimal detection in mobile communication. The majority of the project has been related to sphere decoding. However, to evaluate the performance of the sphere decoding algorithm the performance of the optimal receivers is likewise investigated. It is especially, the Maximum A Posteriori (MAP) detector, which should be investigated in detail. However, also the MLSE detection will be discussed in this thesis.

The principle of SD should be examined in general. However, the emphasis of the investigation of the SD algorithm is related to frequency-selective channels, because such channel conditions will be present in the GSM/EDGE system. Since the SD algorithm uses spheres in its searching for a solution, the problem of choosing the radius will be addressed. Various methods for finding a reasonable radius of the sphere should be studied. Furthermore, different search strategies must be examined.
To get a reasonable idea of the performance of the SD, the proposed methods are tested in different scenarios. Both the BER performance should be determined as well as the complexity of the algorithms.

1.2 Organization of the Thesis

In Section 2 a description of the GSM and EDGE in general terms is given. Here modulation schemes and frame formats for GSM and EDGE are among others found. In Section 3 the multipath radio channel and a signal model of the channel will be treated, which will be used throughout the rest of the thesis. The optimal detectors will be addressed in Section 4 and some simulation results for these is likewise given. Section 5 contains a thorough investigation of sphere decoding. It will primarily be the theoretical issues that are found here, while Section 6 presents the simulation results obtained for the examined SD algorithms. In Section 6 comparisons between the optimal detector and the SD algorithms are given. Finally, Section 7 concludes the thesis and relevant future research topics are presented.

2 GSM and EDGE

In this section a brief introduction to GSM and EDGE is given. It is primarily the modulation technique and the frame format for GSM and EDGE systems which are described, since these subjects are relevant when simulations of receiver performance in GSM and EDGE systems should be carried out.

2.1 GSM

The GSM system relies both on Time Division Multiple Access (TDMA) and Frequency Division Multiple Access (FDMA), which makes it possible to provide network access to several users (subscribers) at the same time. The mobile network is divided into cells, where each cell has a specific frequency for wireless communication (actually it has two, one for uplink and one for downlink). Since the frequency band is a limited resource, the available frequencies are reused in other cells. The frequency reuse will however have the drawback that Co-Channel Interference (CCI) occurs. As the name indicates, it is due to the fact that some of the co-channel cells are using the same frequencies and interference between the signals will arise.
Another type of interference is Adjacent Channel Interference (ACI), which is caused by adjacent channels that have frequencies which overlap their allocated frequency spectrum. In this thesis it will however primarily be CCI which is in focus, since this typically has the most destructive effect on the signal.

2.1.1 Modulation Technique in GSM

The modulation technique used in GSM is Gaussian Minimum Shift Keying (GMSK). To understand GMSK, a description of Minimum Shift Keying (MSK) is first given. This is because GMSK can be viewed as an expansion of the MSK modulation technique.

MSK
MSK modulates the phase of the signal. Unlike e.g. M-PSK, the absolute value of phase shift between two successive symbols will always be the same in MSK. This type of modulation is called Continuous Phase Modulation (CPM). In [4, p. 70] it is shown that this type of modulated signal can be written as

$$s_{MSK}(t) = A \cdot \cos\left(2\pi f_0 t + \varphi_{MSK}(t)\right), \tag{2.1.1}$$

where A is the amplitude of the signal (which is independent of the transmitted bits), f_0 is the carrier frequency. $\varphi_{MSK}(t)$ is the phase of the modulated signal given by

$$\varphi_{MSK}(t) = 2\pi h \sum_{k=-\infty}^{\infty} d_k q_{MSK}(t - kT), \tag{2.1.2}$$

here $d_k \in \{\pm 1\}$ is differentially encoded bits at time k. I.e. $d_k = (I_k \text{ XOR } I_{k-1})$, where $I_k \in \{\pm 1\}$ is the bit. h is called the modulation index, which is $h = ½$ in MSK. T is the symbol period and $q_{MSK}(t)$ is determined by integration of a pulse, $g_{MSK}(t)$,

$$q_{MSK}(t) = \int_0^t g_{MSK}(\tau)\,d\tau, \quad where \quad \lim_{t\to\infty} q_{MSK}(t) = ½. \tag{2.1.3}$$

$g_{MSK}(t)$ is a rectangular pulse, (see e.g. [9, p. 16]), i.e.

$$g_{MSK}(t) = \frac{1}{2T} rect_T\left(t - \frac{t}{2}\right). \tag{2.1.4}$$

An illustration of the pulse shape of $g_{MSK}(t)$ and $q_{MSK}(t)$, which is taken from [9, p. 17], is given in Figure 2.1.

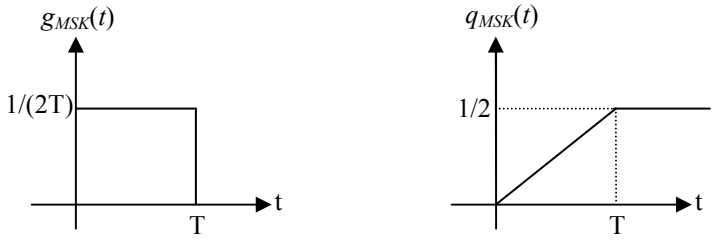

Figure 2.1. Pulse shape of $g_{MSK}(t)$ and $q_{MSK}(t)$.

By using the fact that $d_k \in \{\pm 1\}$ and by looking at (2.1.1) to (2.1.4) it is seen, that the change in phase for a MSK modulated signal, can be expressed as

$$2\pi h \int_0^t g_{MSK}(\tau)\,d\tau\,. \tag{2.1.5}$$

By examining (2.1.5) it becomes clear that the absolute value of the phase shift between two symbols is constant over time, and the difference is $\pm\pi/2$. The constant phase shift between two successive symbols makes it a spectrally efficient modulation method. The constellation diagram for the MSK signal is shown in Figure 2.2, (ref. [8, p. 20]).

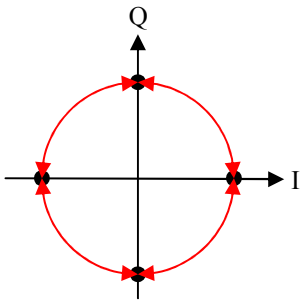

Figure 2.2. Constellation diagram for MSK modulation.

From Figure 2.2 it is seen that the envelope of the signal will not be zero at any time. This is another advantage of MSK modulation, since it makes it possible to design power-efficient amplifiers.

GMSK

To obtain a more bandwidth-efficient modulation method than MSK, GMSK is used in GSM. As mentioned earlier GMSK is closely related to MSK. The difference between the two modulation schemes is that the pulse, $g_{GMSK}(t)$, in GMSK is a rectangular pulse convolved with a Gaussian filter. In [9, p. 18] it is shown that $g_{GMSK}(t)$ is given as

$$g_{GMSK}(t) = rect_T\left(t - \frac{T}{2}\right) * h_{GMSK}(t) \tag{2.1.6}$$

where $h_{GMSK}(t)$ is a Gaussian pulse given as

$$h_{GMSK}(t) = \left(\frac{1}{\sqrt{2\pi}\sigma T}\right)\exp\left(-\frac{t^2}{2\sigma^2 T^2}\right), \quad \sigma = \frac{\sqrt{\ln 2}}{2\pi BT}. \tag{2.1.7}$$

Here B is the 3-dB bandwidth of the Gaussian filter. In GSM the BT product is chosen to be $BT = 0.3$. A consequence of the convolution with the Gaussian filter is that the phase shift of GMSK (in contrast to MSK) is non-linear. Another effect of the filtering is that the power spectrum of the GMSK modulated signal will have a narrower main lope and the side lopes will be attenuated more. The phase function $\varphi_{GMSK}(t)$ of GMSK can be expressed as

$$\varphi_{GMSK}(t) = \pi \int_{-\infty}^{t} \sum_{k=-\infty}^{\infty} d_k g_{GMSK}(\tau - kT)\, d\tau. \tag{2.1.8}$$

From the phase function given in (2.1.8) it is seen that the symbols will not be fully separated from each other. Thus Intersymbol Interference (ISI) will occur in some degree in GMSK modulation.

Linearized GMSK

As shown in the previous subsection, GMSK is a non-linear modulation scheme. This non-linearity complicates the design of the receiver. Therefore, a linearized approximation of GMSK is typically used in the receiver. In [6] it is proven that a constant amplitude binary phase modulation can be represented by a summation of Amplitude Modulated Pulses (AMP). In fact it is shown that only one main pulse gives a good approximation of the GMSK signal. A detailed description of this can be found in [7 and 9, p. 21-29]. It is shown in [7 and 9] that the GMSK modulated signal can be approximated by using the main pulse, $C_0(t)$. In these references a more detailed characteristic of the main pulse is also given. In [8, p. 24] it is shown that the approximation of the GMSK modulated signal becomes

$$s_{GMSK}(t) \approx \sum_{k=-\infty}^{\infty} j^k \cdot I_k \cdot C_0(t - kT). \tag{2.1.9}$$

where j represents the imaginary unit, i.e. $j^2 = -1$. The pulse $C_0(t)$ is one T-period longer than the frequency pulse, $h_{GMSK}(t)$, which means that if the frequency pulse is truncated after $4T$, $C_0(t)$ is truncated limited to $5T$. As in the case of (2.1.8) the approximated GMSK signal in (2.1.9) depends on the previous transmitted symbols, which leads to ISI. This must be taken into account in the receiver. An illustration of the pulse shape of $C_0(t)$ is given in Figure 2.3.

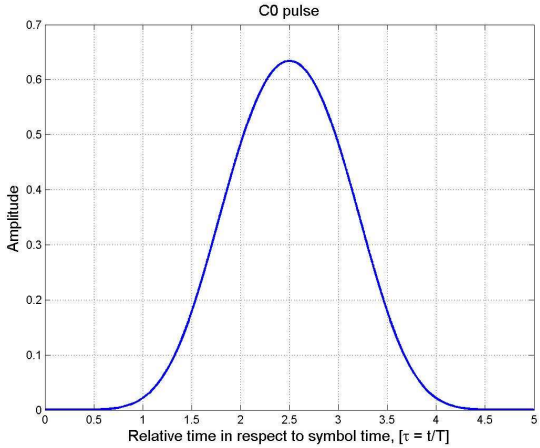

Figure 2.3. Pulse shape of C_0-pulse in respect to the symbol time.

2.1.2 Frame Format

In GSM it has been decided that one TDMA frame contains 8 times slots. Each user is then assigned a specific time slot in which the user is allowed to transmit /receive the signal (it should be noted that there are two frequencies allocated for each user, one for the uplink and one for downlink). A simple illustration of the GSM burst structure is given in Figure 2.4 [2, p.562].

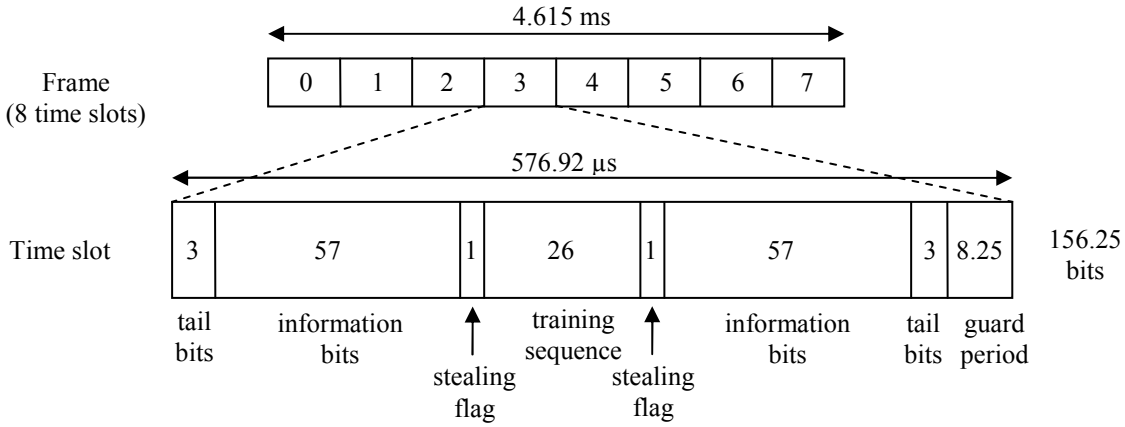

Figure 2.4. The frame structure in GSM.

In Figure 2.4 it is seen that a total of 156.25 bits are transmitted in one time slot. In each time slot there are transmitted 2·57 information bits, while 26 bits are training bits used for channel estimation. In GSM there are 8 different training sequences (TS), which are designed to have good autocorrelation properties [5]. The stealing bits are used to indicate if the TS contains voice or control data, (a thorough description of this is given in [2, pp. 561-563]).

2.2 EDGE

An important condition when designing the EDGE system was that it should have a minimum influence on the existing core network. Hence, it is possible for the EDGE system to operate in the existing GSM and GPRS network nodes [4, p. 52]. To increase the data rates and still use the same allocated frequencies as in GSM, another modulation technique is therefore used in EDGE.

2.2.1 Modulation Technique in EDGE

It is a modified version of the 8-PSK modulation which is used in EDGE. Compared to GMSK, which can be viewed as a binary signalling waveform, the 8-PSK modulation transfers 3 bits when each symbol is transmitted. First general M-PSK modulation is described and then the modulation technique in EDGE is explained.

M-PSK

As the name indicates it is the phase of the signal that is modulated. By mapping a symbol into one of the M phases corresponding to the given symbol, a M-ary Phase Shift Keying (M-PSK) modulation method is obtained. It is ensured that there are equal spaces between the neighbouring symbols in the space diagram by mapping the phases in the following way

$$\varphi_{M-PSK,m} = 2\pi(m-1)/M \qquad , m = 1,2,...,M\,. \tag{2.2.1}$$

The M-PSK symbol can then be expressed as

$$I_m = \exp\left(j\varphi_{M-PSK,m}\right), \tag{2.2.2}$$

The space diagram of a 8-PSK signal is shown in Figure 2.5.

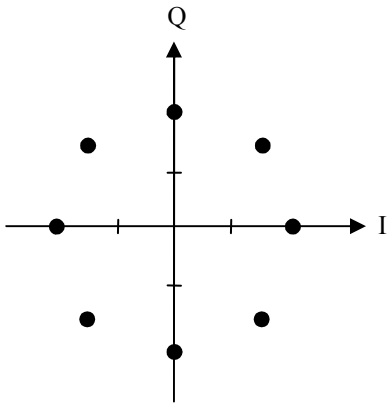

Figure 2.5. Space diagram of a 8PSK signal.

From Figure 2.5 it is seen that the distances from origo to the symbols are the same to all symbols. This is characteristic for the M-PSK signal, regardless of the value of M.

3π/8 rotated 8-PSK

As it was briefly mentioned in the subsection regarding GSM, it is not optimal to have a envelope that becomes close to zero at some time instants, since it prevents the design of power-efficient amplifiers. This will be the case for M-PSK when the phase changes with π, as indicated in Figure 2.6.

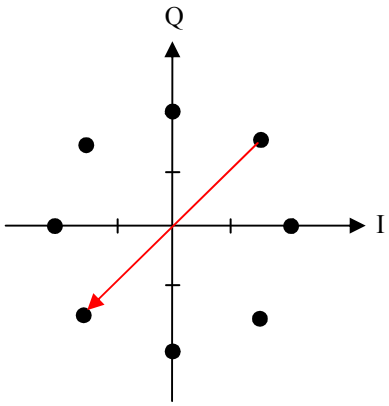

Figure 2.6. Phase shift by π for 8-PSK.

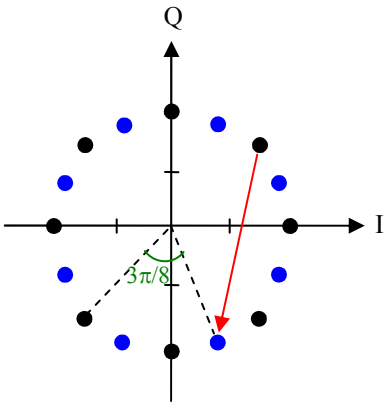

Figure 2.7. 3π/8 rotated 8-PSK.

If the signal is rotated 3π/8 (or any other odd multiple of π/8) between each transmitted symbol, the envelope will never become zero, which is illustrated in Figure 2.7. Therefore, this kind of phase modulation is used in EDGE. To minimize the side-lobes of the 3π/8 rotated 8-PSK signal, it is filtered by $C_0(t)$.

2.2.2 Frame Format

The frame format in EDGE is similar to the one used in GSM. Each time slot has the same duration and the same number of symbols is transmitted, which gives a three times higher bit rate than GSM. Hence, the structure in a time slot is the same as illustrated in Figure 2.4. The TS in EDGE (consisting of 26 symbols, i.e. 78 bits) can be found by replacing a 1 in the GSM TS with 001 and a 0 GSM with 111 in EDGE. Here the symbols 001 and 111 represent ±1 in the constellation diagram, when Gray encoding is used, see e.g. [4, p. 93].

2.3 Summary

The modulation technique in GSM is GMSK. This modulation technique has a constant envelope which leads to a spectrum with narrow main lope and low side lobes. GMSK is a non-linear modulation technique, but a good linear approximation can be obtained using one pulse.

To increase the data rates in the existing GSM system, EDGE has been introduced. It provides a data rate that are three times higher by changing the modulation technique to a modified version of 8-PSK modulation called 3π/8 rotated 8-PSK.

3 Multipath Radio Channel

The channel characteristic often changes considerably over time in a real radio communication channel. This section introduces models for channels with random time-variant impulse responses. The time-variant impulse response is due to the fact that the physical characteristic of the media, which the signal is transmitted over, changes over time. Since reflections of the transmitted signal will occur, it is assumed that the signal is transmitted over a multipath channel. Therefore, when one pulse is transmitted, the received signal will appear as a sequence of pulses. If the same pulse is transmitted again, another sequence of pulses will be received due to the random time-variation of the channel. (This is actually only true if the MS or the scatter objects move, but this will typically always be the case). It should be noted that most of the equations in this section is taken from [1].

3.1 Description of Fading Multipath Channels

In this subsection a model that describes the fading multipath channels in general terms will be presented. As show in [1, p. 151], the transmitted bandpass signal, $g(t)$, can be represented as

$$g(t) = \mathrm{Re}\left[s(t) \cdot \exp(j2\pi f_c t)\right]. \tag{3.1.1}$$

where $s(t)$ is the complex equivalent lowpass signal of the real signal, $g(t)$, called the complex envelope of $g(t)$.

Channel with discrete multipath components
It is assumed that every signal path has a propagation delay and an attenuation factor that changes with time. In [1, p. 802] it is shown that the received signal can be expressed as

$$v(t) = \sum_n \alpha_n(t) \cdot g\left[t - \tau_n(t)\right], \tag{3.1.2}$$

where α_n is the attenuation factor of the signal coming from the nth path and $\tau_n(t)$ is the propagation delay. By combining (3.1.1) and (3.1.2) a new expression for $v(t)$ is obtained

$$v(t) = \mathrm{Re}\left(\left\{\sum_n \alpha_n(t) \cdot \exp\left[-j2\pi f_c \tau_n(t)\right] \cdot s\left[t - \tau_n(t)\right]\right\} \exp(j2\pi f_c t)\right), \tag{3.1.3}$$

which leads to a expression of the equivalent lowpass received signal of the form

$$r(t) = \sum_n \alpha_n(t) \cdot \exp\left[-j2\pi f_c \tau_n(t)\right] \cdot s\left[t - \tau_n(t)\right]. \tag{3.1.4}$$

Therefore, the equivalent lowpass time-variant impulse response of the channel is

$$h(\tau; t) = \sum_n \alpha_n(t) \cdot \exp\left[-j2\pi f_c \tau_n(t)\right] \cdot \delta\left[t - \tau_n(t)\right]. \tag{3.1.5}$$

This model of the channel is appropriate for a channel which is assumed to have discrete multipath components.

To examine the received signal in closer details, it is assumed that $s(t) = 1$ for all t, which corresponds to a transmission of an unmodulated carrier with frequency f_c. In that way (3.1.4) is reduced to

$$r(t) = \sum_n \alpha_n(t) \cdot \exp[-j2\pi f_c \tau_n(t)] = \sum_n \alpha_n(t) \cdot \exp[-j\theta_n(t)], \tag{3.1.6}$$

where $\theta_n(t) = 2\pi f_c \tau_n(t)$. The received signal can be regarded as an addition of n time-variant vectors with amplitudes $\alpha_n(t)$ and phases $\theta_n(t)$. From (3.1.6) it is seen that $\theta_n(t)$ changes by 2π rad when $\tau_n(t)$ changes by $1/f_c$. Since $1/f_c$ is a little value, $\theta_n(t)$ can be changed with even small variations in the transmission medium. On the other hand large variations in the medium are required for $\alpha_n(t)$ to change. As noted earlier, the delays in each signal path are different and the delays will change in a random way. The number of signal paths are assumed to be very large and the received signal, $r(t)$, is therefore modeled as a complex-valued Gaussian random process. Hence, the impulse response of $h(\tau, t)$ is a complex-valued Gaussian random process (in the t variable). Due to the variations in the phases, fading of the signal occurs. This is because the phases $\{\theta_n\}$ associated with the vectors $\{\alpha_n \cdot \exp(-j\theta_n)\}$ sometimes results in the vectors adding destructively, and the received signal will be close to zero. At other times the vectors will add up constructively. Therefore significant amplitude variations will often appear, when a signal is transmitted in a multipath channel. When the impulse response of $h(\tau, t)$ is a zero mean complex-valued Gaussian random process, the envelope $|h(\tau,t)|$ is Rayleigh-distributed and the channel is called a Rayleigh fading channel.

3.1.1 Channel Properties

There are different types of multipath channels. To describe the characteristic of the channels in more details, properties such as multipath spread, coherence bandwidth, coherence time, and Doppler spread are among others used. These things are described shortly here.

Multipath spread

If it is assumed that the lowpass impulse response of the channel, $h(\tau, t)$, is wide sense stationary, it is in [1, p. 804] shown that the autocorrelation function can be defined as

$$\phi_h(\tau_1, \tau_2; \Delta t) = \tfrac{1}{2} E\left[h^*(\tau_1; t) \cdot h(\tau_2; t + \Delta t)\right]. \tag{3.1.7}$$

Normally the two path delays τ_1 and τ_2 are uncorrelated, which means that (3.1.7) can be expressed as

$$\tfrac{1}{2} E\left[h^*(\tau_1; t) \cdot h(\tau_2; t + \Delta t)\right] = \phi_h(\tau_1; \Delta t) \cdot \delta(\tau_1 - \tau_2) \tag{3.1.8}$$

By setting $\Delta t = 0$, the autocorrelation function, $\phi_h(\tau, 0) \equiv \phi_h(\tau)$, describes the average power as a function of the delay τ. By plotting ϕ_h as a function of the delay, the multipath intensity profile of the channel is obtained. The interval of τ in which the multipath intensity profile is approximately nonzero is called the multipath spread, T_m. An illustration of this is shown in Figure 3.1.

Coherence bandwidth

Another autocorrelation function, called the spaced-frequency spaced-time correlation function of the channel is given as

$$\phi_H(\Delta f; \Delta t) = \tfrac{1}{2} E\left[H^*(f_1; t) \cdot H(f_2; t + \Delta t)\right] \tag{3.1.9}$$

where $\Delta f = f_1 - f_2$. This function is used to describe the coherence bandwidth of the channel, $(\Delta f)_h$. This is shown in Figure 3.1.

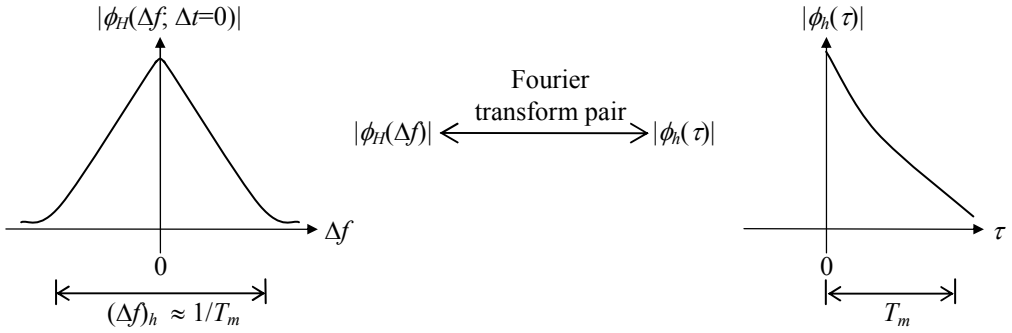

Figure 3.1. Spaced-frequency correlation function and multipath intensity profile.

Figure 3.1 indicates that the two autocorrelation functions are related by the Fourier transformation. This relation is in [1, p. 805] shown to be

$$\phi_H(\Delta f;0) = \int_{-\infty}^{\infty} \phi_h(\tau;0)\exp(-j2\pi f\tau)\,d\tau\,. \tag{3.1.10}$$

Doppler spread

Another effect in the channel, that must be considered, is the Doppler broadening and Doppler shift (which are caused by movement of the mobile station or in the surroundings). By Fourier transforming $\phi_H(\Delta f = 0; \Delta t)$ the Doppler power spectrum, $S_H(\lambda)$ of the channel can be obtained. It is expressed as

$$S_H(0;\lambda) = \int_{-\infty}^{\infty} \phi_H(0;\Delta t)\exp(-j2\pi\lambda\ \Delta t)\,d\Delta t\,. \tag{3.1.11}$$

The interval of λ in which $S_H(\lambda)$ is approximately nonzero is called the Doppler spread, B_d. This is illustrated in Figure 3.2.

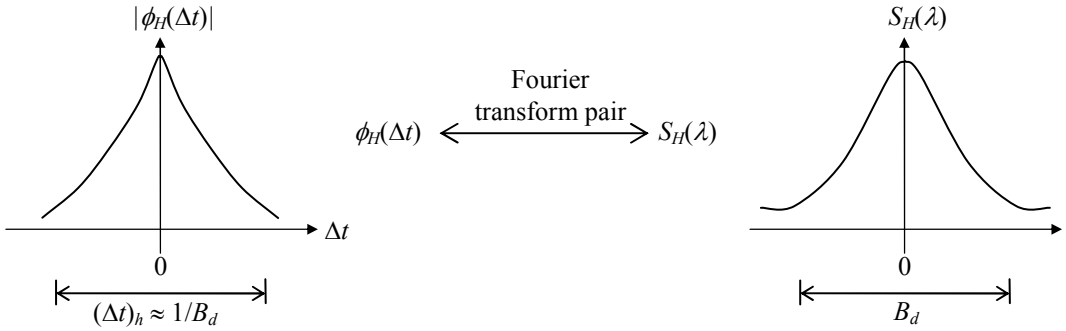

Figure 3.2. Spaced-time correlation function and Doppler power spectrum.

A model that are often used to describe the Doppler power spectrum is Jakes' model. The Doppler power spectrum is here expressed as

$$S_H(\lambda) = \int_{-\infty}^{\infty} J_0(2\pi f_m \Delta t)\exp(-j2\pi\lambda\,\Delta t)\,d\Delta t$$
$$= \begin{cases} \dfrac{1}{\pi f_m}\dfrac{1}{\sqrt{1-(f/f_m)^2}} & for \quad |f| \le f_m \\ 0 & otherwise \end{cases} \qquad (3.1.12)$$

where $f_m = vf_c/c$ and J_0 is the zero-order Bessel function of the first kind, see e.g. [1, p. 809]. A plot of the Doppler spectrum is given in Figure 3.3.

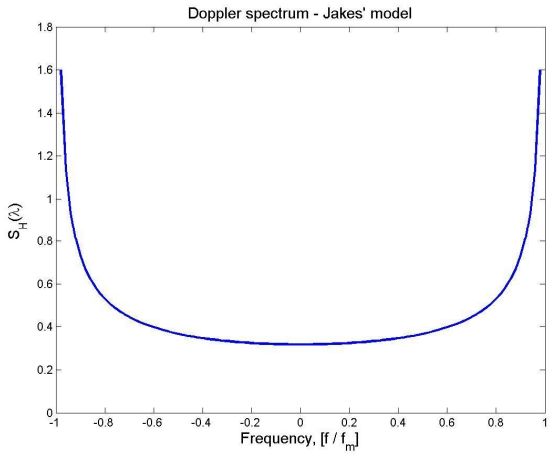

Figure 3.3. Doppler spectrum (Jakes' model).

Coherence time

From (3.1.11) and Figure 3.2 it is seen that there is a connection between $\phi_H(\Delta t)$ and $S_H(\lambda)$. Therefore, the reciprocal of the Doppler spread, B_d, can be used to give an estimate of the coherence time, $(\Delta t)_h$, of the channel, (i.e. $(\Delta t)_h \approx 1/B_d$). This implies that a channel that is changing slowly will have a large coherence time, which corresponds to a small Doppler spread.

The fading characteristic of the channel (i.e. how fast the fading occurs) is related to the coherence time, $(\Delta t)_h$, or the Doppler spread, B_d.

3.1.2 Frequency-Selective, Slowly Fading Channel

If the coherence bandwidth of the channel, $(\Delta f)_h$, is large in comparison to the bandwidth of the transmitted signal, the channel is called a frequency-nonselective channel. If a channel is frequency-selective, $(\Delta f)_h$ is small compared to the bandwidth of the transmitted signal. Thus, the signal will be corrupted considerably by the channel. This can be realized by examining a transmitted signal with the equivalent lowpass signal $s(t)$. The equivalent lowpass received signal, $r(t)$, can be expressed as

$$r(t) = \int_{-\infty}^{\infty} h(\tau;t)\cdot s(t-\tau)\,d\tau = \int_{-\infty}^{\infty} H(f;t)\cdot S(f)\cdot\exp(j2\pi ft)\,df\ . \qquad (3.1.13)$$

From this expression it is possible to see that when $S(f)$ has a bandwidth larger than the coherence bandwidth, $(\Delta f)_h$, $S(f)$ is influenced by different gains and phase shifts across the band.

If the real bandpass signal is assumed to have the bandwidth W, the equivalent lowpass signal $s(t)$ will have the bandwidth $|f| \leq W/2$. When $s(t)$ is band-limited to $|f| \leq W/2$, the sampling theorem says that the signal can be expressed as

$$s(t) = \sum_{n=-\infty}^{\infty} s\left(\frac{n}{W}\right) \frac{\sin\left[\pi W\left(t - n/W\right)\right]}{\pi W\left(t - n/W\right)}. \tag{3.1.14}$$

By Fourier transforming (3.1.14) the following result is obtained

$$S(f) = \begin{cases} \frac{1}{W} \sum_{n=-\infty}^{\infty} s\left(\frac{n}{W}\right) \exp\left(-j2\pi f\, n/W\right), & |f| \leq W/2 \\ 0, & |f| \geq W/2. \end{cases} \tag{3.1.15}$$

As shown in (3.1.13) the equivalent lowpass received signal without noise can be expressed as

$$r(t) = \int_{-\infty}^{\infty} H(f;t) \cdot S(f) \cdot \exp\left(j2\pi f t\right) df, \tag{3.1.16}$$

where $H(f;t)$ is the time-variant transfer function of the channel. By combining (3.1.15) and (3.1.16) the received signal can be represented as

$$\begin{aligned} r(t) &= \frac{1}{W} \sum_{n=-\infty}^{\infty} s\left(\frac{n}{W}\right) \int_{-\infty}^{\infty} H(f;t) \exp\left[j2\pi f\left(t - n/W\right)\right] df \\ &= \frac{1}{W} \sum_{n=-\infty}^{\infty} s\left(\frac{n}{W}\right) \cdot h\left(t - \frac{n}{W}; t\right) \end{aligned}, \tag{3.1.17}$$

where $h(\tau, t)$ is the lowpass time-variant impulse response. Since (3.1.17) represents a convolution sum it is possible to rewrite the equation to

$$r(t) = \frac{1}{W} \sum_{n=-\infty}^{\infty} s\left(t - \frac{n}{W}\right) \cdot h\left(\frac{n}{W}; t\right). \tag{3.1.18}$$

By defining the time-variable channel coefficients as

$$h_n(t) = \frac{1}{W} h\left(\frac{n}{W}; t\right), \tag{3.1.19}$$

a new expression of (3.1.18) is obtained

$$r(t) = \sum_{n=-\infty}^{\infty} h_n(t) \cdot s\left(t - \frac{n}{W}\right). \tag{3.1.20}$$

This implies that the impulse response of the channel is

$$h(\tau;t) = \sum_{n=-\infty}^{\infty} h_n(t) \cdot \delta\left(\tau - \frac{n}{W}\right) \tag{3.1.21}$$

Tapped delay line model

From (3.1.20) and (3.1.21) it is seen that a reasonable model for the frequency-selective channel is to represent it as a tapped delay line with weight coefficients, h_n. The space between the taps is $1/W$ (if the signal is oversampled, the space will be smaller). The transfer function of the channel becomes

$$H(f;t) = \sum_{n=-\infty}^{\infty} h_n(t) \cdot \exp(-j2\pi f\, n/W). \tag{3.1.22}$$

From the equations above, it is possible to see that when an equivalent lowpass signal with a bandwidth of $W/2$ (where W is larger than $(\Delta f)_h$), the resolution in the multipath delay profile is $1/W$.
Of course the tapped delay line model in practice cannot have an infinite number of taps. Since the total multipath spread of the channel is T_m, it is reasonable to reduce the model to

$$r(t) = \sum_{n=0}^{L} h_n(t) \cdot s(t - n/W) + z(t), \tag{3.1.23}$$

where $L = (T_m \cdot W)$ and $z(t)$ is the noise, see e.g. [1, p. 841]. In Figure 5.5 an illustration of the tapped delay line model for the channel is given.

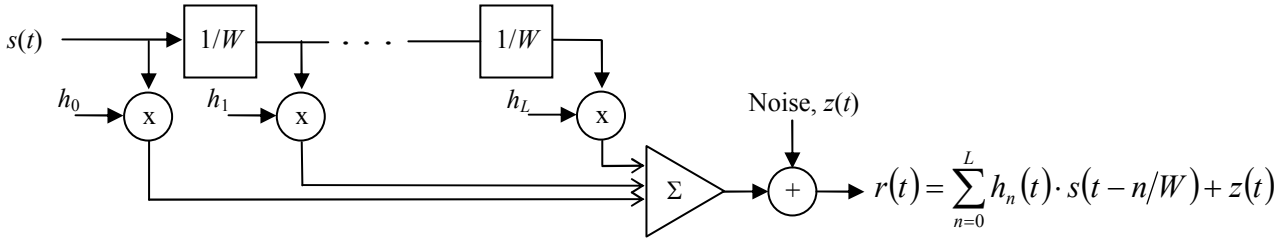

Figure 5.5. Model for frequency-selective channel (tapped delay line).

If it is assumed that the impulse response of $h(\tau, t)$ is a zero mean complex-valued Gaussian random process, the time-variant tap weights are zero mean complex-valued Gaussian random processes. The envelope $|h(t; \tau)| = \alpha_n(t)$ is then Rayleigh-distributed and the phases, $\phi_n(t)$, are uniformly distributed. If the impulse response, on the other hand, is a nonzero mean complex-valued Gaussian random process, the envelope will be Rice-distributed.

Equalizer required
When the bit interval, T_b, is comparable to or smaller than the multipath spread of the channel, T_m, or equivalently when the bandwidth of the transmitted signal exceeds the coherence bandwidth, considerable intersymbol interference occurs. In such a case, it is necessary to use equalizers to reduce the ISI.

3.2 Discrete Model for Multipath Channels

Based on the model for frequency-selective channels described in the previous section, it is possible to create a discrete model for the sampled received signal. In discrete form the received signal in one burst can be expressed as

$$\mathbf{r}_n = \sum_{i=0}^{L-1} \mathbf{h}_i \cdot s_{n-i} + \mathbf{z}_n, \quad \textit{where} \quad n = 1,2,\ldots,N_{BL} + L - 1 \tag{3.2.1}$$

where N_{BL} is the burst length. $\mathbf{h}_i$'s are the coefficients of the channel impulse response. s_n is the transmitted symbol and $\mathbf{z}_i$ is the noise term, $\mathbf{z}_i \in \mathbb{CN}\left(\mathbf{0},\sigma^2\mathbf{I}\right)$. Having the receive dimension N_R, the dimension of $\mathbf{r}_n$ is $N_R \times 1$ vector

$$\mathbf{r}_n = \begin{bmatrix} r_{(n-1)N_R+1} & r_{(n-1)N_R+2} & \cdots & r_{nN_R} \end{bmatrix}^T. \tag{3.2.2}$$

and similarly $\mathbf{h}_i$ with the same dimension is defined as

$$\mathbf{h}_i = \begin{bmatrix} h_{iN_R} & h_{iN_R+1} & \cdots & h_{iN_R+N_R-1} \end{bmatrix}^T. \tag{3.2.3}$$

In the case of multiple users, the signal model described in (3.2.1) is expanded to cover several users. More specifically the received signal will be a superposition of the contribution from all the users. Therefore, the received signal can be expressed as

$$\mathbf{r}_n = \sum_{i=0}^{L-1}\sum_{k=1}^{K} \mathbf{h}_i^{(k)} \cdot s_{n-i}^{(k)} + \mathbf{z}_n \tag{3.2.4}$$

where the superscript (k) indicates the *k'th* user.

Matrix Formulation of the Signal Model

The signal model defined in (3.2.4) consists of linear operations since it contains a summation and a convolution of the complete channel impulse response with the transmitted symbols. This implies that it is possible to express the signal models in matrix form. The received signal can then be expressed as

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \mathbf{z} \tag{3.2.5}$$

where $\mathbf{r} = \begin{bmatrix} \mathbf{r}_1^T & \mathbf{r}_2^T & \ldots & \mathbf{r}_{N_{BL}+L-1}^T \end{bmatrix}^T$ is a column vector of size $N_R(N_{BL}+L\text{-}1)$. The vector $\mathbf{s}$ has the length $K \cdot N_{BL}$ and is given as

$$\mathbf{s} = \begin{bmatrix} s_1^{(1)} & s_1^{(2)} & \ldots & s_1^{(K)} & s_2^{(1)} & s_2^{(2)} & \ldots & s_{N_{BL}}^{(K)} \end{bmatrix}^T. \tag{3.2.6}$$

$\mathbf{H}$ is the channel convolution matrix with the size $N_R(N_{BL}+L\text{-}1)\times(K \cdot N_{BL})$, defined as

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 & 0 & \cdots & \cdots & 0 \\ \mathbf{H}_1 & \mathbf{H}_0 & 0 & \cdots & \vdots \\ \vdots & \mathbf{H}_1 & \ddots & \ddots & \vdots \\ \mathbf{H}_{L-1} & \ddots & \ddots & \ddots & 0 \\ 0 & \mathbf{H}_{L-1} & \ddots & \ddots & \mathbf{H}_0 \\ \vdots & 0 & \ddots & \ddots & \mathbf{H}_1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \mathbf{H}_{L-1} \end{bmatrix}. \tag{3.2.7}$$

where each sub-matrix, $\mathbf{H}_i = \begin{bmatrix} \mathbf{h}_i^{(1)} & \mathbf{h}_i^{(2)} & \ldots & \mathbf{h}_i^{(K)} \end{bmatrix}$, will have the dimension $N_R \times K$. $\mathbf{z}$ is still the noise vector, $\mathbf{z} = \begin{bmatrix} \mathbf{z}_1^T & \mathbf{z}_2^T & \ldots & \mathbf{z}_{N_{BL}+L-1}^T \end{bmatrix}^T$.

GSM Channel Models

In the GSM standard some multipath channel models have been specified. Two of the channel models are the Typical Urban (TUx) and the Hilly Terrain (HTx), which are used for testing in this project. The channel profiles of the TU and the HT channels are shown in Figure 5.6 and Figure 5.7, respectively.

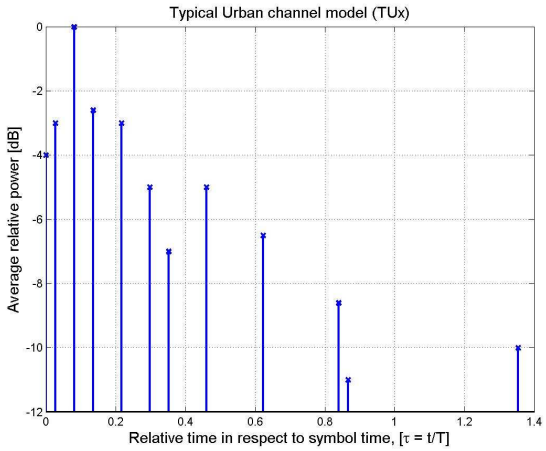

Figure 5.6. TU channel model.

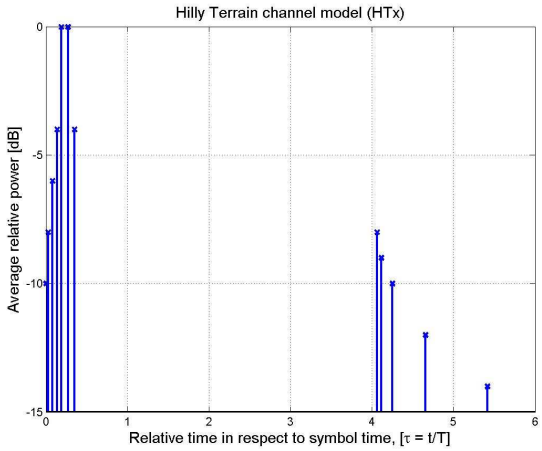

Figure 5.7. HT channel model.

When the C_0-pulse is transmitted in these channel models the pulse shape of the complete impulse response will be as shown in Figure 5.8 and Figure 5.9.

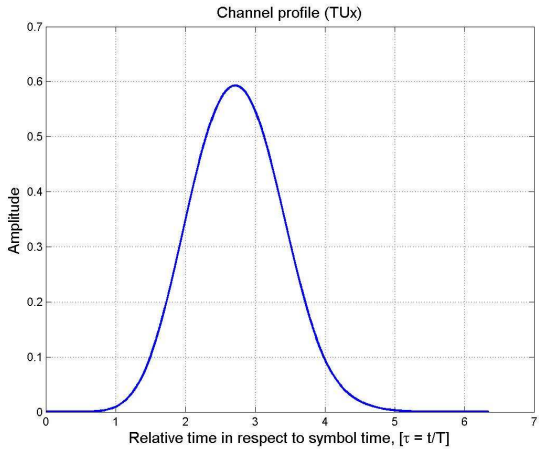

Figure 5.8. Channel profile of TU when the C_0-pulse is transmitted.

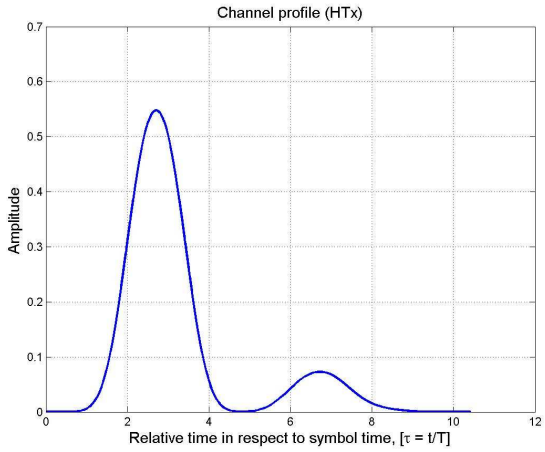

Figure 5.9. Channel profile of HT when the C_0-pulse is transmitted.

From Figure 5.9 it is seen that the channel profile of the HT channel will be modeled as having a longer impulse response than the TU channel.

3.3 Summary

When a signal is transmitted through a radio channel it will typically be corrupted considerably before it reaches the receiver. A channel can be frequency-selective, which will be the case if the transmitted signal has a bandwidth greater than the coherence bandwidth $(\Delta f)_h$ of the channel. If a multipath channel is frequency-selective and slowly fading, the channel is often modeled as a tapped delay line filter. The filter coefficients are typically assumed to be a zero mean complex-

valued Gaussian random process. The magnitudes of the coefficients are often Rayleigh-distributed and the phases are uniformly distributed.
The channel models used in this thesis are the TU and the HT channels, which are both specified in the GSM standard.

4 Traditional Receivers

This section deals with the more traditional receivers. First subsection describes the linear LMMSE receiver. In the same subsection the IQ-LMMSE receiver is shortly described, which is a linear receiver in the IQ-space. In Subsection 4.2 the MAP receiver is described. Finally, simulation results are found in Subsection 4.3 for all the receivers, which are described here.

4.1 The LMMSE Receiver

4.1.1 LMMSE

As the name implies the Linear Minimum Mean Square Error (LMMSE) receiver is a linear receiver that minimizes the Mean Square Error (MSE). More specific, it is the filter coefficients of the receiver that are chosen to minimize the MSE.

Single user case
The signal model in matrix form which is defined in (3.2.1) is used to derive the LMMSE Receiver. When the receiver is linear the estimated symbols can be expressed as

$$\hat{\mathbf{s}} = \mathbf{M}\mathbf{r}\,, \tag{4.1.1}$$

and the task is then to estimate the matrix $\mathbf{M}$. As just mentioned it is the MSE which is used as optimization criteria. This leads to the cost function

$$J = E\left[\left\|\mathbf{s}-\hat{\mathbf{s}}\right\|^2\right] = E\left[\left(\mathbf{s}-\mathbf{M}\mathbf{r}\right)^H\left(\mathbf{s}-\mathbf{M}\mathbf{r}\right)\right] \tag{4.1.2}$$

The optimal solution of $\mathbf{M}$ can be found by taking the derivative of the cost function and setting it equal to zero. Hence,

$$\frac{\partial J}{\partial \mathbf{M}} = 0 \tag{4.1.3}$$

Since J is a quadratic cost function, there will only be one solution to (4.1.3), which will be a minimum. In Appendix 4.1 it is shown that the optimal solution to (4.1.3) is given as

$$\mathbf{M} = \mathbf{H}^H\left(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}\right)^{-1}. \tag{4.1.4}$$

In an Additive White Gaussian Noise (AWGN) channel with no ISI the LMMSE receiver will yield the optimal solution. When there is no noise (i.e. $\sigma = 0$) the LMMSE solution becomes equivalent with the zero forcing solution, (which will be the pseudo inverse of $\mathbf{H}$, i.e. $\mathbf{H}^+$).

Multi user case
In scenarios where there are multiple users the signal model in (3.2.4) is used. By using the notation above this signal model can be expressed as

$$\mathbf{r} = \mathbf{H}\mathbf{s} + \tilde{\mathbf{H}}_I\mathbf{s}_I + \mathbf{z}\,, \tag{4.1.5}$$

where $\widetilde{\mathbf{H}}_I$ is the channel matrix of the interfering users and $\mathbf{s}_I$ are the symbols transmitted from those. When the LMMSE receiver is used to decode the signal of a given user it is done by assuming that there is a desired user and some interfering users. In that way it is possible to regard the signal coming from the interfering users as being noise. Hence, the noise is expressed as

$$\widetilde{\mathbf{z}} = \widetilde{\mathbf{H}}_I \mathbf{s}_I + \mathbf{z}, \tag{4.1.6}$$

Taking the expectation of this new noise term gives

$$E\left[\widetilde{\mathbf{z}}\widetilde{\mathbf{z}}^H\right] = E\left[\left(\widetilde{\mathbf{H}}_I \mathbf{s}_I + \mathbf{z}\right)\left(\widetilde{\mathbf{H}}_I \mathbf{s}_I + \mathbf{z}\right)^H\right] = \widetilde{\mathbf{H}}_I E\left[\mathbf{s}_I \mathbf{s}_I^H\right]\widetilde{\mathbf{H}}_I^H + E\left[\mathbf{z}\mathbf{z}^H\right]. \tag{4.1.7}$$

Like in the derivation of the expression for the SUD it is assumed (without loss of generality) that the transmitted symbols are uncorrelated and have unit power. Using these assumptions (4.1.7) becomes

$$E\left[\widetilde{\mathbf{z}}\widetilde{\mathbf{z}}^H\right] = \widetilde{\mathbf{H}}_I \mathbf{I} \widetilde{\mathbf{H}}_I^H + \sigma^2 \mathbf{I} = \widetilde{\mathbf{H}}_I \widetilde{\mathbf{H}}_I^H + \sigma^2 \mathbf{I} = \boldsymbol{\Sigma}, \tag{4.1.8}$$

where $\boldsymbol{\Sigma}$ is the covariance matrix of the interfering users plus the variance of the AWGN. This new covariance matrix is substituted into (4.1.4) instead of ($\sigma^2\mathbf{I}$) and the LMMSE solution, therefore, becomes

$$\mathbf{M} = \mathbf{H}^H \left(\mathbf{H}\mathbf{H}^H + \boldsymbol{\Sigma}\right)^{-1}. \tag{4.1.9}$$

4.1.2 IQ-Splitting

In [36] it is shown, that a gain in performance can be obtained when the received signal and its complex conjugate are filtered separately and then linearly combined. More specifically, it is shown that both $E\left[\mathbf{r}\mathbf{r}^H\right]$ and $E\left[\mathbf{r}\mathbf{r}^T\right]$ should be taken into account in the receiver to obtain a gain in performance. This will, however, only be true when the received signal can be assumed to a convolution of a real-valued data sequence with the complex channel impulse response. Since the transmitted symbols must be real valued, this technique can be applied for de-rotated GMSK but not for 8-PSK. The reason, why there is no gain in including $E\left[\mathbf{r}\mathbf{r}^T\right]$ for complex signals, is that $E\left[\mathbf{r}\mathbf{r}^T\right]$ becomes zero.

When the derivation of the LMMSE receiver shown in Appendix 4.1, is examined in more details, it is seen that the term, $E\left[\mathbf{r}\mathbf{r}^T\right]$, is not taken into account in the LMMSE (see Equation (A.4.1.7)).

It is possible to modify the LMMSE receiver in such a way that the statistics of $E\left[\mathbf{r}\mathbf{r}^T\right]$ is taken into account. This can be done by splitting the received signal into a real (in-phase) and an imaginary (quadrature) part as

$$\mathbf{r}_{IQ} = \begin{bmatrix} \Re\{\mathbf{r}\} \\ \Im\{\mathbf{r}\} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_I \\ \mathbf{r}_Q \end{bmatrix}. \tag{4.1.10}$$

By defining $\mathbf{H}_I = \Re\{\mathbf{H}\}$ and $\mathbf{H}_Q = \Im\{\mathbf{H}\}$ the received IQ-splitted signal is given as

$$\mathbf{r}_{IQ} = \mathbf{H}_{IQ}\mathbf{s}_{IQ} = \begin{bmatrix} \mathbf{H}_I & -\mathbf{H}_Q \\ \mathbf{H}_Q & \mathbf{H}_I \end{bmatrix}\begin{bmatrix} \mathbf{s}_I \\ \mathbf{s}_Q \end{bmatrix} = \begin{bmatrix} \mathbf{H}_I\mathbf{s}_I - \mathbf{H}_Q\mathbf{s}_Q \\ \mathbf{H}_Q\mathbf{s}_I + \mathbf{H}_I\mathbf{s}_Q \end{bmatrix}. \tag{4.1.11}$$

When the transmitted signal is real valued as it was required, (4.1.11) can be simplified to

$$\mathbf{r}_{IQ} = \begin{bmatrix} \mathbf{H}_I\mathbf{s}_I \\ \mathbf{H}_Q\mathbf{s}_I \end{bmatrix} = \begin{bmatrix} \mathbf{H}_I \\ \mathbf{H}_Q \end{bmatrix}\mathbf{s}_I = \widetilde{\mathbf{H}}_{IQ}\mathbf{s}_I \quad . \tag{4.1.12}$$

The pseudo covariance matrix of $\mathbf{r}_{IQ}$ becomes

$$\begin{aligned} E\left[\mathbf{r}_{IQ}\mathbf{r}_{IQ}^T\right] &= E\left[\begin{bmatrix} \mathbf{r}_I \\ \mathbf{r}_Q \end{bmatrix}\begin{bmatrix} \mathbf{r}_I^T & \mathbf{r}_Q^T \end{bmatrix}\right] \\ &= E\left[\begin{bmatrix} \mathbf{r}_I\mathbf{r}_I^T & \mathbf{r}_I\mathbf{r}_Q^T \\ \mathbf{r}_Q\mathbf{r}_I^T & \mathbf{r}_Q\mathbf{r}_Q^T \end{bmatrix}\right], \end{aligned} \tag{4.1.13}$$

which is the IQ-splitted equivalent to Equation (A.4.1.7) in Appendix 4.1. In the traditional complex-valued LMMSE receiver it is assumed that

$$E\left[\ \mathbf{r}\ \mathbf{r}^T\right] = 0\,. \tag{4.1.14}$$

If (4.1.14) has to be true, it implies that

$$E\left[\ \mathbf{r}\ \mathbf{r}^T\ \right] = 0 = E\left[\ \left(\mathbf{r}_I + i\mathbf{r}_Q\right)\left(\mathbf{r}_I + i\mathbf{r}_Q\right)^T\ \right] = E\left[\ \mathbf{r}_I\mathbf{r}_I^T - \mathbf{r}_Q\mathbf{r}_Q^T\ \right] + iE\left[\ \mathbf{r}_I\mathbf{r}_Q^T - \mathbf{r}_Q\mathbf{r}_I^T\ \right] \tag{4.1.15}$$

and therefore $E\left[\mathbf{r}_I\mathbf{r}_I^T\right] = E\left[\mathbf{r}_Q\mathbf{r}_Q^T\right]$ and $E\left[\mathbf{r}_I\mathbf{r}_Q^T\right] = -E\left[\mathbf{r}_Q\mathbf{r}_I^T\right]$. It can be proved that it is in general not true, when real valued signals are transmitted.
By substitution of $\mathbf{r}_{IQ} = \mathbf{H}_{IQ}\,\mathbf{s}_{IQ}$ into (4.1.13) it is realized that

$$E\left[\ \mathbf{r}_{IQ}\mathbf{r}_{IQ}^T\ \right] = E\left[\ \left(\mathbf{H}_{IQ}\mathbf{s}_{IQ} + \mathbf{z}\right)\left(\mathbf{s}_{IQ}^T\mathbf{H}_{IQ}^T + \mathbf{z}^T\right)\ \right] = \mathbf{H}_{IQ}E\left[\ \mathbf{s}_{IQ}\mathbf{s}_{IQ}^T\ \right]\mathbf{H}_{IQ}^T + \sigma^2\mathbf{I}\,, \tag{4.1.16}$$

where it is assumed that the noise and the symbols are uncorrelated. By assuming that the transmitted signal is real-valued it is seen that

$$E\left[\ \mathbf{s}_{IQ}\mathbf{s}_{IQ}^T\ \right] = E\begin{bmatrix} \mathbf{s}_I\mathbf{s}_I^T & \mathbf{s}_I\mathbf{s}_Q^T \\ \mathbf{s}_Q\mathbf{s}_I^T & \mathbf{s}_Q\mathbf{s}_Q^T \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}. \tag{4.1.17}$$

By substituting (4.1.17) into (4.1.16) the following result is obtained

$$\begin{aligned} E\left[\ \mathbf{r}_{IQ}\mathbf{r}_{IQ}^T\ \right] &= \begin{bmatrix} \mathbf{H}_I & -\mathbf{H}_Q \\ \mathbf{H}_Q & \mathbf{H}_I \end{bmatrix}\begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}\begin{bmatrix} \mathbf{H}_I & -\mathbf{H}_Q \\ \mathbf{H}_Q & \mathbf{H}_I \end{bmatrix}^T \\ &= \begin{bmatrix} \mathbf{H}_I \\ \mathbf{H}_Q \end{bmatrix}\begin{bmatrix} \mathbf{H}_I \\ \mathbf{H}_Q \end{bmatrix}^T = \widetilde{\mathbf{H}}_{IQ}\widetilde{\mathbf{H}}_{IQ}^T + \sigma^2\mathbf{I}. \end{aligned} \tag{4.1.18}$$

It is now seen that the optimization can be carried out in the same way as in Appendix 4.1 The solution of the IQ-LMMSE receiver then becomes

$$\mathbf{M}_{IQ} = \left(\widetilde{\mathbf{H}}_{IQ}\widetilde{\mathbf{H}}_{IQ}^{T} + \sigma^2\mathbf{I}\right)^{-1}\widetilde{\mathbf{H}}_{IQ}^{T}. \tag{4.1.18}$$

From the description above it is seen that the IQ-LMMSE receiver requires an inversion of a matrix, which will have the double size compared to the matrix inversion in the LMMSE. Therefore, the complexity of the IQ-LMMSE receiver is at least doubled.

4.2 The MAP Receiver

In this section a description of the optimum non-linear MAP receiver is given. The receiver is optimum in the sense that it minimizes the probability of a symbol error.

4.2.1 Maximization of the A Posterior Probability

The goal is to obtain a detector that makes a decision on the transmitted signal, **s** based on the received signal, **r**, in such a way, that the posterior probability of a correct decision is maximized. When the system model introduced in (3.2.5) is used, the posterior probability can be expressed as

$$P(\mathbf{s}|\,\mathbf{r}) = \frac{p(\mathbf{r}|\,\mathbf{s})P(\mathbf{s})}{p(\mathbf{r})}. \tag{4.2.1}$$

Here Bayes' rule has been used. $p(\mathbf{r}|\mathbf{s})$ is the conditional PDF of the observed vector given **s** and is called the likelihood for **s**. $P(\mathbf{s})$ is the a priori probability of the transmitted signal. The denominator $p(\mathbf{r})$ is called the evidence and is used to normalize the probability.
From (4.2.1) it is seen that the evidence is independent of the transmitted signal, and therefore, (4.2.1) can be simplified to

$$P(\mathbf{s}|\,\mathbf{r}) \propto p(\mathbf{r}|\,\mathbf{s})P(\mathbf{s}). \tag{4.2.2}$$

When all symbol vectors, **s**, are equally probable (iid) the a priori probability becomes identical for all **s**, and the maximization of $P(\mathbf{s}|\mathbf{r})$ is equivalent with the maximizing of $p(\mathbf{r}|\mathbf{s})$. Hence, the MAP criterion will under these conditions become equivalent to the Maximum Likelihood (ML) criterion. In the case of a Gaussian likelihood the ML optimization criterion minimizes the Euclidean distance (see e.g. [1]), which can be expressed as

$$\hat{\mathbf{s}}_{ML} = \arg\min_{\mathbf{s}} \left\|\mathbf{r} - \mathbf{H}\mathbf{s}\right\|^2. \tag{4.2.3}$$

The optimal Symbol-by-Symbol (SbS) detector selects the symbol at time n, $\hat{s}_n$, which can be expressed as

$$\hat{s}_n = \arg\max_{s_n} P(s_n|\,\mathbf{r}). \tag{4.2.4}$$

The receiver that minimizes the probability for symbol (and bit) error is called the Symbol-by-Symbol Maximum a Posteriori (MAP) receiver and, as it is described in the next subsection, it relies on the forward backward algorithm (also known as the BCJR algorithm [13]). However, it should be noted that other optimality criteria exist. For example Maximum Likelihood Sequence Estimation (MLSE) tries to find the most likely sequence, which is normally done by using the Viterbi algorithm [34]. An alternative way of doing MLSE, which gives the same result, is by using the forward recursion only in the Max-log MAP receiver.

4.2.2 SUD using the MAP Receiver

To keep the derivation of the Symbol-by-Symbol MAP receiver as simple as possible, it is first carried out in the case of Single User Detection (SUD). For Multi User Detection (MUD) the derivation can be carried out in an equivalent way and is described in Subsection 4.2.3 "Optimal MUD – Joint MAP Receiver".

Since the received signal $\mathbf{r}_n$, at time index n, depends on s_n and on the previous L-1 symbols, (see e.g. the signal models in (3.2.1)), it is reasonable to use a Markov model to describe the signal. More specific, a hidden first-order Markov model is used to model the received signal in the MAP receiver, see e.g. [12]. Therefore, the L-1 previous symbols transmitted by the users are collected in a state vector $\mathbf{a}_n$. In the single user case the state vector becomes

$$\mathbf{a}_n = \begin{bmatrix} \hat{s}_{n-1} & \hat{s}_{n-2} & \cdots & \hat{s}_{n-L+1} \end{bmatrix}^T \tag{4.2.5}$$

where $\hat{s}_n$ is the hypothesized value of s_n from the constellation set Ω. For BPSK $|\Omega| = 2$ and for 8-PSK $|\Omega| = 8$. The number of different combinations for the hypothesized values are $Q = |\Omega|^{L-1}$. Therefore, (4.2.5) can be expressed as

$$\mathbf{a}_n(p) = \begin{bmatrix} \hat{s}_{n-1}(p) & \hat{s}_{n-2}(p) & \cdots & \hat{s}_{n-L+1}(p) \end{bmatrix}^T \tag{4.2.6}$$

where $p \in [0, 1, \ldots, Q\text{-}1]$ and $\left[\hat{s}_{n-1}(p),\ldots,\hat{s}_{n-L+1}(p)\right]$ are the hypothesized symbol values for state $\mathbf{a}_n(p)$. When a BPSK signal is transmitted and $L = 3$, the number of different combinations for $\mathbf{a}_n$ are $Q = |\Omega|^2 = 4$, as illustrated in Table 4.1.

$\mathbf{a}_n(p)$	$\left[\hat{s}_{n-1}(p), \hat{s}_{n-2}(p)\right]$
$\mathbf{a}_n(0)$	[-1, -1]
$\mathbf{a}_n(1)$	[-1, +1]
$\mathbf{a}_n(2)$	[+1, -1]
$\mathbf{a}_n(3)$	[+1, +1]

Table 4.1. State vector for $L = 3$.

The next state $\mathbf{a}_{n+1}(q)$ can be expressed as

$$\mathbf{a}_{n+1}(q) = \begin{bmatrix} \hat{s}_n(q) & \hat{s}_{n-1}(q) & \cdots & \hat{s}_{n-L+2}(q) \end{bmatrix}^T . \tag{4.2.7}$$

A trellis diagram indicating the transitions from different states as a function of time can be made. If it is assumed that the initial state is [-1, -1] and the input sequence is [+1 -1 +1 +1] the trellis diagram shown in Figure 4.1 can be obtained.

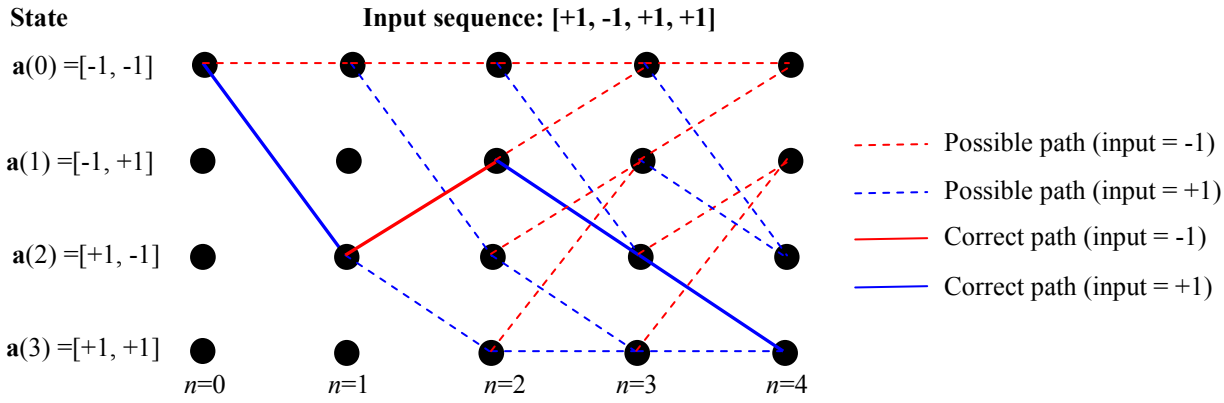

Figure 4.1. Trellis diagram for $L = 3$ when the initial state is [-1, -1].

From Figure 4.1 it is seen that at time instance n=3 (i.e. the channel length, L) there are $|\Omega|$ possible transitions to each state. The task for the MAP receiver is to determine the transitions probabilities from state $\mathbf{a}(p)$ at time index n to state $\mathbf{a}(q)$ at time n+1. When all the transition probabilities have been determined throughout the whole trellis diagram, the probability of a given symbol can be determined by marginalization. This is realized by defining Ω_0 as the sets of all states. Furthermore, define $\Omega(s_{n-1})$ as the set of states where $\hat{s}_{n-1}$ will have a given hypothesized value $\hat{s}_{n-1}$. In [12] it is showed that since the posterior probability can be expressed as

$$P(s_{n-1}|\mathbf{r}) = \frac{P(s_{n-1},\mathbf{r})}{P(\mathbf{r})}, \tag{4.2.8}$$

the probability of a given symbol can be found as

$$P(s_{n-1}|\mathbf{r}) = \frac{\sum_{\mathbf{a}_n(p)\in\Omega(s_{n-1})} P(\mathbf{r},\mathbf{a}_n(p))}{\sum_{\mathbf{a}_n(p)\in\Omega_0} P(\mathbf{r},\mathbf{a}_n(p))}. \tag{4.2.9}$$

To come up with a method for finding this posterior probability another vector, $\xi_n^{(p,q)}$, is introduced. $\xi_n^{(p,q)}$ is defined as the vector that contains all the hypothesized symbols related to the transition from $\mathbf{a}_n(p)$ to $\mathbf{a}_{n+1}(q)$. Hence,

$$\xi_n^{(p,q)} = \begin{bmatrix} \hat{s}_n(q) & \hat{s}_{n-1}(p) & \hat{s}_{n-2}(p) & \cdots & \hat{s}_{n-L+1}(p) \end{bmatrix}^T \tag{4.2.10}$$

It is possible to describe the posterior probability based on the state vectors defined above. Let S_a contain the set of pairs (p, q), where each pair represents a transition from state p to q. Given the observation $\mathbf{r}$, the a posterior probability of symbol $\hat{s}_n(q)$ takes the hypothesized value, $\tilde{s}_n$, and can be expressed as

$$P(\tilde{s}_n|\mathbf{r}) = \sum_{(p,q)\in S_a} P(\mathbf{a}_n(p),\mathbf{a}_{n+1}(q)|\mathbf{r}) = \sum_{(p,q)\in S_a} P(\mathbf{a}_n(p),\mathbf{a}_{n+1}(q),\mathbf{r})\frac{1}{P(\mathbf{r})}. \tag{4.2.11}$$

where $\mathbf{r} = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \cdots & \mathbf{r}_{N_{BL}+L-1} \end{bmatrix}^T$.

Let $\mathbf{r}_{n_1}^{n_2}$ be defined as $\mathbf{r}_{n_1}^{n_2} = \begin{bmatrix}\mathbf{r}_{n_1} & \cdots & \mathbf{r}_{n_2}\end{bmatrix}^T$, $n_1, n_2 \in [1, 2,\ldots,N_{BL}+L-1]$, and $n_1 < n_2$. Based on this definition it is possible to rewrite (4.2.11) to

$$\begin{aligned} P(\tilde{s}_n|\mathbf{r}) &= \frac{1}{P(\mathbf{r})}\sum_{(p,q)\in Sa} P\left(\mathbf{a}_n(p),\mathbf{a}_{n+1}(q),\mathbf{r}_1^{n-1},\mathbf{r}_n,\mathbf{r}_{n+1}^{N_{BL}+L-1}\right) \\ &= \frac{1}{P(\mathbf{r})}\sum_{(p,q)\in Sa} P\left(\mathbf{a}_n(p),\mathbf{r}_1^{n-1}\right)P\left(\mathbf{a}_{n+1}(q),\mathbf{r}_n\middle|\mathbf{a}_n(p),\mathbf{r}_1^{n-1}\right)P\left(\mathbf{r}_{n+1}^{N_{BL}+L-1}\middle|\mathbf{a}_n(p),\mathbf{a}_{n+1}(q),\mathbf{r}_1^{n-1},\mathbf{r}_n\right) \end{aligned}, \quad (4.2.12)$$

as described in [8]. Using the first order Markov property (4.2.12) can be simplified to

$$P(\tilde{s}_n|\mathbf{r}) = \frac{1}{P(\mathbf{r})}\sum_{(p,q)\in Sa} P\left(\mathbf{a}_n(p),\mathbf{r}_1^{n-1}\right)P\left(\mathbf{a}_{n+1}(q),\mathbf{r}_n\middle|\mathbf{a}_n(p)\right)P\left(\mathbf{r}_{n+1}^{N_{BL}+L-1}\middle|\mathbf{a}_{n+1}(q)\right). \quad (4.2.13)$$

By introducing three new variables the expression becomes

$$P(\tilde{s}_n|\mathbf{r}) = \frac{1}{P(\mathbf{r})}\sum_{(p,q)\in Sa} \alpha_n(p)\cdot\gamma_n(p,q)\cdot\beta_{n+1}(q), \quad (4.2.14)$$

where

$$\alpha_n(p) = P\left(\mathbf{a}_n(p),\mathbf{r}_1^{n-1}\right), \quad (4.2.15)$$

is the forward variable, which takes information in the past to time n into account. The variable

$$\gamma_n(p,q) = P\left(\mathbf{a}_{n+1}(q),\mathbf{r}_n\middle|\mathbf{a}_n(p)\right) = P\left(\mathbf{r}_n\middle|\mathbf{a}_{n+1}(q),\mathbf{a}_n(p)\right)P\left(\mathbf{a}_{n+1}(q)\middle|\mathbf{a}_n(p)\right), \quad (4.2.16)$$

is the transition variable which describes the probability of (p, q). Finally the variable

$$\beta_{n+1}(q) = P\left(\mathbf{r}_{n+1}^{N_{BL}+L-1}\middle|\mathbf{a}_{n+1}(q)\right), \quad (4.2.17)$$

is the backward variable, which takes the information from the future (time $N_{BL} + L$ -1) back to time n+1 into account. In e.g. [15] it is shown that the forward and backward variables can be recursively determined as

$$\alpha_{n+1}(q) = \sum_{p\in Q}\alpha_n(p)\cdot\gamma_n(p,q) \quad \textit{and} \quad \beta_n(p) = \sum_{q\in Q}\beta_{n+1}(q)\cdot\gamma_n(p,q) \quad (4.2.18)$$

According to [8] the transition variable in (4.2.16) can be rewritten to

$$\gamma_n(p,q) = P\left(\mathbf{r}_n\middle|\mathbf{a}_{n+1}(q),\mathbf{a}_n(p)\right)P\left(\hat{s}_n(q)\right) \quad (4.2.19)$$

In (4.2.19) $P\left(\mathbf{r}_n\middle|\mathbf{a}_{n+1}(q),\mathbf{a}_n(p)\right)$ represents the likelihood of a transition from $\mathbf{a}_n(p)$ to $\mathbf{a}_{n+1}$(q) given $\mathbf{r}_n$. If the noise is AWGN with zero mean and variance σ^2, the likelihood becomes

$$P\left(\mathbf{r}_n\middle|\mathbf{a}_{n+1}(q),\mathbf{a}_n(p)\right) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{N_R}\cdot\exp\left[\frac{-1}{2\sigma^2}\left(\mathbf{r}_n-\tilde{\mathbf{r}}_n^{(p,q)}\right)^H\left(\mathbf{r}_n-\tilde{\mathbf{r}}_n^{(p,q)}\right)\right]. \quad (4.2.20)$$

where $\tilde{\mathbf{r}}_n^{(p,q)}$ represents the expected output for a given $\boldsymbol{\xi}_n^{(p,q)}$. The computation of $\tilde{\mathbf{r}}_n^{(p,q)}$ is in a while explained in more details, but first a final expression for the MAP receiver is given. By combining (4.2.13), (4.2.18) and (4.2.19) a new expression for the posterior probability becomes

$$\begin{aligned} P(\tilde{s}_n|\mathbf{r}) &= \frac{1}{P(\mathbf{r})} \sum_{(p,q)\in Sa} \alpha_n(p)\beta_{n+1}(q)P(\mathbf{r}_n|\mathbf{a}_{n+1}(q),\mathbf{a}_n(p))P(\hat{s}_n(q)) \\ &= \frac{P(\tilde{s}_n)}{P(\mathbf{r})} \sum_{(p,q)\in Sa} \alpha_n(p)\beta_{n+1}(q)P(\mathbf{r}_n|\mathbf{a}_{n+1}(q),\mathbf{a}_n(p)) \end{aligned} \tag{4.2.21}$$

where $P(\tilde{s}_n)$ is the prior of the symbol. According to [8] the equality in (4.2.21) is true since there is a summation over all $(p, q) \in S_a$, where S_a represents all the transitions that correspond to $\hat{s}_n(q) = \tilde{s}_n$.

Implementation issues
When the MAP receiver is implemented it is recommended to carry out the calculations in the logarithmic domain to avoid underflow [14]. The underflow will occur even for a limited number of states in the computation of forward and backward variables, since the recursions are carried out for time index $n = [0,\ldots, L + N_{BL}]$. The implementation in the logarithmic domain also has the advantage that no exponential evaluation has to be carried out when the transition probability $\gamma_n(p,q)$ is computed. In the recursions given in (4.2.18) there is an addition of two variables, which should be handled in the logarithmic domain. If $C = A + B$ the calculation in the logarithmic domain becomes

$$\ln(C) = \max\{\ln(A), \ln(B)\} + \ln(1 + \exp(-|\ln(A) - \ln(B)|)).$$

The last term is called the correction term. This term is often neglected, and the receiver is then referred to as the max-log MAP receiver. It is described in [14] that the error is insignificant for $A<<B$ and $A>>B$, and it can not be larger than $\ln 2$, which is the case when $A = B$.

Complexity
The complexity of the MAP receiver depends on the number of states. As shown in (4.2.5), the state vector increases with channel memory, L-1, i.e. the number of states becomes $|\Omega|^{L-1}$. In each state the number of operations is $O(N_R|\Omega|)$ and, hence, the complexity of the MAP receiver for SUD becomes $O(N_R|\Omega|^L)$ per symbol.

4.2.3 Optimal MUD – Joint MAP Receiver

As it will be shown in Subsection 4.3 "Simulation Results", the Bit Error Rate (BER) performance of the MAP receiver relying on SUD is significantly reduced when interference from other users is present. Likewise, it will be shown that whitening of the signal under such conditions will improve the performance of the MAP receiver. However, whitening is in general terms not the optimal way of doing the detection of multiple users. The optimal way of doing the MUD is by jointly detecting symbols of all the users and then marginalize the interfering users from the received signal. Given the observation, **r** the Joint MAP receiver maximizes the joint posterior probability of the symbols by taking all the users into account.
Normally, it is only interesting to detect one user, who is denoted the desired user, and hence, the posterior probability of the desired user should be calculated. By performing a marginalization over all the other users, the desired posterior probability can be found from the joint posterior probability.

The way the Markov model handles multiple users is by extending the state space. Without loss of generality it is assumed that the channel impulse response length is the same for all the users.

Hence, the Markov model has to take the L-1 previous symbols from the K users into account. The state vector then becomes

$$\mathbf{a}_n = \left[\hat{s}_{n-1}^{(1)} \quad \hat{s}_{n-2}^{(1)} \quad \cdots \quad \hat{s}_{n-L+1}^{(1)} \quad \hat{s}_{n-1}^{(2)} \quad \hat{s}_{n-2}^{(2)} \quad \cdots \quad \hat{s}_{n-L+1}^{(K)}\right]^T. \tag{4.2.22}$$

The number of different combinations for the hypothesized values now become $Q = |\Omega|^{(L\text{-}1)K}$ and (4.2.22) can be expressed as

$$\mathbf{a}_n(p) = \left[\hat{s}_{n-1}^{(1)}(p) \quad \hat{s}_{n-2}^{(1)}(p) \quad \cdots \quad \hat{s}_{n-L+1}^{(1)}(p) \quad \hat{s}_{n-1}^{(2)}(p) \quad \hat{s}_{n-2}^{(2)}(p) \quad \cdots \quad \hat{s}_{n-L+1}^{(K)}(p)\right]^T, \tag{4.2.23}$$

where $p \in [0, 1, \ldots, Q\text{-}1]$. The next state $\mathbf{a}_{n+1}(q)$ can be expressed in the same way as

$$\mathbf{a}_{n+1}(q) = \left[\hat{s}_{n}^{(1)}(q) \quad \hat{s}_{n-1}^{(1)}(q) \quad \cdots \quad \hat{s}_{n-L+2}^{(1)}(q) \quad \hat{s}_{n}^{(2)}(q) \quad \hat{s}_{n-1}^{(2)}(q) \quad \cdots \quad \hat{s}_{n-L+2}^{(K)}(q)\right]^T, \tag{4.2.24}$$

where $q \in [0, 1, \ldots, Q\text{-}1]$. The vector, $\boldsymbol{\xi}_n^{(p,q)}$, that contains all the hypothesized symbols related to the transition (p, q), will have a length of $(L{\cdot}K)$ and is given as

$$\boldsymbol{\xi}_n^{(p,q)} = \left[\hat{s}_{n}^{(1)}(q) \quad \hat{s}_{n-1}^{(1)}(p) \quad \cdots \quad \hat{s}_{n-L+1}^{(1)}(p) \quad \hat{s}_{n}^{(2)}(q) \quad \hat{s}_{n-1}^{(2)}(p) \quad \cdots \quad \hat{s}_{n-L+1}^{(K)}(p)\right]^T. \tag{4.2.25}$$

Based on the new definitions of the state vectors the derivation of the Joint MAP receiver can be carried out in the same way as for the single user case given in Subsection 4.2.2 "SUD using the MAP Receiver". Therefore, the expressions (4.2.15), (4.2.16), and (4.2.17) for the variables $\alpha_n(p)$, $\gamma_n(p,q)$, and $\beta_{n+1}(q)$ will still be valid. Likewise, expression for the joint posterior probability will still be

$$P(\tilde{s}_n|\mathbf{r}) = \frac{P(\tilde{s}_n)}{P(\mathbf{r})} \sum_{(p,q)\in Sa} \alpha_n(p)\beta_{n+1}(q)P(\mathbf{r}_n|\mathbf{a}_{n+1}(q),\mathbf{a}_n(p)). \tag{4.2.26}$$

Once again it is assumed that the noise is white Gaussian with zero mean and variance σ^2. Hence, the likelihood of $\mathbf{r}_n$ given a transition from $\mathbf{a}_n(p)$ to $\mathbf{a}_{n+1}$(q), $P(\mathbf{r}_n|\mathbf{a}_{n+1}(q),\mathbf{a}_n(p))$, can be calculated as

$$P(\mathbf{r}_n|\mathbf{a}_{n+1}(q),\mathbf{a}_n(p)) = \left(\frac{1}{\sqrt{2\pi\sigma^2}}\right)^{N_R} \cdot \exp\left[\frac{-1}{2\sigma^2}\left(\mathbf{r}_n - \tilde{\mathbf{r}}_n^{(p,q)}\right)^H\left(\mathbf{r}_n - \tilde{\mathbf{r}}_n^{(p,q)}\right)\right]. \tag{4.2.27}$$

The expected output, $\tilde{\mathbf{r}}_n^{(p,q)}$, for a given state transition, $\boldsymbol{\xi}_n^{(p,q)}$ is computed as

$$\tilde{\mathbf{r}}_n^{(p,q)} = \mathbf{H}'\boldsymbol{\xi}_n^{(p,q)}, \tag{4.2.28}$$

and $\mathbf{H}'$ is a $N_R{\times}(L{\cdot}K)$ matrix containing the filter coefficients given as

$$\mathbf{H}' = \left[\mathbf{h}_0^{(1)} \quad \mathbf{h}_1^{(1)} \quad \cdots \quad \mathbf{h}_{L-1}^{(1)} \quad \mathbf{h}_0^{(2)} \quad \mathbf{h}_1^{(2)} \quad \cdots \quad \mathbf{h}_0^{(K)} \quad \mathbf{h}_1^{(K)} \quad \cdots \quad \mathbf{h}_{L-1}^{(K)}\right], \tag{4.2.29}$$

where $\mathbf{h}_i$ has the same dimension as described in (3.2.3).

Complexity

For MUD the number of states (and hereby also the number of computations) increases with the power of the number of users as well as the length of the channel. In each of the $|\Omega|^{(L\text{-}1)K}$ states the number of operations is $O(N_R|\Omega|^K)$. Thus, complexity of the MAP receiver for joint detection is $O(N_R|\Omega|^{LK})$ per symbol.

4.3 Simulation Results

Simulation Setup
To evaluate the BER performance of the receivers they are tested in various scenarios. It is primarily the type of the channel that is changed in the different tests, but in some tests the interference conditions are also changed. Simulations concerning interfering scenarios are only carried out in the Typical Urban (TU) channel type, since it is mainly in urban areas that co-channel interference occur. Furthermore, it is assumed that the channels are synchronous and only co-channel interference is considered.
In all test scenarios the oversampling factor is $N_{sps} = 2$ and the frame formats and modulation types used are identical to the ones that are used in the GSM and the EDGE system, respectively. It is assumed that frequency hopping is made between each received burst and that the channel impulse response is perfectly known. The filter length of both the LMMSE and the IQ-LMMSE receiver has been set to 32 symbol periods in all the simulations in the thesis. Ideally, the filter length should be infinite, but this is of course not possible. Therefore, it is assumed that 32 taps in the filter gives a reasonable filtration.
The number of errors that have to occur for each SNR level or CIR level are set to 1000 in all the simulations. Besides, the minimum number of transmitted bursts have been set to 1000 as well. This ensures that there are a reasonable number of errors to average over when the BER performance is determined.

In the simulations carried out in this thesis the SNR and CIR are defined in the following way. The SNR is defined as the average received signal power from the desired user divided by the noise power (in the frequency band of interest). If E_s represents the power of the received signal and if the double-side AWGN power density is $N_0/2$, it is in [8, p. 11] shown that the SNR can be expressed as

$$SNR = \frac{E_s}{N_0 B_s} . \tag{4.3.1}$$

Here it is assumed that the signal has a double-side bandwidth of $2B_s$. The CIR is defined as the average received signal power from the desired user divided by the average interference power from other users. Let the number of users be K, and denote the desired user as user 1. In [8, p. 11] it is likewise shown that the CIR can be expressed as

$$CIR = \frac{E_s^{(1)}}{\sum_{k=2}^{K} E_s^{(k)}} . \tag{4.3.2}$$

Furthermore, when the definitions above is used, the Signal-to-Interference plus Noise Ratio (SINR) is calculated in the following way

$$SINR = \frac{E_s^{(1)}}{N_0 B_s + \sum_{k=2}^{K} E_s^{(k)}} . \tag{4.3.3}$$

Test cases for GSM

The different test scenarios for GSM are listed below. In all the GSM simulations below the C_0-pulse is modeled as having the symbol length five. When referring to the channel length L, it is the complete channel impulse response, including the pulse shape. I.e. in a perfect channel (having a delta function as the impulse response), the complete channel impulse response will be $L = 5$.

1. SNR performance in an ideal channel with AWGN. Hence, the ISI in this test scenario is solely from the C_0-pulse. This gives an impulse response of length, $L = 5$. No interference from other users.
2. SNR performance in a TU0 channel with complete channel length, $L = 7$. No interference from other users.
3. SNR performance in a Hilly Terrain (HT0) channel with complete channel length, $L = 11$. No interference from other users.
4. CIR performance at a constant SNR level at 40 dB in a TU50 channel with complete channel length, L = 7. Interference from one user. This test scenario is primarily used as a test of the implementation, since it directly will be possible to compare the performance results with the ones found in [12].
5. CIR performance at a constant SNR level at 10 dB in a TU0 channel with complete channel length $L = 7$. Interference from one user.

In Figure 4.2 the result of the first test scenario is illustrated. The BER performance for all types of receivers is the same for a signal-to-noise ratio below -5 dB. It can be seen that the LMMSE receiver is most affected by the ISI introduced by the GMSK modulation. This is the reason why the performance of the LMMSE receiver will be worse for higher SNRs than the other receivers. In the simulated SNR interval (-15 to 5 dB) the performance of the IQ-LMMSE, the MAP, and the Max-Log MAP receiver is identical. The performance corresponds to the theoretical lower bound of the performance for a BPSK signal in an AWGN channel, (see. e.g. [1]). This implies that all the receivers work properly.

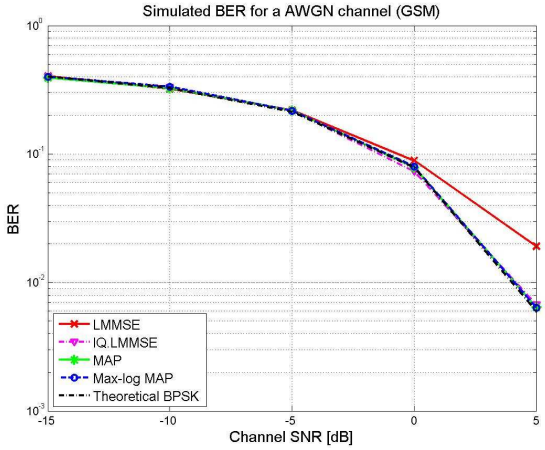

Figure 4.2. GSM simulation. BER performance in an ideal channel with AWGN.

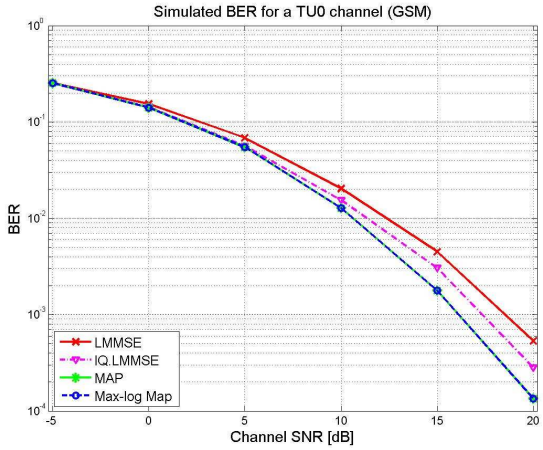

Figure 4.3. GSM simulation. BER performance in a TU0 channel (L = 7).

In Figure 4.3 the performance in the TU channel is shown. Here, the ISI is more noticeable, and, therefore, it can be expected that the best performance is obtained by the MAP receiver. This is exactly what Figure 4.3 shows. From the figure it is also seen that the Max-Log MAP receiver gives identical performance. Furthermore, it is seen that the BER performance of the IQ-LMMSE is better than LMMSE. The performance in a HT channel is shown in Figure 4.4. As in the case of the TU channel the (Max-Log) MAP receiver leads to the best performance.

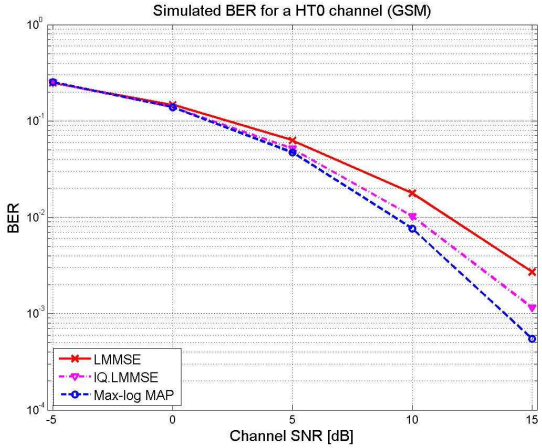

Figure 4.4. GSM simulation. BER performance in a HT0 channel (L = 11).

In Figure 4.5 the BER is plotted as a function of the CIR for a TU50 channel, when there is interference from one interferer. Figure 4.5 indicates that the best BER performance is obtained by the IQ-LMMSE receiver. The performance is considerably better than the other receivers. Actually both the IQ-LMMSE receiver and the LMMSE receiver give better performance than the MAP and the Max-Log MAP receiver. This is because the LMMSE and the IQ-LMMSE receiver take information regarding the interfering user (i.e. the covariance matrix) into account.

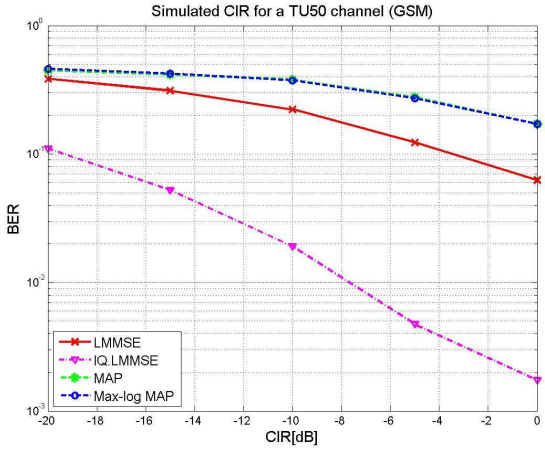

Figure 4.5. GSM simulation. BER performance in CIR scenario.
TU50 channel with one interferer and a 40dB SNR level.

A way of increasing the performance of the MAP receiver is by whitening the signal, which is shortly described in Appendix 4.2 “Whitening of Signals”. By taking the whitening approach into account in the MAP receiver it is possible to suppress some of the interference from another user or multiple users. This is exactly what is seen in Figure 4.6, where the BER performance gain of the MAP receiver has been greatly improved. However, it is possible to obtain even better performance by using MUD. In the case of perfect knowledge of the channel of all users, this will lead to the optimal solution.

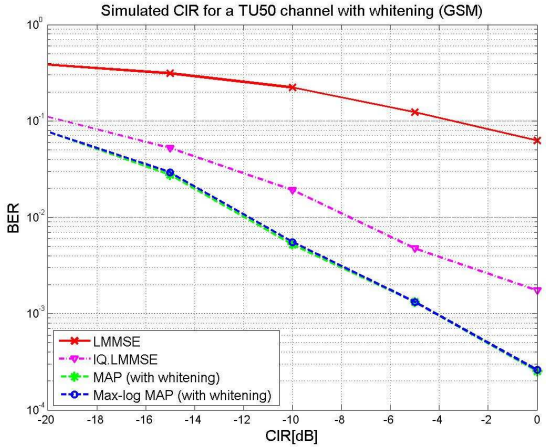

Figure 4.6. GSM simulation. BER performance in CIR scenario. TU50 channel with one interferer and a 40dB SNR level when the signal has been whitened, $N_m = 3$.

From the simulations above it is clear that the BER will decrease as the CIR level is increased for receivers relying on SUD (with or without whitening). This is expected, since the signal from the desired user will be more dominant as the CIR level is increased. For the MUD receiver it can be a little different. As a final comment to Figure 4.6 it should be noted that the BER performance of the MAP receivers are identical to the performances given in [12]. Hence, it is a good evidence that the receivers have been implemented correctly.

In the case, where there is one interfering user, is now discussed for MUD. In the case where the power level of the interfering user is considerably larger than the desired user, it will be possible to determine the symbols of the interfering user quite precisely. By performing the marginalization over the interfering user, it will be possible to estimate the posterior probability of the desired user reasonable well, since the receiver is quite sure about the interference. Hence, in poor CIR conditions the Joint MAP receiver will still perform well, which is of course a great advantage. It should be noted that this will only be true as long as there is a reasonable good channel estimate.

When the noise is small compared to the signal (i.e. at high SNRs) the worst case for the Joint MAP receiver will be at a CIR level approximately around 0dB. This is because there will be most uncertainty about both of the users in this case.

In Figure 4.7 a plot of the BER performance in a TU0 channel of the Joint MAP receiver is given. The figure illustrates the phenomenon's described above.

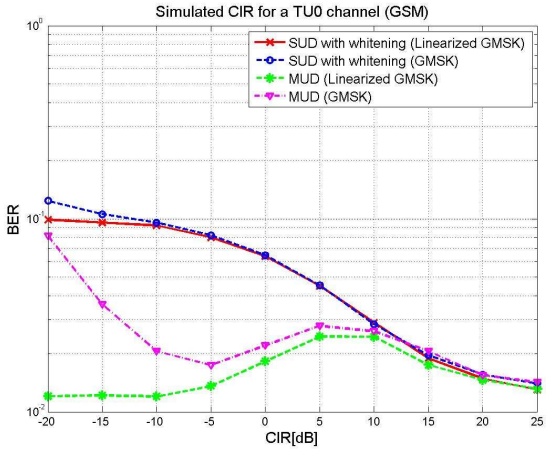

Figure 4.7. GSM simulation. CIR plot that compares the performance of SUD and MUD in a TU0 channel. SNR level at 10dB.

In Figure 4.7 the green curve illustrates MUD detection when a linearized C_0-pulse is used in the transmitter. Since the channel is perfectly known, the impulse response in the transmitter and the receiver will be the same. The worst performance for the green curve is seen at the CIR level at 5-10dB. This is because the SNR level is 10dB and, hence, the greatest uncertainty will be at this level. Hence, there is a good agreement with what has been described above for the green curve. However, when the linearized C_0-pusle is no longer used in the transmitter, there will be an approximation error in the pulse of both users. Thus at CIR = -20dB the approximation error will become dominant, which leads to a significant degradation in the BER performance as the magenta curve illustrates. The red and blue curves show the corresponding BER performance for SUD using the Max-log MAP with whitening, for linearized GMSK and GMSK, respectively. Observe, that the performance of the SUD using whitening and the MUD using Joint MAP will be the same for high CIR ratios. This is also expected, since the signal from the interfering user will have little influence on the received signal.

Test cases for EDGE
As shown in the GSM simulations above, the performance of the exact MAP and the Max-log MAP receiver is roughly the same. Therefore, only the performance for the Max-log MAP is investigated in the rest of the simulations.
The test cases specified for the GSM simulation will require a large computation load in the case of a EDGE simulation. When the channel length is $L = 7$ the complexity for each symbol will approximately be $O(N_R|\Omega|^{LK}) = 2\cdot 8^7 = 4.2\cdot 10^6$ operations (assumes that an oversampling factor of two is used, i.e. $N_R = 2$). Hence, it turns out to be very time consuming to carry out the simulations under these conditions (actually it turns out to be impossible to carry out the simulations on the 2 Gbyte RAM computers available, when $L = 7$, since it requires allocation of memory above the available amount of RAM). Therefore, the maximal channel length has been set to $L = 5$ in the EDGE simulations. This require about $65\cdot 10^3$ operations per symbol. The different test scenarios for EDGE are listed below.

1. SNR performance in an ideal channel with AWGN. Hence, the ISI in this test scenario is solely from the C_0-pulse. This gives an impulse response of the length, $L = 5$. No interference from other users.
2. SNR performance in a TU0 channel with a modified C_0-pulse. The complete channel length is, $L = 5$ in the receiver. No interference from other users. (See explanation regarding the modified C_0-pulse below).

In the TU channel the channel length will be $L = 7$ if no modification of the C_0-pulse in the receiver is made. It turns out that the contribution from the first and last tap in the C_0-pulse will be less than two percentage of the energy of the pulse (when oversampling is disregarded). Hence, it will often be a reasonable approximation to cut off the tails of the C_0-pulse in the receiver. However, some performance degradation in the receiver occurs, but it is assumed that the result will be close to the true one. When the tails of the pulse have been cut off the channel length is $L = 5$.

The result of the first test case is illustrated in Figure 4.8. From the EDGE simulation it is seen that the ISI (introduced by the pulse) have more influence on the BER performance of both the LMMSE and the Max-Log MAP receiver compared to the GSM case. This is due to the increase in the alphabet size, hence several types of decision errors can occur. It can also be seen that the BER performance of the Max-log MAP receiver is significantly better than the LMMSE receiver. Hence, a large performance gain can be obtained if optimal SbS detection is applied. It should be stated that the theoretical curve is an approximation where only errors to the closest constellation points are

considered. This is a reasonable approximation for large SNR, but will still be a lower bound since no ISI is considered in derivation of the theoretical approximation. The derivation of the approximation is given in [1]. Due to limitation of simulation time, the number of errors, which have to occur at the SNR = 18dB, have been set to 400. Hence, there can be some uncertainty for the BER at SNR = 18dB in Figure 4.8.

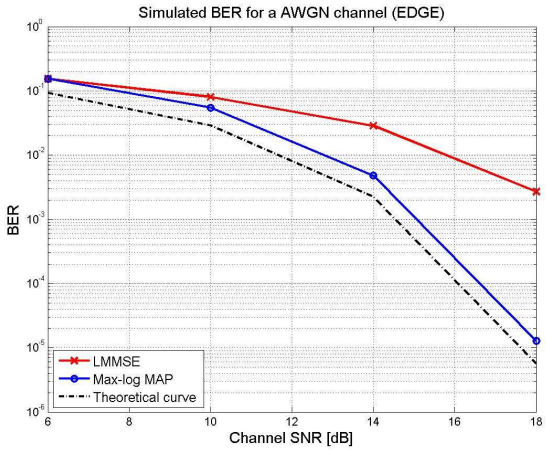

Figure 4.8. EDGE simulation. BER performance in an ideal channel with AWGN.

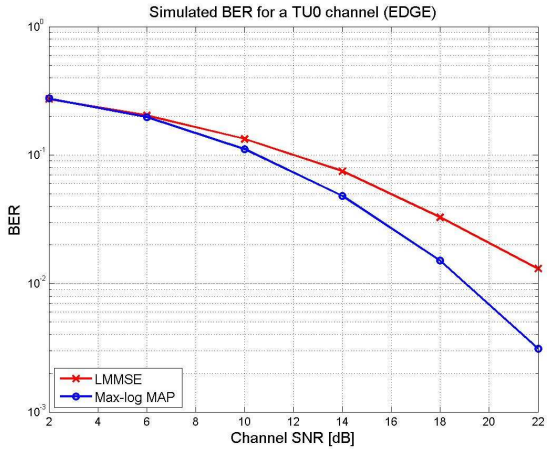

Figure 4.9. EDGE simulation. SNR plot in a TU channel (modified C_0-pulse)

In Figure 4.9 the BER performance in a TU0 channel is given when the modified C_0-pulse is used. From both Figure 4.8 and Figure 4.9 it is seen that considerably gain in BER performance can be achieved by using the (Max-log) MAP receiver.

4.4 Summary

The LMMSE receiver has been derived. It is a linear receiver, which minimizes the mean squared error. A gain in BER performance can in frequency-selective channels be obtained by using the nonlinear IQ-LMMSE receiver instead. However, it has a complexity will be considerably larger, since the covariance matrix will be twice as large in each dimension.

Some of the optimal receivers in frequency-selective channels have been introduced. Especially the MAP receiver has been investigated. It uses a forward and a backward recursion to calculate the transition probabilities between the state in the trellis diagram. Based on the probabilities of the state transitions it is possible to calculate the probabilities of the symbols.

When there is no interference from other users the simulation results show that the best BER performance is always obtained with the MAP receiver. However, when there is considerably CCI, the traditional MAP receiver gives a very poor performance. This is because the MAP receiver does not take the noise from interfering users into account (i.e. the covariance matrix). Hence, either a whitening approach or the Joint MAP receiver is needed in interfering scenarios. Then the latter method will lead to the best performance when it is assumed that there is perfect knowledge of the channel.

5 Sphere Decoding

This section deals with Sphere Decoding (SD) which can be used to reduce the complexity of the receiver that gives the MLSE solution. In the first subsection the fundamental idea behind SD is introduced. In the next subsection it is described how Sphere Decoding can be applied in frequency-selective channels. As it will be described, the choice of radius for the sphere has a significant impact on the complexity of the receiver, and, therefore, this subject is addressed in the following subsection. The way of searching through the trellis diagram does also have influence on the complexity and will, likewise, be discussed. In Subsection 5.5 the usage of the LMMSE solution in combination with SD is proposed. Finally, it is discussed how to perform MUD using SD in Subsection 5.6.

5.1 Basic Concept of Sphere Decoding

In this subsection it is described how the exact Maximum Likelihood Sequence Estimation (MLSE) solution can be found by using SD.

5.1.1 The Integer Least-Squares Problem

The Sphere Decoding algorithm address the problem of finding the integer least-squares solution in a clever way. Stated in another way, the SD algorithm tries to solve the least-squares problem when the unknown vector consists of integer elements. This is also referred to as the *closest lattice point problem*. In general the integer least-squares problem can be expressed as

$$\hat{\mathbf{s}} = \min_{\mathbf{s}\in\mathbb{CZ}^M} \|\mathbf{x} - \mathbf{H}\mathbf{s}\|^2, \tag{5.1.1}$$

where $\mathbf{x}$ denotes the received signal, $\mathbf{x} \in \mathbb{C}^N$, and $\mathbf{s}$ is the vector of interest, $\mathbf{s} \in \mathbb{CZ}^M$. It should here be noted that, $\mathbb{CZ}^M$, represents an integer lattice [19]. The matrix $\mathbf{H}$ is given as $\mathbf{H} \in \mathbb{C}^{N\times M}$ and in this subsection it is assumed that the matrix does not have any special structure.
In communication, (as well as in many other practical problems), the search space will have a finite size, and therefore the problem which needs to be solved can be expressed as

$$\hat{\mathbf{s}} = \min_{\mathbf{s}\in D\subset\mathbb{CZ}^M} \|\mathbf{x} - \mathbf{H}\mathbf{s}\|^2, \tag{5.1.2}$$

where $D \subset \mathbb{CZ}^M$. In Figure 5.1 and Figure 5.2 a simplified illustration of space of **s** and **Hs** is given.

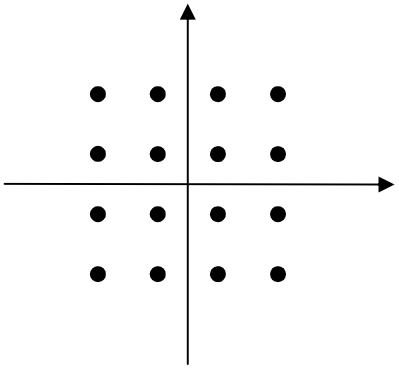

Figure 5.1. Simple illustration of space of s.

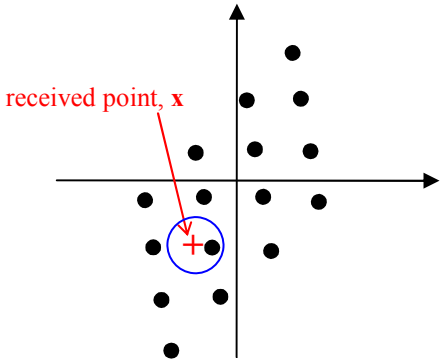

Figure 5.2. Simple illustration of space of Hs.

As illustrated in Figure 5.2, the integer least-squares problem can geometrically be represented as finding the closest lattice point in a skewed lattice, **Hs**, to the received signal, **x** [18]. In general the integer least-squares problem is much more complicated to solve than the "standard" least squares problem, where the solution can be found by calculating the pseudo-inverse matrix, $\mathbf{H}^{+}$. In fact the exact methods for solving the integer least squares problem will in general have an expected complexity that is exponential (in respect to the dimensions), if the received signal, **x**, is arbitrary, see [18]. However, when it is assumed that the received point is a lattice point with additive Gaussian noise added, it is in [18] proved that the expected complexity is polynomial over a wide range of SNRs. As described in [18] the worse-case complexity will still be exponential though.

5.1.2 Principle of SD

The basic principle of the SD algorithm is quite simple. As illustrated in Figure 5.3 it tries to solve the problem by only searching over a limited number of lattice points inside a sphere around the received point.

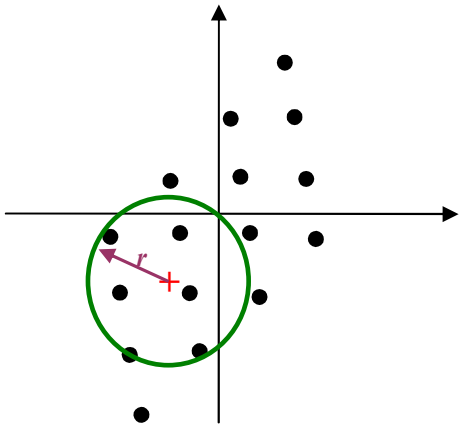

Figure 5.3. Principle of Sphere Decoding with radius, *r*.

From Figure 5.3 it is seen that the closest lattice point in the circle will of course also be the closest point in the full grid. Notice that the radius of the circle denoted by *r* which from now on will refer to the radius and not the received signal as it has previously been the case.
In general, the search space will be multi-dimensional and the circle in Figure 5.3 will then be a hyper-sphere in this multi dimensional space. Even though the idea behind SD is quite simple, there are two new issues which must be dealt with.

- First of all a method is needed to tell which points are inside the sphere, since the calculation of distances from the received point to all the lattice points again will be exponential in complexity. Then there is no advantage in performing SD.
- Secondly, a method to choose the radius, *r*, of the sphere is needed. The size of the radius will have a great impact on the complexity of the algorithm. If the chosen radius is too small there will be no lattice point inside the sphere. On the other hand, a too large radius will include too many points and the complexity of the algorithm will still be nearly exponential.

As described in [18] a natural choice of the radius will be the *covering radius* of the lattice. This is defined to be the smallest radius the circle can have, if it should cover the whole lattice space, when it is centered at each of the lattice point. An illustration of this covering radius is given in Figure 5.4.

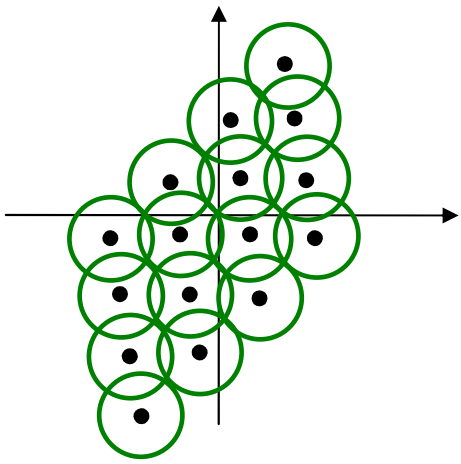

Figure 5.4. Covering radius of the lattice.

From Figure 5.4 it is seen that the covering radius is the smallest radius, which ensures that minimum one lattice point is inside the sphere, for any received point, **x**. However, the problem of finding the covering radius is also exponential in complexity, [18]. Therefore, it is not a trivial issue to determine a proper radius of the sphere and this issue will be discussed in more detail in Subsection 5.3 “Choosing the Radius”. A way to handle the first issue will now be described.

QR Factorization

Since it is complicated to determine which points are inside a hyper-sphere in the *M*-dimensional space, the problem needs to be simplified. As an initial search, the one-dimensional space can be examined instead. This corresponds to finding the lattice points in a given interval. The points that lie inside the interval are candidates to lie inside the sphere, while the rest can be disregarded. When the points that lie in the one-dimensional interval have been found, the dimension can be increased by one and it is tested if the remaining points lie in the new interval. This approach can be repeated for each dimension until dimension *M* is reached. Therefore, the SD algorithm can be thought of as creating a tree where the branches at each level represent the lattice points that are inside the sphere. This is illustrated in Figure 5.5.

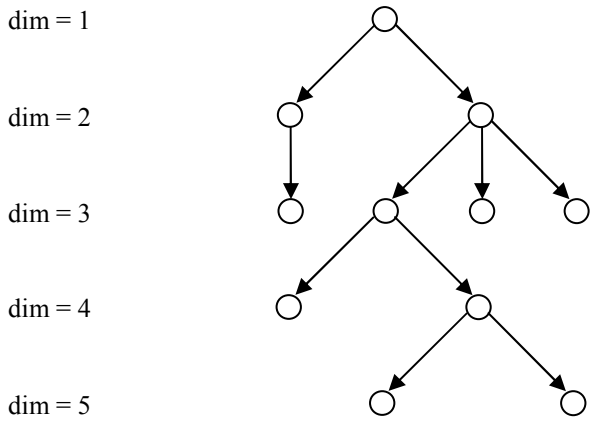

Figure 5.5. Example of branches of a tree in the case of a 5 dimensional search space.

In Figure 5.5 an example of a search in a 5 dimensional space is given. It is seen that only two lattice points lie inside the 5 dimensional hyper-sphere. From the example it is clear that the complexity of the algorithm depends on the number of points that the tree includes. Thus, the better the pruning of the tree is, the less the complexity will be. However, a too rough pruning will result in an empty sphere, i.e. no solution is found. The approach of comparing the radius of the hyper-sphere with the distance from the received point to the lattice points can be expressed as

$$r^2 \geq \left\| \mathbf{x} - \mathbf{Hs} \right\|^2 . \tag{5.1.3}$$

It is this problem that the SD algorithm solves and it is in this thesis denoted the Sphere Decoding (SD) Constraint. It is assumed that there are at least as many equations as unknowns, i.e. $N \geq M$. If the latter is not the case, the SD algorithm can only be used to solve the linear system of equations up to dimension N and a full search in the rest of the dimensions (i.e. $M - N$) will then have to be carried out. Furthermore, it is assumed that $\mathbf{H}$ has full (row) rank.
To be able to search in one dimension it is necessary to make a QR factorization of the matrix, $\mathbf{H}$. Then (5.1.3) becomes

$$r^2 \geq \left\| \mathbf{x} - \mathbf{QRs} \right\|^2 . \tag{5.1.4}$$

where $\mathbf{Q}$ is an $N \times N$ unitary matrix and $\mathbf{R}$ is an $N \times M$ upper triangular matrix with positive real diagonal elements. For every matrix a QR factorization exists. The unitary matrix has the special property that it is nonsingular (see e.g. [29]) and

$$\mathbf{Q}^H = \mathbf{Q}^{-1}, \tag{5.1.5}$$

which implies that

$$\mathbf{QQ}^H = \mathbf{I} . \tag{5.1.6}$$

Therefore, the columns of $\mathbf{Q}$ are orthonormal. This will also be the case for the rows, since $\mathbf{Q}^H$ will also be an unitary matrix, see [29]. Furthermore, the Euclidean norm will not be altered when a matrix is multiplied with an unitary matrix. Hence, it is possible to rewrite (5.1.4) to

$$\begin{aligned} r^2 &\geq \left\|\mathbf{x}-\mathbf{QRs}\right\|^2 = \left(\mathbf{x}-\mathbf{QRs}\right)^H\left(\mathbf{x}-\mathbf{QRs}\right) \\ r^2 &\geq \left(\mathbf{x}-\mathbf{QRs}\right)^H \mathbf{Q}\mathbf{Q}^H\left(\mathbf{x}-\mathbf{QRs}\right) \\ r^2 &\geq \left(\mathbf{Q}^H\mathbf{x}-\mathbf{Rs}\right)^H\left(\mathbf{Q}^H\mathbf{x}-\mathbf{Rs}\right) \\ r^2 &\geq \left\|\mathbf{Q}^H\mathbf{x}-\mathbf{Rs}\right\|^2. \end{aligned} \tag{5.1.7}$$

When $N > M$ there are more equations than unknowns and (5.1.7) can be expressed as

$$r^2 \geq \left\| \begin{bmatrix} \mathbf{Q}_1^H \\ \mathbf{Q}_2^H \end{bmatrix}\mathbf{x} - \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0}_{(N-M)\times M} \end{bmatrix}\mathbf{s} \right\|^2, \tag{5.1.8}$$

where $\mathbf{R}_1$ is an $M \times M$ upper triangular matrix (with positive real diagonal elements), $\mathbf{Q}_1^H$ is the first M rows of $\mathbf{Q}^H$, and, likewise, $\mathbf{Q}_2^H$ is the last N-M rows. Due to the zeros in the $\mathbf{R}$ matrix, (5.1.8) is equal to

$$r^2 \geq \left\|\mathbf{Q}_1^H\mathbf{x}-\mathbf{R}_1\mathbf{s}\right\|^2 + \left\|\mathbf{Q}_2^H\mathbf{x}\right\|^2, \tag{5.1.9}$$

and by introducing the new variables $\breve{\mathbf{x}} = \mathbf{Q}_1^H\mathbf{x}$ and $\breve{r}^2 = r^2 - \left\|\mathbf{Q}_2^H\mathbf{x}\right\|^2$ the expression is simplified to

$$\breve{r}^2 \geq \left\|\breve{\mathbf{x}}-\mathbf{R}_1\mathbf{s}\right\|^2. \tag{5.1.10}$$

In the following, the elements in $\mathbf{R}_1$ are denoted R_{ij} (not to be confused with the radius) and, likewise, the elements in $\breve{\mathbf{x}}$ and $\mathbf{s}$ is denoted $\breve{x}_i$ and s_i, respectively. Equation (5.1.10) is then given as

$$\breve{r}^2 = \left\| \begin{bmatrix} \breve{x}_1 \\ \breve{x}_2 \\ \vdots \\ \breve{x}_{M-1} \\ \breve{x}_M \end{bmatrix} - \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1(M-1)} & R_{1M} \\ 0 & R_{22} & \cdots & R_{2(M-1)} & R_{2M} \\ \vdots & \ddots & \ddots & \vdots & \vdots \\ 0 & \cdots & 0 & R_{(M-1)(M-1)} & R_{(M-1)M} \\ 0 & \cdots & 0 & 0 & R_{MM} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \vdots \\ s_{M-1} \\ s_M \end{bmatrix} \right\|^2. \tag{5.1.11}$$

Since $\mathbf{R}_1$ has an upper triangular structure, it will be possible to examine a single dimension at a time, and find out which lattice points lie in the sphere. Thus, if a lattice point lies inside the sphere it definitely has to fulfil

$$\breve{r}^2 \geq \left\|\breve{x}_M - R_{MM}s_M\right\|^2. \tag{5.1.12}$$

For simplicity it is for a moment assumed that all terms in (5.1.12) are real numbers. In next section the complex case is treated. When the terms are real it is easy to see that

$$\begin{aligned} \breve{r}^2 &\geq \left(\breve{x}_M - R_{MM}s_M\right)^2 \\ \breve{r} &\geq \pm\left(\breve{x}_M - R_{MM}s_M\right). \end{aligned} \tag{5.1.13}$$

When s_M is defined to be an integer number, the interval for s_M is determined as

$$\left\lceil \frac{-\breve{r}+\breve{x}_M}{R_{MM}} \right\rceil \le s_M \le \left\lfloor \frac{\breve{r}+\breve{x}_M}{R_{MM}} \right\rfloor. \tag{5.1.14}$$

Now a stronger requirement can be introduced for each s_M satisfying (5.1.14)

$$\breve{r}^2 \ge \left(\breve{x}_{M-1} - R_{(M-1)(M-1)} s_{M-1} - R_{(M-1)M} s_M\right)^2 + \left(\breve{x}_M - R_{MM} s_M\right)^2. \tag{5.1.15}$$

If $\breve{r}_{M-1}^2 = \breve{r}^2 - \left(\breve{x}_M - R_{MM} s_M\right)^2$ and $\breve{x}_{(M-1)|M} = \breve{x}_{M-1} - R_{(M-1)M} s_M$ the interval for $s_{M\text{-}1}$ is given as

$$\left\lceil \frac{-\breve{r}_{M-1}+\breve{x}_{(M-1)|M}}{R_{(M-1)(M-1)}} \right\rceil \le s_{M-1} \le \left\lfloor \frac{\breve{r}_{M-1}+\breve{x}_{(M-1)|M}}{R_{(M-1)(M-1)}} \right\rfloor. \tag{5.1.16}$$

It is of course possible to continue this approach until the full dimensional space is reached, i.e. when boundaries for s_1 have been found. In that way it is possible to set up boundaries for lattice points for each dimension, while the requirements they have to fulfil will be tightened as the dimensions increase. Therefore, the lattice points that fulfil the sphere decoding constraint will be found in a computational efficient way.
As described in the beginning of this section, in [18] it is shown that the average complexity will be polynomial instead of exponential. This will also be the case when the signals are complex, (when the complexity is measured in complex operations instead). Actually, the complexity is roughly cubic over the range of high SNRs. However, both for the real-valued and complex signals the complexity will only be polynomial, when the received point is not arbitrary, but instead assumed to be a lattice point distorted by AWGN. This can be expressed as

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{z} \tag{5.1.17}$$

where $\mathbf{z} \in \mathrm{CN}\left(\mathbf{0}, \sigma_c^2 \mathbf{I}\right)$, here σ_c^2 is the variance of the complex noise. As described in Section 3.2 "Discrete Model for Multipath Channels", the received signals in radio channels are assumed to be corrupted by AWGN. Hence, the assumption, which leads to (5.1.17) used in [18 and 19], will be fulfilled. Therefore, it is very interesting to examine if the SD algorithm can reduce the expected complexity of the optimal detector so it becomes polynomial.

5.2 Sphere Decoding in Frequency-Selective Channels

As described in Section 3.2 "Discrete Model for Multipath Channels" the model for the received signal will lead to a block banded block Toeplitz channel convolution matrix, **H**, given in (3.2.7). It is shown in [19, 20 and 21] that it is possible to use sphere decoding directly on **H**, without performing the QR factorization in the case of SUD. In the rest of this thesis it will be assumed that **H** will have this block banded block Toeplitz structure.
Recall that the model for the received signal in the single user scenario is

$$\mathbf{x}_n = \sum_{i=0}^{L-1} \mathbf{h}_i \cdot s_{n-i} + \mathbf{z}_n, \qquad n = 1,2,\ldots, N_{BL} + L - 1. \tag{5.2.1}$$

In this case the channel matrix, **H**, in (3.2.7) becomes

$$\mathbf{H}_{SUD} = \begin{bmatrix} \mathbf{h}_0 & 0 & \cdots & \cdots & 0 \\ \mathbf{h}_1 & \mathbf{h}_0 & 0 & \cdots & \vdots \\ \vdots & \mathbf{h}_1 & \ddots & \ddots & \vdots \\ \mathbf{h}_{L-1} & \ddots & \ddots & \ddots & 0 \\ 0 & \mathbf{h}_{L-1} & \ddots & \ddots & \mathbf{h}_0 \\ \vdots & 0 & \ddots & \ddots & \mathbf{h}_1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \mathbf{h}_{L-1} \end{bmatrix}. \tag{5.2.2}$$

A necessary condition to satisfy the SD constraint given in (5.1.3) is

$$r^2 \geq \left\| \mathbf{x}_1 - \mathbf{h}_0 s_1 \right\|^2. \tag{5.2.3}$$

It is seen that there is a clear parallel between (5.1.12) and (5.2.3). Therefore, it will be possible to examine one complex dimension at a time. To ease the derivation of the SD algorithm for the banded Toeplitz matrix with complex elements a slightly different notation of the signals of interest is needed. Let the received signal from time index 1 to *n* be represented as

$$\mathbf{x}_1^n = \left[\mathbf{x}_1{}^T, \mathbf{x}_2{}^T, \ldots, \mathbf{x}_n{}^T \right]^T, \quad where \quad n \leq N_{BL} + L - 1. \tag{5.2.4}$$

In (5.2.4) each vector has the dimension $\mathbf{x}_n \in \mathrm{C}^{N_R \times 1}$, hence $\mathbf{x}_1^n \in \mathrm{C}^{(n \cdot N_R) \times 1}$. N_R is the number of receive dimensions, (i.e. it could be the oversampling factor or the number of receive antennas). Likewise, the vector containing transmitted symbols in the time index from 1 to *n* is defined as

$$\mathbf{s}_1^{n'} = \left[\mathbf{s}_1^T, \mathbf{s}_2^T, \ldots, \mathbf{s}_{n'}^T \right]^T, \tag{5.2.5}$$

where *n*' = min(*n*, N_{BL}) and each vector has the dimension $\mathbf{s}_i \in C^{N_T \times 1}$. N_T is the number of transmit dimensions, (i.e. it could be the number of users). Thus, the size of the vector does not increase when $n > N_{BL}$, (i.e. when all symbols in burst have been transmitted).

Correspondingly, the channel convolution matrix, $\mathbf{H}_1^n$, is defined to be a $(n \cdot N_R) \times (n \cdot N_T)$ matrix for $n \leq N_{BL}$, and to be a $(n \cdot N_R) \times (N_{BL} \cdot N_T)$ matrix for $N_{BL} < n \leq N_{BL} + L - 1$. Hence, in the end of a burst (i.e. at time index $n = N_{BL} + L - 1$), the matrix is given as

$$\mathbf{H}_1^{N_{BL}+L-1} = \mathbf{H} = \begin{bmatrix} \mathbf{H}_0 & 0 & \cdots & \cdots & 0 \\ \mathbf{H}_1 & \mathbf{H}_0 & 0 & \cdots & \vdots \\ \vdots & \mathbf{H}_1 & \ddots & \ddots & \vdots \\ \mathbf{H}_{L-1} & \ddots & \ddots & \ddots & 0 \\ 0 & \mathbf{H}_{L-1} & \ddots & \ddots & \mathbf{H}_0 \\ \vdots & 0 & \ddots & \ddots & \mathbf{H}_1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \mathbf{H}_{L-1} \end{bmatrix}, \tag{5.2.6}$$

where each sub-matrix, $\mathbf{H}_i$, will have the dimension $N_R \times N_T$. By using the introduced notation, the necessary condition to satisfy the SD constraint at time index *n* then becomes

$$r^2 \geq \left\| \mathbf{x}_1^n - \mathbf{H}_1^n \mathbf{s}_1^{n'} \right\|^2 . \tag{5.2.7}$$

For simplicity the radius is kept fixed in the derivation below, but as it will be shown in Section 5.3, the radius can be a function of the time index, *n*.
From (5.2.7) it is seen that the requirement on the symbols are tightened, when the time index is increased, as described in the general case in Section 5.1 "Basic concept of Sphere Decoding".

5.2.1 Combining the Trellis Structure with the SD

As noted in Section 4.2.1 "Maximization of the Posterior Probability", the MLSE solution is given as

$$\hat{\mathbf{s}}_{MLSE} = \arg\min_{\mathbf{s}} \left\| \mathbf{x} - \mathbf{H}\mathbf{s} \right\|^2 .$$

Hence, the cost function that is to be minimized is

$$C_{N_{BL}} = \left\| \mathbf{x}_1^n - \mathbf{H}_1^n \mathbf{s}_1^{n'} \right\|^2 = \sum_{n=1}^{N_{BL}} \left\| \mathbf{x}_n - \sum_{i=0}^{L-1} \sum_{k=1}^{K} \mathbf{h}_i^{(k)} s_{n-i}^{(k)} \right\|^2 . \tag{5.2.8}$$

Notice that this cost function can be calculated recursively as

$$C_n = C_{n-1} + \left\| \mathbf{x}_n - \sum_{i=0}^{L-1} \sum_{k=1}^{K} \mathbf{h}_i^{(k)} s_{n-i}^{(k)} \right\|^2 , \quad where \quad C_0 = 0. \tag{5.2.9}$$

By using the Viterbi algorithm (or equivalently, the forward Max-Log MAP algorithm), the most probable path through the trellis diagram is found. Therefore, both methods find the sequence which minimizes the cost function in (5.2.8). Alternatively, the SD algorithm can be used to optimize the same cost function.
As noted in Section 4.2, the complexity of the Viterbi algorithm grows exponentially with the length of the channel. However, it is linear in the length of the data sequence as shown above and in [21], which is not necessarily the case for the SD algorithm. Furthermore, the SD algorithm does not exploit the trellis structure. Therefore, the idea in [20 and 21] is to combine the two approaches and hereby use the good properties of each of the algorithms.

Viterbi Algorithm with SD Constraint
When the traditional Viterbi algorithm is implemented, only the smallest path through the trellis diagram to a given state is kept as illustrated in Figure 5.6.

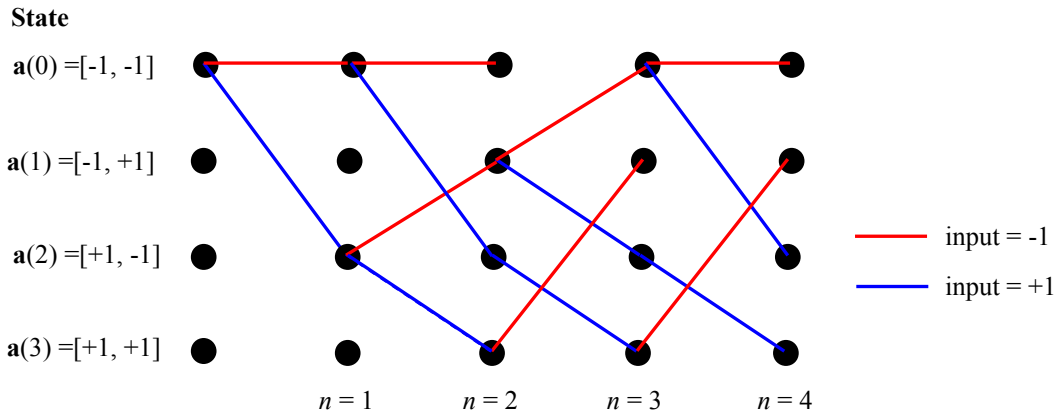

Figure 5.6. Trellis diagram in the traditional Viterbi algorithm.

If additionally a SD constraint has to be fulfilled, it is clear from (5.2.8) that the cost function in a given state has to be smaller than the radius, i.e.

$$C_n = \left\| \mathbf{x}_1^n - \mathbf{H}_1^n \mathbf{s}_1^{n'} \right\|^2 \leq r^2 . \tag{5.2.10}$$

If this is not the case the state is pruned away, and the state space will be reduced. This is illustrated in Figure 5.7.

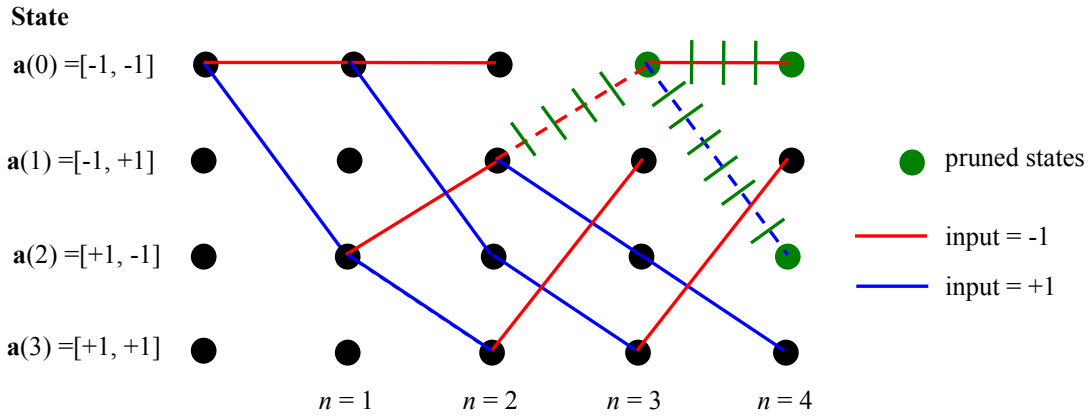

Figure 5.7. Pruned trellis diagram in the Viterbi algorithm with SD constraint.

From Figure 5.7 it is seen that the number of paths in the trellis diagram have been reduced. The SD constraint on top of the Viterbi algorithm can therefore be regarded as a dynamical way of pruning the state space. This way of pruning the state space can also be used in the MAP receiver, which indirectly calculates the distance from the received signal to the expected output in the given state, see e.g. (4.2.20). Then the distance covered in the trellis diagram (i.e. the cost function) additionally must be stored for all the states that have not been pruned away. Since some of the states in the trellis diagram are pruned away, it will no longer be the exact posterior probability, which is found in the SbS MAP receiver, but an approximation.

Implementation Issues

In the implementation of the combined algorithm it should be noted that special conditions occur in the beginning and end of a GSM or EDGE burst. For example the first symbol must fulfil

$$r^2 \geq \left\| \mathbf{x}_1 - \mathbf{h}_0 s_1 \right\|^2 .$$

However, the received signal will be affected by the tail bits as well as the signal conditions in the ramp up phase of the burst. These "disturbances" must be removed from the signal. In case of a perfect knowledge of the channel and if the bits before time index $n = 1$ can be assumed to be e.g. +1, it is possible to remove the disturbances by subtracting the channel coefficients of "no interest" from $\mathbf{x}$, i.e.

$$\mathbf{x}_{1,start\_up} = \mathbf{x}_1 - \sum_{i=1}^{L-1} \mathbf{h}_i , \quad if \quad s_j = 1, \quad for \quad j = \left[-L+1, \ldots, 0 \right].$$

Then $\mathbf{x}_{1,start\_up}$ is used instead of $\mathbf{x}_1$. The same type of problem must be dealt with at the end of the burst.

5.2.2 Bounding the Symbols

As described in the previous subsection, [20] proposes to prune the trellis diagram dynamically by comparing the radius with the cost function at each time index. However, since it is possible to determine the boundaries of the symbols, as explained in Section 5.1 "Basic Concepts of Sphere Decoding in Frequency Flat Channels", the complexity can be reduced even further. Thus, based on these boundaries it is possible to reduce the number of state transitions, for which the cost function has to be calculated. In this subsection the prediction of the states that fulfil the SD constraint is derived. More precisely, it is shown that by comparing the cost function of the current state with the radius of the sphere, the boundaries of the next symbol (and hereby the possible state transitions) can be found.

As noted in the beginning of this subsection, the boundaries of the symbols can only be found in the case of a single user, unless the triangular factorization of the $\mathbf{H}$ matrix is computed, which is described in Subsection 5.6 "MUD using SD". The derivation of the boundaries for the symbols in the next state relies on the basic idea of SD. It should be noted that derivation has been assisted by L.P.B. Christensen and Z. Safar.

Single User Detection

As already mentioned in the previous section, an important observation (given in (5.2.10)) is that the cost function has to be smaller than the radius to fulfil the SD constraint. In the derivation below a potential variation of the radii have been taken into account as well, which is the reason why there is an index on the radius. In the case where the same radius is kept throughout the search, the radius will simply be the same for all time indices. Equation (5.2.10) can in the case of variation of the radius be written as

$$r_n^2 \geq \left\| \mathbf{x}_1^n - \mathbf{H}_1^n \mathbf{s}_1^{n'} \right\|^2 = C_n , \quad where \quad n' = \min\left(n, N_{BL}\right). \tag{5.2.11}$$

As mentioned above it is assumed that $N_T = 1$. It is now feasible to reformulate (5.2.11) as

$$r_n^2 \geq \left\| \mathbf{x}_n - \sum_{i=0}^{L-1} \mathbf{h}_i s_{n-i} \right\|^2 + C_{n-1} = C_n . \tag{5.2.12}$$

Equation (5.2.12) can now be rewritten to

$$
\begin{aligned}
r_n^2 - C_{n-1} &\geq \left\| \mathbf{x}_n - \sum_{i=0}^{L-1} \mathbf{h}_i s_{n-i} \right\|^2 \\
r_n^2 - C_{n-1} &\geq \left\| \mathbf{x}_n - \left(\mathbf{h}_0 s_n + \mathbf{h}_1 s_{n-1} + \ldots + \mathbf{h}_{L-1} s_{n-L+1} \right) \right\|^2 \\
r_n^2 - C_{n-1} &\geq \left\| \mathbf{x}_n - \left(\mathbf{h}_1 s_{n-1} + \ldots + \mathbf{h}_{L-1} s_{n-L+1} \right) - \mathbf{h}_0 s_n \right\|^2 .
\end{aligned} \tag{5.2.13}
$$

By introducing $\tilde{r}_n^2 = r_n^2 - C_{n-1}$ and $\tilde{\mathbf{x}}_n = \mathbf{x}_n - \left(\mathbf{h}_1 s_{n-1} + \ldots + \mathbf{h}_{L+1} s_{n-L+1} \right)$, (5.2.13) can be simplified to

$$
\tilde{r}_n^2 \geq \left\| \tilde{\mathbf{x}}_n - \mathbf{h}_0 s_n \right\|^2 . \tag{5.2.14}
$$

From (5.2.14) it is realized that bounding of the symbols can be made in a general way at arbitrary time indices based on the knowledge of the cost function at the previous time index. Hence, in a given state in the Markov model, the transitions that fulfil the SD constraint can be computed without calculating the cost function of all the potential transitions (which is equal to $|\Omega|$ for SUD). First if a state transition fulfils the SD constraint, the cost function is found.
When there is no oversampling of the signal it is quite simple to solve (5.2.14) in respect to s_n. In the case of multiple receive dimensions (i.e. oversampling) it becomes more complicated. Then a QR factorization of the vector $\mathbf{h}_0$ is needed. Let the vector, $\mathbf{h}_0$, be represented as

$$
\mathbf{h}_0 = \left\| \mathbf{h} \right\| \hat{\mathbf{h}}, \quad \textit{where} \quad \left\| \hat{\mathbf{h}} \right\| = 1 . \tag{5.2.15}
$$

A unitary matrix $\tilde{\mathbf{Q}}$ can now be constructed such that it contains the column vectors

$$
\tilde{\mathbf{Q}} = \begin{bmatrix} \hat{\mathbf{h}} & \mathbf{q}_1 & \cdots & \mathbf{q}_{N_R - 1} \end{bmatrix} . \tag{5.2.16}
$$

The column vectors $\mathbf{q}_i$'s should have unit length and be orthogonal to $\hat{\mathbf{h}}$ and to each other. Hence,

$$
\left\| \mathbf{q}_i \right\| = 1 \quad \textit{and} \quad \hat{\mathbf{h}}^H \mathbf{q}_i = 0 \quad \textit{for} \quad i \in \left[1, \ldots, N_R - 1 \right] . \tag{5.2.17}
$$

This implies that

$$
\hat{\mathbf{h}} = \tilde{\mathbf{Q}} \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \quad \textit{and} \quad \tilde{\mathbf{Q}}^H \hat{\mathbf{h}} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} . \tag{5.2.18}
$$

In the case where the receive dimensions are either $N_R = 2$ or $N_R = 4$ there exists an analytical expression for the $\tilde{\mathbf{Q}}$ matrix, see e.g. [31]. If the oversampling factor is larger than $N_R > 4$ and $\hat{\mathbf{h}}$ contains complex values it is shown in [31] that no analytical expression of $\tilde{\mathbf{Q}}$ exists. Instead e.g. the Gramm-Schmidt procedure can be used to find the elements in $\tilde{\mathbf{Q}}$. This procedure is among others described in [29, pp. 15-16].
In the case of an oversampling factor of two, which is used in all the simulations in this thesis, the $\mathbf{q}_1$ vector can be constructed in two ways, i.e.

$$
\textit{Let} \quad \hat{\mathbf{h}} = \begin{bmatrix} \hat{h}_1 \\ \hat{h}_2 \end{bmatrix} \quad \textit{then} \quad \mathbf{q}_1 = \begin{bmatrix} -\hat{h}_2^* \\ \hat{h}_1^* \end{bmatrix} \quad \textit{or} \quad \mathbf{q}_1 = \begin{bmatrix} \hat{h}_2^* \\ -\hat{h}_1^* \end{bmatrix} . \tag{5.2.19}
$$

As mentioned earlier the goal is to isolate s_n in (5.2.14). Based on (5.2.15), the equation is rewritten as

$$\begin{aligned}\widetilde{r}_n^{\,2} &\geq \left\|\widetilde{\mathbf{x}}_n - \mathbf{h}_0 s_n\right\|^2 \\ &\geq \|\mathbf{h}\|^2 \left\|\frac{\widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|} - \hat{\mathbf{h}} s_n\right\|^2 .\end{aligned} \tag{5.2.20}$$

(5.2.20) is then divided by $\|\mathbf{h}\|^2$ and (5.2.16) and (5.2.17) are used (recall that the unitary matrix does not change the Euclidean norm). The expression then becomes

$$\begin{aligned}\frac{\widetilde{r}_n^{\,2}}{\|\mathbf{h}\|^2} &\geq \left\|\frac{\widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|} - \hat{\mathbf{h}} s_n\right\|^2 \\ \frac{\widetilde{r}_n^{\,2}}{\|\mathbf{h}\|^2} &\geq \left\|\widetilde{\mathbf{Q}}^H \frac{\widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|} - \widetilde{\mathbf{Q}}^H \hat{\mathbf{h}} s_n\right\|^2 \\ \frac{\widetilde{r}_n^{\,2}}{\|\mathbf{h}\|^2} &\geq \left\|\frac{1}{\|\mathbf{h}\|}\begin{bmatrix} \hat{\mathbf{h}}^H \widetilde{\mathbf{x}}_n \\ \mathbf{q}_1^H \widetilde{\mathbf{x}}_n \\ \vdots \\ \mathbf{q}_{N_R-1}^H \widetilde{\mathbf{x}}_n \end{bmatrix} - \begin{bmatrix} s_n \\ 0 \\ \vdots \\ 0 \end{bmatrix}\right\|^2\end{aligned} \tag{5.2.21}$$

It is now seen that s_n only occurs in one of the terms, and therefore (5.2.21) is equivalent to

$$\frac{\widetilde{r}_n^{\,2}}{\|\mathbf{h}\|^2} \geq \left|\frac{\hat{\mathbf{h}}^H \widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|} - s_n\right|^2 + \frac{1}{\|\mathbf{h}\|^2}\sum_{i=1}^{N_R-1}\left|\mathbf{q}_i^H \widetilde{\mathbf{x}}_n\right|^2 \tag{5.2.22}$$

The last term is then subtracted on both sides and (5.2.22) then becomes

$$\left|\frac{\hat{\mathbf{h}}^H \widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|} - s_n\right|^2 \leq \frac{1}{\|\mathbf{h}\|^2}\left(\widetilde{r}_n^{\,2} - \sum_{i=1}^{N_R-1}\left|\mathbf{q}_i^H \widetilde{\mathbf{x}}_n\right|^2\right) \tag{5.2.23}$$

From (5.2.23) it is now possible to predict the symbols that fulfil the SD constraint. Since the signals are complex, the valid solutions lie inside a circle, (and not just in an interval in 1D, as it was the case for real signals). Therefore, the solutions are found by using a circle search in the complex plane. In the case of a GSM signal, the task is to find potential intersections with the real axis, since the symbols are real-valued. In the case of an EDGE signal the intersections that should be examined will be between two circles (where the one, which represents the symbols in the IQ-space, will be the unit circle). From (5.2.23) it is seen that the solutions that fulfil the SD constraint lie inside the circle with the parameters

$$\begin{aligned}&Radius,\ \widetilde{R} = \sqrt{\frac{1}{\|\mathbf{h}\|^2}\left(\widetilde{r}_n^{\,2} - \sum_{i=2}^{N_R-1}\left|\mathbf{q}_i^H \widetilde{\mathbf{x}}_n\right|^2\right)} \\ &Centre,\ \widetilde{C}\left(\Re\left\{\frac{\hat{\mathbf{h}}^H \widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|}\right\}; \Im\left\{\frac{\hat{\mathbf{h}}^H \widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|}\right\}\right).\end{aligned} \tag{5.2.24}$$

The symbols satisfying the SD constraint are predicted based on the circle in (5.2.24). For an EDGE signal this is done by examining the angles of the intersections of the circles. In Appendix 5.1 "Boundaries for Symbols that Fulfil the SD Constraint" the prediction of the symbols are described in more details.

Prediction of GSM symbols

In the case of a GSM signal there can be three scenarios.

1. None of the symbols fulfil the SD constraint. This will be the case when the circle has no intersection with the real axis, or when the interval that contains the intersections does not include either ±1.
2. One of the symbols fulfils the SD constraint. There are normally two intersections between the circle and the real axis and the interval contain either -1 OR +1. However, the circle can in theory intersect the axis in either ±1 and in such a case the symbol will also fulfil the SD constraint.
3. Both symbols fulfil the SD constraint. There are two intersections between the circle and the real axis and the interval contains both -1 AND +1.

Prediction of EDGE symbols

In the case of an EDGE signal there are basically also three scenarios.

1. None of the symbols fulfil the SD constraint. There is no intersection between the two circles and the circle in (5.2.24) does not contain the unit circle. Alternatively, the intersections between the two circles will be in an interval where there are no EDGE symbols. The latter is illustrated in Figure 5.8.
2. One or several of the symbols (but not all symbols) fulfil the SD constraint. In this case there will often be two intersections between the circles. This case can be seen in Figure 5.9. Theoretically, there can be one intersection between the circles, but this will rarely be the case in practice. In such a case the intersection should be exactly in one of the EDGE symbols to fulfil the SD constraint.
3. All symbols fulfil the SD constraint. In this case there will often be any intersection between the two circles. However, in a special case there can be one intersection between the circles, but the circle in (5.2.24) does still contain the unit circle. This is illustrated in Figure 5.10.

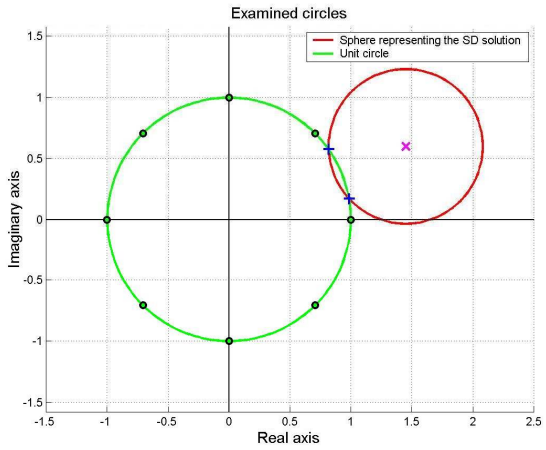

Figure 5.8. None of the EDGE symbols fulfil the SD constraint. The blue markers show the calculated intersections between the two circles.

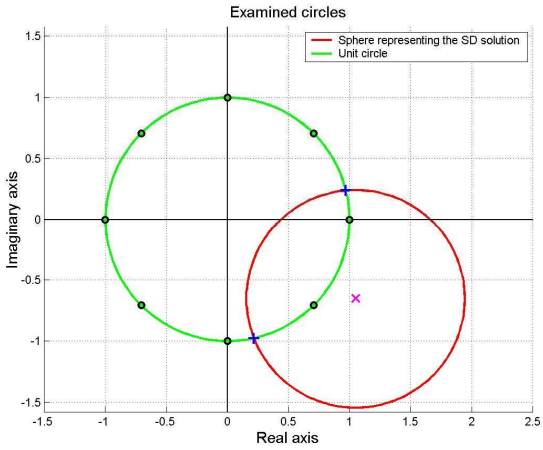

Figure 5.9. Two EDGE symbols fulfil the SD constraint.

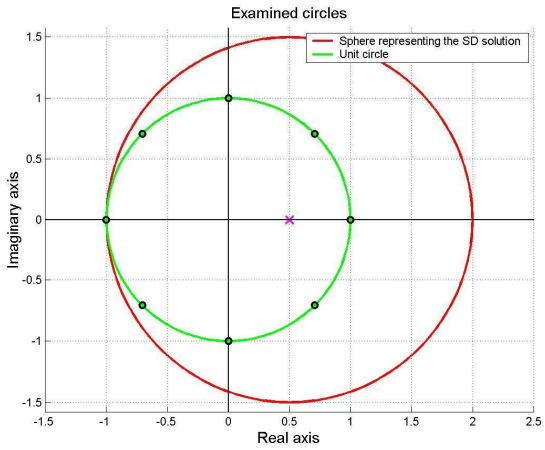

Figure 5.10. All the EDGE symbols fulfil the SD constraint.

5.3 Choice of Radius

One of the essential issues in sphere decoding is how to choose the radius, since the complexity depends on this. As mentioned earlier a too large radius will lead to too many points and the complexity of the algorithm will still be nearly exponential to the number of dimensions. This is due to the fact that almost all states will fulfil the SD constraint when the radius is too large. On the other hand, a radius that is too small will cause the SD algorithm to prune all the states away and no solution will be found.

5.3.1 Static Radius

As described in (5.1.17) in Section 5.1 "Basic Concept of SD", it is assumed that the received signal is an integer point in the multi-dimensional space, which has been disturbed by additional white Gaussian noise. Equation (5.1.17) is repeated here

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{z},$$

where $\mathbf{z} \in \mathrm{CN}\left(\mathbf{0}, \sigma_c^2 \mathbf{I}\right)$. When this assumption has been made, it is possible to derive the statistical estimate of the probability of finding a point that lies inside the hyper-sphere, as illustrated in [18]. Observe that

$$\begin{aligned} \left\|\mathbf{x} - \mathbf{H}\hat{\mathbf{s}}\right\|^2 &= \left\|\mathbf{H}\left(\mathbf{s} - \hat{\mathbf{s}}\right) + \mathbf{z}\right\|^2 \\ &\le \left\|\mathbf{H}\left(\mathbf{s} - \hat{\mathbf{s}}\right)\right\|^2 + \left\|\mathbf{z}\right\|^2. \end{aligned} \tag{5.3.1}$$

where $\hat{\mathbf{s}}$ is the estimated symbol vector. If the ML solution in fact is the correct solution, i.e. $\hat{\mathbf{s}} = \mathbf{s}$, (5.3.1) becomes

$$\left\|\mathbf{x} - \mathbf{H}\hat{\mathbf{s}}\right\|^2 = \left\|\mathbf{z}\right\|^2, \tag{5.3.2}$$

which will be a good approximation at high SNR. It is now possible to choose the radius based on the statistic of the noise variance, since (5.3.2) is a χ^2 random variable with $2 \cdot N_{BL} \cdot N_R$ degrees of freedom. The reason why there is a factor of two is because the signals are complex. In Appendix

5.2 "Derivation of the Chi-square Distribution" it is shown that the χ^2- distribution will have the probability density function (PDF)

$$p(\lambda) = \frac{1}{\left(2\sigma^2\right)^{(\nu/2)}\Gamma\left(\nu/2\right)} \cdot (\lambda)^{(\nu/2-1)} \cdot \exp\left(-\frac{\lambda}{2\sigma^2}\right), \tag{5.3.3}$$

where σ^2 is the variance of the noise per dimension and ν is the degrees of freedom. Thus in [18] it is shown that, the radius can be chosen statistically as

$$1-\varepsilon = \int_0^{r^2} \frac{1}{\left(2\sigma^2\right)^{(\nu/2)}\Gamma\left(\nu/2\right)} \cdot (\lambda)^{(\nu/2-1)} \cdot \exp\left(-\frac{\lambda}{2\sigma^2}\right) d\lambda, \quad \textit{where} \quad 0 \le \varepsilon \le 1. \tag{5.3.4}$$

Here $1-\varepsilon$ is the probability of having a point inside the sphere. For example if $1-\varepsilon = 0.99$ is chosen, the probability of having minimum one point inside the sphere is 0.99. An important observation is that the radius is chosen solely based on the noise term and not on the **H** matrix. (Recall from the discussion on the covering radius in Section 5.1 that the problem of choosing the radius based on **H** easily becomes exponential in complexity). It can, therefore, be stated that the complexity of SD depends on data, in contrast to the unconstrained LS-problem, which depends on the dimension of **H**, see e.g. [18].

Procedure for Updating the Radius
When a search with the radius based on ε does not have any solution, (i.e. lattice points that fulfil the SD constraint), the radius must be increased and the search must be restarted. In [18] it is proposed that the increase of the radius is done by picking the new radius based on ε^2. If this search does not give a solution either, the radius is determined based on ε^4 and so forth.

With the updating procedure described above, in [18 and 19] it is proved that the expected complexity will be polynomial (and often roughly cubic over a wide range of SNRs). However, when $\sigma^2 \to \infty$ the received point, **x**, becomes arbitrary and as mentioned earlier the expected complexity will then be exponential. For a theoretical proof of this, the interested reader can consult [18].

Implementation Issues
If there is enough memory available in the application that exploits the SD algorithm, and the algorithm is restarted at time index $n > 1$, it is recommended that the cost function of the states that fulfil the SD constraint (from time index 1 to n-1) are stored. The states that already have fulfilled the SD constraint in the previous search radius will also fulfil the SD constraint in the new search and, therefore, they do not have to be evaluated. Though it requires some bookkeeping in the implementation, it will reduce the computational complexity.

As described above the radius is calculated based on (5.3.4). In an implementation (in MATLAB) it is recommended to use the Inverse Chi-square function to calculate this integral. In Appendix 5.3 "The Inverse Chi Square Distribution" a more detailed description is given.

5.3.2 Increasing Radii Based on the Degrees of Freedom

It turns out that the static radius approach described in the subsection above often will be very loose in the beginning of a search when the dimension of the received point, **x**, is large. This will be the

case for a GSM or EDGE burst. Here the degrees of freedom will vary from $2 \cdot N_R$ in the beginning of a burst to $2 \cdot N_R \cdot N_{BL} = 2 \cdot N_R \cdot 148$ (tail bits and TS are here included). A plot of the squared radius as a function of the degrees of freedom is given in Figure 5.11.

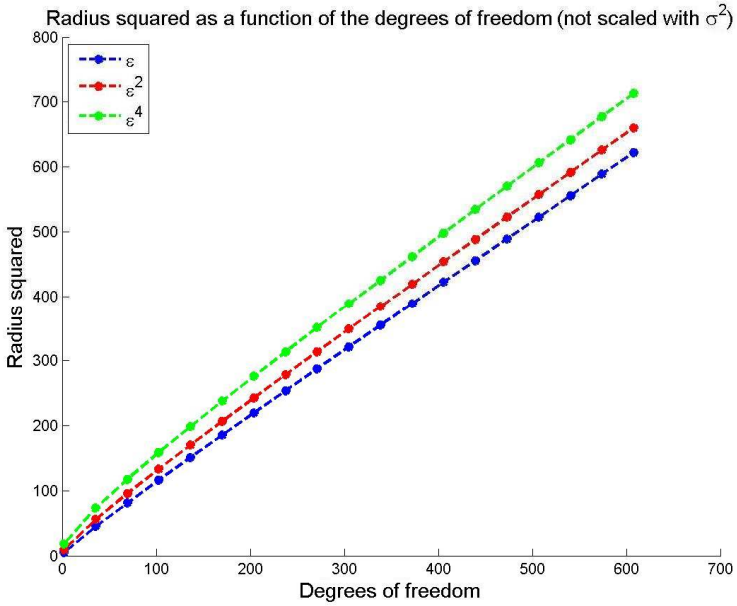

Figure 5.11. Squared radius as a function of the degrees of freedom, $\varepsilon = 0.1$.

From Figure 5.11 it is clear that Degrees of Freedom (DF) has big influence on the size of the radius. Therefore, it is likely that there is no pruning of the state space in the beginning of a burst if only one radius is used, and the complexity will then be exponential in the beginning.

An alternative approach is to let the radius increase as a function of the DF (i.e. as a function of the time index). Hence, the size of the radii will be increased as shown in Figure 5.11. That way there will be a reasonable coherence between the size of the radius and the DF for the cost function. It should though be stated that a procedure that relies on increasing the radius does not perform exact MLSE decoding. This is because the asymmetry in the search region can lead to situations where the closest lattice point to **x** does not lie in this region, [22]. However, in [22] it is likewise stated that the approach can give performance as close to MLSE as desired through the choice of parameters, (i.e. ε).

5.3.3 Increasing Radii Search

From a statistical point of view, the method of choosing the radii as a function of DF, can be improved. For example the search region (i.e. radii) from the previous dimensions should be taken into account when the radius of the current dimension is chosen. This is exactly what is done in [22], where an approach called Increasing Radii Search is described. In this subsection this method will be explained in more detail.
To ease the derivation a new vector is introduced

$$\mathbf{y} = \mathbf{x} - \mathbf{H}\mathbf{s}\,. \tag{5.3.5}$$

Let $\mathbf{y}_i$ be a vector of size N_R which contains the rows from $N_R \cdot (i-1)+1$ to $N_R \cdot i$ in vector $\mathbf{y}$, and let λ_i be given as

$$\lambda_i = \left\| \mathbf{y}_i \right\|^2, \quad where \quad i \in \left[1, \ldots, N_{BL} + L - 1\right]. \tag{5.3.6}$$

Furthermore, let $W = N_{BL} + L - 1$. With this notation the SD constraint with the static radius can be expressed as

$$\lambda_1 + \lambda_2 + \ldots + \lambda_W \le r^2 . \tag{5.3.7}$$

The Increasing Radii Search instead have to fulfil

$$\begin{aligned} \lambda_1 &\le r_1^2 \\ \lambda_1 + \lambda_2 &\le r_2^2 \\ &\vdots \\ \lambda_1 + \lambda_2 + \ldots + \lambda_W &\le r_W^2 . \end{aligned} \tag{5.3.8}$$

Recall that each λ_i can be described by the χ^2-distribution with the PDF

$$p(\lambda_i) = \frac{1}{\left(2\sigma^2\right)^{(\nu/2)}\Gamma\left(\nu/2\right)} \cdot (\lambda_i)^{(\nu/2-1)} \cdot \exp\left(-\frac{1}{2}\frac{\lambda_i}{\sigma^2}\right), \tag{5.3.9}$$

where ν is the degrees of freedom and σ^2 is the variance of the noise per dimension. Since the signal is complex the "real" degrees of freedom will be $\nu = 2 \cdot N_R$. Then (5.3.9) becomes

$$p(\lambda_i) = \frac{1}{\left(2\sigma^2\right)^{N_R}\Gamma(N_R)} \cdot (\lambda_i)^{(N_R-1)} \cdot \exp\left(-\frac{\lambda_i}{2\sigma^2}\right) \tag{5.3.10}$$

The random variables λ_i's are assumed to be independent and, therefore, the joint PDF can be expressed as

$$\begin{aligned} p(\lambda_1, \lambda_2, \ldots, \lambda_W) &= p(\lambda_1) \cdot p(\lambda_2) \cdot \ldots \cdot p(\lambda_W) \\ &= \left(\frac{1}{\left(2\sigma^2\right)^{N_R}\Gamma(N_R)}\right)^W \cdot (\lambda_1 \cdot \lambda_2 \cdot \ldots \cdot \lambda_W)^{(N_R-1)} \cdot e^{-\frac{1}{2\sigma^2}(\lambda_1+\lambda_2+\ldots+\lambda_W)} . \end{aligned} \tag{5.3.11}$$

The probability of finding minimum one solution, i.e. a set of λ_1, λ_2, … , λ_W that satisfies (5.3.8) can be found. This is done by integrating the joint PDF in the search interval, i.e. over the chosen radii $r_1^2, r_2^2, \ldots, r_W^2$. The probability of finding a solution, 1-ε, in a search interval is given as

$$1 - \varepsilon = \int_0^{r_1^2} \int_0^{r_2^2-\lambda_1} \cdots \int_0^{r_W^2-(\lambda_1+\lambda_2+\ldots+\lambda_{W-1})} p(\lambda_1, \lambda_2, \ldots, \lambda_W)\, d\lambda_W \cdots d\lambda_2 d\lambda_1 , \tag{5.3.12}$$

where ε is still defined to be close to zero. Hence, it is possible to find a set of radii based on the statistic of the noise and the probability of finding a solution.

When the receive dimension is N_R = 1

For simplicity it is first assumed that the receive dimension is, N_R = 1. Hence (5.3.11) is reduced to

$$p(\lambda_1, \lambda_2, \ldots, \lambda_W) = \left(\frac{1}{2\sigma^2}\right)^W \cdot e^{-\frac{1}{2\sigma^2}(\lambda_1+\lambda_2+\ldots+\lambda_W)} . \tag{5.3.13}$$

The expression for the PDF is inserted in (5.3.12), which gives

$$1-\varepsilon=\int_0^{r_1^2}\int_0^{r_2^2-\lambda_1}\cdots\int_0^{r_W^2-(\lambda_1+\lambda_2+\ldots+\lambda_{W-1})}\left(\frac{1}{2\sigma^2}\right)^W\cdot e^{-\frac{1}{2\sigma^2}(\lambda_1+\lambda_2+\ldots+\lambda_W)}\,d\lambda_W\cdots d\lambda_2 d\lambda_1. \tag{5.3.14}$$

By introducing a new variable,

$$\psi_i=\sum_{j=1}^{i}\lambda_j, \tag{5.3.15}$$

it is possible to simplify expression (5.3.14) to

$$1-\varepsilon=\int_0^{r_1^2}\int_{\psi_1}^{r_2^2}\cdots\int_{\psi_{W-1}}^{r_W^2}\left(\frac{1}{2\sigma^2}\right)^W\cdot e^{-\frac{1}{2\sigma^2}\psi_W}\,d\psi_W\cdots d\psi_2 d\psi_1. \tag{5.3.16}$$

This expression can be further simplified by defining the variables

$$\tau_i=\frac{1}{2\sigma^2}\psi_i\Rightarrow d\psi_i=2\sigma^2 d\tau_i \quad and \quad \tilde{r}_i^2=\frac{1}{2\sigma^2}r_i^2.$$

Equation (5.3.16) can then be expressed as

$$1-\varepsilon=\int_0^{\tilde{r}_1^2}\int_{\tau_1}^{\tilde{r}_2^2}\cdots\int_{\tau_{W-1}}^{\tilde{r}_W^2}e^{-\tau_W}\,d\tau_W\cdots d\tau_2 d\tau_1, \tag{5.3.17}$$

It is now quite simple to integrate out the integration variable $d\tau_W$, which will lead to the following result

$$1-\varepsilon=\int_0^{\tilde{r}_1^2}\int_{\tau_1}^{\tilde{r}_2^2}\cdots\int_{\tau_{W-2}}^{\tilde{r}_{W-1}^2}\left[-e^{-r_W^2}+e^{-\tau_{W-1}}\right]d\tau_{W-1}\cdots d\tau_2 d\tau_1 \tag{5.3.18}$$

From (5.3.18) it can be observed that the first term in the integral is a constant and the last term corresponds to the new integration variable. This integration variable will have the same form as in (5.3.17). Thus, there is a recursive structure in the integral where the number of terms will increase when each of the integration variables are integrated out. Because of this recursive structure in the integral, it is possible to compute it in a recursive manner. In [22, p. 16] it is shown that the integral, I_W, in (5.3.18) can be found by

$$1-\varepsilon=I_W=I_{W-1}-e^{-\tilde{r}_W^2}\tilde{J}_{W-1}, \tag{5.3.19}$$

where

$$\tilde{J}_{W-1}=\int_0^{\tilde{r}_1^2}\int_{\tau_1}^{\tilde{r}_2^2}\cdots\int_{\tau_{W-2}}^{\tilde{r}_{W-1}^2}d\tau_{W-1}\cdots d\tau_2 d\tau_1. \tag{5.3.20}$$

The recursive way of calculating the integral in (5.3.20) is given in [22, p. 17] as

$$\tilde{J}_k=\sum_{l=0}^{k-1}(-1)^{k-l+1}\frac{\tilde{r}_{l+1}^{2(k-l)}}{(k-l)!}\tilde{J}_l \tag{5.3.21}$$

By the definition of $\tilde{J}_k$ the value of $\tilde{J}_0$ can be determined to be

$$\widetilde{J}_1 = \int_0^{\widetilde{r}_1^2} 1\, d\tau_1 = \widetilde{r}_1^{\,2} = (-1)^{1-0+1} \frac{\widetilde{r}_1^{\,2(1-0)}}{(1-0)!}\widetilde{J}_0 = \widetilde{r}_1^{\,2}\widetilde{J}_0 \Rightarrow$$
$$\widetilde{J}_0 = 1. \tag{5.3.22}$$

Recall that $\widetilde{r}_i^{\,2} = \frac{1}{2\sigma^2} r_i^2$, and by substitution of this into (5.3.21) the expression becomes

$$\widetilde{J}_k = \sum_{l}^{k-1} (-1)^{k-l+1} \frac{\left(\frac{1}{2\sigma^2} r_{l+1}^2\right)^{(k-l)}}{(k-l)!}\widetilde{J}_l, \quad where \quad \widetilde{J}_0 = 1. \tag{5.3.23}$$

By defining $I_0 = 1$ and using (5.3.19) the integral in (5.3.17) becomes

$$I_W = 1 - \sum_{k=1}^{W} e^{-\frac{1}{2\sigma^2} r_k^2} \widetilde{J}_{k-1}. \tag{5.3.24}$$

By combining (5.3.19) and (5.3.24) the equation below is obtained

$$\varepsilon = \sum_{k=1}^{W} e^{-\frac{1}{2\sigma^2} r_k^2} \widetilde{J}_{k-1}. \tag{5.3.25}$$

Hence, based on a specific value of ε the radii can be determined by using numerical method (e.g. bisection or Newton-Raphson's method).

When the receive dimensions are $N_R = 2$
When the receive dimensions are larger than one, (e.g. $N_R = 2$) it becomes much more complicated to find an analytical solution of (5.3.12). The PDF in (5.3.11) in the case of $N_R = 2$ now becomes

$$p(\lambda_1, \lambda_2, \ldots, \lambda_W) = \left(\frac{1}{(2\sigma^2)^2 \Gamma(2)}\right)^W \cdot (\lambda_1 \cdot \lambda_2 \cdot \ldots \cdot \lambda_W) \cdot e^{-\frac{1}{2\sigma^2}(\lambda_1 + \lambda_2 + \ldots + \lambda_W)}. \tag{5.3.26}$$

The integral that has to be solved is now

$$1 - \varepsilon = \left(\frac{1}{(2\sigma^2)^2 \Gamma(2)}\right)^W \int_0^{r_1^2} \int_{\lambda_1}^{r_2^2} \cdots \int_{(\lambda_1 + \lambda_2 + \ldots + \lambda_{W-1})}^{r_W^2} (\lambda_1 \cdot \lambda_2 \cdot \ldots \cdot \lambda_W) \cdot e^{-\frac{1}{2\sigma^2}(\lambda_1 + \lambda_2 + \ldots + \lambda_W)} d\lambda_W \cdots d\lambda_2 d\lambda_1. \tag{5.3.27}$$

When the integral was evaluated for the receive dimension $N_R = 1$, the PDF had the convenient property of only consisting of an exponential function in the multi dimensional space. This is not the case for the integral in (5.3.27), since the term $(\lambda_1 \cdot \lambda_2 \cdot \ldots \cdot \lambda_W)$ also appears.
In [32, Eq. (17.25.4)] it is shown that

$$\int x^n e^{ax} dx = \frac{e^{ax}}{a}\left(x^n - \frac{n x^{n-1}}{a} + \frac{n(n-1)x^{n-2}}{a^2} - \ldots \frac{(-1)^n n!}{a^n}\right), \quad if\ n = postive\ integer, \tag{5.3.28}$$

which in the multi dimensional space will make it quite difficult to find the corresponding $\widetilde{J}_k$ - function. Even though from an analytical point of view it would be very interesting to solve this integral, it has been assessed that the gain of solving (5.3.27) from a practical point of view would be limited. This is due to the fact that numerical problems prevent the method for being

applied in GSM and EDGE bursts. Besides, it is possible to find a reasonable good estimate of the radii by using the receive dimension $N_R = 1$ and instead setting the number of radii to $2W$. Then every second radius can be picked out. Hence an estimate of the W radii with the receive dimensions $N_R = 2$ is obtained. As a final remark it should be noted that there will not be a problem for $N_R \geq 2$ when a QR factorization of the matrix is performed, since one dimension is examined at a time. This will be describe in more detail in Section 5.6.

Implementation

In a GSM or EDGE system the number of radii that should be estimated in the Increasing Radii Search (IRS) approach is equivalent to the burst length, i.e. 148 radii should be found if the tail bits and the TS are also taken into account. Hence, it becomes quite difficult to solve (5.3.25) even numerically. Therefore, it is recommended to use a slightly different approach. Instead of keeping ε fixed for the whole burst, it can be increased as a function of the time index in the burst. Let ε_i's be defined as

$$\varepsilon_i = \sum_{k=1}^{i} e^{-\frac{1}{2\sigma^2} r_k^2} \widetilde{J}_{k-1} \tag{5.3.29}$$

Then r_1 can be found based on ε_1 as

$$r_1^2 = -2\sigma^2 \ln\left(\frac{\varepsilon_1}{\widetilde{J}_0}\right) = 2\sigma^2\left[\ln\left(\widetilde{J}_0\right) - \ln\left(\varepsilon_1\right)\right]. \tag{5.3.30}$$

Based on r_1 and ε_2, the value of r_2 can be found and so forth. The general expression calculation of the radii then becomes

$$r_i^2 = 2\sigma^2\left[\ln\left(\widetilde{J}_{i-1}\right) - \ln\left(\varepsilon_i - \varepsilon_{i-1}\right)\right], \quad where \quad i = 1,\ldots,W. \tag{5.3.31}$$

Experiments have been performed with three ways of increasing ε_i to see how the choice of ε affects the size of the radius. The three methods are shown in Figure 5.12 below.

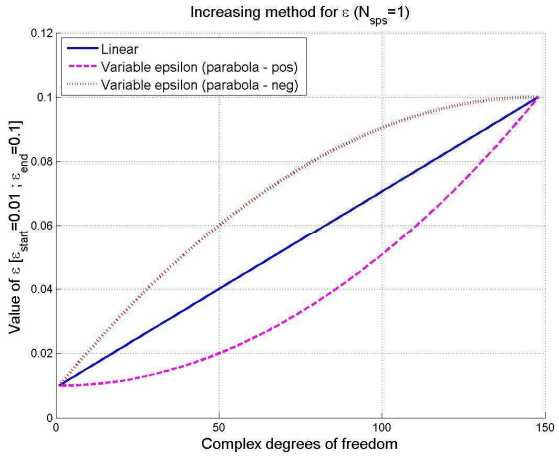

Figure 5.12. Illustration of different approaches of increasing ε_i (ε_1 = 0.01 and ε_{end} = 0.1).

In Figure 5.12 the blue curve represents a linear increase of ε_i. Hence,

$$\varepsilon_i = \alpha i + \beta, \quad where \quad i = 1,\ldots,W\,, \tag{5.3.32}$$

where the parameters α and β are determined by specifying the start and the end value of ε_i. The magenta and red curves are parabolas with respectively positive and negative coefficients of a, where a is the parameter in the equation for a parabola given as

$$\varepsilon_i = ai^2 + bi + c, \quad where \quad i = 1,\ldots,W \,. \tag{5.3.33}$$

To calculate these parameters for the negative parabola it is, furthermore, assumed that parabola has its maximum for ε_W. Since the global extremum is found in the coordinate

$$Global\ extremum\ in: \left\{ i;\ \ \varepsilon_i \right\} = \left\{ \frac{-b}{2a} \ ;\ \ \frac{-\left(b^2\right)+4ac}{4a} \right\}, \tag{5.3.34}$$

the parameters can be found. For the positive parabola it is likewise assumed that it has the minima for the start value.

Numerical problems

From (5.3.23) it can be seen that the numerical value of $\widetilde{J}_k$ will rapidly increase since it among others contains the squared radius raised to the power of (k–l). Additionally, it is observed that the sign changes each time the variable l is incremented in the summation. Hence, to avoid numerical problems it is recommended to summarize the positive and negative terms separately and finally do the subtraction. Furthermore, it is advised to perform the calculation of $\widetilde{J}_k$ in the logarithmic domain so the large numerical values of $\widetilde{J}_k$ can be handled.

Even though the implementation is done in the logarithmic domain and the positive and negative terms are computed separately, the computation of $\widetilde{J}_k$ can become numerically unstable. As shown in Figure 5.13 and Figure 5.14, this will be the case when the complex degrees of freedom become larger than approximately 55.

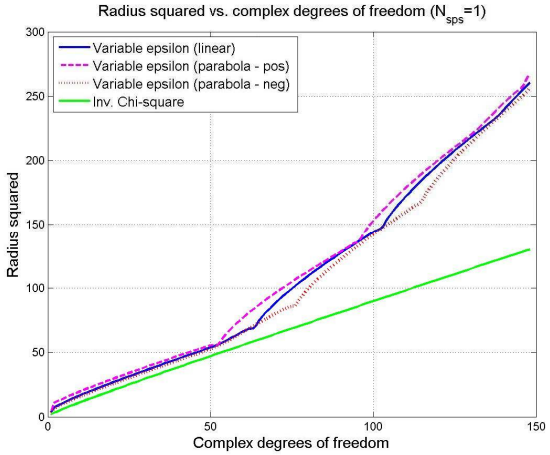

Figure 5.13. Squared radius as a function of the complex degrees of freedom for the IRS approaches. ε_1= 0.01, ε_{end} = 0.1, and SNR = 1dB.

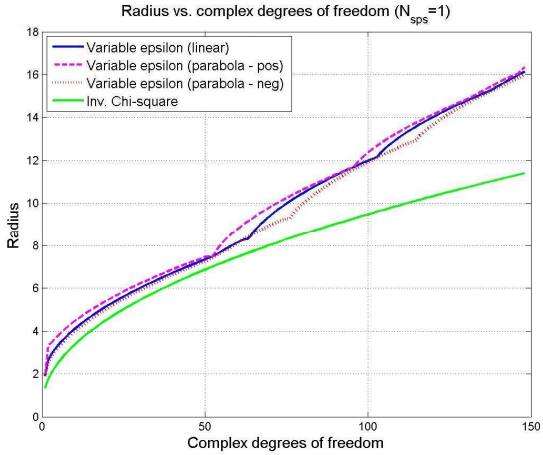

Figure 5.14. Radius as a function of the complex DF for the IRS approaches. ε_1 = 0.01, ε_{end} = 0.1, and SNR = 1dB.

In Figure 5.13 and Figure 5.14 it is seen that the three methods using increasing ε_i's give radii which are quite close each other. Besides, it is observed that, when the complex degrees of freedom is below 55, the three methods give radii, which are comparable with the one determined by the inverse Chi-square distribution. However, as the degrees of freedom is increased further the radii of the methods using increasing ε_i's grow more rapidly than the method using the inverse Chi-square

distribution. Thus, when the degrees of freedom is larger than approximately 55 the squared radius of the methods using ε_i's will give incorrect radii. This in undesirable since a too large radius will prevent the SD algorithm to prune the state space. In Figure 5.15 and Figure 5.16 a zoom of the area where the radii start to grow rapidly is given.

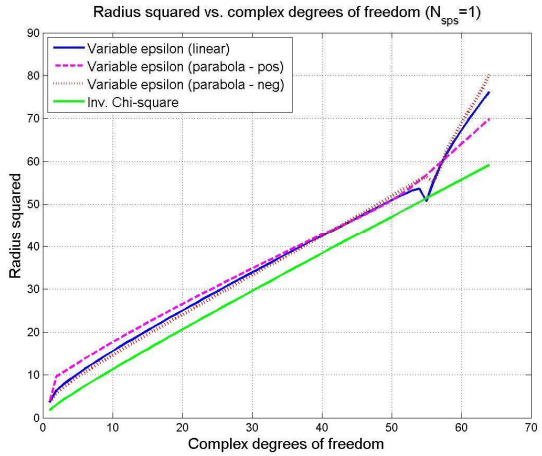

Figure 5.15. Squared radius as a function of the complex DF for the IRS approaches. $\varepsilon_1 = 0.01$, $\varepsilon_{end} = 0.1$, and SNR = 1dB.

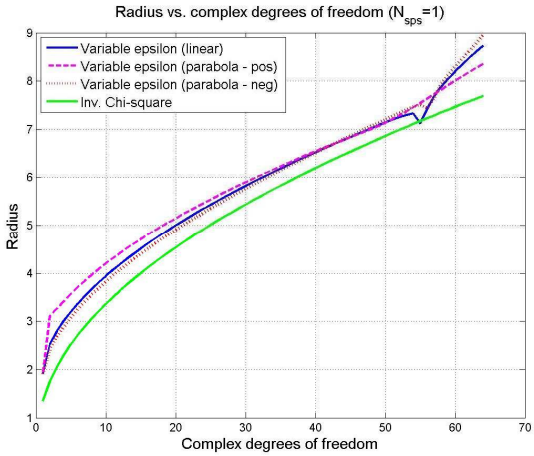

Figure 5.16. Radius as a function of the complex DF for the IRS approaches. $\varepsilon_1 = 0.01$, $\varepsilon_{end} = 0.1$, and SNR = 1dB.

To explain why the method using ε_i's becomes numerically unstable, it is necessary to look at the value of $\widetilde{J}_k$ and the positive and negative terms which are used to calculate it. In Figure 5.17 the value of $\widetilde{J}_k$ is plotted, while the difference between the positive and negative (logarithmic) terms is plotted in Figure 5.18 and Figure 5.19.

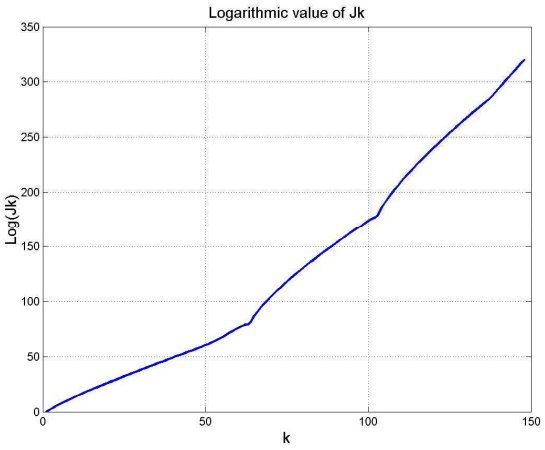

Figure 5.17. Logarithmic value of $\widetilde{J}_k$ as a function of *k*.

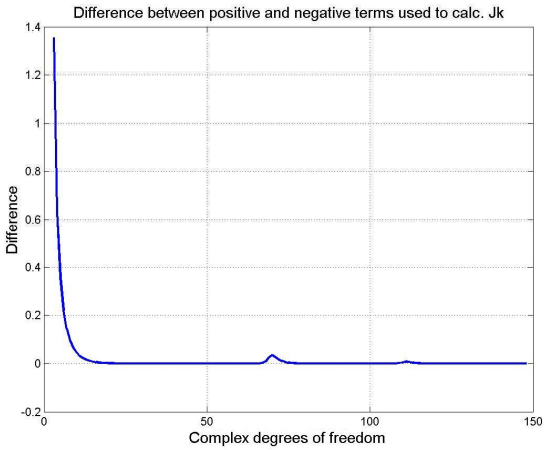

Figure 5.18. Difference between positive and negative terms used to calculate $\widetilde{J}_k$.

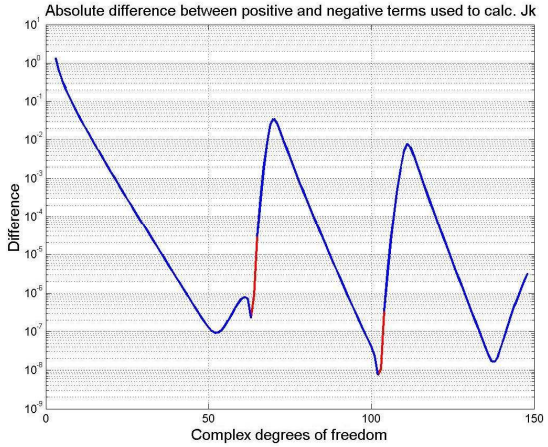

Figure 5.19. Absolute difference between positive and negative terms which are used to calculate $\widetilde{J}_k$ (logarithmic scale).

It is seen that the function $\widetilde{J}_k$ basically has two curvatures (at approximately DF=63 and DF=103). From Figure 5.19 it is seen that the absolute difference between the positive and negative terms becomes less than 10^{-7} when the complex degrees of freedom is larger than 50. When the degrees of freedom is increased further a ripple effect can be observed, which indicates numerical problems.
The red part of the curve illustrates values where the negative terms become larger than the positive terms, which is also an indication of numerical problems. By comparing Figure 5.17 and Figure 5.19 it is seen that the two curvatures occur exactly when the negative terms become larger than the positive terms. To overcome the numerical problems it is necessary to use a precision that is better than the default precision that MATLAB relies on[1].
Since a GSM or EDGE burst consist of 148 symbols, which lead to 148 complex degrees of freedom in the end of the burst, the method described above cannot be used to calculate the radii of the spheres, unless a better precision is used. However, an important observation is that the method using an inverse Chi-square distribution leads to radii that are quite close to the methods using ε_i's, which are the optimal ones viewed from a statistically point of view. Thus, it is possible to use the inverse Chi-square distribution instead.

5.4 Search Strategies

In this section two types of search strategies are described. Furthermore, a modification in one of them is proposed, which can reduce the complexity.

5.4.1 Finke-Pohst

This search strategy is the traditional way of searching in sphere decoding, and it is this strategy that [18] uses to determine the expected complexity of SD. As described in (5.1.16) the interval of the symbols at dimension *k* is found by

[1] MATLAB uses per default the IEEE 754 floating point standard. The IEEE double precision floating-point format is a 64-bit word. However, to the writer of this thesis knowledge, MATLAB does contain a special toolbox, which allow the user to define one's own floating point class. This might result in the numerical problems first occur later (so the radii in a GSM/EDGE burst can be found), but has not been investigated further.

$$\left\lceil \frac{-\breve{r}_k + \breve{x}_{k|k+1}}{R_{kk}} \right\rceil \leq s_k \leq \left\lfloor \frac{\breve{r}_k + \breve{x}_{k|k+1}}{R_{kk}} \right\rfloor . \tag{5.4.1}$$

The Finke-Pohst strategy then starts at the lower bound of the interval for s_M and increases s_M until the upper bound is reached. In the trellis structure it corresponds to examine the states (that have not been pruned away) in a random order. This could e.g. be done by picking out the valid states based on their state number in a numerically increasing or decreasing order. This search strategy is illustrated in Figure 5.20.

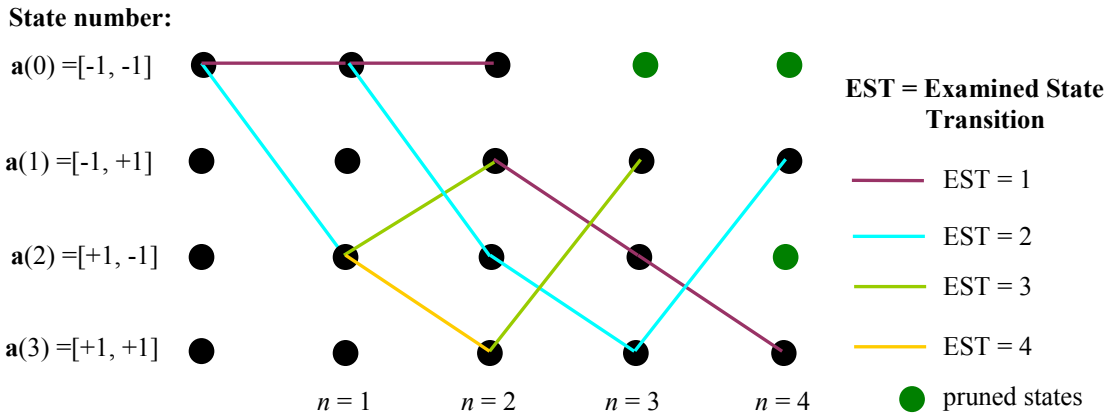

Figure 5.20. Example of Finke-Pohst search strategy in the trellis diagram when the valid states are picked out in a numerically increasing order based on their state number. (EST stands for Examined State Transition, i.e. the order in which the state transitions are examined).

It should be noted that the states, which are pruned away during the search (based on the bounding of the symbols described in Section 5.2.2) are not illustrated in Figure 5.20.

5.4.2 Schnorr-Euchner

An alternative search strategy proposed in [24] is to examine the point in the middle of the interval first. Hence, the search in dimension k is started with the point

$$s_k = \left\lceil \frac{\breve{x}_{k|k+1}}{R_{kk}} \right\rfloor . \tag{5.4.2}$$

Then the next closest point to the middle is examined, (which would either be the lattice point on the right or left of (5.4.2) in the case of a real-valued signal).
In the trellis structure this corresponds to examining the value of the cost function of all the states in the following way. The state with the lowest cost function is examined first. Then the state with the second lowest cost function is chosen and so forth. Besides, the state transition for a given state, which is closest to the received point, will be examined first. The state transition that is closes to the received point is determined based on the bounding of the symbols (described in Section 5.2.2). The Schnorr-Euchner search strategy is illustrated in Figure 5.21.

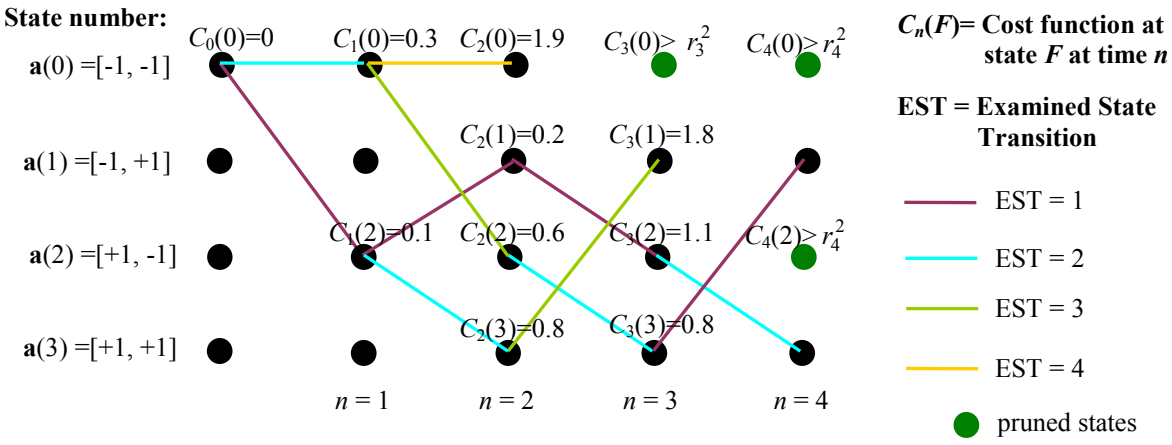

Figure 5.21. Example of Schnorr-Euchner search strategy in the trellis. $C_n(F)$ denotes the cost function in state number F at time index n.

From Figure 5.21 it is seen that the search through the trellis based on the Schnorr-Euchner approach will be different from the Finke-Pohst. However, when no additional constraint is added to the two search strategies this will lead to the same result in the end. Furthermore, the complexity in this case will be exactly the same. On the other hand when an additional constraint concerning the maximum number of allowed state transitions is added, there will be a difference in complexity.

Modification of the Schnorr-Euchner Search Strategy

To reduce the complexity of the SD algorithm, which in worst-case is exponential, an extra constraint can be added to the algorithm. Because of the Schnorr-Euchner search strategy, it is possible to examine only the most likely paths through the trellis diagram.

This will be a great advantage in an implementation of the algorithm in a real application (i.e. a mobile phone). This is due to the rigorous requirement on the peak complexity of the algorithm. The constraint will, however, imply that the algorithm is not guaranteed to find the MLSE solution. The fewer the maximum number of allowable states is set to, the larger the probability of not finding the MLSE solution will be. An illustration of the Schnorr-Euchner search strategy is shown in Figure 5.22, when the maximum number of allowable states is one.

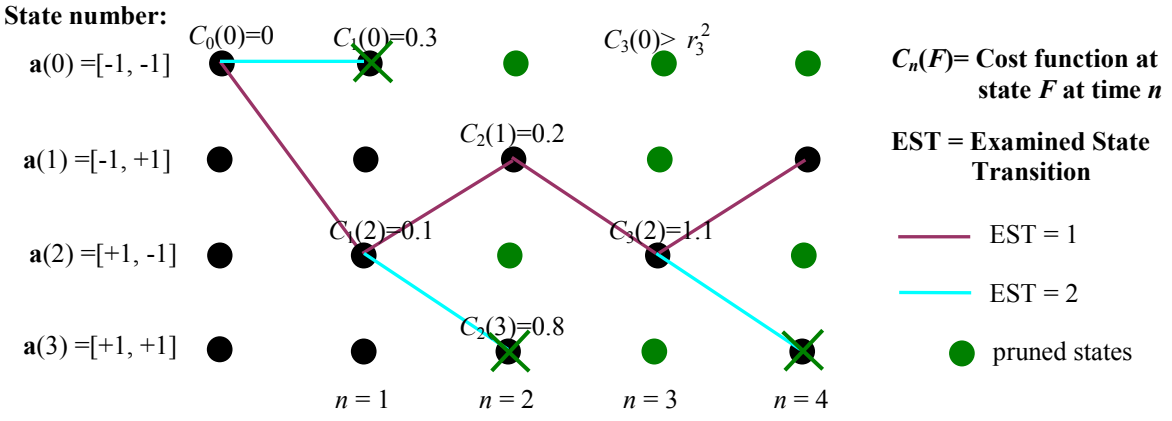

Figure 5.22. Example of Schnorr-Euchner search strategy in the trellis. The maximum number of allowable states is set to one in this example.

From Figure 5.22 it is seen that the number of states, which are pruned away are larger compared with Figure 5.21. It should be stated that Figure 5.22 only serves as an illustration of the principle, since it is a very extreme case to keep only one state in the trellis diagram.
Due to the prediction of which state transitions that fulfil the SD constraint, it will be possible to determine the most likely state transitions. Hence, it is possible to specify the maximum number of allowable state transitions from a given state. This will also reduce the complexity.

5.5 Using the LMMSE Solution in Combination with SD

As mentioned earlier (see e.g. Section 5.3.1) the complexity of the SD algorithm is roughly cubic for high SNR. It is precisely at high SNRs that the difference in BER performance between e.g. the LMMSE receiver and the MAP receiver is significant. Hence, at high SNRs the gain of using the MAP receiver with SD constraint will be great. At low SNRs the complexity of the SD algorithm is exponential and the difference in BER performance of the two receivers mentioned above will not be as significant. Thus, receivers exploiting the sphere decoding algorithm complements the linear receivers in a quite good way. Under low SNRs the linear detector can be used. When the SNR increases, the MAP receiver (or alternatively the Viterbi algorithm) employing the SD algorithm can be used, which will then give a considerable gain in performance. Therefore, in some applications it is a good idea to combine linear detectors with SD.
An approach is to first obtain the LMMSE solution and then try to find the MLSE solution using the SD algorithm. If it turns out that the complexity of the MLSE solution is too high, the receiver should instead output the LMMSE solution. On the other hand, if the MLSE solution is found within reasonable complexity, this solution is given as the output.

Another idea is to use the LMMSE solution to gain some a priori knowledge of the radius of the sphere. This is because the radius of the MLSE solution per definition must be upper bounded by the LMMSE solution, since it finds the closest lattice point to the received point **x**. The upper bound of the radius will especially at poor SNR levels be useful, since the radius from a statistical point of view has to be large, (recall that the radius is scaled based on the noise variance). It could though be argued that the method becomes unnecessary when the modified Schnorr-Euchner search strategy with a maximum number of allowed states is used. Therefore, the method should rather be considered as a supplement to algorithms, which perform exact MLSE detection.

5.6 MUD using SD

When the number of transmit dimensions is larger than one, which is the case in multi user detection, the **H** matrix is a block Toeplitz matrix as shown in (3.2.7). In that case the boundaries of the symbols cannot directly be found based on **H**. Instead some kind of QR factorization is needed. However, it is desirable to avoid a full QR factorization of **H**. Furthermore, the structure of the Toeplitz matrix should preferably be kept, since this ensures that the trellis structure in e.g. the MAP receiver still can be used.
It should be mentioned that the derivation below has been assisted by L. P. B. Christensen. Besides, it is noted that in the derivation it is assumed that the number of receive dimensions is at least as large as the transmit dimensions, i.e. $N_R \geq N_T$. This assumption is made since the linear system of equations will otherwise be underdetermined. (Recall from Section 5.1 “Basic Concept of Sphere Decoding” that the SD algorithm can only be used for solving the system up to dimension $N_R\left(N_{BL}+L-1\right)$ and a full search would have to be carried out for the rest of the dimensions).

5.6.1 QL Factorization

As explained above, the goal is to preserve the Toeplitz structure of **H** and at the same time be able to bound the symbols of all users. Hence, the **H** matrix should be transformed into a new matrix, $\widetilde{\mathbf{H}}$, which has the form

$$\widetilde{\mathbf{H}} = \begin{bmatrix} \widetilde{\mathbf{H}}_0 & 0 & \cdots & \cdots & 0 \\ \widetilde{\mathbf{H}}_1 & \widetilde{\mathbf{H}}_0 & 0 & \cdots & \vdots \\ \vdots & \widetilde{\mathbf{H}}_1 & \ddots & \ddots & \vdots \\ \widetilde{\mathbf{H}}_{L-1} & \ddots & \ddots & \ddots & 0 \\ 0 & \widetilde{\mathbf{H}}_{L-1} & \ddots & \ddots & \widetilde{\mathbf{H}}_0 \\ \vdots & 0 & \ddots & \ddots & \widetilde{\mathbf{H}}_1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & \cdots & 0 & \widetilde{\mathbf{H}}_{L-1} \end{bmatrix}. \tag{5.6.1}$$

where the sub-matrix $\widetilde{\mathbf{H}}_0$ (still having the size $N_R \times N_T$) is lower triangular. Furthermore, $\widetilde{\mathbf{H}}_0$ must be of full row rank. This leads exactly to the property of $\widetilde{\mathbf{H}}$, which is desired. The elements in $\widetilde{\mathbf{H}}_0$ are denoted

$$\widetilde{\mathbf{H}}_0 = \begin{bmatrix} \widetilde{h}_{1,1}^{0} & 0 & \cdots & 0 \\ \widetilde{h}_{2,1}^{0} & \widetilde{h}_{2,2}^{0} & \ddots & \vdots \\ \vdots & \vdots & \ddots & 0 \\ \widetilde{h}_{N_T,1}^{0} & \widetilde{h}_{N_T,2}^{0} & \cdots & \widetilde{h}_{N_T,N_T}^{0} \\ \mathbf{0}_{(N_R-N_T)\times 1} & \mathbf{0}_{(N_R-N_T)\times 1} & \cdots & \mathbf{0}_{(N_R-N_T)\times 1} \end{bmatrix}, \tag{5.6.2}$$

where $\mathbf{0}_{(N_R-N_T)\times 1}$ is a vector containing $N_R - N_T$ zeros. To obtain such a $\widetilde{\mathbf{H}}_0$ it is necessary to make a QL factorization of $\mathbf{H}_0$, which is basically equivalent to a QR factorization of $\mathbf{H}_0$. The only difference is that the triangular matrix in the QL factorization will be lower triangular, which also leads to another unitary matrix. Hence, $\widetilde{\mathbf{H}}_0$ is found as

$$\mathbf{H}_0 = \mathbf{Q}_0 \mathbf{L}_0, \quad \textit{where} \quad \mathbf{L}_0 = \widetilde{\mathbf{H}}_0, \tag{5.6.3}$$

where $\mathbf{Q}_0$ has the size $N_R \times N_R$.

When the mapping from $\mathbf{H}_0$ to $\widetilde{\mathbf{H}}_0$ is found, the mapping between **H** and $\widetilde{\mathbf{H}}$ can be made. To find this mapping the Kronecker product is needed. In [17] it is shown that the Kronecker product of matrix **A** with size $m \times n$ and matrix **B** becomes

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} & \cdots & a_{2n}\mathbf{B} \\ \vdots & \vdots & \cdots & \vdots \\ a_{m1}\mathbf{B} & a_{m2}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix}. \tag{5.6.4}$$

Let an identity matrix, **I**, have the size (N_{BL}+L-1)×(N_{BL}+L-1). The Kronecker product of **I** and $\mathbf{Q}_0^H$ will then be a matrix of size N_R·(N_{BL}+L-1)×N_R·(N_{BL}+L-1) given as

$$\mathbf{I} \otimes \mathbf{Q}_0^H = \begin{bmatrix} \mathbf{Q}_0^H & \mathbf{0}_{N_R \times N_R} & \cdots & \mathbf{0}_{N_R \times N_R} \\ \mathbf{0}_{N_R \times N_R} & \mathbf{Q}_0^H & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_{N_R \times N_R} \\ \mathbf{0}_{N_R \times N_R} & \cdots & \mathbf{0}_{N_R \times N_R} & \mathbf{Q}_0^H \end{bmatrix}. \tag{5.6.5}$$

Therefore, it is possible to transform **H** into $\widetilde{\mathbf{H}}$ by

$$\widetilde{\mathbf{H}} = \left(\mathbf{I} \otimes \mathbf{Q}_0^H\right)\mathbf{H}. \tag{5.6.7}$$

Hence, it is possible to obtain the convenient structure of $\widetilde{\mathbf{H}}$, by multiplication of $\mathbf{Q}_0^H$ with each sub-matrix in **H**. Since the channel convolution matrix is rotated, the received signal must also be rotated, before it is given as an input to the receiver. Thus, the new received signal can be expressed as

$$\mathbf{y} = \left(\mathbf{I} \otimes \mathbf{Q}_0^H\right)\mathbf{x} = \widetilde{\mathbf{H}}\mathbf{s} + \left(\mathbf{I} \otimes \mathbf{Q}_0^H\right)\mathbf{z}. \tag{5.6.8}$$

Since $\mathbf{Q}_0$ is unitary, the statistics of the transformed noise term will not be changed. The SD constraint then becomes

$$r^2 \geq \left\| \mathbf{y} - \widetilde{\mathbf{H}}\mathbf{s} \right\|^2. \tag{5.6.9}$$

5.6.2 Bounding the Symbols

To ease the derivation of the bounding of the symbols some new variable definitions are needed. Let the received signal $\mathbf{y}_n \in \mathbb{C}^{(N_R \times 1)}$ at time index n contain the elements

$$\mathbf{y}_n = \begin{bmatrix} y_{(n-1)N_R+1} & y_{(n-1)N_R+2} & \cdots & y_{nN_R} \end{bmatrix}^T, \quad where \quad n = 1, 2, ..., N_{BL} + L - 1, \tag{5.6.10}$$

and let the vector $\mathbf{y}_{i_1:i_2} \in \mathbb{C}^{((i_2+1-i_1)\times 1)}$ be defined as

$$\mathbf{y}_{i_1:i_2} = \begin{bmatrix} y_{i_1} & y_{i_1+1} & \cdots & y_{i_2} \end{bmatrix}^T, \quad where \quad i_1 \leq i_2 \quad and \quad i_1, i_2 \in \left[1, 2, ..., N_R \cdot \left(N_{BL} + L - 1\right)\right]. \tag{5.6.11}$$

Recall from (5.2.5) that the symbol vector, **s**, is given as

$$\mathbf{s}_1^{n'} = \left[\mathbf{s}_1^T, \mathbf{s}_2^T, \ldots, \mathbf{s}_{n'}^T\right]^T, \quad where \quad n' = \min\left(n, N_{BL}\right),$$

here vector $\mathbf{s}_n^T = \begin{bmatrix} s_n^{(1)} & s_n^{(2)} & \cdots & s_n^{(N_T)} \end{bmatrix}$. Let the vector $\mathbf{s}_{i_1:i_2} \in \mathbb{C}^{((i_2+1-i_1)\times 1)}$ contain the i_1'*th* to the i_2'*th* elements of vector $\mathbf{s}_1^{n'}$. Finally let the matrix $\widetilde{\mathbf{H}}_{1:i} \in \mathbb{C}^{(i \times N_T)}$ contain the first i rows of $\widetilde{\mathbf{H}}$.

When the Receive Dimensions are larger than the Transmit Dimensions

Without loss of generality, it can be assumed that the number of transmit dimensions and received dimensions are the same, (i.e. $N_R = N_T$). This is because the first N_T samples in the received signal,

$\mathbf{y}_n$, at time index *n* will contain the transmitted information, while the last of the $N_R - N_T$ samples in $\mathbf{y}_n$ only contain noise, i.e.

$$\mathbf{y}_n = \begin{bmatrix} \mathbf{y}_{n,signal} \\ \mathbf{y}_{n,noise} \end{bmatrix}, \tag{5.6.12}$$

where $\mathbf{y}_{n,signal} \in \mathbb{C}^{(N_T \times 1)}$ and $\mathbf{y}_{n,noise} \in \mathbb{C}^{((N_R - N_T) \times 1)}$. From (5.6.2) it is seen, that (5.6.12) in fact must be true. Therefore, the noise samples should be disregarded. This is done by either altering the optimization problem in (5.6.9) a bit by excluding the rows in $\mathbf{y}$ and $\widetilde{\mathbf{H}}\mathbf{s} + \mathbf{z}$, that solely contain noise, (this also changes the DF, when choosing the radius). Alternatively, a permutation matrix is multiplied on both sides of (5.6.9) which ensures that all noise samples will be located in the (N_R – N_T) bottom rows. Now the method in Section 5.1 (more precisely the steps from (5.1.8) to (5.1.10)) can be used on the (N_R – N_T) bottom rows. Hence, the problem will then be similar to the case where $N_R = N_T$.

When the Receive Dimensions and the Transmit Dimensions are the same
Since the MUD problem has now been reformulated, the traditional SD algorithm is used. First a loose necessary condition has to be fulfilled, which is used to bound the first symbol. Then the requirements are tightened as the dimensions are increased, just as described in Subsection 5.1.
If (5.6.9) is compared with (5.6.2), it is seen how the bounding of the symbols should be carried out. Based on the first received sample $\mathbf{y}_{1:1}$, the boundaries of symbol one (i.e. $\mathbf{s}_{1:1} = s_1^{(1)}$) are calculated. It can be expressed as

$$\begin{aligned} r_1^2 &\geq \left\| \mathbf{y}_{1:1} - \widetilde{\mathbf{H}}_{1:1}\mathbf{s}_{1:1} \right\|^2 \\ r_1^2 &\geq \left\| \mathbf{y}_{1:1} - \widetilde{h}_{1,1}^0 \mathbf{s}_{1:1} \right\|^2 \\ r_1^2 &\geq \left\| \widetilde{h}_{1,1}^0 \right\|^2 \left\| \frac{\mathbf{y}_{1:1}}{\widetilde{h}_{1,1}^0} - \mathbf{s}_{1:1} \right\|^2 \\ \frac{r_1^2}{\left\| \widetilde{h}_{1,1}^0 \right\|^2} &\geq \left\| \frac{\mathbf{y}_{1:1}}{\widetilde{h}_{1,1}^0} - \mathbf{s}_{1:1} \right\|^2 . \end{aligned} \tag{5.6.13}$$

By introducing the variables $\widetilde{\widetilde{r}}_1 = r_1^2 \Big/ \left\| \widetilde{h}_{1,1}^0 \right\|^2$ and $\widetilde{\widetilde{\mathbf{y}}}_{1:1} = \frac{\mathbf{y}_{1:1}}{\widetilde{h}_{1,1}^0}$ (5.6.13) becomes

$$\widetilde{\widetilde{r}}_1 \geq \left\| \widetilde{\widetilde{\mathbf{y}}}_{1:1} - \mathbf{s}_{1:1} \right\|^2, \tag{5.6.14}$$

From (5.6.14) it is seen that the boundaries of the symbol $\mathbf{s}_{1:1} = s_1^{(1)}$ are found by a circle search in the complex plane as described in Section 5.2.2 “Bounding the Symbols”. When the boundaries of the first symbol have been found, the next symbol, $\mathbf{s}_{2:2} = s_1^{(2)}$, has to fulfil

$$
\begin{aligned}
r_2^2 &\geq \left\| \mathbf{y}_{1:2} - \widetilde{\mathbf{H}}_{1:2}\mathbf{s}_{1:2} \right\|^2 \\
r_2^2 - \left\| \mathbf{y}_{1:1} - \widetilde{h}_{1,1}^0 \mathbf{s}_{1:1} \right\|^2 &\geq \left\| \mathbf{y}_{2:2} - \widetilde{h}_{2,1}^0 \mathbf{s}_{1:1} - \widetilde{h}_{2,2}^0 \mathbf{s}_{2:2} \right\|^2 \\
r_2'^2 &\geq \left\| \mathbf{y}'_{2:2} - \widetilde{h}_{2,2}^0 \mathbf{s}_{2:2} \right\|^2,
\end{aligned}
\tag{5.6.15}
$$

where $r_2'^2 = r_2^2 - \left\| y_{1:1} - \widetilde{h}_{1,1}^0 \mathbf{s}_{1:1} \right\|^2$ and $\mathbf{y}'_{2:2} = \mathbf{y}_{2:2} - \widetilde{h}_{2,1}^0 \mathbf{s}_{1:1}$. By comparing (5.6.15) with (5.6.13) it is clear that boundaries of $s_1^{(2)}$ can be found using the circle search again. This approach is repeated for each dimension. If the radius should be chosen based on the degrees of freedom of the Chi-square distribution, it is seen that the DF increase with one each time the dimension is increased.

5.7 Summary

The basic concepts of sphere decoding have been treated. It is illustrated how the SD algorithm finds the MLSE solution in a computational efficient way. Furthermore, it is described that the algorithm, over a range of SNRs, has an expected complexity which is polynomial in the number of dimensions of the received signal. However, it will only be true when the received point is assumed to be disturbed by additive white Gaussian noise. The worst-case complexity will though be exponential.

One of the important issues in sphere decoding is related to how an appropriate radius of the sphere is found, since the radius has significant influence on the complexity. Some approaches using increasing radii have been treated. They do not guarantee the MLSE solution but offer a considerably reduction in complexity. From a statistical point of view the IRS approach gives an elegant description of how to choose the radius in a given dimension based on the choice of radii in the previous dimensions. Unfortunately, when the number of complex dimensions (i.e. degrees of freedom) become larger than approximately 50, the method turns out to be numerically unstable when IEEE double precision floating-point format is used. This leads to a poor estimate of the radii and, therefore, an implementation using a higher floating point resolution, than the one MATLAB uses per default, is needed.

It has been shown that it is possible to combine the SD algorithm with the more traditional receivers, (e.g. the MLSE or MAP receiver). When the traditional receivers also exploit the SD constraint, the state space will be pruned dynamically, and it is actually possible to predict, which state transitions fulfil the SD constraint, before the cost function is calculated.
The way of examining the symbols that fulfil the SD constraint will in some cases have an influence on the complexity. Therefore, the Schnorr-Euchner search strategy is proposed. Using this search strategy the maximum number of states in the trellis diagram can be specified. It is also possible to limit the number of state transitions from state to state. Both these modifications ensure that the peak complexity can be specified. However, a receiver relying on these modifications will not always give the MLSE solution.

The use of the LMMSE solution in combination with SD has also shortly been discussed. This can be interesting when the goal is to make exact MLSE.

Finally MUD using sphere decoding has also been described. In the case of multiple users it turns out that a kind of QL factorization of the channel convolution matrix is needed before it is possible to calculate the boundaries of the symbols.

6 Results

In this section the simulation results of the MAP receiver exploiting the SD algorithm are presented. In first subsection a description of the general simulation setup is described. Then the simulations concerning GSM are presented. Finally, in Subsection 6.3 the simulation results concerning EDGE are found. In the cases where it is possible to determine the performance of the optimal receiver, the performance of this is included in the plots.
Since there are quite a lot of simulation results which are presented, it has been decided to group the simulation results of both GSM and EDGE into subsections. In both the GSM and EDGE subsection, there will be some subsections concerning different simulation scenarios. It should be stated that the focus in the simulation results have been on the EDGE. This is because it is especially for EDGE the complexity of the optimal receiver becomes unacceptable high.

6.1 Simulation Setup

Setup Parameters in General
The simulation setup is similar to the one used in Section 4.3, but to make it convenient for the reader, it is again described shortly here.
In all test scenarios the oversampling factor is $N_{sps} = 2$, (i.e. $N_R = 2$) and the frame formats and modulation types used are identical to the ones that are used in the GSM and the EDGE system. It is assumed that the signal is stationary in a burst, (i.e. block fading) and that the channel impulse response is perfectly known. Frequency hopping is made between each bursts. In all simulations concerning sensitivity the "true" GMSK modulation is used in the transmitter. However, in simulations concerning interfering scenarios the linearized GMSK approximation is used in the transmitter. Whenever this is the case it will be mentioned in the simulations. The filter length of both the LMMSE and the IQ-LMMSE receiver has been set to $N_f = 32$ symbol periods in all the simulations in this thesis. Ideally, the filter length should be infinite, but this is of course not possible. Therefore, it is assumed that the filter length is $N_f = 32$ taps gives a reasonable filtration.

Setup for Parameters Concerning Sphere Decoding
In all the simulations below the value of ε is set to $\varepsilon = 0.1$ and the inverse Chi-square distribution is used to calculate the radius based on the degrees of freedom. In simulations where the static radius is used, it is the maximum number of DF that are used (as described in Section 5.3 "Choice of Radius"). If no solution is found, the search is restarted and the radius is calculated based on ε^2. In the cases where this will still not lead to any solution the radius is calculated based on ε^4, and so forth. Before the search is restarted, the paths in the trellis diagram which have not been pruned away are stored. Hence, it will not be necessary to calculate the cost function of these states again since they already fulfil the SD constraint with a smaller radius.

It should be stated that the abbreviation "SE" is used in the plots, when the Schnorr-Euchner search strategy, (described in Section 5.4, "Search Strategies"), is used. In all plots concerning the SE search strategy, increasing radii have been used. To ease the description of the receivers that uses the modified SE search strategy, a notation "*SE_S = X*" is introduced. *SE_S = X* refers to that *X* is

the maximum number of states, which is allowed in the receiver. This notation is used whenever the modified SE search strategy is commented in the figures below.
Due to time limitations, the technique to bound the symbols in the case of multiple users has not been implemented. Hence, in the simulations using MUD, all state transitions are examined for the states that have not been pruned away. If the state transitions do not fulfill the SD constraint they are pruned away afterwards.

6.1.1 Complexity Exponent

Complexity of the Optimal and Suboptimal Receivers
In the plots regarding the complexity of the algorithm, a complexity exponent is used. It is defined as the logarithm to the number of operations used to run through the trellis diagram divided by the logarithm of the burst length. This can be expressed as

$$e_c = \frac{\ln\left(\sum_{i=1}^{N_{BL}} N_{state\_trans}(i) \cdot C_{l,state\_trans}\right)}{\ln(N_{BL})}, \tag{6.1.1}$$

where $N_{state\_trans}(i)$ is the number of examined state transitions at time index i. $C_{l,state\_trans}$ is the number of complex operations that is used in an examination of a single state transition. Hence, the complexity exponent tells something about the number of operations used to decode a burst. I.e. for a given complexity exponent, e_c, the complexity of decoding the burst is $N_{BL}^{e_c}$.
When the complexity exponent is calculated in this thesis, the burst length is set to $N_{BL} = 142$, (here the tail bits are disregarded but the stealing flag and the TS is included in the calculation).
It turns out to be quite difficult to determine the exact number of complex operations which is used in an examination of a single state transition. Thus, a rough estimate, where only the major operations are taken into account, has been made instead. Therefore, the complexity exponent is only used to get a reasonable idea of the complexity and is not an exact measurement.
Recall from Section 4.2, "The MAP receiver", that the number of symbols representing a state transition from state $\mathbf{a}_n(p)$ to state $\mathbf{a}_{n+1}(q)$ are

$$\boldsymbol{\xi}_n^{(p,q)} = \left[\hat{s}_n(q) \quad \hat{s}_{n-1}(p) \quad \hat{s}_{n-2}(p) \quad \cdots \quad \hat{s}_{n-L+1}(p)\right]^T.$$

The expected output signal from each state transition is therefore calculated as a matrix multiplication with the channel impulse response. Hence, this will require $N_R \cdot L$ operations. Furthermore, N_R operations are needed to find the distance between the received signal and the expected output signal. Thus, the number of complex operations per examined state transition are roughly

$$C_{l,state\_trans} = N_R \cdot (L+1).$$

The number of examined state transitions have been determined exactly, and therefore, it can be used to give a precise description of the complexity of the receiver. Therefore, the examined number of state transitions will also appear in the plot of the complexity.

Complexity of the LMMSE Receiver
For comparison the complexity exponent of the LMMSE solution will also be calculated here. For general matrices with no structure, the complexity of finding the filter coefficients for the LMMSE filter will be $O\left(\left[N_R \cdot N_f\right]^3\right)$. However, since the channel matrix, $\mathbf{H}$, is block banded block Toeplitz,

the special structure can be exploited. For a Toeplitz matrix with no blocks the Levison-Durbin algorithm can be used. This require $O(N^2)$ operations when the matrix has size $N \times N$ (see. e.g. [35]). Since the channel matrix is block banded block Toeplitz, each sub-block of size $N_R \times N_R$ must be transformed, this requires N_R^3 operations. Therefore, the total number of operations used to calculate the filter coefficients are $O(N_R^3 \cdot N_f^2)$. For the actual filtering of the signal $N_R \cdot N_f$ operations are used for each symbol. Hence, the complexity exponent becomes

$$e_{c,LMMSE} = \frac{\ln\left(N_R^3 \cdot N_f^2 + N_{BL} \cdot N_R \cdot N_f\right)}{\ln\left(N_{BL}\right)}. \tag{6.1.2}$$

For the simulation setup in this thesis the complexity of the LMMSE receiver will therefore be

$$e_{c,LMMSE} = \frac{\ln\left(2^3 \cdot 32^2 + 142 \cdot 2 \cdot 32\right)}{\ln\left(142\right)} \approx 1.97.$$

Since this result does not depend on the channel length, the complexity exponent for the LMMSE will be the same for all simulations below.

6.2 GSM

The different test scenarios for GSM are listed below. In all the GSM simulations below, the C_0-pulse is modeled as having the symbol length five. When there is referred to channel length L, it is the complete channel impulse response, including the pulse shape. I.e. in a perfect channel (having a delta function as the impulse response), the complete channel impulse response will be $L = 5$.

1. SNR performance in an ideal channel with AWGN. Hence, the ISI in this test scenario is solely from the C_0-pulse. This gives an impulse response of length, $L = 5$. No interference from other users.
2. SNR performance in a TU0 channel with complete channel length, $L = 7$. No interference from other users.
3. SNR performance in a HT0 channel with complete channel length, $L = 11$. No interference from other users.
4. CIR performance at a constant SNR level at 10dB in a TU0 channel with complete channel length $L = 7$. Uses linearized GMSK pulse in transmitter. One interfering user.

6.2.1 AWGN

The BER performance in an ideal channel with AWGN is illustrated in Figure 6.1. It is seen that the performance of the receivers, using both a static radius and the IRS approach, gives a performance equal to the optimal one. Furthermore, it is seen that the receivers using the modified SE search strategy with *SE_S* = 16 and *SE_S* = 32 also gives a performance, which corresponds to the optimal one.

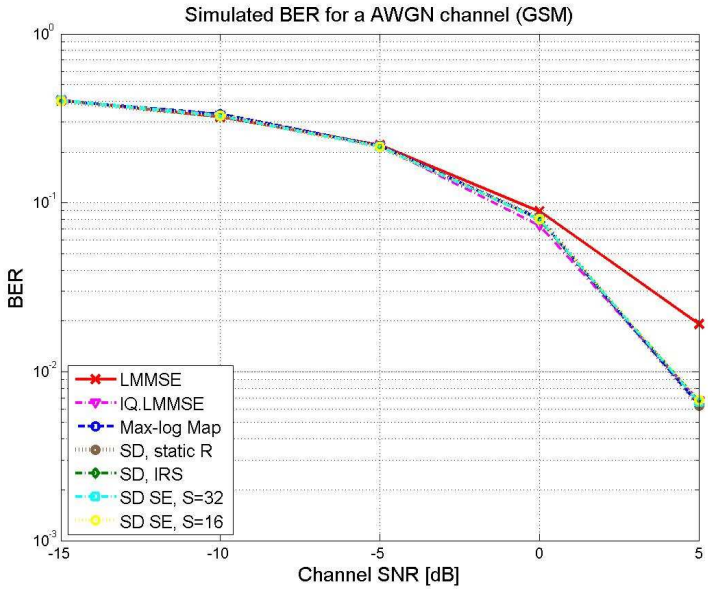

Figure 6.1. BER performance in an ideal channel with AWGN. Hence, the only ISI in this simulation is from the C_0-pulse.

The number of state transitions, which are examined in the optimal receiver is 2^5, leading to a complexity exponent of

$$e_c = \frac{\ln\left(\sum_{i=1}^{N_{BL}} N_{state\_trans}(i)\cdot C_{l,state\_trans}\right)}{\ln(N_{BL})} = \frac{\ln\left(2^5\cdot 2(5+1)\cdot 142\right)}{\ln(142)} \approx 2.20.$$

Since this is a rather low complexity, it is not as critical to reduce the complexity in this case as it will be in the other test scenarios.

6.2.2 TU0

The BER performance in the TU0 channel is plotted in Figure 6.2. It is seen that the performance of the SD algorithm using either the static radius or increasing radii is similar to the optimal MAP detector. Furthermore, it is seen that the modified SE search strategy with $SE\_S = 16$ and $SE\_S = 32$ likewise is capable of obtaining a BER performance which corresponds to the optimal one. Hence, even a receiver using only 16 states in the trellis diagram is able to give better performance than both the LMMSE and the IQLMMSE.

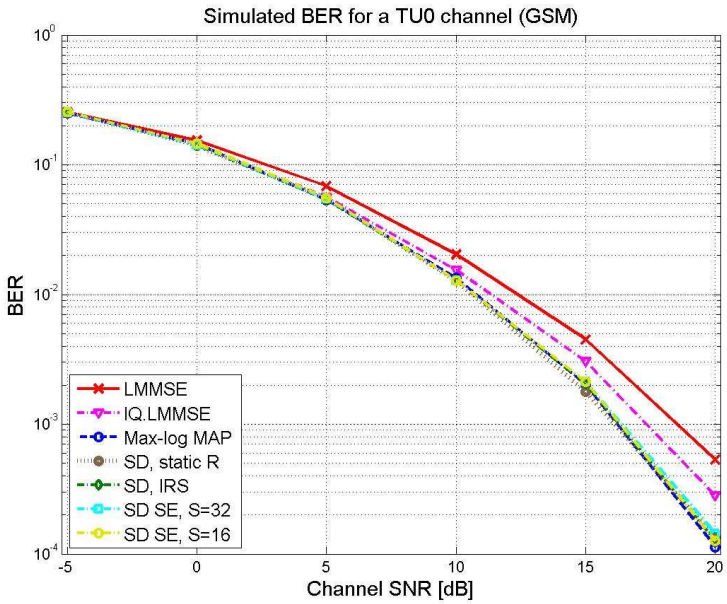

Figure 6.2. GSM simulation. BER performance in TU0.

The number of state transitions, which are examined in the optimal receiver is 2^7. Thus, the complexity exponent of the optimal receiver will be

$$e_{c,opt,TU} = \frac{\ln\left(2^7 \cdot 2(7+1)\cdot 142\right)}{\ln(142)} \approx 2.54.$$

The average number of examined states for the receivers is illustrated in Figure 6.3 and in Figure 6.4 the average complexity exponent as a function of SNR is plotted.

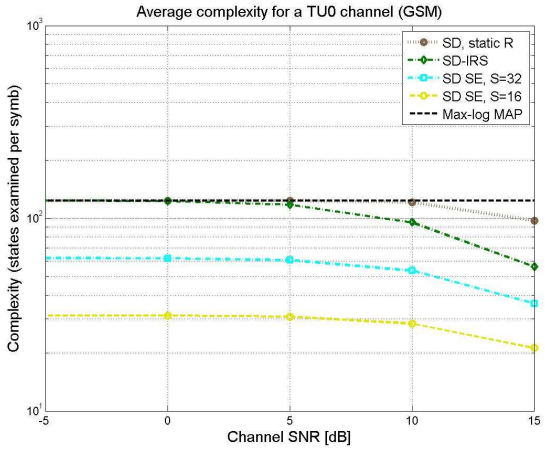

Figure 6.3. Average number of examined state transitions in TU0 as a function of SNR.

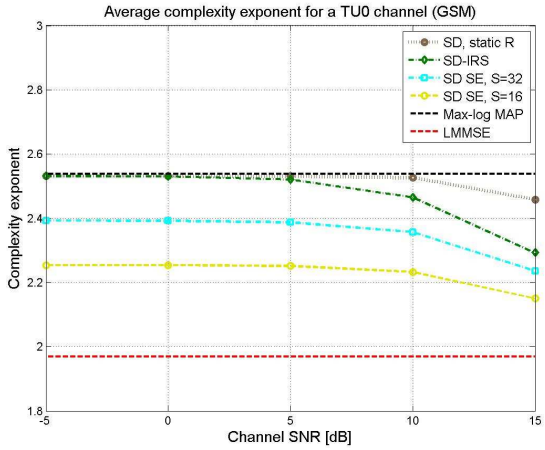

Figure 6.4. Average complexity exponent in TU0 as a function of SNR.

From Figure 6.3 and Figure 6.4 it is seen that the SD algorithm using the static radius will only give a minor reduction in complexity at SNR > 10dB. It is observed that the complexity is reduced more when increasing radii are used instead. Furthermore, it is seen that the constraint on the number of maximum allowable states, efficiently limit the complexity of the algorithm.

6.2.3 HT0

In Figure 6.5 the BER performance of the receivers in the HT0 channel is given. As it has been observed in the other BER simulations, the performance of the SD algorithm using static radius and increasing radii is the same as the Max-log MAP receiver. The modified SE search strategy with *SE_S* = 128 and *SE_S* = 64 also gives performance corresponding to the optimal one. However, when the maximum number of allowable states are decreased to *SE_S* = 32 there will be an degradation in performance. The performance will though still be better than the one obtained with the IQ-LMMSE receiver. When the number of allowable states is reduced even further to *SE_S* = 16 a considerably loss in performance is observed. This is an indication of the requirement to the maximum allowable states has been tightened too much. The performance of *SE_S* = 16 will now be worse than the IQ-LMMSE.

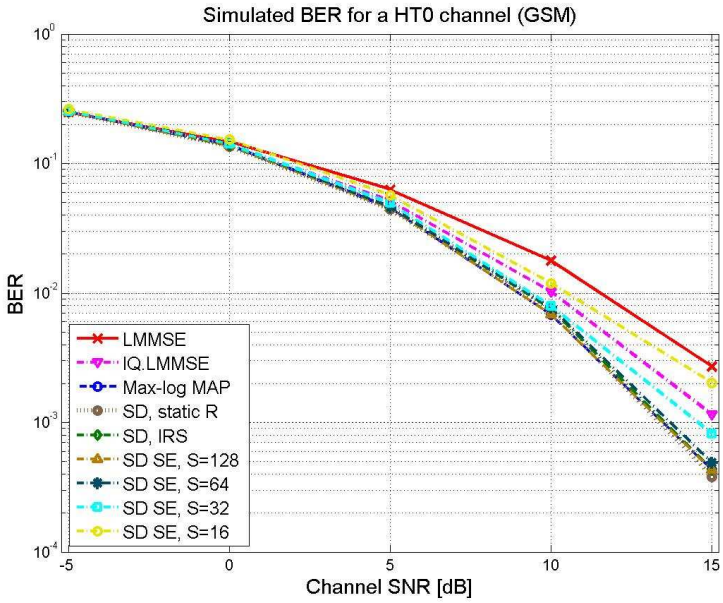

Figure 6.5. GSM simulation. BER performance in HT0.

The number of states in the trellis diagram of the optimal receiver is $2^{10} = 1024$, which leads to 2048 examined state transitions per symbol. The complexity exponent of the optimal receiver becomes

$$e_{c,opt,HT} = \frac{\ln\left(2^{11} \cdot 2(11+1) \cdot 142\right)}{\ln(142)} \approx 3.18.$$

The distribution of the number of examined states for SNR = 15dB is shown in Figure 6.6. The corresponding complexity exponent is plotted in Figure 6.7.

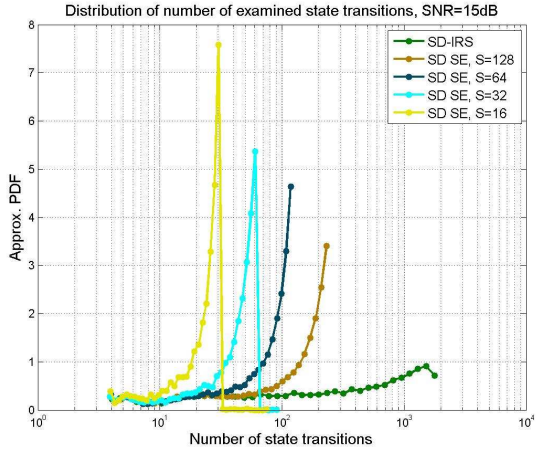

Figure 6.6. Distribution of the number of examined state transitions in HT0 when SNR = 15dB.

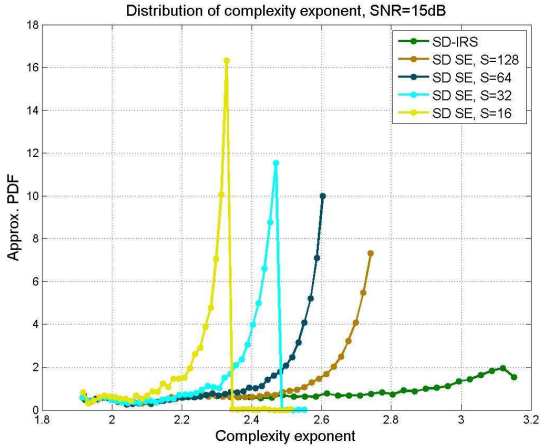

Figure 6.7. Distribution of the complexity exponent in HT0 when SNR = 15dB.

From Figure 6.6 and Figure 6.7 it is seen that the complexity of the SD algorithm using IRS (without max constraint on the number of states) will sometimes have a complexity which is just as large as the optimal receiver. On the other hand, the modified SE search strategy effectively limits the complexity from growing too large.

The average number of examined states per symbol for the receivers is illustrated in Figure 6.8. In Figure 6.9 the average complexity exponent is plotted as a function of the SNR.

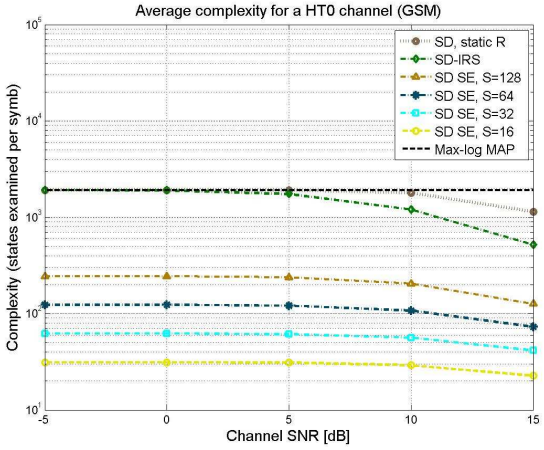

Figure 6.8. Average number of examined state transitions in HT0 as a function of SNR.

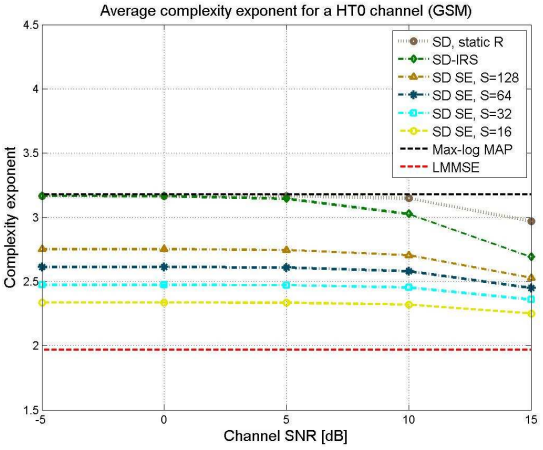

Figure 6.9. Average complexity exponent in HT0 as a function of SNR.

From Figure 6.8 and Figure 6.9 it is once again seen that the SD algorithm using static radius or IRS (without constraint on the maximum number of allowable states) over a wide range of SNRs have a complexity, which is close to the optimal receiver.

6.2.4 CIR Scenario in TU0

In the CIR simulation below the linearized GMSK modulation has been used in the transmitter, and the signal to noise ratio has been set to SNR = 10dB. Figure 6.10 shows the result of the CIR simulation. The yellow curve illustrates the case of SUD using the SD algorithm with increasing radii. There have not been used any whitening. In this case the SINR has been used instead of the SNR to estimate the radius of the sphere, since the interfering user is considered as being noise. From the figure it is seen that a considerably gain in BER performance is obtained by whitening the signal (see the red curve). It should be stressed that SD has not been used to create the red curve.

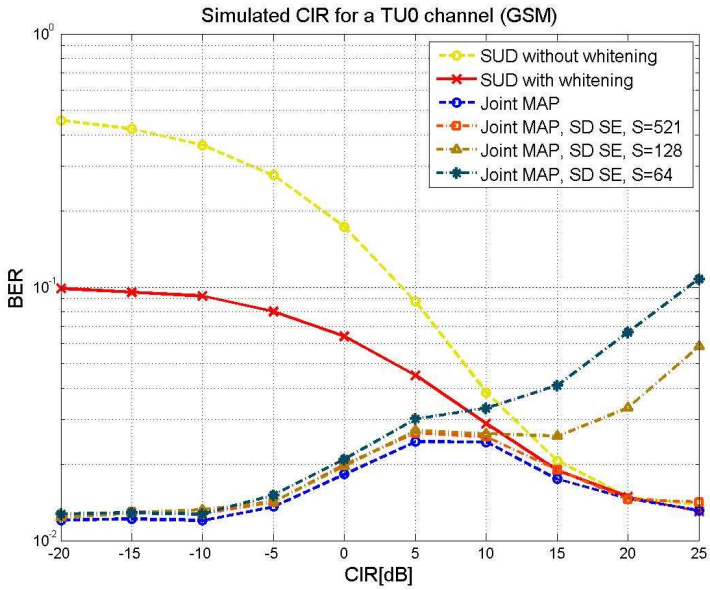

Figure 6.10. BER performance in CIR scenario with a constant SNR level at 10dB. Illustrates the performance using SUD with and without whitening and using MUD.

From Figure 6.10 it is seen that a substantial gain is obtained at low CIR using joint detection. For low CIR, all the receivers using SD with the modified SE search strategy will have comparable performance with the optimal Joint MAP detector. However, as the CIR increases the performance of the receivers, with *SE_S* = 64 and *SE_S* = 128, will be distorted more and more. This might seem surprising, but in a closer examination it makes good sense. Recall that the SNR is determined in respect to the desired user and that the signal from the interfering user in joint detection is not considered as being noise. Therefore, it is the SNR of the desired user, which only has influence on the radius of the sphere. When the CIR ratio becomes either very large or very small (e.g. 20dB or -20dB) the condition number of the channel matrix, **H**, becomes large. This leads to a very skewed lattice **Hs**, which implies that many points will be located close to each other. When the interfering user is dominant, the noise level will be small compared to the signal received from the interfering user. Therefore, the radius will be small compared with the interfering signal. Thus, it is likely that only a few lattice points are inside the sphere. The signal of the interfering user will on the contrary be small compared to the noise level at high CIR. Thus, many lattice points are likely to lie inside the sphere. Because of the constraint on the maximum number of states, it is likely that the true point will be disregarded (i.e. pruned away). A simplified illustration of the explanation above is given in Figure 6.11, Figure 6.12, and Figure 6.13.

In the example it is assumed that two users are using Amplitude Shift Keying (ASK) with 4 levels, ($s_1, s_2 \in \{-3, -1, 1, 3\}$).The possible transmitted symbols at each time index can then be represented as shown in Figure 6.11.

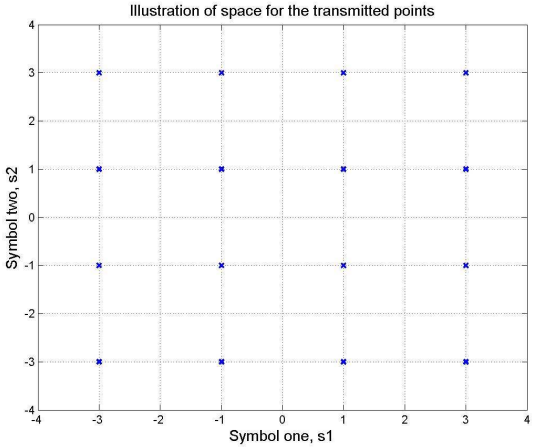

Figure 6.11. Symbol space of the transmitted symbols in the case of ASK with 4 levels.

In the case of no channel memory (i.e. $L = 1$) and a receive dimension of $N_R = 2$, the received signal is given as

$$\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1 & \mathbf{h}_2 \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \end{bmatrix} + \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}, \quad where \quad z_1, z_2 \; is \; noise \, .$$

For simplicity it is assumed that both the channel and the noise is real-valued. Then the received signal and the size of the radius can then e.g. be as shown in Figure 6.12 at low CIR. Figure 6.13 illustrates the situation where the CIR is high. Notice that the scaling of axis in Figure 6.12 and Figure 6.13 are not the same.

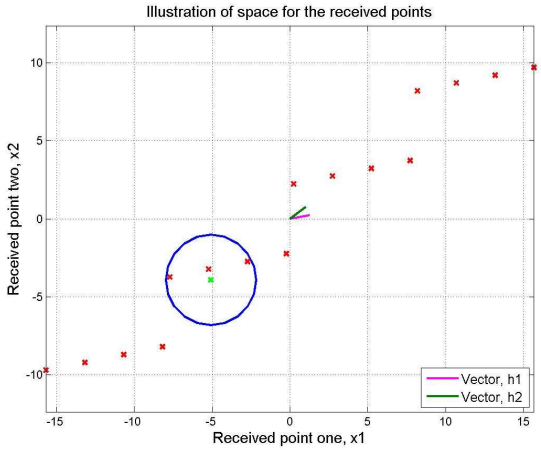

Figure 6.12. Simplified illustration of the lattice Hs compared with the radius at low CIR.

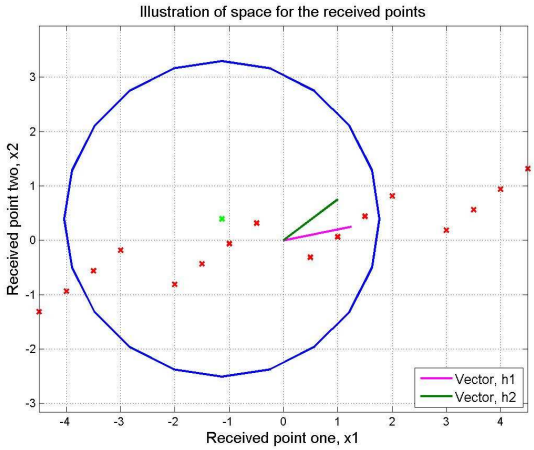

Figure 6.13. Simplified illustration of the lattice Hs compared with the radius at high CIR.

From Figure 6.13 it is clear that MUD detection with a constraint on the maximum number of allowable states in the trellis diagram will have problems at high CIR. However, the problem can be solved by regarding interfering user as being noise and then perform SUD.

The number of examined state transitions per symbol will for the Joint MAP detector be $2^{(2\cdot 7)} = 16384$. The complexity exponent of the optimal receiver becomes

$$e_c = \frac{\ln\left(2^{14} \cdot 2(14+1) \cdot 142\right)}{\ln(142)} \approx 3.64.$$

In Figure 6.14 the average number of examined state transitions is shown and in Figure 6.15 the corresponding complexity exponent is shown.

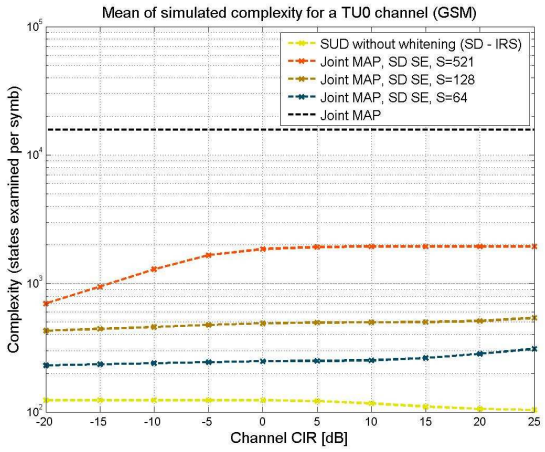

Figure 6.14. Average number of examined state transitions in CIR scenario as a function of CIR.

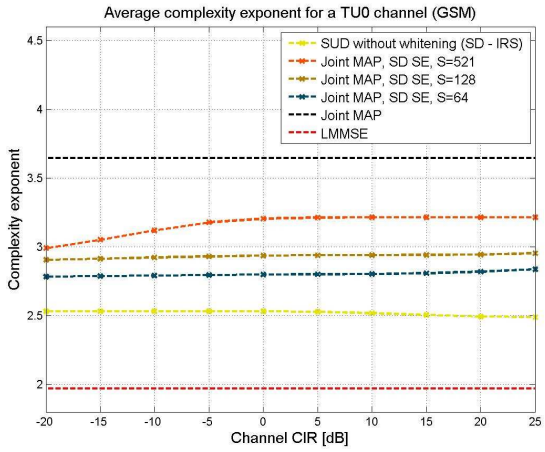

Figure 6.15. Average complexity exponent in CIR scenario as a function of CIR.

From Figure 6.14 and Figure 6.15 it is seen that SUD will have a considerable lower complexity compared to MUD. It is also seen that the MUD receivers using the SD will have the smallest complexity when the CIR is low. This is in agreement with what have been described above.

6.3 EDGE

From the results in the previous subsection, it is clear that the SD algorithm using a static radius has a considerably higher complexity than SD algorithms using increasing radii. Therefore, only SD algorithms using increasing radii are tested in the EDGE simulations. The different test scenarios for EDGE are listed below.

1. SNR performance in an ideal channel with AWGN. Hence, the ISI in this test scenario is solely from the C_0-pulse. This gives an impulse response of length, $L = 5$. No interference from other users.
2. SNR performance in a TU0 channel with a modified C_0-pulse, leading to channel length, $L = 5$ in receiver. No interference from other users.
3. SNR performance with full C_0-pulse in a TU0 channel leading to channel length, $L = 7$. No interference from other users.
4. SNR performance in a HT0 channel with a modified C_0-pulse, leading to channel length, $L = 9$ in receiver. No interference from other users.

It should be stated that the transmitter in all the simulations uses the normal C_0-pulse. Hence, it will only be in the receiver that a potential modification of the pulse is made.

6.3.1 AWGN

In Figure 6.16 the BER performance in an ideal channel with AWGN noise is given. An approximation of the best possible performance is also given. It should be stated that the theoretical curve is only an approximation where only errors to the closest constellation points are considered. This is a reasonable approximation for large SNR, but will still be a lower bound, since no ISI is considered in derivation of the theoretical approximation. The derivation of the approximation is

given in [1]. It should be stressed that the number of errors, which have to occur, have been set to 400 for the last point (i.e. SNR = 18). Hence, for SNR = 18dB there might be some uncertainty.

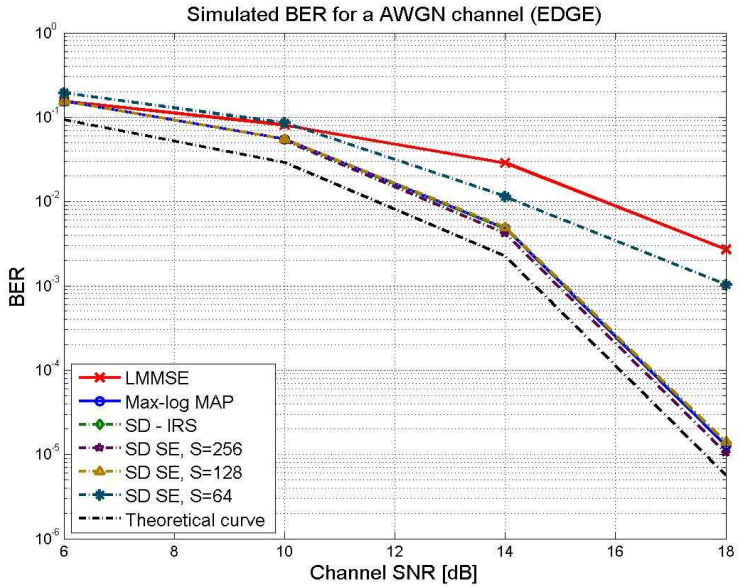

Figure 6.16. BER performance in an ideal channel with AWGN. Hence, the only ISI in this simulation is from the C_0-pulse.

From Figure 6.16. it is seen that the performance of the MAP receiver, which exploits the SD algorithm using the increasing radius, is exactly the same as the MAP receiver. This could also be expected, since the SD algorithm in this case, guarantees the MLSE solution. Furthermore, it is observed that the receiver using $SE\_S = 128$ is also capable of giving performance identical to the optimal. However, when the number of states for the receiver are reduced to $SE\_S = 64$, there will be a considerably degradation in performance. Actually, the performance will even be worse than the LMMSE receiver at SNR less than 10dB.
The number of examined state transitions for the optimal receiver will for each time index be $8^5 = 32768$ states. This leads to the complexity exponent

$$e_{c,opt,AWGN} = \frac{\ln\left(8^5 \cdot 2 \cdot (5+1) \cdot 142\right)}{\ln(142)} \approx 3.60 .$$

In Figure 6.17 the distribution of the number of examined state transitions for the different methods are plotted at SNR = 10dB. The corresponding distribution of the complexity exponents are given in Figure 6.18.

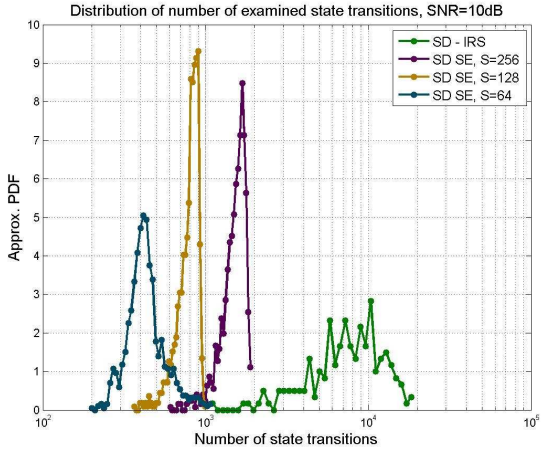

Figure 6.17. Distribution of the number of examined state transitions in an ideal channel with AWGN when SNR = 10dB.

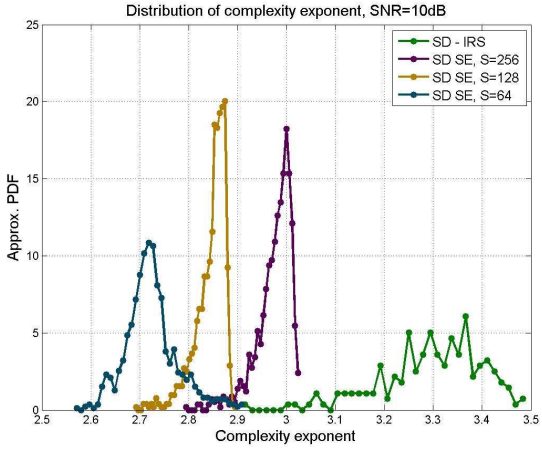

Figure 6.18. Distribution of the complexity exponent in an ideal channel with AWGN when SNR = 10dB.

From Figure 6.17 and Figure 6.18 it is clear that the complexity of the receiver using only the increasing radii method is significantly higher than the ones using the modified Schnorr-Euchner search strategy. Figure 6.19 and Figure 6.20 illustrates the distributions when the SNR = 18dB.

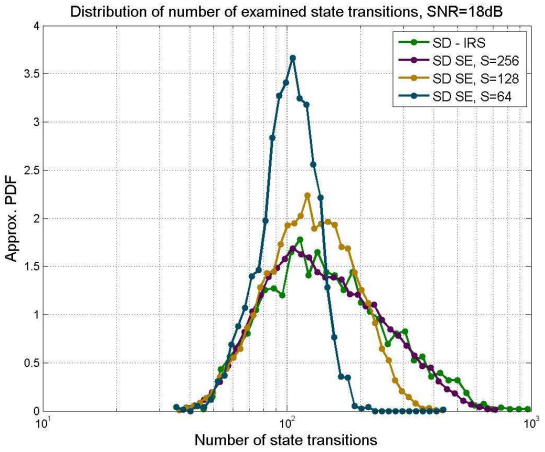

Figure 6.19. Distribution of the number of examined state transitions in AWGN when SNR = 18dB.

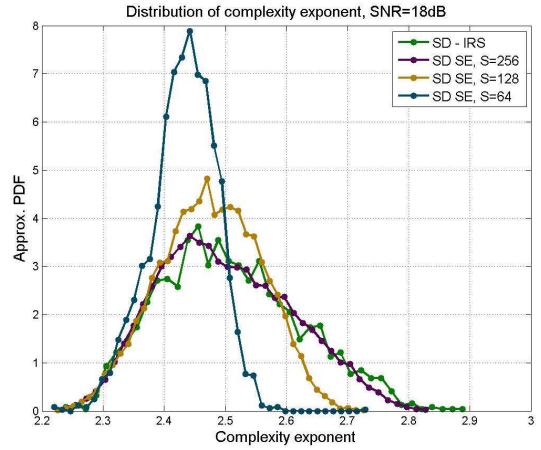

Figure 6.20. Distribution of the complexity exponent in AWGN when SNR = 18dB.

It is observed that the complexity will be considerable lower at SNR = 22dB compared to SNR = 10dB, which could also be expected. Plots of the average number of state transitions and the average complexity exponent are given in Figure 6.21 and Figure 6.22.

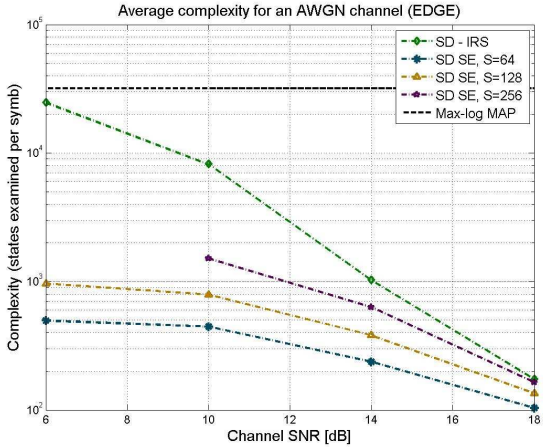

Figure 6.21. Average number of examined state transitions in AWGN as a function of SNR.

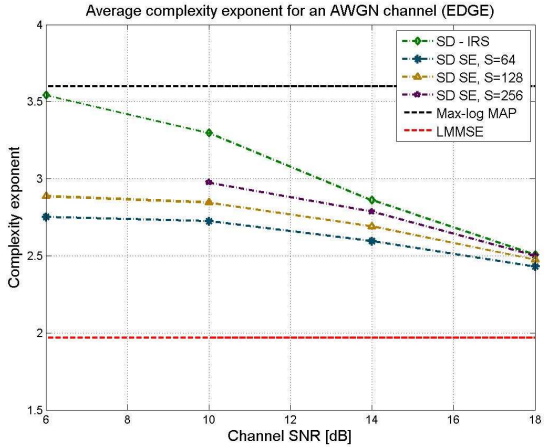

Figure 6.22. Average complexity exponent in AWGN as a function of SNR.

From Figure 6.21 and Figure 6.22 it is seen that the average complexity of the receiver using increasing radii only will be close to the optimal one at low SNR.

6.3.2 TU0

In the TU0 channel tests with both the normal C_0-pulse and a modified C_0-pulse (where the tails have been cut off) have been made. This is to examine if there is any difference in the performance. First the performance with the modified C_0-pulse is investigated, since it here will be possible to compare with the performance of the optimal receiver.

Modified C_0-pulse in receiver

In Figure 6.23 the BER performance is plotted for the different receivers. A zoom of the same plot is given in Figure 6.24. From the figures it is clear, that the receiver using only IRS (i.e. without max constraint on the number of states), has a performance corresponding to the optimal one. This is also true for some of the receivers using the modified SE search strategy. When $SE\_S \geq 128$ the receiver gives performance corresponding to the optimal receiver. However, when $SE\_S = 64$ the performance begins to be affected.

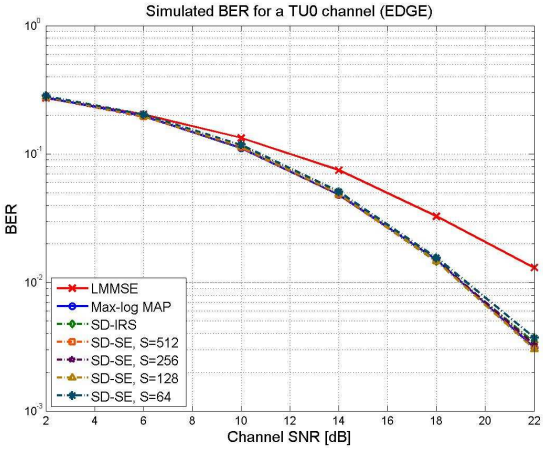

Figure 6.23. BER performance in TU0 with modified C_0-pulse.

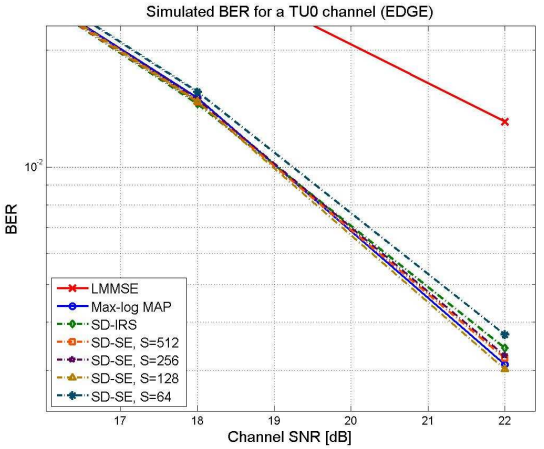

Figure 6.24. Zoom of BER performance in TU0 with modified C_0-pulse.

The complexity of the optimal receiver is for the modified C_0-pulse equal to the one described in the case of an AWGN channel without modification of the C_0-pulse. Hence, the number of

examined state transitions per symbol is $8^5 = 32768$ and the complexity exponent is $e_{c,opt,TU,modC_0} \approx 3.60$. The distribution of the number of examined state transitions and the complexity exponent for SNR = 2dB can be seen in Figure 6.25 and Figure 6.26, respectively.

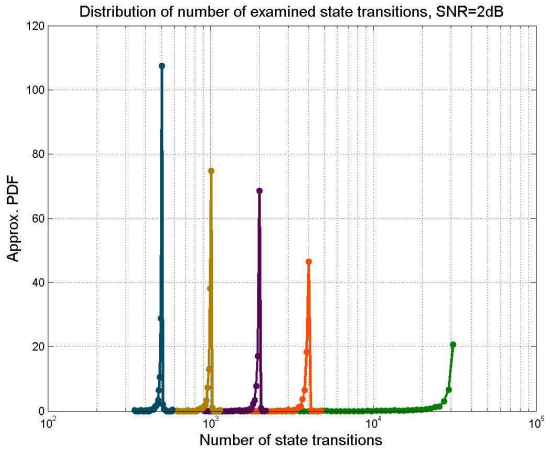

Figure 6.25. Distribution of the number of examined state transitions in TU0 when SNR = 2dB.

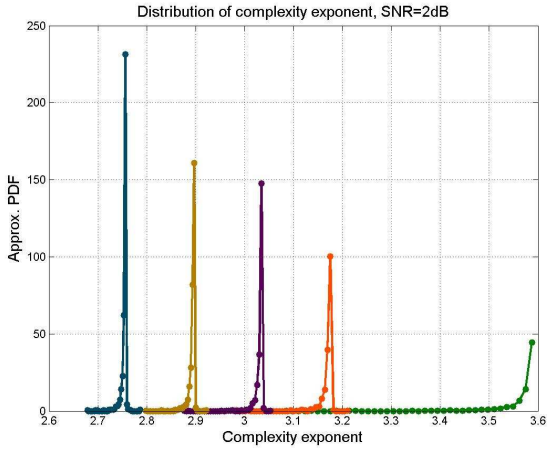

Figure 6.26. Distribution of the complexity exponent in TU0 when SNR = 2dB.

From Figure 6.25 and Figure 6.26 it is seen that the complexity of the receiver using the IRS approach, without max constraint on the number of states, often has a complexity corresponding to the optimal one. This is because the radius of the sphere becomes too large and will include all the points in the state space. As the SNR increases the radius of the sphere becomes smaller and the SD algorithm is able to prune some of the states away. This can be seen in Figure 6.27 and Figure 6.28 where the SNR = 22dB.

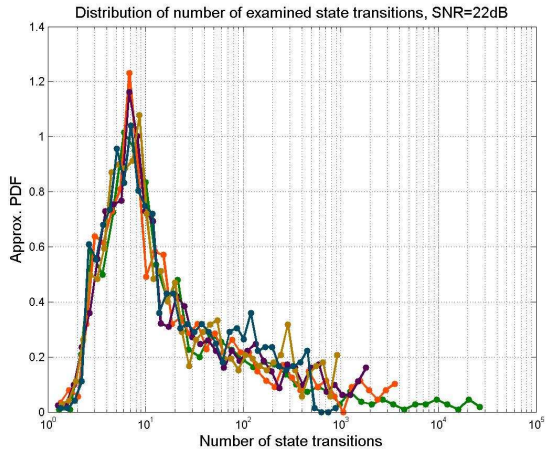

Figure 6.27. Distribution of the number of examined state transitions in TU0 when SNR = 22dB.

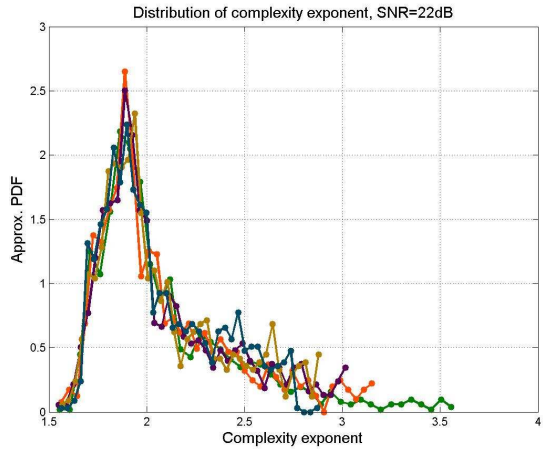

Figure 6.28. Distribution of the complexity exponent in TU0 when SNR = 22dB.

Figure 6.27 and Figure 6.28 indicates that the complexity of all the suboptimal receivers are roughly the same when SNR = 22dB. Hence, at high SNR the state space has been pruned efficiently. This can also be seen when the number of examined state transitions or the complexity exponents are examined. From Figure 6.29, which illustrates the average complexity of the algorithm, it is seen that the number of examined state transitions is minimized as the SNR increases. This can also be observed from Figure 6.30, where the complexity exponents of all the receivers using SD will converge towards the same complexity as SNR increases.

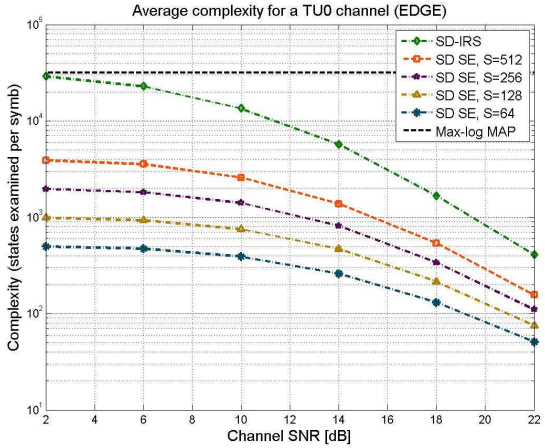

Figure 6.29. Average number of examined state transitions in TU0 as a function of SNR.

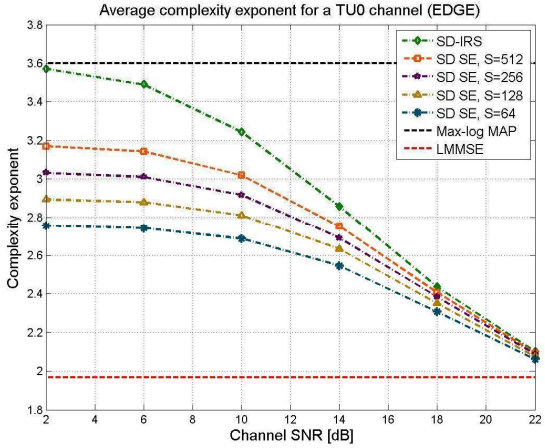

Figure 6.30. Average complexity exponent in TU0 as a function of SNR.

It is interesting to see how much the complexity is reduced, when the state transitions that fulfill the SD constraint are predicted. This should be compared with the method which first exams all transitions from a state that have not been pruned away and then does the pruning. Hence, both the methods that are compared use the sphere of the radius to prune the state space, but the latter will first do it after the examination of all transitions from a state. The prediction method (which is also called the bounding method) is the one, described in Section 5.2.2 "Bounding the Symbols", which is the one used in all other simulations (with exception of the ones concerning MUD). Figure 6.31 and Figure 6.32 shows the complexity of the two methods when the same modified SE search strategy has been used and $SE\_S = 512$.

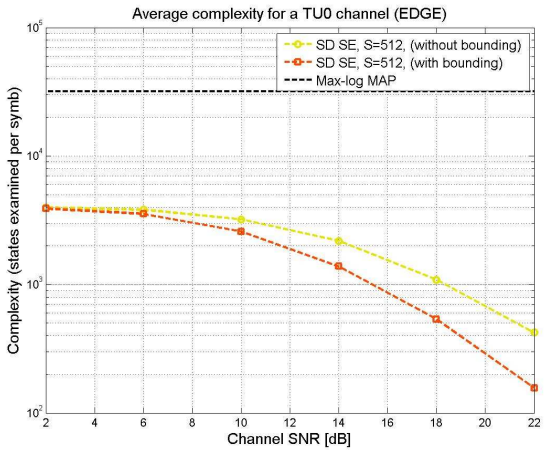

Figure 6.31. Average number of examined state transitions with and without bounding of the symbols (TU0).

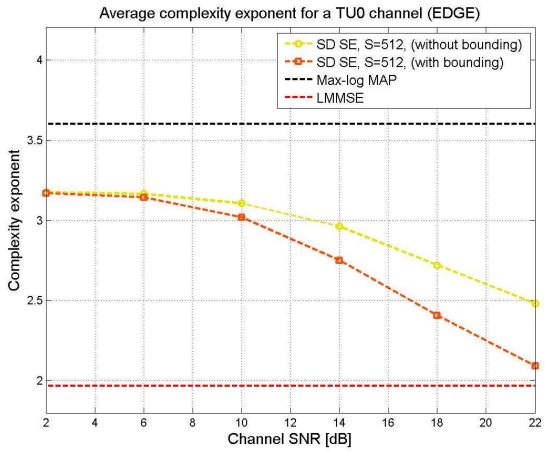

Figure 6.32. Average complexity exponent with and without bounding of the symbols (TU0).

It is seen that there is a significant reduction in the complexity for the method using bounding of the symbols at high SNR. This is because the SD constraint will be tightened as the SNR increases.

As described in Section 5.4 "Search Strategies", it is also possible to limit the number of branches (i.e. state transitions) which are allowed. A plot which shows the performance of this is given in Figure 6.33

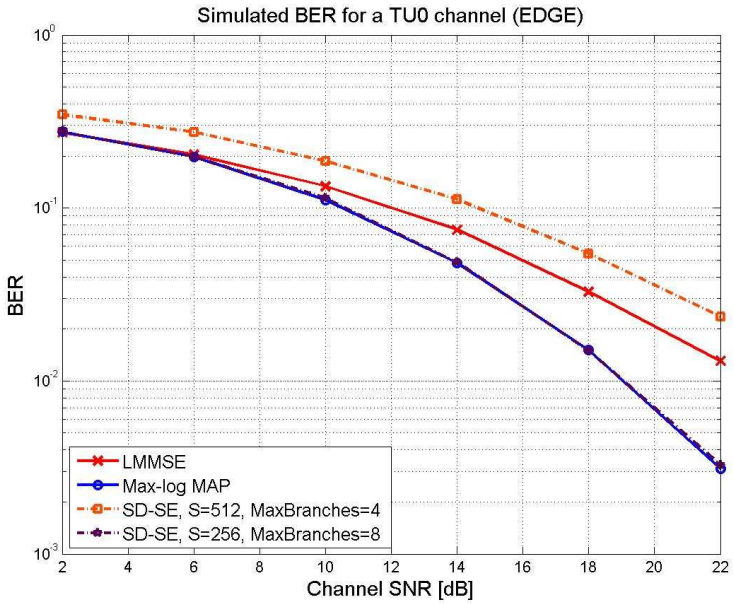

Figure 6.33. Comparison of the BER performance in TU0 when the number of branches from a state have been limited.

The orange curve in Figure 6.33 illustrates the case where only half of the possible state transitions have been allowed, (i.e. 4) for each state. It is clear that this gives a poor performance even though $SE\_S = 512$. For comparison, the case with $SE\_S = 256$ has also been plotted in the figure. For the latter there is no limit of the possible number of the state transitions coming from a given state. These two approaches ought to have comparable complexity. The reason, why the first mentioned method gives such a poor performance, is that the receiver prunes away states before the energy of the whole C_0-pulse has been observed. Recall, from the description of state transitions in the Markov model, that it is the new received symbol that has influence on the transition from a given state. Because of the pulse shape of the C_0-pulse, the first symbol $\hat{s}_n(q)$ will have small energy compared to symbols in the middle of

$$\xi_n^{(p,q)} = \left[\hat{s}_n(q) \quad \hat{s}_{n-1}(p) \quad \hat{s}_{n-2}(p) \quad \cdots \quad \hat{s}_{n-L+1}(p)\right]^T.$$

(Here the notation from Section 4.2, "The MAP Receiver" has been used). Thus, with the given pulse shape, it will be unwise to set a restriction on the maximum number of branches from a state.

Unmodified C_0-pulse

In Figure 6.34 a plot of the BER performance is given, when the C_0-pulse has not been modified. Since the complexity of the optimal receiver is too high to implement, the modified C_0-pulse has been used for this receiver (i.e. the one which has also been used above).

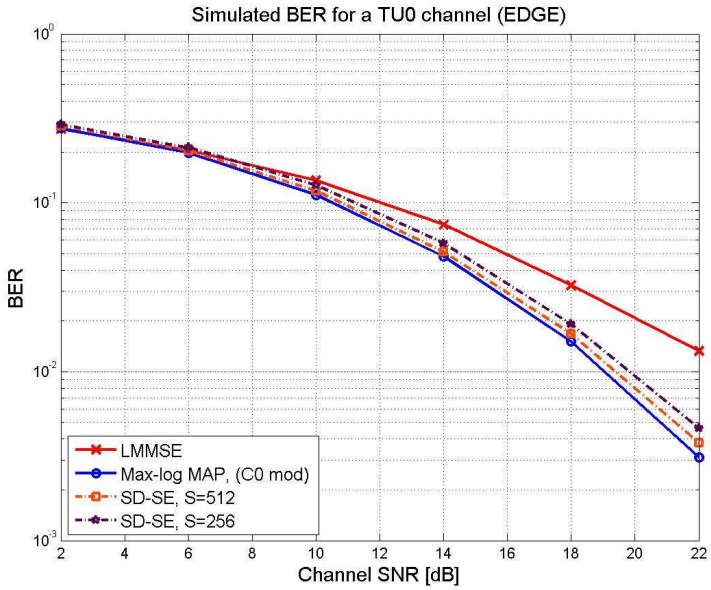

Figure 6.34. Comparison of the BER performance in TU0 with full C_0-pulse.

From the Figure 6.34 it is seen that the performance of the receivers using the modified SE search strategy with $SE\_S = 512$ and $SE\_S = 256$ will not have as good a performance now as the optimal one. Hence, by using the full C_0-pulse a performance degradation of the suboptimal receivers is seen. This is a quite interesting observation, since it might be expected that the full knowledge of the overall channel impulse response will lead to a slightly better BER. This could at least be expected for the traditional MAP receiver. The reason why this is not the case for the sub-optimal receiver, is to be found again in the pulse shape of the C_0-pulse. Since it in the beginning and end of the pulse shape has values close to zero, many of the states will nearly give the same output when they are multiplied with the channel impulse response. If they are considered as points in a hyper dimensional space, this would imply, that there would be series of clusters in the space. Each of these clusters would then contain many points. Hence, it becomes difficult for the receiver to decide which points to choose. Therefore, it is in this case necessary to increase the number of allowable states (i.e. increase SE_S).
It is likely that the problems observed in Figure 6.33 and Figure 6.34 could be solved by pre-filtering the received signal. A minimum-phase pre-filter can lead to a desirable impulse response of the channel, where the majority of the energy is in the beginning of the signal.

6.3.3 HT0

For the simulation of the HT0 channel a modified pulse is used in the receiver. Hence, as in some of the simulations in the TU0 channel, the tails of the C_0-pulse has been cut off. It is unrealistic to simulate receivers relying on IRS solely and therefore only receivers using the modified SE search strategy with a maximum of the number of states has been used. Since, it is unrealistic to determine the BER performance of the traditional MAP receiver (even in the case of a modified C_0-pulse), a theoretical curve of the optimal receiver in a HT channel is plotted instead. The theoretical curve is given in paper [33], which is written by W. H. Gerstacker and R. Schober. For comparison with other suboptimal receivers, a simulation result of the performance of the best receiver in [33] is also reproduced here.

In [33] it is assumed that the channel is time invariant in one burst and that there is perfect knowledge of the channel. Thus, the general simulation setup seems to be the same, but of course there can be slightly different configurations in the simulation anyway. Hence, one must be careful with jumping to conclusions, but the simulation can serve as an indication of the performance.
Before the performance result is presented, the receiver, which leads to the best simulated curve in [33] is described. It uses a reduced state equalization relying on delayed decision-feedback sequence estimation (DDFSE). In the reproduced simulation the number of states in the trellis diagram is set to 64, (i.e. $L = 3$, where $|\Omega|^{L-1}$ is the number of states and $|\Omega|$ is the alphabet size). This implies that only the first 3 taps of the impulse response of the channel is used for modeling the trellis diagram. To be able to obtain reasonable performance with DDFSE, a minimum-phase pre-filter has been applied.
The BER performance in the HT0 channel is shown in Figure 6.35.

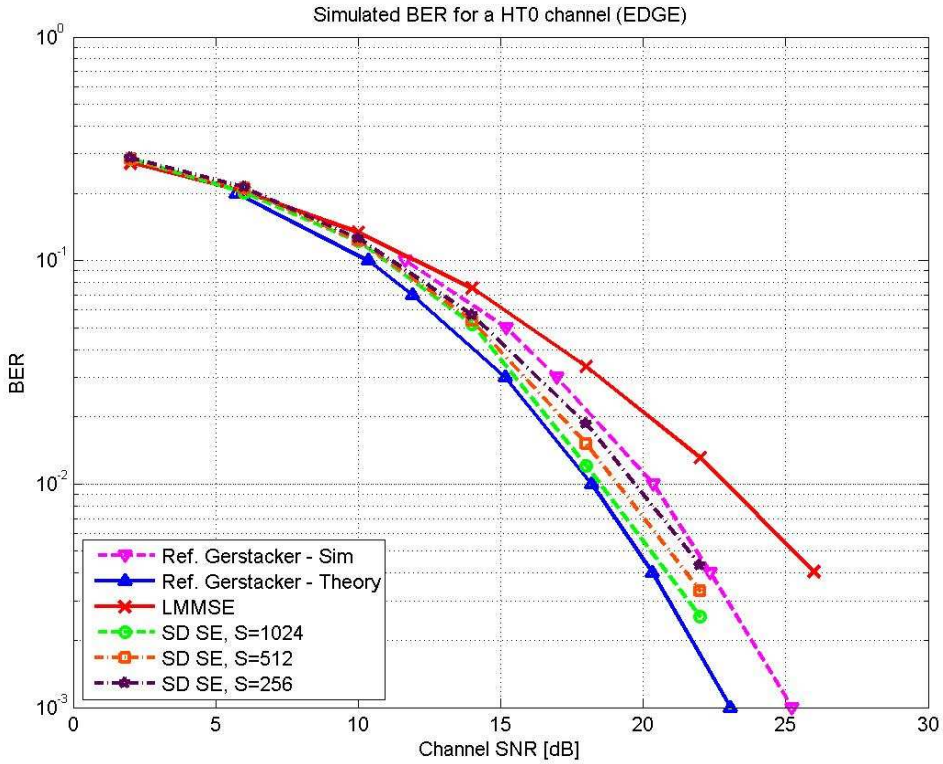

Figure 6.35. BER performance in HT0 with modified C_0-pulse.

From Figure 6.35 it is seen that the receiver using $SE\_S = 1024$ states will give the best performance. The performance is close to, but not as good as, the theoretical curve, however, it must be noted that the theoretical curve is a lower bound. Both the receivers, where $SE\_S = 1024$ and $SE\_S = 512$, gives performance that is better than the best obtained performance in [33]. It is seen that the receiver with $SE\_S = 256$ gives a performance close to the simulated curve in [33] for $L = 3$.

The number of examined state transitions at each time index will for the optimal receiver be 8^9, which is more than $134 \cdot 10^6$ states. Hence, it is clear that it would require enormous amount of computation power to find the optimal solution. The complexity exponent will be

$$e_{c,opt,HT,modC_0} = \frac{\ln\left(8^9 \cdot 2 \cdot (9+1) \cdot 142\right)}{\ln(142)} \approx 5.38 .$$

The complexity of the receiver proposed in [33] is not specifically described. However, it will be interesting to examine the complexity exponent in [33] anyway. To come up with an estimate of the complexity it is assumed that the calculation of the minimum-phase pre-filter coefficients and the pre-filtering will have the same complexity as the LMMSE receiver. Since, the filter length is neither described in [33] it is assumed that it has a symbol length of 32 as in the case of the LMMSE filter. Since the number of states in the trellis diagram is set to 64 the number of examined state transitions per symbol will be 512.

$$e_{c,Gerstacker} = \frac{\ln\left(\overbrace{N_R^3 \cdot N_f^2 + N_{BL} \cdot N_R \cdot N_f}^{pre-filter} + \sum_{i=1}^{N_{BL}} N_{state\_trans}(i) \cdot N_R \cdot (L+1)\right)}{\ln(N_{BL})}$$
$$= \frac{\ln\left(2^3 \cdot 32^2 + 142 \cdot 2 \cdot 32 + 142 \cdot 512 \cdot 2 \cdot (3+1)\right)}{\ln(142)} \approx 2.68.$$

In Figure 6.36 and Figure 6.37 plots of the distribution of the number of state transitions and the complexity exponent for the receivers are given, respectively, when the SNR = 2dB.

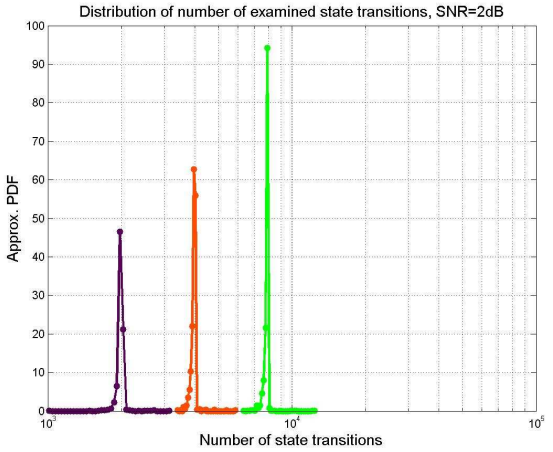

Figure 6.36. Distribution of the number of examined state transitions in HT0 when SNR = 2dB.

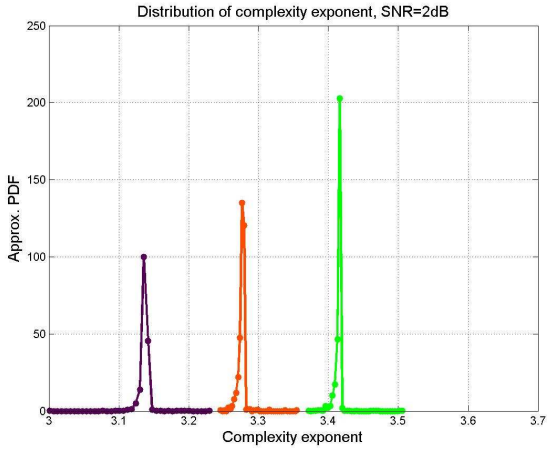

Figure 6.37. Distribution of the complexity exponent in HT0 when SNR = 2dB.

From Figure 6.36 and Figure 6.37 it is seen that the limit of the maximum number of states are reached for almost every time. Figure 6.38 and Figure 6.39 illustrates the same results for SNR = 10dB.

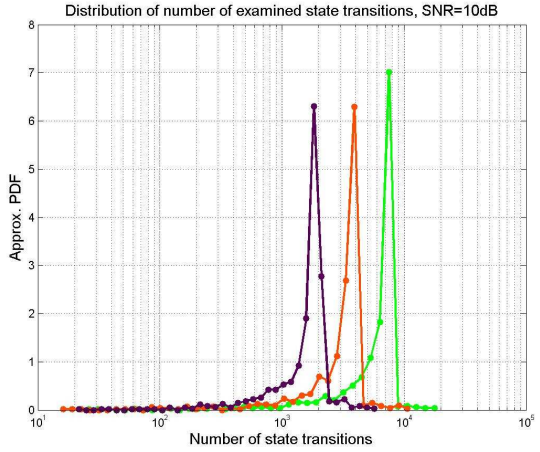

Figure 6.38. Distribution of the number of examined state transitions in HT0 when SNR = 10dB.

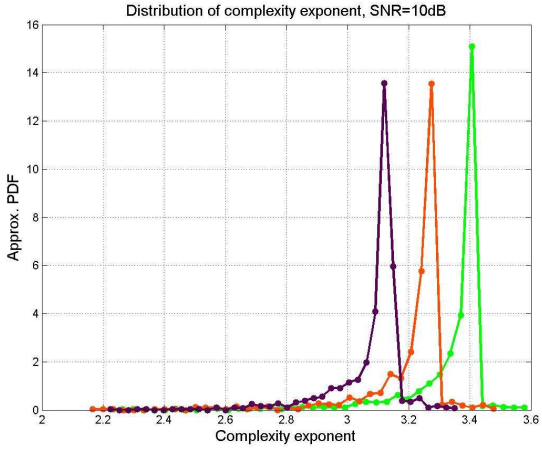

Figure 6.39. Distribution of the complexity exponent in HT0 when SNR = 10dB.

From Figure 6.38 and Figure 6.39 it is clear, that the complexity of the receivers still reaches the upper limit most frequently at SNR = 10dB. For SNR = 22dB the number of states needed will typically be less than the upper limit, *SE_S*. In Figure 6.40 and Figure 6.41 it is observed that the complexity of the three receivers are often the same.

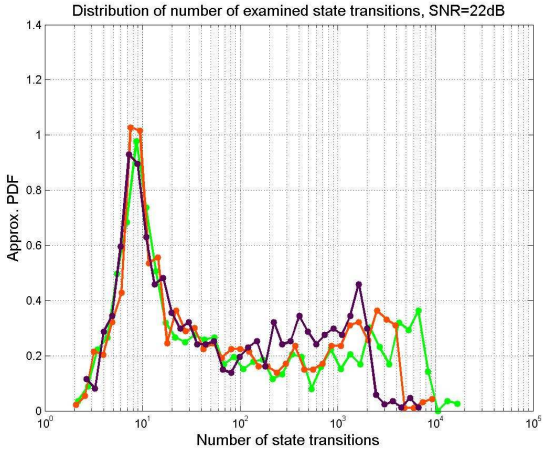

Figure 6.40. Distribution of the number of examined state transitions in HT0 when SNR = 22dB.

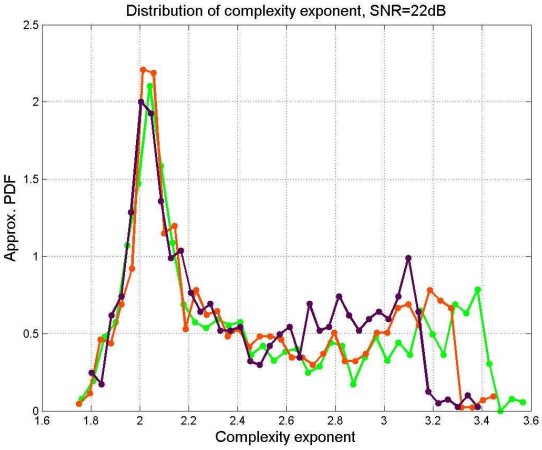

Figure 6.41. Distribution of the complexity exponent in HT0 when SNR = 22dB.

The plot of the average number of state transitions for all SNRs is found in Figure 6.42, while the average complexity exponent is plotted in Figure 6.43.

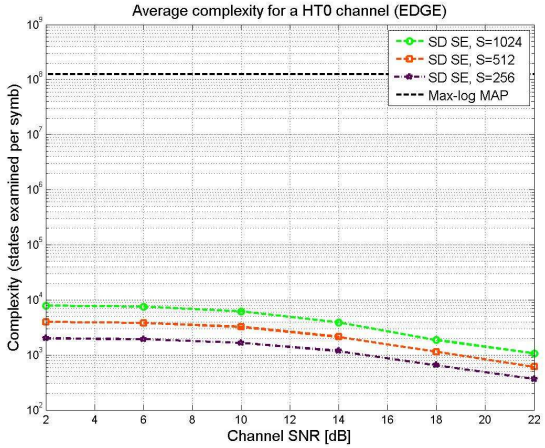

Figure 6.42. Average number of examined state transitions in HT0 as a function of SNR.

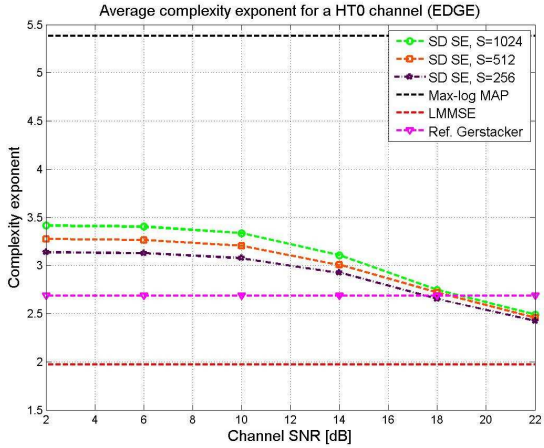

Figure 6.43. Average complexity exponent in HT0 as a function of SNR.

Observe from Figure 6.43 that the average complexity exponent converge through the same level as the SNR increases, this corresponds to what was seen in Figure 6.41. Hence, when the SNR = 22dB the complexity exponent will in average be 2.5 for all receivers. However, recall that the BER performance at SNR = 22dB for the three receivers are not identical. This implies that the number of states that are needed to model the trellis diagram reasonable, are sometimes larger. When the complexity exponents of the SD SE algorithms are compared with the approximated complexity exponent in [33], it is seen that the SD SE algorithms require more complexity at low SNR. At high SNR the complexity of the SD SE algorithms will be less than [33]. Furthermore, by comparing the complexity of the SD SE algorithms with the optimal one, it is realized that the complexity reduction is enormous. More specifically, the reduction in the number of examined state transitions is a factor of 16384 at SNR = 2dB, and a factor of approximately 131000 at SNR = 2dB (if the SE_S = 1024 is compared with the optimal receiver).
Another important observation is that it is in fact possible to achieve a complexity exponent which is lower that 3.5 in the HT0 channel.

The plots in Figure 6.44 to Figure 6.46 shows the frequency of the number of times the search in the trellis are restarted for SNRs of 2, 10 and 22dB, respectively. (Recall, that the search is restarted when there are no points inside the sphere).

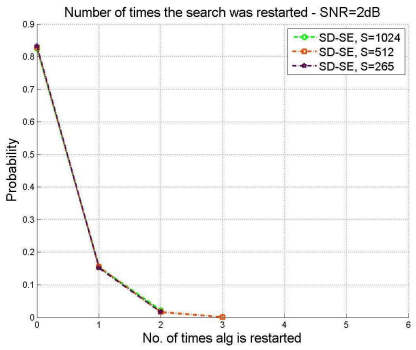

Figure 6.44. Frequency of the number of times the search are restarted. SNR = 2dB.

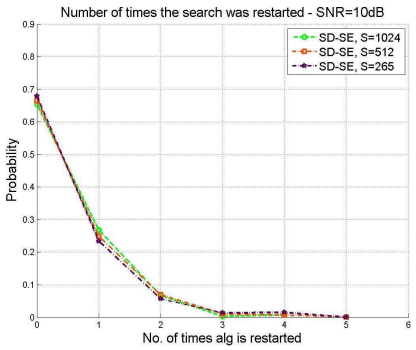

Figure 6.45. Frequency of the number of times the search are restarted. SNR = 10dB.

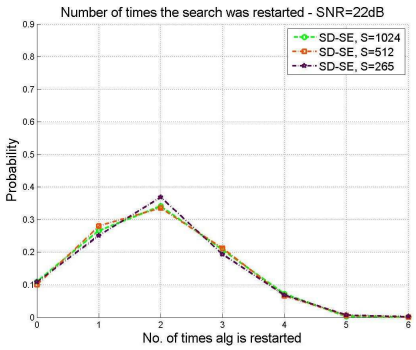

Figure 6.46. Frequency of the number of times the search are restarted. SNR = 22dB.

From Figure 6.44 to Figure 6.46 it is seen that the frequency of number of times the search is restarted, does not depend on the maximum allowed number of states, but rather on the SNR. When the SNR = 2dB it is most likely that the algorithm is not restarted. On the other hand, this is not the

case for SNR = 22dB. Here, it is most likely it will be restarted twice. This is a bit surprising since the radius is chosen based on the noise statistics. However, one possible explanation could be that the radius constraint is too loose at low SNRs, where many points are inside the hyper-sphere. Recall for Section 5.3 "Choice of Radius" that the MLSE solution is given as

$$\begin{aligned}\left\|\mathbf{x}-\mathbf{H}\hat{\mathbf{s}}_{MLSE}\right\|^2 &= \left\|\mathbf{H}\left(\mathbf{s}-\hat{\mathbf{s}}_{MLSE}\right)+\mathbf{z}\right\|^2 \\ &\leq \left\|\mathbf{H}\left(\mathbf{s}-\hat{\mathbf{s}}_{MLSE}\right)\right\|^2+\left\|\mathbf{z}\right\|^2.\end{aligned}$$

In cases of low SNRs, it is therefore likely that there are lattice points, which are closer to the received point than the transmitted point. As a final remark to the number of times the algorithm is restarted it can be noted that frequency of the restarts depends on the value of ε. Recall that 1-ε is the probability of having minimum one point inside the sphere. Since, the method for choosing the radii is based on the DF, they are chosen independently for each dimension. The probability of having a point inside the sphere will therefore in best case be $(1-\varepsilon)$, and will in worst case be $(1-\varepsilon)^{N_{BL}}$ (in the case of the choice of N_{BL} radii).

6.4 Summary

A MAP receiver which exploits the SD algorithm has been implemented. Experiments with different modifications of the SD search strategy have been carried out. Among others the Schnorr-Euchner search strategy with a maximum number of allowable states have been designed and tested.
The GSM simulations showed that the complexity can be reduced using sphere decoding. However, the method using a static radius gives over a wide range of SNR a complexity which is just as large as the optimal one. In all the simulations with the SD algorithm using increasing radii, the performance were similar to the optimal one but had a lower complexity. Therefore, it is recommended that an increasing radii scheme should be used instead of a static radius.

Even though the increasing radii method reduces the complexity in both GSM and EDGE at high SNR, it will still lead to a complexity, which is just as large as the optimal receiver at poor SNR. This is a great disadvantage of sphere decoding. The modified Schnorr-Euchner search strategy, where the maximum number of states is specified, seems to solve this problem. The modified Schnorr-Euchner search strategy has proved to give a BER performance which in many of the simulations is comparable to the optimal one. However, this will only be the case as long as the number of maximum states is not reduced too much. If the latter is the case, a considerably increase of the BER is observed. Thus, a critical issue in the modified Schnorr-Euchner search strategy is to choose the maximum number of allowable states reasonable.

One important observation is that the modified SE algorithm has problems when the first taps in the complete channel impulse response has low energy. This will be the case in GSM and EDGE because of the pulse shape of the C_0-pulse. Therefore, the number of states which are needed to obtain BER performance close to the optimal one, is likely to be reduced if a minimum-phase pre-filter is used.

From the simulations it is also seen that the BER increases considerably when there is a limit on the maximum number of state transitions from a state. It is very likely also because of the pulse shape of the C_0-pulse. Thus, some states will be pruned away even though only a small amount of energy

of the symbol have been observed. Therefore, pre-filtering is also likely to have a beneficial effect on this problem.

Furthermore, simulations have proved that the bounding technique used to predict, which state transitions fulfil the SD constraint, will give a significant reduction in the complexity, especially at high SNR.

7 Conclusion

In this thesis, both optimal and near-optimal detectors have been examined. It has been shown that considerable gain in the BER performance (in ISI scenarios) can be obtained by using the MAP detector, which minimizes the probability of bit errors, compared with linear detectors. Performance for both optimal SUD and optimal MUD have been studied. In interfering scenarios, it turns out that significant gain in BER performance can be obtained by using MUD. However, there is one great disadvantage of the optimal detector. It has been illustrated that the complexity of the optimal receiver is exponentially in the channel length and the number of transmitters (e.g. users). Especially, for higher order modulation techniques, such as 8-PSK, the complexity will be enormous, which prevents the optimal detector from being implemented in a MS.

Due to the complexity problem with the optimal detector, alternative ways of achieving near-optimal performance becomes interesting. The task of the near-optimal detectors is to reduce the complexity as much as possible without affecting the BER performance too much.

Sphere Decoding addresses the problem of solving the integer least-squares problem in a computationally efficient way (which corresponds to finding the MLSE solution). This is done by only searching over a limited number of lattice points inside a hyper-sphere around the received point. The search is carried out for a single dimension at a time. To make this feasible a triangular structure of the channel matrix is needed. Hence, in the general case a QR factorization of the channel matrix is necessary. However, in frequency-selective channels, the QR factorization can sometimes be avoided.

One of the important issues, which must be dealt with in SD, is how to choose the radius of the sphere. This is because the radius of the sphere has significant influence on the complexity and the performance. Three methods, which all determine the radius based on the noise statistics, have been described. From the view point of computational efficiency, especially methods using increasing radii throughout the search in the multi-dimensional space are interesting. However, they do not guarantee to find the MLSE solution.

It has been shown that sphere decoding can be combined with e.g. the MAP receiver. Then the SD algorithm can be used to prune away states, which do not fulfil the SD constraint. This can be considered as dynamical pruning of the state space. If the SD algorithm is used in combination with the MAP receiver, it will be possible to obtain approximated posteriors of the symbols.
The order of examining states that fulfil the SD constraint can lead to a reduction in the complexity when only the most probable paths in the trellis diagram are examined. This search method is called the Schnorr-Euchner (SE) search strategy.

Simulation results have been presented for the MAP receiver exploiting the SD algorithm. The simulation results have shown that the BER performance is similar to the one obtained by the MAP receiver. A reduction in complexity is observed at high SNR, but will not be that significant when a static radius is used in SD. Thus, it is recommended to use a increasing radii method instead. Such a method will lead to a considerable reduction in complexity at high SNRs, especially in the case of

EDGE simulations. However, the complexity will still be the same as the optimal receiver at low SNR. This is the greatest disadvantage of the SD algorithm.
A modified Schnorr-Euchner search strategy, where the maximum number of states in the trellis diagram is specified, seems to solve this problem. Hence, the complexity will then be upper bounded. This implies that the complexity at low SNR will be the one specified by the maximum number of allowed states, while the complexity will be reduced further at high SNR. This modified search strategy has proved to give a performance comparable with the optimal receiver as long as the number of states is adapted reasonably with respect to the channel impulse response.
In all the EDGE simulations it has been indicated, that near-optimal BER performance can be achieved with a complexity exponent that is less than $e_c < 3.5$ for the modified SE search strategy. In a HT channel this corresponds to a complexity reduction in the number of examined state transitions, which is more than a factor of 16000 compared with the optimal receiver.
Based on the simulation results, it can be concluded that sphere decoding is an efficient method to obtain near-optimal detection performance, and may be considered to be implemented in a MS.

7.1 Future Research

The following list of research topics could be interesting to examine in the future.

- First of all it would be interesting to evaluate the complexity and the performance in GSM and EDGE when QR factorization of the channel matrix is made. Then it will also be possible to bound the symbols in the case of MUD.

- As described in the simulation results, the pulse shape of the C_0-pulse will have a negative effect on the number states, which are needed in the modified SE search strategy. This is because only a limited amount of energy of the pulse is taken into account when the modified SE search strategy prunes the state space. Thus, it would be interesting to see how pre-filtering with a minimum-phase filter will affect the number of states, which are needed to obtain near-optimal performance.

- It would also be relevant to extend the implemented MAP receiver with SD constraint to be able to do near-optimal detection in channels with Gauss-Markov noise. This is relevant since the interference can be modelled as being Gauss-Markov noise.

- Finally, it would be interesting to examine the performance in situations when the channel is not perfectly known, (i.e. when the channel is estimated). It would also be interesting to examine the performance in other interference scenarios than the one examined in this thesis.

8 References

[1] J. G. Proakis, “Digital Communications”. McGraw-Hill, 4th edition, 2001.

[2] T. S. Rappaport, “Wireless Communications, Principles and Practice”. Prentice Hall, 2nd edition, 2002.

[3] S. Verdu, “Multiuser Detection”. Cambridge University Press, 1st edition, 1998.

[4] E. Seurre, P. Savelli and P. J. Pietri, “EDGE for Mobile Internet”. Artech House, 1st edition, 2001.

[5] P.A. Murphy, “GSM Physical Layer”. July, 1996.
Link: http://www.ece.ucdavis.edu/~murphy/

[6] P. A. Laurent, “Exact and Approximate Construction of Digital Phase Modulations by Superposition of Amplitude Modulation Pulses (AMP)”. IEEE Transactions on Communications, Vol. COM-34, No. 2, February 1986.

[7] R. Booth, “A Note on the Application of the Laurent Decomposition to the GSM GMSK Signaling Waveform”. Tropian, Inc.
Link: http://www.tropian.com/wps/wselaurent1.pdf

[8] L. Nie, “Comparison of Exact and Approximate Multi-User Detection for GSM”. Master’s thesis, Informatics and Mathematical Modelling, DTU, 2005.

[9] T. Arildsen and J. Blauendahl, “Advanced Channel Estimation and Multiuser Detection in GSM”. Master’s thesis, Aalborg University, Institute of Electronic Systems, Department of Communication Technology, 2004

[10] L. R. Rabiner, “A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition”. Proceedings of the IEEE, 77(2):257–286, February 1989.

[11] A. Meng, “An introduction to Markov and Hidden Markov Models”. Note for 02457 Non-Linear Signal Processing, Informatics and Mathematical Modelling, DTU, October 2003.

[12] L. P. B. Christensen, “Minimum Symbol Error Rate Detection in Single-Input Multiple-Output Channels with Markov Noise”. IEEE 6th Workshop on Signal Processing Advances in Wireless Communications, pp. 236–240, June 2005.

[13] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, “Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate”. IEEE Transactions on Information Theory, pp. 284–287, March 1974.

[14] W. Koch and A. Baier, “Optimum and Sub-optimum Detection of Coded Data Disturbed by Time-Varying Intersymbol Interference”. IEEE, pp. 1679–1684, 1990.

[15] J. R. Barry, “The BCJR Algorithm for Optimal Equalization”. Scool of Electrical and Computer Engineering, Georgia Institute of Technology, March 2000.

[16] P. A. Murphy, “Review of MAP State Sequence Estimation, MAP State Estimation, and MAP State Transition Estimation for Finite-Interval, Discrete-Time, Finite-State Markov Processes Observed in Additive White Gaussian Noise with Application to Detection of ISI-corrupted PAM Signals”. Department of Electrical and Computer Engineering, University of California, December 1996.

[17] K. B. Petersen and M.S. Pedersen, “The Matrix Cookbook”. DTU, IMM, October 2005. Available on the internet at: http://www.ece.mcmaster.ca/~reilly/ece712/imm3274.pdf

[18] B. Hassibi and H. Vikalo, “On the Sphere-Decoding Algorithm. I. Expected Complexity”. IEEE Transactions on Signal Processing, pp. 2806-2818, August 2005.

[19] B. Hassibi and H. Vikalo, “On the Sphere-Decoding Algorithm. II. Generalizations, Second-Order Statistics, and Applications to Communications”. IEEE Transactions on Signal Processing, pp. 2819–2834, August 2005.

[20] B. Hassibi, H. Vikalo and U. Mitra, “Sphere-Constrained ML Detection for Frequency-Selective Channels”. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), April 2003.

[21] B. Hassibi, H. Vikalo and U. Mitra, “Sphere-Constrained ML Detection for Channels with Memory”. Conference Record of the Thirty-Seventh Asilomar Conference of Signal, Systems and Computers 2003, pp. 672–676, November 2003.

[22] R. Gowaikar and B. Hassibi, “Statistical Pruning for Maximum Likelihood Decoding”. California Institute of Technology, Department of Electrical Engineering. June, 2003. Available on the internet at: http://www.systems.caltech.edu/EE/Faculty/babak/pubs/sphere.html (Title on web: Efficient Maximum-Likelihood Decoding via Statistical Pruning).

[23] U. Ficke and M. Pohst, “Improved Methods for Calculating Vectors of Short Length in a Lattice, Including a Complexity Analysis”. Mathematics of Computation, pp. 463–471, April 1985.

[24] C. P. Schnorr and M. Euchner, “Lattice Basis Reduction: Improved Practical Algorithms and Solving Subset Sum Problems”. Mathematical Programming, pp. 181–199, 1994.

[25] M. O. Damen, H.E. Gamal and G. Caire, “On Maximum-Likelihood Detection and the Search for the Closest Lattice Point”. IEEE Transactions on Information Theory, pp. 2389–2402, October 2003.

[26] E. Agrell, T. Eriksson, A.Vardy and K. Zeger, “Closest Point Search in Lattices”. IEEE Transactions on Information Theory, pp. 2201–2214, August 2002.

[27] H. Vikalo, B. Hassibi and T. Kailath, “Iterative Decoding for MIMO Channels via Modified Sphere Decoding”. IEEE Transactions on Wireless Communications, pp. 2299–2311, November 2004.

[28] W. Zhao and G.B. Giannakis, “Sphere Decoding Algorithms with Improved Radius Search”. Wireless Communications and Networking Conference, 2004. IEEE WCNC, pp. 2290–2294, March 2004.

[29] R.A. Horn and C.R. Johnson, “Matrix Analysis”. Cambridge University Press, 1985.

[30] S.M. Kay, “Fundamentals of Statistical Signal Processing Detection Theory”. Prentice Hall, 1998.

[31] V. Tarokh, H. Jafarkhani and A.R. Calderbank, “Space-Time Block Codes from Orthogonal Designs”. IEEE Transactions on Information Theory, pp. 1456–1467, July 1999.

[32] M. R. Spiegel and J. Liu, “Mathematical Handbook of Formulas and Tables”. Schaum’s Outline Series, 2nd edition, McGraw-Hill, 1999.

[33] W. H. Gerstacker and R. Schober, “Equalization Concepts for EDGE”. IEEE Transactions on Wireless Communications, pp. 190–199, January 2002.

[34] A. D. Viterbi, “Error Bounds for Convolutional Codes and an Asymptotically Optimum Decoding Algorithms”. IEEE Transactions on Information Theory, pp. 260-269, April 1967.

[35] Description of the Levinson-Durbin algorithm on Wikipedia.
See link at: http://en.wikipedia.org/wiki/Levinson_recursion

[36] W. H. Gerstacker, R. Schober and A. Lampe, “Receivers With Widely Linear Processing for Frequency-Selective Channels. IEEE Transactions on Wireless Communications, pp. 1512–1515, September 2003.

[37] S. Haykin, “Adaptive Filter Theory”. Prentice Hall, 4th edition, 2002.

[38] J. Justensen, and K. Larsen, ”Notes for 34230 Telecommunication”. COM, DTU, 2002.

9 List of Appendices

Appendix	Name	Description
4.1	Derivation of the LMMSE Receiver for SUD	Derives the optimal solution for the LMMSE receiver in the case of SUD.
4.2	Whitening of Signals	Describes a whitening approach originally proposed by [12].
5.1	Boundaries for Symbols that Fulfil the SD Constraint	Describes in more detail the method of searching in circles of the symbols that fulfil the sphere decoding constraint.
5.2	Derivation of the Chi-square Distribution	Derives a generalized expression for the Chi-square distribution when the squared random variables do not have unit variance.
5.3	The inverse Chi-square Distribution	It is shown how the inverse Chi-square distribution can be used to calculate the radius of the sphere based on the DF.

Appendix 4.1. Derivation of the LMMSE Receiver for SUD

In this appendix the expression for the optimum MSE solution is given. The signal models used to derive the LMMSE receiver for SUD are

$$\mathbf{r} = \mathbf{Hs} + \mathbf{z} \qquad \text{and} \qquad \hat{\mathbf{s}} = \mathbf{Mr}\,, \tag{A.4.1.1}$$

where $\mathbf{z}$ is complex Gaussian noise, $\mathcal{CN}(0, \sigma^2\mathbf{I})$. The task is to calculate the matrix $\mathbf{M}$. By using the MSE of the estimated symbols as the optimization criteria, the following quadratic cost function is obtained

$$J = E\left[\ \|\mathbf{s} - \hat{\mathbf{s}}\|^2\ \right] = E\left[\ (\mathbf{s} - \mathbf{Mr})^H(\mathbf{s} - \mathbf{Mr})\right] \tag{A.4.1.2}$$

Before the cost function is set equal to zero, it is convenient to rewrite (A.4.1.2).

$$\begin{aligned} J &= E\left[\ (\mathbf{s} - \mathbf{Mr})^H(\mathbf{s} - \mathbf{Mr})\ \right] = E\left[\ (\mathbf{s}^H - \mathbf{r}^H\mathbf{M}^H)(\mathbf{s} - \mathbf{Mr})\ \right] \\ &= E\left[\ \mathbf{s}^H\mathbf{s} - \mathbf{s}^H\mathbf{Mr} - \mathbf{r}^H\mathbf{M}^H\mathbf{s} + \mathbf{r}^H\mathbf{M}^H\mathbf{Mr}\ \right] \\ &= E\left[\ \mathbf{s}^H\mathbf{s}\ \right] - 2E\left[\ \Re\{\mathbf{s}^H\mathbf{Mr}\}\ \right] + E\left[\ \mathbf{r}^H\mathbf{M}^H\mathbf{Mr}\ \right] \end{aligned} \tag{A.4.1.3}$$

Since J is a scalar and thereby all the expectation terms are scalars, it is possible to take the trace of the matrix,

$$\begin{aligned} J &= E\left[\ \mathbf{s}^H\mathbf{s}\ \right] - 2Tr\left(E\left[\ \Re\{\mathbf{s}^H\mathbf{Mr}\}\ \right]\right) + Tr\left(E\left[\ \mathbf{r}^H\mathbf{M}^H\mathbf{Mr}\ \right]\right) \\ &= E\left[\ \mathbf{s}^H\mathbf{s}\ \right] - 2Tr\left(\mathbf{M}E\left[\ \Re\{\mathbf{rs}^H\}\ \right]\right) + Tr\left(\mathbf{M}E\left[\mathbf{rr}^H\ \right]\mathbf{M}^H\right). \end{aligned} \tag{A.4.1.5}$$

The transmitted symbols, $\mathbf{s}$, are assumed to be uncorrelated (due to interleaving), which means that

$$E\left[\ \mathbf{s}^H\mathbf{s}\ \right] = \mathbf{I}\,. \tag{A.4.1.6}$$

By assuming that the noise and the symbols are uncorrelated and using (A.4.1.6) a simplification of $E\left[\ \mathbf{rr}^H\ \right]$ can be found,

$$\begin{aligned} E[\mathbf{rr}^H] &= E\left[\ (\mathbf{Hs} + \mathbf{z})(\mathbf{Hs} + \mathbf{z})^H\ \right] = E\left[\ (\mathbf{Hs} + \mathbf{z})(\mathbf{s}^H\mathbf{H}^H + \mathbf{z}^H)\right] \\ &= E\left[\ \mathbf{Hss}^H\mathbf{H}^H\ \right] + E\left[\ \mathbf{zz}^H\ \right] = E\left[\ \mathbf{HH}^H\ \right] + \sigma^2\mathbf{I}. \end{aligned} \tag{A.4.1.7}$$

Using the same assumptions $E\left[\ \mathbf{rs}^H\ \right]$ can be simplified to

$$E\left[\ \mathbf{rs}^H\ \right] = E\left[\ (\mathbf{Hs} + \mathbf{z})\mathbf{s}^H\ \right] = E\left[\ \mathbf{Hss}^H\ \right] = \mathbf{H} \tag{A.1.8}$$

The simplified expressions of (A.4.1.6), (A.4.1.7), and (A.4.1.8) are now inserted into (A.4.1.5). This gives

$$J = E\left[\ \mathbf{s}^H\mathbf{s}\right] - 2Tr\left(\mathbf{M}\Re\{\mathbf{H}\}\right) + Tr\left(\mathbf{M}\left(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}\right)\mathbf{M}^H\right) \tag{A.1.9}$$

The optimal solution of $\mathbf{M}$ can be found by taking the derivative of the cost function,

$$\begin{aligned}\frac{\partial J}{\partial \mathbf{M}} &= \frac{-\partial 2Tr\left(\mathbf{M}\Re\{\mathbf{H}\}\right)}{\partial \mathbf{M}} + \frac{\partial Tr\left(\mathbf{M}\left(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}\right)\mathbf{M}^H\right)}{\partial \mathbf{M}} \\ &= -2\mathbf{H}^H + \mathbf{M}\left(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}\right)^H + \mathbf{M}\left(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}\right) \\ &= -2\mathbf{H}^H + 2\mathbf{M}\left(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}\right)\end{aligned} \tag{A.4.1.10}$$

and setting it equal to zero,

$$\mathbf{M} = \mathbf{H}^H\left(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}\right)^{-1}. \tag{A.4.1.11}$$

By expressing (A.4.1.11) as

$$\mathbf{M} = \mathbf{H}^H\left(\mathbf{H}\mathbf{H}^H + \sigma^2\mathbf{I}\right)^{-1} = \mathbf{I}\mathbf{H}^H\left(\mathbf{H}\mathbf{I}\mathbf{H}^H + \sigma^2\mathbf{I}\right)^{-1}, \tag{A.4.1.12}$$

it is possible to use relation 3.2.1 in [17], since $\mathbf{I}$ is positive definite. Hence,

$$\mathbf{I}\mathbf{H}^H\left(\mathbf{H}\mathbf{I}\mathbf{H}^H + \sigma^2\mathbf{I}\right)^{-1} = \left(\sigma^2\mathbf{I}^{-1} + \mathbf{H}^H\mathbf{I}^{-1}\mathbf{H}\right)^{-1}\mathbf{H}^H\mathbf{I}^{-1} \tag{A.4.1.13}$$

In that way the optimal solution can also be expressed as

$$\mathbf{M} = \left(\mathbf{H}^H\mathbf{H} + \sigma^2\mathbf{I}\right)^{-1}\mathbf{H}^H. \tag{A.4.1.15}$$

Appendix 4.2. Whitening of Signals

This Appendix briefly describes the whitening approach used in some of the simulations. The whitening approach is proposed by L. P. B. Christensen in [12] and the equations in the Appendix comes from [12] too.

In [12] it is described how interference from other users can be approximated by Markov noise. The reason why the noise term of the interfering users can not be assumed to be AWGN, is because the signal path has memory. Thus, a performance gain can be expected in interference scenarios, by assuming that the noise is Markov noise instead of just being AWGN.

In Section 4.2 it is shown that the optimal Symbol-by-Symbol detection in the AWGN case is obtained by using the forward-backward algorithm. Recall, that it is only the length of the channel impulse response which has influence on the number of states in the Markov model. However, this is not the case for Markov noise. Now the noise process will also be coupled in time and, thus the noise will have influence on the size of the Markov model. If the noise has a memory length of N_m, it is shown in [12] that the total number of states needed in the Markov model will be $|\Omega|^{L-1+N_m}$, where L is the length of the channel. In [12] it is shown that the expression of the observation probability is given as

$$P\left(\mathbf{r}_{n-1}^{n-1}\middle|\mathbf{r}_{n-1-N_M}^{n-2},\mathbf{a}_n(q),\mathbf{a}_{n-1}(p)\right)=\frac{P\left(\mathbf{r}_{n-1-N_m}^{n-1}\middle|\mathbf{a}_n(q),\mathbf{a}_{n-1}(p)\right)}{P\left(\mathbf{r}_{n-1-N_m}^{n-2}\middle|\mathbf{a}_n(q),\mathbf{a}_{n-1}(p)\right)}, \tag{A.4.2.1}$$

where $\mathbf{a}_n(q)$ denotes the *q'th* state in the Markov model and $\mathbf{r}_{n_1}^{n_2}$ is the received signal stacked in the same way as in Section 4.2.
It is shown in [12] that when the noise is assumed to be zero-mean Gaussian, only the covariance matrix of the noise is needed in order to find the probability in (A.4.2.1)
Let the noise samples, $\mathbf{z}_{n_1}^{n_2}$, be stacked in the same way as the received signal, $\mathbf{r}_{n_1}^{n_2}$. When the upper part of (A.4.2.1) should be evaluated the following covariance matrix is needed

$$\mathbf{\Sigma}=E\left[\mathbf{z}_{n-N_m}^{n}\left(\mathbf{z}_{n-N_m}^{n}\right)^H\right]. \tag{A.4.2.2}$$

Similarly, to determine the lower part of (A.4.2.1) the covariance matrix

$$\breve{\mathbf{\Sigma}}=E\left[\mathbf{z}_{n-N_m}^{n-1}\left(\mathbf{z}_{n-N_m}^{n-1}\right)^H\right], \tag{A.4.2.3}$$

is needed. Observe that the latter covariance matrix is included in (A.4.2.2). By defining

$$\mathbf{W}=\mathbf{\Sigma}^{-1}-\begin{bmatrix}\breve{\mathbf{\Sigma}}^{-1} & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix} \tag{A.4.2.4}$$

it will be possible to reformulate (A.4.2.1) to

$$\begin{aligned}&-2\ln\left(P\left(\mathbf{r}_{n-1}^{n-1}\middle|\mathbf{r}_{n-1-N_m}^{n-2},\mathbf{a}_n(q),\mathbf{a}_{n-1}(p)\right)\right)+C\\&\quad=\left(\mathbf{r}_{n-1-N_m}^{n-1}-\hat{\mathbf{r}}_{n-1-N_m}^{n-1}(p,q)\right)^H\mathbf{W}\left(\mathbf{r}_{n-1-N_m}^{n-1}-\hat{\mathbf{r}}_{n-1-N_m}^{n-1}(p,q)\right).\end{aligned} \tag{A.4.2.5}$$

Here $\hat{\mathbf{r}}_{n-1-N_m}^{n-1}(p,q)$ is the expected output from the state transition from $\mathbf{a}_{n-1}(p)$ to $\mathbf{a}_n(q)$ and C is a normalization constant. To see the exact definition of C, [12] can be consulted. It is proved in [12] that the rank of $\mathbf{W}$ is not full and therefore it is possible to express $\mathbf{W}$ as two vector outer products. Hence,

$$\mathbf{W} = \mathbf{F}\mathbf{F}^H. \tag{A.4.2.6}$$

By substituting (A.4.2.6) into (A.4.2.6) the expression becomes

$$\begin{aligned} &-2\ln\left(P\left(\mathbf{r}_{n-1}^{n-1}\middle|\mathbf{r}_{n-1-N_m}^{n-2},\mathbf{a}_n(q),\mathbf{a}_{n-1}(p)\right)\right)+C \\ &\quad=\left(\mathbf{r}_{n-1-N_m}^{n-1}-\hat{\mathbf{r}}_{n-1-N_m}^{n-1}(p,q)\right)^H\mathbf{F}\mathbf{F}^H\left(\mathbf{r}_{n-1-N_m}^{n-1}-\hat{\mathbf{r}}_{n-1-N_m}^{n-1}(p,q)\right) \\ &\quad=\left\|\mathbf{F}^H\left(\mathbf{r}_{n-1-N_m}^{n-1}-\hat{\mathbf{r}}_{n-1-N_m}^{n-1}(p,q)\right)\right\|^2. \end{aligned} \tag{A.4.2.7}$$

From (A.4.2.7) it is seen that the noise can be whitened by using the whitening filter $\mathbf{F}^H$.
Let the whitened received signal be

$$\tilde{\mathbf{r}}_{n-1}^{n-1} = \mathbf{F}^H\mathbf{r}_{n-1}^{n-1}, \tag{A.4.2.8}$$

and let the expected output from a specific state transition be

$$\tilde{\hat{\mathbf{r}}}_{n-1}^{n-1}(p,q) = \mathbf{F}^H\hat{\mathbf{r}}_{n-1}^{n-1}(p,q). \tag{A.4.2.9}$$

By using the definitions in (A.4.2.8) and (A.4.2.9) it is possible to express (A.4.2.7) as

$$-2\ln\left(P\left(\mathbf{r}_{n-1}^{n-1}\middle|\mathbf{r}_{n-1-N_M}^{n-2},\mathbf{a}_n(q),\mathbf{a}_{n-1}(p)\right)\right)+C=\left\|\tilde{\mathbf{r}}_{n-1-N_m}^{n-1}-\tilde{\hat{\mathbf{r}}}_{n-1-N_m}^{n-1}(p,q)\right\|^2. \tag{A.4.2.10}$$

From (A.4.2.10) it is realized that the problem has now been transformed into the traditional detection problem when AWGN is present. However, the channel length is now $L+N_m$. Thus, the complexity of the receiver will be $O(N_R|\Omega|^{L+Nm})$.

Appendix 5.1. Boundaries for Symbols that Fulfil the SD Constraint

In this Appendix it is described in more detail how the symbols that fulfil the Sphere Decoding (SD) constraint are found. In Section 5.2 it is shown that this is equivalent with finding the symbols that lie inside the circle with the parameters

$$\begin{aligned} &\textit{Squared radius}, \widetilde{R}^2 = \frac{1}{\|\mathbf{h}\|^2}\left(\widetilde{r}_n^2 - \sum_{i=2}^{N_{sps}-1}\left|\mathbf{q}_i^H \widetilde{\mathbf{x}}_n\right|^2\right) \\ &\textit{Centre}, \widetilde{C}\left(\Re\left\{\frac{\hat{\mathbf{h}}^H \widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|}\right\}; \Im\left\{\frac{\hat{\mathbf{h}}^H \widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|}\right\}\right). \end{aligned} \tag{A.5.1.1}$$

The general formula for circle with centre in $\widetilde{C}(a,b)$ and radius $\widetilde{R}$ is given as

$$(x-a)^2 + (y-b)^2 = \widetilde{R}^2, \tag{A.5.1.2}$$

where x and y is real-valued numbers and where

$$a = \Re\left\{\frac{\hat{\mathbf{h}}^H \widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|}\right\} \quad \text{and} \quad b = \Im\left\{\frac{\hat{\mathbf{h}}^H \widetilde{\mathbf{x}}_n}{\|\mathbf{h}\|}\right\} \tag{A.5.1.3}$$

Figure A.5.1.1. shows an illustration of this.

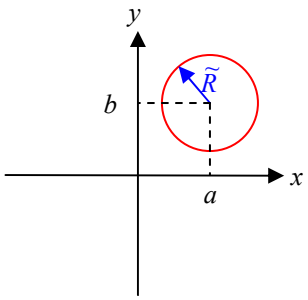

Figure A.5.1.1. Illustration of circle parameters.

GSM symbols

It is quite simple to determine the interval of the symbols in the case of a GSM signal, since it is only the intersections between the circle in (A.5.1.1) and the real axis that should be found. To solve this problem the circle in (A.5.1.2) becomes

$$\begin{aligned} (x-a)^2 + (0-b)^2 &= \widetilde{R}^2 \\ x^2 - 2ax + a^2 + b^2 &= \widetilde{R}^2, \end{aligned} \tag{A.5.1.4}$$

Therefore, the intersections with the real axis are found by solving the quadratic formula in (A.5.1.4). The symbols that lie in the intersection interval will fulfil the SD constraint.

EDGE symbols

When it should be determined which EDGE symbols that fulfil the SD constraint, the intersections of two circles are examined. The EDGE symbols lie on the unit circle and therefore, they lie on the circle

$$x^2 + y^2 = 1. \tag{A.5.1.5}$$

When the intersections of the two circles have been calculated the angles of these intersections are found. Thus, the angle interval that fulfils the SD constraint can be found and therefore also the symbols.

By using (A.5.1.2) and (A.5.1.3) two equations with two unknowns are obtained. (A. 5.1.2) can be rewritten to

$$x^2 - 2ax + a^2 + y^2 - 2by + b^2 = \widetilde{R}^2. \tag{A.5.1.6}$$

By having (A.5.1.5) in mind, (A.5.1.6) is equivalent with

$$\begin{aligned} 1 - 2ax + a^2 - 2by + b^2 &= \widetilde{R}^2 \\ -2ax - 2by &= \widetilde{R}^2 - \left(1 + a^2 + b^2\right) \end{aligned} \tag{A.5.1.7}$$

If a ≠ 0

If it is assumed that $a \neq 0$, (A.5.1.7) can be reduced to

$$x = \frac{1 + a^2 + b^2 - \widetilde{R}^2}{2a} - \frac{b}{a}y = K_1 - K_2 y, \tag{A.5.1.8}$$

where $K_1 = \frac{1 + a^2 + b^2 - \widetilde{R}^2}{2a}$ *and* $K_2 = \frac{b}{a}$. The expression for *x* in (A.5.1.8) is now inserted into (A.5.1.5), which gives

$$\begin{aligned} \left(K_1 - K_2 y\right)^2 + y^2 &= 1 \\ K_1^2 - 2K_1K_2 y + K_2^2 y^2 + y^2 &= 1 \\ \left(K_2^2 + 1\right)y^2 + \left(-2K_1K_2\right)y + \left(K_1^2 - 1\right) &= 0. \end{aligned} \tag{A.5.1.9}$$

It is now clear that the potential intersections of the two circles are found by solving the quadratic formula in (A.5.1.9). If the discriminant,

$$D = \left(-2K_1K_2\right)^2 - 4\left(K_2^2 + 1\right)\left(K_1^2 - 1\right), \tag{A.5.1.10}$$

is negative, there is no intersection of the circles. If *D* = 0, there will only be one intersection. In that case it should be checked if the unit circle lie inside the circle that represents the SD solutions. This is illustrated in Figure A.5.1.2. From a theoretical point of view, another special case can also appear when D = 0, since the intersection could be exactly in an EDGE symbol. However, it can be discussed if resources should be used to check this scenario, since the solution will be in the boundary of the interval. Furthermore, in practice this case will seldom occur.

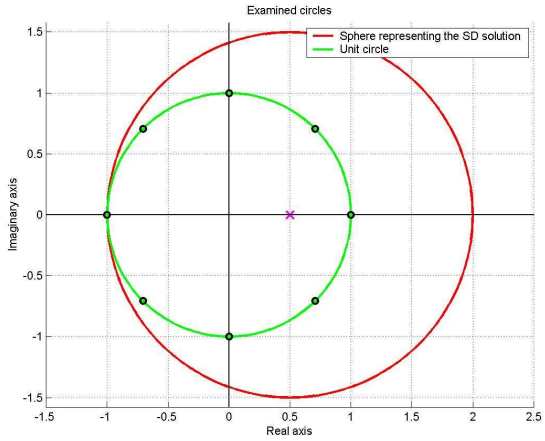

Figure A.5.1.2. Illustration of case where there is only one intersection and all the symbols fulfil the SD constraint.

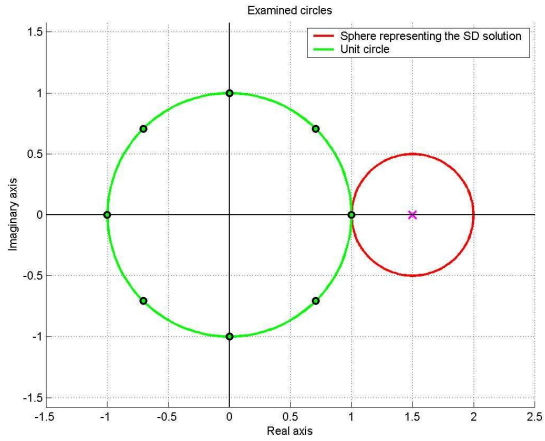

Figure A.5.1.3. Illustration of case where there is only one symbol that fulfil the SD constraint.

If a = 0

When a is in fact zero (or very close to zero) Equation (A.5.1.7) is simplified to

$$\begin{aligned} -2by &= \widetilde{R}^2 - \left(1+b^2\right) \\ by &= \frac{1+b^2-\widetilde{R}^2}{2} \end{aligned} \qquad \text{(A.5.1.11)}$$

If $b \neq 0$ the solution of (A.5.1.11) is

$$y = \frac{1+b^2-\widetilde{R}^2}{2b}. \qquad \text{(A.5.1.12)}$$

If b = 0 (or very close to zero) there will be a special case, where circle representing the SD solutions will have centre in (0,0). Therefore, it is easy to check if the circle contains the EDGE symbols by looking at the radius, $\widetilde{R}$.
As just mentioned in the brackets above, numerical values of a and b which are close to zero should be treated as being zero, duo to the fact that there are limited precision available in the computer. Hence, it is recommended that absolute values of a and b below e.g. 10^{-3} are treated as zero.

Appendix 5.2. Derivation of the Chi-square Distribution

The Chi-square distribution is well described in many books concerning statistic, see e.g. [30]. The Chi-square distribution stem from a summation of the squared Gaussian random variables, $\tilde{y}_i$, with unit variance. Hence,

$$\tilde{x} = \sum_{i=1}^{\nu} \tilde{y}_i^2 , \quad where \quad \tilde{y}_i \sim N(0,1) , \tag{A.5.2.1}$$

here ν is the degrees of freedom and $\tilde{x}$ is the random variable that is Chi-square distributed. The probability density function (PDF) of the Chi-square distribution with ν degrees of freedom is given as

$$p(\tilde{x}) = \frac{1}{2^{\nu/2}\Gamma\left(\frac{\nu}{2}\right)} \tilde{x}^{(\nu/2-1)} \cdot \exp\left(-\frac{1}{2}\tilde{x}\right), \tag{A.5.2.2}$$

where Γ(·) is the gamma function. It is desired to find an expression for the Chi-square distribution when the squared random variables, y_i, do not have unit variance. Hence, the case where the random variables has variance σ^2 per dimension i.e.

$$x = \sum_{i=1}^{\nu} y_i^2 , \quad where \quad y_i \sim N(0, \sigma^2) \tag{A.5.2.3}$$

The more generalized expression of the Chi-square distribution can be found by substituting $\tilde{x}$ in (A.5.2.2) with $\left(x/\sigma^2\right)$. However, it should still be ensured that the cumulative density function (CDF) of this new distribution goes towards one. Therefore, a normalization constant Z is introduced in the beginning. The PDF is given as

$$p(x) = \frac{1}{Z} \frac{1}{2^{\nu/2}\Gamma\left(\nu/2\right)} \left(\frac{x}{\sigma^2}\right)^{(\nu/2-1)} \cdot \exp\left(-\frac{1}{2}\frac{x}{\sigma^2}\right), \tag{A.5.2.4}$$

and the normalization constant, Z, should then be determined. Equation (A.5.2.4) can be rewritten to

$$\begin{aligned} p(x) &= \frac{1}{Z} \frac{1}{2 \cdot 2^{(\nu/2-1)}\Gamma\left(\nu/2\right)} \cdot \left(\frac{1}{\left(\sigma^2\right)^{(\nu/2-1)}}\right) \cdot (x)^{(\nu/2-1)} \cdot \exp\left(-\frac{1}{2}\frac{x}{\sigma^2}\right) \\ &= \frac{1}{Z} \frac{1}{2 \cdot \left(2\sigma^2\right)^{(\nu/2-1)}\Gamma\left(\nu/2\right)} \cdot (x)^{(\nu/2-1)} \cdot \exp\left(-\frac{1}{2}\frac{x}{\sigma^2}\right) \end{aligned} \tag{A.5.2.5}$$

The normalization constant is determined by ensuring that the CDF $\to 1$ for $x \to \infty$, i.e.

$$1 = \int_0^{\infty} p(x)\,dx = \frac{1}{Z} \frac{1}{2 \cdot \left(2\sigma^2\right)^{(\nu/2-1)}\Gamma\left(\nu/2\right)} \cdot \int_0^{\infty} (x)^{(\nu/2-1)} \cdot \exp\left(-\frac{1}{2}\frac{x}{\sigma^2}\right) dx \tag{A.5.2.6}$$

In [32, Eq. (18.76)] it is shown that the integral is equal to

$$\int_0^\infty (x)^{(\nu/2-1)} \cdot \exp\left(-\frac{1}{2}\frac{x}{\sigma^2}\right) dx = \frac{\Gamma\left(\nu/2\right)}{\left(\frac{1}{2\sigma^2}\right)^{(\nu/2)}} \tag{A.5.2.7}$$

The result in (A.5.2.7) is substituted into (A.5.2.6) and we then get

$$1 = \frac{1}{Z}\frac{1}{2\cdot\left(2\sigma^2\right)^{(\nu/2-1)}\Gamma\left(\nu/2\right)} \cdot \frac{\Gamma\left(\nu/2\right)}{\left(\frac{1}{2\sigma^2}\right)^{(\nu/2)}} = \frac{1}{Z}\frac{1}{\left(1/\sigma^2\right)\cdot\left(2\sigma^2\right)^{(\nu/2)}\left(\frac{1}{2\sigma^2}\right)^{(\nu/2)}} = \frac{1}{Z}\sigma^2, \tag{A.5.2.8}$$

and, therefore, the normalization constant must be $Z = \sigma^2$. Hence, from (A.5.2.5) it is seen that the PDF of the Chi-square distribution is

$$\begin{aligned} p(x) &= \frac{1}{\sigma^2}\frac{1}{2\cdot\left(2\sigma^2\right)^{(\nu/2-1)}\Gamma\left(\nu/2\right)} \cdot (x)^{(\nu/2-1)} \cdot \exp\left(-\frac{1}{2}\frac{x}{\sigma^2}\right) \\ &= \frac{1}{\left(2\sigma^2\right)^{(\nu/2)}\Gamma\left(\nu/2\right)} \cdot (x)^{(\nu/2-1)} \cdot \exp\left(-\frac{1}{2}\frac{x}{\sigma^2}\right) \end{aligned} \tag{A.5.2.9}$$

Appendix 5.3. The Inverse Chi-square Distribution

This Appendix shows how the inverse of Chi-square Distribution can be used to evaluate an integral over a Chi-square Distribution when the squared random variables do not have unit variance. This is done so the MATLAB-function '*chi2inv.m*' can be used directly to compute the integral.

The Inverse Chi-square Distribution in MATLAB is defined as

$$1-\varepsilon = \int_0^{\beta} \frac{1}{2^{(\nu/2)}\Gamma\left(\nu/2\right)} \cdot t^{(\nu/2-1)} \cdot e^{(-t/2)}\, dt\,. \tag{A.5.3.1}$$

In MATLAB the function call that corresponds to (A.5.3.1) would be

$$\beta = chi2inv\left(1-\varepsilon, \nu\right). \tag{A.5.3.2}$$

From (A.5.3.1) it is seen that the function assumes that the Chi-square Distribution stem from a summation of the squared random variables with unit variance. Since, this is not the case for the Chi-square Distribution which is used to determine the radius of the sphere, a correspondence needs to be established.

The integral that should be evaluated is given as

$$1-\varepsilon = \int_0^{r^2} \frac{1}{\left(2\sigma^2\right)^{(\nu/2)}\Gamma\left(\nu/2\right)} \cdot \lambda^{(\nu/2-1)} \cdot e^{\left(-\frac{\lambda}{2\sigma^2}\right)}\, d\lambda\,. \tag{A.5.3.3}$$

By substitution with the variable $\alpha = \dfrac{\lambda}{\sigma^2} \Rightarrow d\lambda = \sigma^2 d\alpha$, (A.5.3.2) becomes

$$\begin{aligned}
1-\varepsilon &= \int_0^{r^2/\sigma^2} \frac{1}{\left(2\sigma^2\right)^{(\nu/2)}\Gamma\left(\nu/2\right)} \cdot \left(\sigma^2\alpha\right)^{(\nu/2-1)} \cdot e^{\left(-\frac{\alpha}{2}\right)} \cdot \sigma^2\, d\alpha \\
&= \int_0^{r^2/\sigma^2} \frac{\sigma^2 \cdot \left(\sigma^2\right)^{(\nu/2-1)}}{\left(2\sigma^2\right)^{(\nu/2)}\Gamma\left(\nu/2\right)} \cdot \alpha^{(\nu/2-1)} \cdot e^{\left(-\frac{\alpha}{2}\right)}\, d\alpha \\
&= \int_0^{r^2/\sigma^2} \frac{\left(\sigma^2\right)^{(\nu/2)}}{2^{(\nu/2)} \cdot \left(\sigma^2\right)^{(\nu/2)}\Gamma\left(\nu/2\right)} \cdot \alpha^{(\nu/2-1)} \cdot e^{\left(-\frac{\alpha}{2}\right)}\, d\alpha \\
&= \int_0^{r^2/\sigma^2} \frac{1}{2^{(\nu/2)} \cdot \Gamma\left(\nu/2\right)} \cdot \alpha^{(\nu/2-1)} \cdot e^{\left(-\frac{\alpha}{2}\right)}\, d\alpha
\end{aligned} \tag{A.5.3.4}$$

It is now possible to see that when the function '*chi2inv.m*' called with the arguments: (1-ε, ν), as shown in (A.5.3.2) the output β only needs to be scaled with the variance to find the radius. Hence

$$\beta = \frac{r^2}{\sigma^2} \Leftrightarrow r^2 = \sigma^2 \cdot \beta\,. \tag{A.5.3.5}$$