Technische Universiteit Eindhoven

Faculty of Electrical Engineering
Section of Telecommunication Technology and Electromagnetics

Thesis

Extending the Capacity of Next Generation Wireless LANs Using Space Division Multiplexing combined with OFDM

A. van Zelst

Coaches: Dr.Ir. P.F.M. Smulders (TUE) and Dr.Ir. R.D.J. van Nee (Lucent Technologies)
Supervisor: Prof.Dr.Ir. G. Brussaard
Period: January – October 1999

Abstract

With the increasing amount of bandwidth-hungry services like video on demand and video conferencing, the need for broadband communication systems is obvious. Furthermore, with the growing demand for flexible workplaces at the office (office hotelling) and university, wireless broadband communication systems are no luxury anymore.

The main goals in developing new wireless communication systems are increasing the bit rate and increasing system capacities. Because the available frequency spectrum is limited, this means that future systems should be characterised by improved spectrum efficiency. Recent information theory research has revealed that the multipath wireless channel is capable of enormous capacities, provided that the multipath scattering is sufficiently rich and is properly exploited. A possible way to exploit the multipath scattering properly is Space Division Multiplexing (SDM) or Space Division Multiple Access (SDMA). Basically, these techniques transmit different signals simultaneously on different transmit antennas. The parallel streams of data are mixed-up in the air, but can be recovered at the receiver by using the SDM algorithms proposed in this report, like the Zero Forcing or Minimum Mean Square Error method, with or without Decision Feedback Decoding, or the Maximum Likelihood Decoding (MLD) technique. When the multipath is exploited properly, both the data rate and the Signal-to-Noise Ratio performance can be increased. For BLAST (Bell laboratories LAyered Space Time), a system based on Minimum Mean Square Error decoding with Decision Feedback, it has be shown that spectrum efficiencies of 20 - 40 Bit/s/Hz can be achieved in an indoor propagation environment at realistic Signal-to-Noise Ratios and error ratios.

In this report it is shown that Maximum Likelihood Decoding (based on detection) achieves a diversity order equal to the number of receive antennas. This is (much) better than the diversity order of Zero Forcing for which it is shown in literature that the diversity order equals the number of receive antennas minus the number of transmit antennas plus one. The big disadvantage of MLD is that its complexity grows exponentially with the number of transmit antennas. For a reasonable number of transmit antennas (≤ 5), however, its complexity is shown to be comparable with other Space Division Multiplexing methods.

Furthermore, in this thesis, single carrier Space Division Multiplexing based on MLD is combined with the multicarrier technique Orthogonal Frequency Division Multiplexing (OFDM) to make the system more robust against delay spread impairments. A basic introduction to OFDM is given and it is described how to combine SDM with OFDM. This system is simulated in C++ using the parameters of the new IEEE 802.11 OFDM standard for wireless LANs. The delay spread channel is modelled by exponentially-decayed Rayleigh fading. By performing Monte Carlo simulations, Bit Error Rate performances for different antenna configurations and different delay spreads are shown in this report.

To achieve a better performance, forward error correction coding based on a convolutional code is implemented. However, if a soft-decision input Viterbi decoder, which should get soft-decision values from the receiver detection algorithm, is used to decode this convolutional code, an even better performance is obtained. Therefore, the MLD algorithm is adapted in order to produce soft-decision outputs. A simple intuitive adaptation and a complex adaptation based on the log-likelihood ratio are proposed and their performance is compared. From all the results presented in this report, it can be concluded that Space Division Multiplexing in combination with OFDM is a promising solution for increasing the bit rate and system capacity.

Table of contents

Abbreviations

AAA	Adaptive Antenna Array
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
BLAST	Bell Laboratories Layered Space-Time
C_ADD	Complex Addition
C_MUL	Complex Multiplication
CIR	Channel Impulse Response
DFB	Decision Feedback
DFT	Discrete Fourier Transform
FFT	Fast Fourier Transform
ICI	Inter Carrier Interference
IDFT	Inverse Discrete Fourier Transform
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
i.i.d.	independently and identically distributed
ISI	Inter Symbol Interference
LAN	Local Area Network
LNA	Low Noise Amplifier
MAC	Medium Access Control
MAP	Maximum a-posteriori Probability
MAPU	Multi-Antenna Processing Unit
MMSE	Minimum Mean Square Error
MLD	Maximum Likelihood Decoding
MRC	Maximal Ratio Combining
MSE	Mean Square Error
OD	Optimal Detection
OFDM	Orthogonal Frequency Division Multiplexing
PDF	Power Delay Profile
PHY	Physical Layer
PSK	Phase Shift Keying
PER	Packet Error Rate
R_ADD	Real Addition
R_MUL	Real Multiplication
RMS	Root Mean Square
RX	Receiving
SDM	Space Division Multiplexing
SDMA	Space Division Multiple Access
SNR	Signal-to-Noise Ratio
TX	Transmitting
QAM	Quadrature Amplitude Modulation
QPSK	Quadrature Phase Shift Keying
WLAN	Wireless Local Area Network
ZF	Zero Forcing

1 Introduction

It is a well-known fact that the amount of information transported over communication systems grows rapidly. Not only the file sizes increase, but also bandwidth-hungry applications such as video on demand and video conferencing require increasing data rates to transfer the information in a reasonable amount of time or to establish real-time connections. To support this kind of services, broadband communication systems are required.

Another well-known fact is that the mobility of communication systems is more and more demanded. One example is the enormous increase of mobile phone users worldwide. Another example is the trend of the work place becoming increasingly mobile. A term often used in this context is office hotelling. Hotelling is a form of alternative officing in which employees who work out of the office for significant periods of time can call ahead (just as they do in making hotel reservations) and reserve workspace. They select an office from a specially-designated block of workspaces when they come into the company's office facilities. This allows organisations to reduce space costs by relying on technology to create the appearance of a permanent office. Ideally, the employee should be completely mobile within the office, and still be able to connect his personal computer to the computer network. Thus, a wireless computer network is desirable.

Another place where wireless computer networking is in demand is at the university campus. A growing number of universities offer their students the possibility to buy a laptop from university. The way of teaching is adapted to the availability of laptops. To make this kind of studying efficient, students need to be able to connect to the computer network of the university in order to download the study material that the professor offers. However, it is undoable to adapt the computer network infrastructure in such a way that the students can plug-in in every (lecture-)room at the university. Here too, it seems that wireless network access is essential. As an example where a wireless network already is installed, the Carnegie Mellon University in Pittsburgh can be mentioned [21].

The above considerations justify research into new broadband wireless communication systems. Large-scale penetration of such systems into our daily lives will require significant reductions in cost and increases in bit rate and/or system capacity. Recent information theoretical studies have revealed that the multipath wireless channel is capable of huge capacities, provided that multipath scattering (see Figure 1-1) is sufficiently rich and is properly exploited through the use of the spatial dimension ([15] and [37]). Appropriate solutions for exploiting the multipath properly, could be based on new techniques that recently appeared in literature, which are based on Space Division Multiplexing (SDM). Basically, these techniques transmit different datastreams on different transmit antennas simultaneously. By designing an appropriate processing architecture to handle these parallel streams of data, the data rate and/or the Signal-to-Noise Ratio (SNR) performance can be increased.

In this thesis, a number of Space Division Multiplexing algorithms will be compared. The one with the best performance will be combined with a spectrally efficient transmission technique called Orthogonal Frequency Division Multiplexing (OFDM) to avoid Inter Symbol Interference (ISI).

1.1 Receiver diversity techniques

Before SDM techniques were developed, receiver diversity was seen as a powerful communication receiver technique that provided wireless link improvement at relatively low cost [36]. Diversity techniques try to solve the problem that transmission errors occur when the transmission channel attenuation is large, i.e., when the channel is in a deep fade. Diversity exploits the random nature of

radio propagation by finding independent (or at least highly uncorrelated) signal channels for communication. This concept can be explained in a simple way; if one radio channel undergoes a deep fade, another independent channel may have a strong signal. By having multiple independent channels to select from (or to combine), the Bit Error Rate performance in the presence of noise is improved.

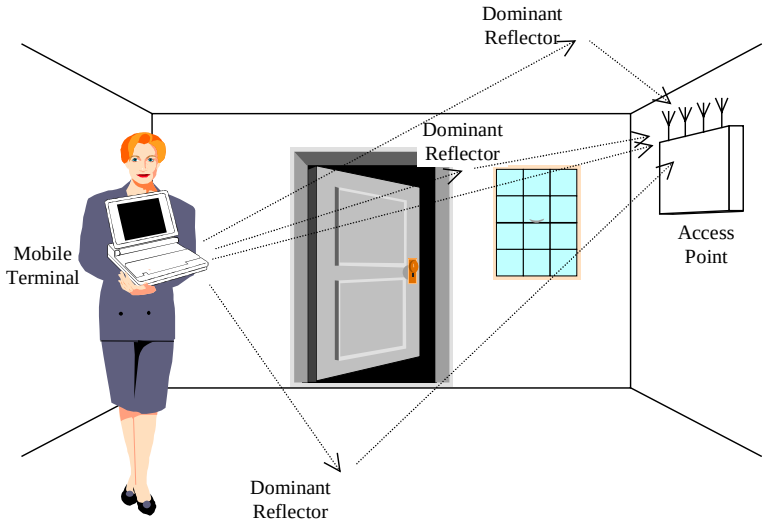

Figure 1-1: multipath fading.

There are several ways in which the receiver can be provided with a number of independently fading channels (say L) that bear the same information signal ([34] and [36]):

- **Frequency diversity**. The same information-bearing signal is transmitted on L different frequencies. At the receiver, the carrier frequency that has suffered least from fading is used to estimate the transmitted signal, or the carrier frequencies are weighted (according to the fading they have experienced) and combined and this combination of signals is used to recover the transmitted signal.

- **Time diversity**. Time diversity transmits the signal in L different time slots to achieve L independently fading versions of the same information-bearing signal. The time spacing between two successive time slots must exceed the coherence time of the channel, so that multiple repetitions of the signal will be received with independent fading conditions.

On one hand the above techniques requires only a single antenna and a single receiver. On the other hand their performance is inferior to techniques which use multiple antennas and radios, which will be described below. This can be explained by the fact that frequency diversity uses an L times larger bandwidth to transmit and time diversity uses L times more time slots to transmit a single data symbol. Thus, the efficiency for both diversity types is reduced by a factor L compared to the techniques described below.

- **Polarisation diversity**. At the transmitter, the same information signal can be transmitted on two antennas, which polarise the radio wave differently. Due to the fact that the reflection coefficient for each polarisation is different (which results in different amplitudes and phases for each reflection), the measured horizontal and vertical polarisation paths between a mobile terminal and an access point are assumed uncorrelated. At the receiver, the horizontal and vertical polarised waves can be combined. While this only provides two diversity branches (L = 2) it does allow the antenna elements to be co-located (i.e. less than half a radio wave length apart). An important constraint is that the diversity gain might deteriorate when another type of

diversity is applied at one end of the telecommunication link (e.g., mobile terminal) and dual polarisation at the other end (e.g., access point).

- **Space diversity**. Space diversity, also known as antenna diversity, is one of the most popular forms of diversity used in wireless systems. Space diversity is based on the fact that if a system operates in multipath scattering that is sufficiently rich, the signals received by spatially separated antennas are uncorrelated for antenna separations of half a radio wavelength or more. If L antennas are used at the receiver, it can be assumed that L independently fading channels are present. These L channels are combined at the receiver to improve the estimation of the transmitted signal.

At the receiver, the L received signals can be combined in several ways, namely:

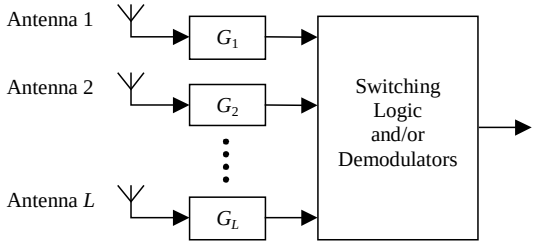

Figure 1-2: Block diagram for selection diversity.

- **Selection diversity**. A block diagram of the selection diversity method is shown in Figure 1-2. For selection diversity, the gains of all L branches are adjusted to provide the same average Signal-to-Noise Ration (SNR) for each branch. The receiver branch having the highest instantaneous SNR is connected to the demodulator[1]. The antenna signals themselves could be sampled and the best one sent to a single demodulator. In practice, the branch with the largest $(S + N)/N$ is used (where S represents the received signal power and N denotes the noise power), since it is difficult to measure the SNR.

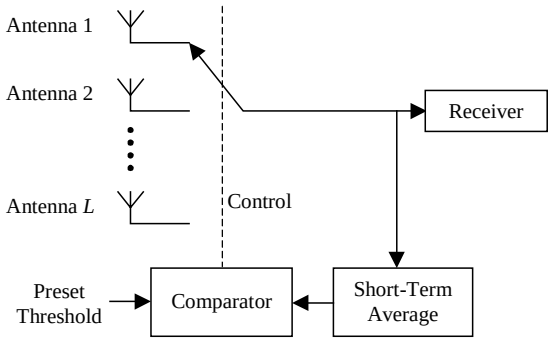

Figure 1-3: Block diagram of scanning diversity.

- **Feedback or scanning diversity**. Scanning diversity is similar to selection diversity except that instead of always using the best of the L signals, the L signals are scanned in a fixed sequence until one is found with an SNR above a predetermined threshold. The scanning process is resumed when this signal falls below the threshold. Until that time, it is selected and routed to the receiver (see Figure 1-3). The other combination techniques have a (somewhat)

[1] Note that in case of frequency diversity and time diversity, the L received signals are not obtained from L different antennas (as shown in Figure 1-2) but from L different frequencies and time slots, respectively. Keeping this in mind, all the combining techniques described here are applicable to frequency diversity and time diversity as well.

better performance, but the advantage of this technique is that it has a very low implementation complexity since only one receiver is required. For packet communications, another way of scanning is to only initiate it at every packet preamble.

- **Maximal Ratio Combining**. In this method, the signals from the *L* branches are weighted with the complex conjugate of the corresponding sub-channel and added (see Figure 1-4). This procedure generally requires an individual receiver for each antenna element, which is a big disadvantage compared to selection diversity. However, Maximal Ratio Combining (MRC) produces an output SNR equal to the sum of the individual SNRs, thus, it has the advantage that, when none of the individual signals have an acceptable SNR by themselves, it may produce a combined output which does haven an acceptable SNR. This technique is optimal because it can be seen as a special case of Maximum Likelihood Decoding which, under given assumptions, is proven to be optimal in Section 2.6

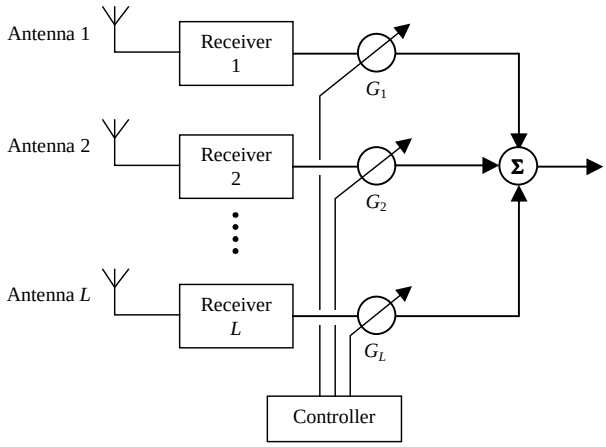

Figure 1-4: Block diagram of Maximal Ratio Combining

- **Equal Gain Combining**. An Equal Gain Combiner is obtained when the gains used in MRC are all set to unity. The disadvantage is that (like MRC) an individual receiver for each antenna element is needed. However, the possibility of producing an acceptable signal from a number of unacceptable inputs is still retained, and the performance is only marginally inferior to MRC and usually superior to selection diversity.

The above described diversity techniques are all based on the desire to provide wireless link improvement. However, the bandwidth efficiency (i.e., the ratio between the effective bit rate and the bandwidth), given in bps/Hz (bits per second per Hz), is not increased. For time diversity and frequency diversity the bandwidth efficiency is even reduced. This can be easily seen, namely, for time diversity, the bandwidth stays the same, but the effective bit rate in reduced by a factor *L* and for frequency diversity, the effective bit rate stays the same but the radio frequency bandwidth is *L* times larger. If in addition to diversity we also want to obtain a more bandwidth efficient wireless communication system, we have to find other techniques.

1.2 Space Division Multiplexing

A promising solution for significant increase of the bandwidth efficiency is the exploitation of the spatial dimension. Recent information theory research has revealed that the multipath wireless channel is capable of enormous capacities, provided that the multipath scattering is sufficiently rich ([14], [15], [37], [38], [41] and [58]). The multipath scattering can be properly exploited through the use of an appropriate processing architecture. The diagonally-layered space-time architecture

proposed in [14], known as diagonal BLAST (Bell Laboratories Layered Space Time) or D-BLAST, is such an approach. However, the diagonal approach is very complex and hard to implement. Therefore, a simplified version of BLAST, known as vertical BLAST or V-BLAST is proposed in [60]. A prototype has been built by which it is demonstrated that bandwidth efficiencies of 20 – 40 bps/Hz can be achieved in an indoor propagation environment at realistic SNRs and error rates.

These techniques can be captured under the more general term Space Division Multiplexing (SDM) or Space Division Multiple Access (SDMA). SDM techniques exploit the spatial dimension using multiple antennas at receiver as well as at the transmitter. Basically, these techniques transmit different signals on different transmit antennas simultaneously, with the goal of increasing the capacity and the SNR performance. At the receiver these different signals can be recovered by the Space Division Multiplexing techniques described in this report. For proper recovering of the transmitted signals, multiple antennas are required at the receiver as well. The difference between SDM and SDMA is that the latter allows different users to transmit simultaneously on a single antenna each), whereas in SDM a single user transmits simultaneously on multiple antennas. Hybrid schemes where several users transmit simultaneously on different antennas may also be possible.

One can naturally ask in which way SDM(A) techniques differ from traditional multiple access techniques. Some of these differences are worth pointing out [60]: First, unlike code-division or other spread-spectrum multiple access techniques, the total channel bandwidth utilised by a SDM(A) system is only a small fraction in excess of the symbol rate, i.e. similar to the excess bandwidth required by a conventional transmission technique like Quadrature Amplitude Modulation (QAM). Second, unlike Frequency Division Multiple Access (FDMA), each transmitted signal occupies the entire system bandwidth. Finally, unlike Time Division Multiple Access (TDMA), the entire system bandwidth is used simultaneously by all of the transmitters all of the time. These differences together are precisely what give SDM(A) the potential to realise higher bandwidth efficiencies than the other multiple-access techniques.

1.3 Target applications

One of the target areas we think will benefit from the research presented in this thesis is the area of the wireless local area networks (WLANs). Since the beginning of the nineties, WLANs for the 900-MHz, 2.4-GHz and 5-GHz ISM (Industrial, Scientific and Medical) bands have been available, based on a range of proprietary techniques [53]. In June 1997 the Institute of Electrical and Electronics Engineers (IEEE) approved an international interoperability standard, called IEEE 802.11 [24]. This standard specifies a number of Medium Access Control (MAC) procedures and three different Physical Layers (PHY). Two of these PHYs are radio-based and use the 2.4-GHz band and the other PHY uses infrared light. All PHYs support a data rate of 1 Mbps and optionally 2 Mbps.

User demand for higher bit rates and the international availability of the 2.4-GHz band has spurred the development of a higher speed extension to the 802.11 standard. In July 1998, a proposal was selected for standardisation, which describes a PHY providing a basic rate of 11 Mbps and a fall back rate of 5.5 Mbps. This PHY can be seen as a fourth option, to be used in conjunction with the MAC that is already standardised. Practical products, however, are expected to support both the high-speed 11 and 5.5 Mbit/s rate modes as well as the 1 and 2 Mbps modes. A product based on this IEEE 802.11 standard is WaveLAN™ of Lucent Technologies (see www.wavelan.com). The modem, that is implemented as PC card, is shown in Figure 1-5.

Figure 1-5: An IEEE 802.11 modem in the form of a PC card.

Since January 1997, in the United States 300 MHZ in the 5-GHz band came available for the use of a new category of unlicensed equipment called Unlicensed National Information Infrastructure (UNII) devices [13]. In July 1998, the IEEE 802.11 standardisation group selected Orthogonal Frequency Division Multiplexing (OFDM) as transmission technique for the recently available spectrum in the 5-GHz band. This adds a fifth PHY option to the 802.11 standard, with data rates ranging from 6 up to 54 Mbps ([25] and [46]).

Besides IEEE 802.11, two other standardisation groups were formed. In Europe, the European Telecommunication Standards Institute formed the standardisation group Broadband Radio Access Networks (ETSI BRAN) [12] and in Japan equipment manufacturers, service providers and the Ministry of Post and Telecommunications are co-operating in the Multimedia Mobile Access Communication (MMAC) project [29]. ETSI BRAN is now working on extensions of the ETSI HYPERLAN type 1 standard. Three extensions are under development: HIPERLAN/2, a wireless indoor LAN with a Quality of Service provision; Hiperlink, a wireless indoor backbone; and HiperAccess, an outdoor, fixed wireless network providing access to a wired infrastructure. The MMAC project is started to define new wireless standards similar to those of IEEE 802.11 and ETSI BRAN. Additionally, MMAC is also looking into the possibility for ultra-high-speed wireless indoor LANs supporting large-volume data transmission at speeds up to 156 Mbps using frequencies in the 30- to 300-GHz band.

Following the IEEE 802.11 selection of OFDM as basis for transmission in the 5-GHz band, ETSI BRAN and MMAC also adopted OFDM for their physical layer standards. The three standardisation groups have worked in close co-operation since then to make sure that differences among the various standards are kept to a minimum, thereby enabling the manufacturing of equipment that can be used world-wide.

So, since there is a lot of effort going on to adopt OFDM as a world-wide standard for Wireless LANs, the research to next generation WLANs with even higher data rates, but based on these OFDM standards, is a logical consequence. In this thesis, a system is proposed that combines Space Division Multiplexing with OFDM as a spectrum efficient transmission technique, to achieve higher data rates and to obtain sufficient delay-spread robustness.

The rest of this thesis is organised as follows: in Chapter 2 a signal model for the wireless link is given. Various Space Division Multiplexing techniques are described and their performance is compared. In Chapter 3 the complexity of these Space Division Multiplexing techniques is determined and compared. Chapter 4 gives a summary of the Orthogonal Frequency Division Multiplexing transmission technique. In Chapter 5 this technique is extended with the SDM method

with the best performance, i.e. Maximum Likelihood Decoding (MLD). In this chapter a convolutional forward error correction code is applied to the data transmitted by the OFDM system with SDM. A Viterbi decoder is used to decode the convolutional code. From [17] it can be concluded that Viterbi decoders perform better if they get soft-decision values as input. Therefore, the MLD method is somewhat adapted to provide soft-decision values as output, which again are used as input for the soft-decision input Viterbi decoder. Finally, conclusions and future work are discussed in Chapter 6.

1.4 Related work

As mentioned before, there are different ways to improve the performance and/or capacity of wireless communication systems. Diversity is one of the techniques that is often used to improve the system performance. There are two basic types of diversity, namely, receiver diversity and transmitter diversity. The main difference is that for a well performing transmitter diversity system, the transmitter needs some information about the radio channel, thus, if the downlink and uplink carrier frequencies are different, feedback is required from the receiver. For receiver diversity, the transmitter does not need any information about the radio channel, and no feedback is required.

There are many publications on receiver diversity for single-carrier wireless systems. An overview of different receiver diversity techniques is given in Section 1.1. The reader is referred to the literature mentioned in this section and the references therein. Furthermore, in [28] the feasibility of polarisation equal gain combining and selection diversity in indoor cells at 1800 MHz is investigated. The combination of selection diversity with Maximal Ratio Combining is given in [39]. Other applications of receiver diversity are Smart Antenna Arrays ([43]). With an (adaptive) antenna array at the receiver and additional processing it is possible to do beamforming, i.e., “listen” to the direction the (multipath) signals come from. An adaptive array is based on an internal feedback control loop that updates the complex weight coefficients of the array. After weighting the received signals, the output of each antenna element is combined in order to minimise the variation in amplitude of the received signals. In this way beams are formed to receive the desired signals. It is also possible to place nulls in the direction of interferers. Beamforming alogrihms (or direction of arrival (DOA) estimation algorithms) include the MUSIC algorithm, ESPRIT algorithm and the MODE estimate algorithm. An extensive list of references on DOA estimation is given in [38].

Transmitter diversity for single-carrier wireless systems has widely been researched. As indicated in [59] and the references therein, transmitter diversity can be based on linear transforms. However, neither transmitter diversity nor receiver diversity increase the system capacity, whereas the Space Division Multiplexing techniques proposed in this report do. Furthermore, it will be shown in this thesis that the latter techniques are, just like transmitter and receiver diversity, able to achieve performance improvement by exploiting diversity.

Currently, a lot of research is ongoing to apply transmitter or receiver diversity to multi-carrier techniques like Orthogonal Frequency Division Multiplexing (OFDM). The advantage of OFDM over an equalised single carrier system is that, for a given delay spread and bit rate, the OFDM receiver is less complex to implement [53]. The basic principles of OFDM are discussed in [6], [44] and [56]. A number of transmitter diversity techniques for OFDM are proposed in [30], [42] and [61]. Furthermore, research on applying receiver diversity techniques to OFDM is reported in [4], [7], [18], [26] and [54]. In [4], a coherent diversity receiver for OFDM signals is presented. The received signals are optimally combined using Maximal Ratio Combining. In [7] the OFDM subchannels are spread over multiple transmit antennas to randomise the fading across the OFDM bandwidth. At the receiver two branch antenna diversity is used. To realise the full potential of the diversity, Reed-Solomon coding across subchannels is used. The error correction ability of this

code benefits from the randomisation of the fading by spreading the OFDM subchannels across multiple transmit antennas. In [26] this principle is extended with convolutional coding. In [18], at the receiver the OFDM frequency band is divided in subbands (consisting of a multiple of subcarriers) and Maximal Ratio Combining is performed on each subband. It is shown that the Signal-to-Noise Ratio performance increases compared to using the same MRC weights for all subcarriers of the OFDM signal. Ideally, MRC should be performed per subcarrier. In [54] a method is described on how to combine adaptive antennas and OFDM for operation in a faded delay-spread channel. All these techniques show (significant) SNR performance improvement, all without increasing the capacity. In this report OFDM is combined with Space Division Multiplexing and it is shown that a system based on SDM is capable of high bandwidth efficiencies, which allows the transmission of higher bit rates using the same bandwidth.

In [37] a method is described to extend multitone (i.e., multicarrier) systems like OFDM with SDM. However, in this technique the singular value decomposition of the channel matrix is applied which implies that, both the receiver and the transmitter need to know the channel. The advantage of the SDM techniques, described in this thesis, is that the transmitter does not need to know the channel.

Closely related to the Space Division Multiplexing techniques are the so-called Space-Time codes. In Space-Time coding the design of channel codes is considered to improve the data rate and/or the reliability of communications over fading channels using multiple transmit antennas. Data is encoded by a channel code and the encoded data is split into a number of parallel streams that are simultaneously transmitted on different transmit antennas ([1], [31] and [49]). The code words, that have a space and time dimension, are chosen such that both the coding gain and diversity gain are maximised. In [1] the application of Space-Time codes to an OFDM system is described and it is shown that this combined system performs well at reasonable SNRs. The simple transmitter diversity system proposed in [2] can be seen as a simple space-time code approach. A disadvantage of Space-Time coding is that if the radio channel is changing, new Space-Time codes need to be found for which, again, both the coding gain and diversity gain are maximised. This involves a lot of processing and, therefore, we have looked to other methods to increase data-rates.

2 Multi-Antenna Link: Signal model and SDM techniques

One way to increase the capacity of wireless communication systems is by exploiting the spatial dimension using multiple antennas at the receiver as well as at the transmitter ([15] and [37]). The idea is to transmit several streams of data simultaneously over the air. Of course these parallel streams of data get mixed up in the air. However, at the receiver it is possible to "decode" this mixture of data. This type of transmission is called Space Division Multiplexing (SDM). In this chapter, a number of different decoding algorithms for SDM will be discussed and their performance will be compared. In order to describe the different decoding methods, first a signal model for the multi-antenna link will be stated.

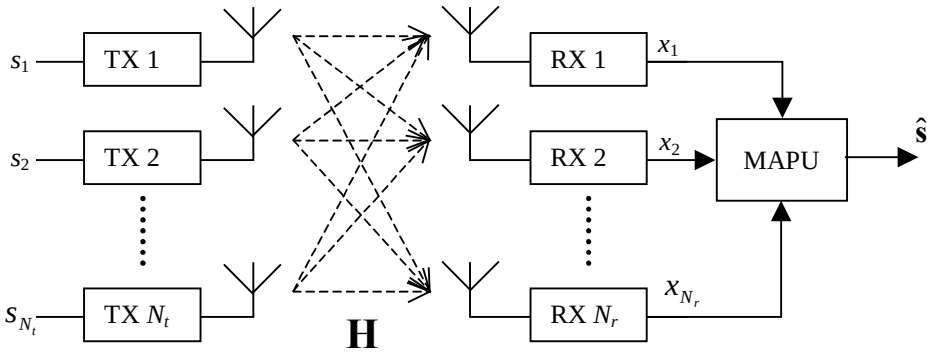

Figure 2-1: The physical model of a system with Space Division Multiplexing (SDM).

2.1 Signal model

We consider a communication system comprising N_t transmit (TX) and N_r receive (RX) antennas that operates in a Rayleigh flat-fading environment and exploits the spatial dimension by using Space Division Multiplexing (SDM) (see Figure 2-1, where MAPU stands for Multi Antenna Processing Unit). A channel is said to be a *Rayleigh fading channel* when the channel impulse response is modelled as a zero-mean complex-valued gaussian process [34]. In this case, the envelope of the channel impulse response at any time instant has a Rayleigh probability distribution and the phase is uniformly distributed in the interval $(0,2\pi)$. This can be shown as follows. Suppose we model the channel impulse response H as a zero-mean complex gaussian variable, like:

$$H = A + jB \tag{2.1}$$

where A and B are zero-mean statistically independent real guassian variables, each having a variance $\sigma^2/2$. The variance of the complex gaussian variable H can be shown to be:

$$\begin{aligned} \mathrm{var}(H) &= E\left[H^2\right] - E\left[H\right]^2 = E\left[(A + jB)(A + jB)\right] \\ &= E\left[A^2 + 2jAB + B^2\right] = E\left[A^2\right] + E\left[B^2\right] = \sigma^2 \end{aligned} \tag{2.2}$$

Then, in order to find the distribution of the envelope we introduce a new variable:

$$Y = A^2 + B^2 \tag{2.3}$$

From the analysis of [34] it follows that Y is chi-square-distributed with two degrees of freedom. Hence, the probability density function (pdf) of Y is:

$$p_Y(y) = \frac{1}{\sigma^2} e^{-y/\sigma^2}, \; y \geq 0 \tag{2.4}$$

Now, suppose we define a random variable R to denote the envelope of the channel impulse response:

$$R = \sqrt{A^2 + B^2} = \sqrt{Y} \tag{2.5}$$

In order to find the pdf of R, we need to make a change of variables in the Probability Distribution Function (PDF) of Y (given by $F_Y(y)$), to first find the PDF of R:

$$F_R(r) = P\{R \leq r\} = P\{\sqrt{Y} \leq r\} = P\{Y \leq r^2\} = F_Y(r^2) \tag{2.6}$$

where the fact is used that $r \geq 0$. Then the pdf of R is given by:

$$p_R(r) = \frac{d}{dr} F_R(r) = \frac{d}{dr} F_Y(r^2) = 2r \cdot p_Y(r^2) = \frac{2r}{\sigma^2} e^{-r^2/\sigma^2}, \; r \geq 0 \tag{2.7}$$

This is the pdf of a Rayleigh-distributed random variable, thus the envelope of H has a Rayleigh probability distribution. The fact that the phase of H is uniformly distributed in the interval $(0,2\pi)$ can be seen intuitively and will not be shown here.

Once the channel characteristics are specified, we can introduce the signal-model. Suppose that at discrete times, the transmitter sends an N_t-dimensional (complex) signal vector **s** (i.e., it transmits N_t parallel streams of data), and the receiver records an N_r-dimensional complex vector **x**. Then the following signal model describes the relation between **s** and **x**:

$$\mathbf{x} = \mathbf{H}\mathbf{s} + \mathbf{v} \tag{2.8}$$

where **H** is an $N_r \times N_t$ complex propagation matrix that is constant with respect to the symbol time and assumed known at the receiver (e.g. via transmitting training sequences) and the vector **v** (N_r-dimensional) represents additive receiver noise. The vector **s** is assumed to have zero-mean, uncorrelated random variables with variance equal to σ_s^2. The total power of **s** (i.e., $E[\mathbf{s}^*\mathbf{s}]$) is assumed to be P. Thus, the covariance matrix of **s** equals:

$$E[\mathbf{s}\mathbf{s}^*] = \sigma_s^2 \mathbf{I}_{N_t} = \frac{P}{N_t} \mathbf{I}_{N_t} \tag{2.9}$$

where * denotes the conjugate transpose of a vector or matrix and the matrix **I** with subscript N_t represents the identity matrix with dimension N_t. In the rest of this report * will be used to denote the conjugate transpose of a vector or matrix, unless indicated otherwise. Note that the total transmitted power does not depend on the number of transmit antennas but is assumed fixed at P.

As said before, the channel matrix is denoted by **H**. It is assumed the channel matrix **H** has independently and identically distributed (i.i.d.), zero-mean, complex Gaussian, unit variance entries (the variance of each entry is $\sigma_c^2 = 1$). In other words, each component has a Rayleigh distribution (like H) with $\sigma^2 = \sigma_c^2 = 1$. Thus, the system operates in a Rayleigh flat-fading environment.

The vector $\mathbf{v}$ is N_r-dimensional and represents additive receiver noise. The vector $\mathbf{v}$ is assumed to have zero-mean, uncorrelated random variables with variance σ_v^2 and a covariance matrix equal to:

$$E\left[\mathbf{v}\mathbf{v}^*\right] = \sigma_v^2 \mathbf{I}_{N_r} \tag{2.10}$$

Furthermore, it is assumed that the vectors **s** and $\mathbf{v}$ are independent and thus the following holds: $E[\mathbf{s}\mathbf{v}^*] = 0$. With the above described assumptions about the power of the signal and the noise, the expected Signal-to-Noise Ratio (SNR) per receiving antenna, i.e. the SNR for each component of **x**, can be found and is equal to:

$$\rho = \frac{E_s}{N_0} = \frac{N_t \sigma_s^2 \sigma_c^2}{\sigma_v^2} = \frac{P}{\sigma_v^2} \tag{2.11}$$

where E_s stands for the signal power per receive antenna and N_0 denotes the noise power per receive antenna.

In order to explain the different Space Division Multiplexing techniques, the following notations will be used:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_{N_r} \end{bmatrix}, \ \mathbf{s} = \begin{bmatrix} s_1 \\ \vdots \\ s_{N_t} \end{bmatrix} \text{ and } \mathbf{H} = \begin{bmatrix} \mathbf{H}_1 \\ \vdots \\ \mathbf{H}_{N_r} \end{bmatrix} = \begin{bmatrix} \mathbf{h}_1 & \dots & \mathbf{h}_{N_t} \end{bmatrix} \tag{2.12}$$

where x_i and s_i represent the i-th element of **x** and **s** respectively. $\mathbf{H}_i$ and $\mathbf{h}_i$ denote the i-th row and i-th column of **H**, respectively.

In the following sections, a number of SDM decoding techniques for the receiver will be discussed and compared. One way to compare the different techniques is by comparing their bit error rate (BER) performance. The BER performance depends on the SNR and, thus, can be visualised by a figure in which the BER is depicted against the SNR. Simulations are performed in order to obtain these BER characteristics. Furthermore, for some SDM techniques, theoretical representations (or upperbounds) of the BER characteristics will be deduced, in order to verify the simulation results.

2.2 The Zero Forcing Algorithm

The first decoding technique forms the basis of the following decoding algorithms to be described in this report (except for the Maximum Likelihood Decoding): it is the so-called Zero Forcing (ZF) algorithm. This algorithm is based on a conventional adaptive antenna array (AAA) technique, namely, linear combinatorial nulling [60]. In this technique, each substream in turn is considered to be the desired signal, and the remaining data streams are considered as “interferers”. Nulling of the interferers is performed by linearly weighting the received signals so that all interfering terms are cancelled. For Zero Forcing, nulling of the “interferers” can be performed by choosing weight vectors $\mathbf{d}_i$ (with $i = 1, 2, \dots, N_t$) such that

$$\mathbf{d}_i^T \mathbf{h}_j = \begin{cases} 0, & j \neq i \\ 1, & j = i \end{cases} \tag{2.13}$$

where ^{T} stands for the transpose of a vector or matrix and $\mathbf{h}_j$ denotes the j-th column of the channel matrix $\mathbf{H}$. However, when we take a closer look to this criterion, solving the weight vectors is equal to finding a matrix $\mathbf{D}$ such that:

$$\mathbf{D} \cdot \mathbf{H} = \mathbf{I} \tag{2.14}$$

where $\mathbf{D}$ is a matrix that represents the linear processing in the receiver. The i-th row of $\mathbf{D}$ is equal to the transpose of the i-th weight vector $\mathbf{d}_i$ and $\mathbf{I}$ is the identity matrix. So, by forcing the "interferers" to zero, each desired element of $\mathbf{s}$ can be estimated. If $\mathbf{H}$ is not square, $\mathbf{D}$ equals the *pseudo-inverse* of $\mathbf{H}$:

$$\mathbf{D} = \mathbf{H}^{+} = \left(\mathbf{H}^{*}\mathbf{H}\right)^{-1}\mathbf{H}^{*} \tag{2.15}$$

where $^+$ represents the *pseudo-inverse*. In order for the pseudo-inverse to exist, N_t must be less than or equal to N_r, because for N_t larger than N_r, $\mathbf{H}^{*}\mathbf{H}$ is singular and its inverse does not exist [45]. Furthermore, note that in order for the inverse to exist, the columns of $\mathbf{H}$ must be independent. Regarding the i.i.d. assumption of the elements of $\mathbf{H}$, independence is usually an approximation, which is justifiable if the antenna spacing is chosen equal to or larger than $\lambda/2$ [15] (where λ represents the wavelength of the transmission frequency) and if the system operates in a rich-scattered environment, which can be modelled by Rayleigh flat-fading. Thus, for $N_r \geq N_t$ and if the inverse of $\mathbf{H}^{*}\mathbf{H}$ exists, the estimates of $\mathbf{s}$ (given by $\mathbf{s}_{\text{est}}$) can be found by:

$$\begin{aligned} \mathbf{s}_{\text{est}} &= \mathbf{D}\mathbf{x} \\ &= \left(\mathbf{H}^{*}\mathbf{H}\right)^{-1}\mathbf{H}^{*}\mathbf{x} \end{aligned} \tag{2.16}$$

or, equivalently:

$$\mathbf{s}_{\text{est}} = \mathbf{H}^{+}\mathbf{x} \tag{2.17}$$

Using Formula (2.17), $(s_{\text{est}})_i$, i.e. the i-th component of $\mathbf{s}_{\text{est}}$, can be written as:

$$\left(s_{\text{est}}\right)_i = \mathbf{H}_i^{+}\mathbf{x} \tag{2.18}$$

where $\mathbf{H}_i^{+}$ represents the i-th row of $\mathbf{H}^{+}$, which, according to Formula (2.13), is equal to the transpose of the i-th weight vector $\mathbf{d}_i$. Note that $\mathbf{d}_i$ is a so-called *nulling vector* [60]. As a final step, $(s_{\text{est}})_i$ can be sliced to the nearest Quadrature Amplitude Modulation (QAM) constellation point, these sliced signals are denoted by $\hat{\mathbf{s}}$. In this way, all N_t elements of $\mathbf{s}$ can be decoded at the receiver.

The in this section described ZF algorithm can be simulated and its BER performance can be obtained (see Section 2.7). To verify the simulations it is useful to deduce a theoretical representation of the BER performance of a system based on the ZF algorithm. A theoretical representation can be obtained using the fact that the diversity order of the average error probability of such a system with antenna configuration (N_t,N_r), is equal to a system with Maximal Ratio Combining (MRC), one transmitting (TX) antenna and N_r-N_t+1 receiving (RX) antennas [58]. Note that a diversity order of one means that the BER improves by a factor of 10^1 if the SNR is increased by 10 dB. In case of a diversity order of two, if the SNR is increased by 10 dB, the BER improves 10^2 times, etc. With the knowledge that the ZF algorithm can be compared with MRC, an exact representation of the ZF solution can be deduced from the diversity analysis performed in [34]. Here, a summary will be given.

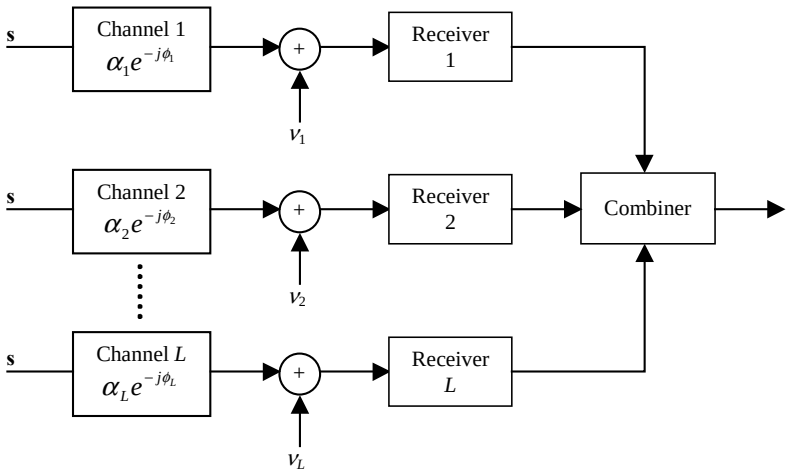

Figure 2-2: Model of a digital communication system with diversity.

Suppose we have a system, based on MRC, comprising one transmit and L receive antennas (see Figure 2-2) and assume that the system operates in a Raleigh flat-fading environment, then the channel elements can be written as:

$$h_k = \alpha_k e^{-j\phi_i} \tag{2.19}$$

where h_k represents the channel from the TX antenna to the k-th RX antenna, α_k denotes the channel attenuation factor and the phase shift is given by ϕ_k. Furthermore, we assume that the total channel is perfectly known at the receiver. Now, let us consider the performance of BPSK with Lth-order diversity. From Section 1.1 it is known that a MRC system combines the received signals by multiplying them with the conjugate of the corresponding channel element. Thus, for a system with MRC operating in a Raleigh flat-fading environment the output can be expressed as a single decision variable in the form:

$$U = \mathrm{Re}\left(\sum_{k=1}^{L} h_k^* x_k \right) = \mathrm{Re}\left(\sum_{k=1}^{L} \alpha_k^2 \mathbf{s} + \sum_{k=1}^{L} h_k^* \nu_k \right) \tag{2.20}$$

where ν_k denotes the complex Gaussian noise and **s** is the transmitted symbol that, for BSPK, is equal to $\sqrt{E_b}$ or $-\sqrt{E_b}$, where E_b denotes the mean transmitted power. Thus, **s** only consists of one element in case of MRC (i.e., the system has only one transmit antenna). The decision variable U is used in the decoder to recover the transmitted **s**. If $U > 0$ then the decoder assumes that $\mathbf{s} = \sqrt{E_b}$ is transmitted and if $U < 0$ then it is assumed that $\mathbf{s} = -\sqrt{E_b}$ is transmitted. For a fixed set of $\{\alpha_k\}$ and the assumption that $\mathbf{s} = \sqrt{E_b}$ is transmitted, the decision variable U is Gaussian and with the mean noise power per receive antenna equal to N_0, the probability that U is less than zero is:

$$P_2(\gamma_b) = Q\left(\sqrt{2\gamma_b}\right) \tag{2.21}$$

where Q is the function that is frequently used for the area under the tail of the Gaussian pdf and defined as:

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2} dt \quad \text{and } x \geq 0 \tag{2.22}$$

The SNR per bit, γ_b, is given as:

$$\gamma_b = \sum_{k=1}^{L} \frac{E_b}{N_0}\alpha_k^2 = \sum_{k=1}^{L} \gamma_k \tag{2.23}$$

where γ_k is the instantaneous SNR on the k-th channel. Because γ_b can be seen as a chi-square-distributed random variable with $2L$ degrees of freedom, the probability density function $p(\gamma_b)$ is [34]:

$$p(\gamma_b) = \frac{1}{(L-1)!\gamma_c^L}\gamma_b^{L-1}e^{\gamma_b/\gamma_c} \tag{2.24}$$

where γ_c is the average SNR per channel. In order to obtain the total BER, we need to average the conditional error probability given by Formula (2.21) over the fading channel statistics:

$$P_2 = \int_0^{\infty} P_2(\gamma_b)p(\gamma_b)d\gamma_b \tag{2.25}$$

There is a closed-form solution for this integral, which can be expressed as:

$$P_2 = \left[\frac{1}{2}(1-\mu)\right]^L \sum_{k=0}^{L-1}\binom{L-1+k}{k}\cdot\left[\frac{1}{2}(1+\mu)\right]^k \tag{2.26}$$

where, by definition,

$$\mu = \sqrt{\frac{\gamma_c}{1+\gamma_c}} \tag{2.27}$$

The conditional error probability in case $\mathbf{s} = -\sqrt{E_b}$ is transmitted is also equal to Formula (2.26), thus, this leads to a total error probability of $P_2/2 + P_2/2 = P_2$. When γ_c satisfies the condition $\gamma_c >> 1$, the term $(1 + \mu)/2 \approx 1$ and the term $(1 - \mu)/2 \approx 1/4\gamma_c$. Furthermore,

$$\sum_{k=0}^{L-1}\binom{L-1+k}{k} = \binom{2L-1}{L} \tag{2.28}$$

Therefore, when γ_c is sufficiently large (greater than 10 dB), the probability of error in Formula (2.26) can be approximated as

$$P_2 \approx \left(\frac{1}{4\gamma_c}\right)^L \binom{2L-1}{L} \tag{2.29}$$

From, this formula we observe that the error rate decreases inversely with the L-th power of the SNR, i.e., the error rate decreases with an L-th order of diversity.

Formula (2.26) can be extended to a multi-transmit antenna case as follows: as described before, in [58] it is shown that the diversity order of an (N_t,N_r) system based on ZF is equal to $N_r–N_t+1$. So,

substituting $L = N_r - N_t + 1$ in Formula (2.26) gives the exact BER for a ZF solution of an (N_t, N_r) system with BPSK. Note that, if ρ represents the SNR per receiving antenna, the SNR per channel equals:

$$\gamma_c = \frac{\rho}{N_t} \tag{2.30}$$

The theoretical result, given in (2.26), will be used in Section 2.7 to verify the BER performance that the simulation of the ZF algorithm gives as output.

2.3 The Minimum Mean Square Error solution

Another approach in estimation theory to the problem of estimating a random vector **s** on the basis of observations **x** is to choose a function **g**(**x**) that minimises the Mean Square Error (MSE) [36]:

$$\varepsilon^2 = E[(\mathbf{s}-\mathbf{s}_{\text{est}})^*(\mathbf{s}-\mathbf{s}_{\text{est}})] = [(\mathbf{s}-\mathbf{g}(\mathbf{x}))^*(\mathbf{s}-\mathbf{g}(\mathbf{x}))] \tag{2.31}$$

An exact function **g**(**x**) is usually hard to obtain, however, is we restrict this function to be a linear function of the observations, an exact solution can be achieved. Using linear processing, the estimates of **s** can be found by:

$$\mathbf{s}_{\text{est}} = \mathbf{D}\mathbf{x} \tag{2.32}$$

Now, to obtain the linear Minimum Mean Square Error (MMSE), **D** must be chosen such that the Mean Square Error ε^2 is minimised:

$$\varepsilon^2 = E[(\mathbf{s}-\mathbf{s}_{\text{est}})^*(\mathbf{s}-\mathbf{s}_{\text{est}})] = E[(\mathbf{s}-\mathbf{D}\mathbf{x})^*(\mathbf{s}-\mathbf{D}\mathbf{x})] \tag{2.33}$$

Since $\mathbf{a}^*\mathbf{a} = \text{tr}(\mathbf{a}\mathbf{a}^*)$, where tr(·) stands for the trace of a matrix, the MSE can be rewritten as:

$$\begin{aligned} E[(\mathbf{s}-\mathbf{D}\mathbf{x})^*(\mathbf{s}-\mathbf{D}\mathbf{x})] &= \text{tr}\left(E[(\mathbf{s}-\mathbf{D}\mathbf{x})(\mathbf{s}-\mathbf{D}\mathbf{x})^*]\right) \\ &= \text{tr}\left(\mathbf{Q}_s - \mathbf{Q}_{sx}\mathbf{D}^* - \mathbf{D}\mathbf{Q}_{xs} + \mathbf{D}\mathbf{Q}_x\mathbf{D}^*\right) \end{aligned} \tag{2.34}$$

where, $\mathbf{Q}_s = E[\mathbf{s}\mathbf{s}^*]$, $\mathbf{Q}_{sx} = E[\mathbf{s}\mathbf{x}^*]$, $\mathbf{Q}_{xs} = E[\mathbf{x}\mathbf{s}^*]$ and $\mathbf{Q}_x = E[\mathbf{x}\mathbf{x}^*]$. In order to minimise this MSE over **D**, we first need to make some remarks.

Remark that the covariance matrices are Hermitian: $\mathbf{Q} = \mathbf{Q}^*$ [45]. To prove this, suppose that $\mathbf{Q} = E[\mathbf{a}\mathbf{a}^*]$ then:

$$\mathbf{Q}^* = \left(E[\mathbf{a}\mathbf{a}^*]\right)^* = E[\left(\mathbf{a}\mathbf{a}^*\right)^*] = E[\left(\mathbf{a}^*\right)^*\mathbf{a}^*] = E[\mathbf{a}\mathbf{a}^*] = \mathbf{Q} \tag{2.35}$$

where the fact is used that $(\mathbf{AB})^* = \mathbf{B}^*\mathbf{A}^*$. Furthermore, remark that the covariance matrices are nonnegative definite. A Hermitian matrix **Q** is said to be *nonnegative definite* if $\mathbf{z}^*\mathbf{Q}\mathbf{z} \geq 0$ for all nonzero complex vectors **z**. Suppose again that $\mathbf{Q} = E[\mathbf{a}\mathbf{a}^*]$, then the proof goes as follows:

$$\mathbf{z}^*\mathbf{Q}\mathbf{z} = \mathbf{z}^*E[\mathbf{a}\mathbf{a}^*]\mathbf{z} = E[\mathbf{z}^*\mathbf{a}\mathbf{a}^*\mathbf{z}] = E[\left|\mathbf{a}^*\mathbf{z}\right|^2] \geq 0 \tag{2.36}$$

Furthermore, remark that each eigenvalue of a nonnegative definite matrix is a nonnegative real number [22]: let $\mathbf{Q}$ be nonnegative definite, let λ be an eigenvalue of $\mathbf{Q}$, let $\mathbf{z}$ be an eigenvector of $\mathbf{Q}$ associated with λ, and calculate $\mathbf{z}^*\mathbf{Q}\mathbf{z} = \mathbf{z}^*\lambda\mathbf{z} = \lambda|\mathbf{z}|^2$. Therefore, $\lambda = \mathbf{z}^*\mathbf{Q}\mathbf{z}/|\mathbf{z}|^2$ is real and nonnegative since it is a ratio of a real nonnegative and a real positive number.

Finally, remark that the eigenvectors of a Hermitian matrix $\mathbf{Q}$, if they come from different eigenvalues, are orthogonal to one another [45]. The proof of this remark starts with the information given: $\mathbf{Q}\mathbf{x} = \lambda_1\mathbf{x}$ and $\mathbf{Q}\mathbf{y} = \lambda_2\mathbf{y}$ and $\mathbf{Q} = \mathbf{Q}^*$, where $\mathbf{x}$ and $\mathbf{y}$ are the eigenvectors for the eigenvalues λ_1 and λ_2, respectively. Then, it can be shown that:

$$(\lambda_1\mathbf{x})^*\mathbf{y} = (\mathbf{Q}\mathbf{x})^*\mathbf{y} = \mathbf{x}^*\mathbf{Q}\mathbf{y} = \mathbf{x}^*(\lambda_2\mathbf{y}) \tag{2.37}$$

The outside numbers are $\lambda_1\mathbf{x}^*\mathbf{y} = \lambda_2\mathbf{x}^*\mathbf{y}$, since the λ's are real. Now we use the assumption $\lambda_1 \neq \lambda_2$, which forces the conclusion that $\mathbf{x}^*\mathbf{y} = 0$, which again shows that $\mathbf{x}$ and $\mathbf{y}$ are orthogonal. Note, however, that any multiples $\mathbf{x}/a$ and $\mathbf{y}/b$ would be equally good as eigenvectors. Suppose we choose $a = \|\mathbf{x}\|$ and $b = \|\mathbf{y}\|$, so that $\mathbf{x}/a$ and $\mathbf{y}/b$ are unit vectors; the eigenvectors have been normalised to have length one. Since they were already orthogonal, they are now *orthonormal*. If these orthonomal eigenvectors are chosen to be the columns of $\mathbf{U}$, then we have $\mathbf{U}^*\mathbf{Q}\mathbf{U} = \mathbf{\Lambda}$, where $\mathbf{\Lambda}$ is a diagonal matrix with the eigenvalues of $\mathbf{Q}$ on the diagonal and $\mathbf{U}$ is said to be a unitary matrix. For unitary matrices the following condition holds: $\mathbf{U}^*\mathbf{U} = \mathbf{I}$.

Now, if we go back to the MMSE problem, we see that we may write $\mathbf{Q}_x$ (because $\mathbf{Q}_x$ is Hermitian and nonnegative definite) as follows:

$$\mathbf{Q}_x = \mathbf{U}^*\mathbf{\Lambda}_x\mathbf{U} = \mathbf{U}^*\mathbf{\Lambda}_x^{1/2}\mathbf{\Lambda}_x^{1/2}\mathbf{U} = \left(\mathbf{\Lambda}_x^{1/2}\mathbf{U}\right)^*\mathbf{\Lambda}_x^{1/2}\mathbf{U} = \mathbf{A}^*\mathbf{A} \tag{2.38}$$

where $\mathbf{\Lambda}_x^{1/2}$ is a diagonal matrix with the square-roots of the eigenvalues on the diagonal (this is allowed because the eigenvalues are real and nonnegative) and $\mathbf{A}$ is said to be the "square-root" of $\mathbf{Q}_x$. Finally, we can rewrite the MMSE problem to a form, from which a solution for $\mathbf{D}$ can be obtained that minimises the Mean Square Error:

$$\begin{aligned}
\varepsilon^2 &= \mathrm{tr}\left(\mathbf{Q}_s - \mathbf{Q}_{sx}\mathbf{D}^* - \mathbf{D}\mathbf{Q}_{xs} + \mathbf{D}\mathbf{Q}_x\mathbf{D}^*\right) \\
&= \mathrm{tr}\left(\mathbf{Q}_s - \mathbf{Q}_{sx}\mathbf{A}^{-1}\mathbf{A}\mathbf{D}^* - \mathbf{D}\mathbf{A}^*\mathbf{A}^{*-1}\mathbf{Q}_{xs} + \mathbf{D}\mathbf{A}^*\mathbf{A}\mathbf{D}^*\right) \\
&= \mathrm{tr}(\mathbf{Q}_s - \mathbf{Q}_{sx}\mathbf{A}^{-1}\mathbf{A}^{*-1}\mathbf{Q}_{xs} + \mathbf{Q}_{sx}\mathbf{A}^{-1}\mathbf{A}^{*-1}\mathbf{Q}_{xs} \\
&\qquad - \mathbf{Q}_{sx}\mathbf{A}^{-1}\mathbf{A}\mathbf{D}^* - \mathbf{D}\mathbf{A}^*\mathbf{A}^{*-1}\mathbf{Q}_{xs} + \mathbf{D}\mathbf{A}^*\mathbf{A}\mathbf{D}^*) \\
&= \mathrm{tr}\left(\mathbf{Q}_s - \mathbf{Q}_{sx}\mathbf{A}^{-1}\mathbf{A}^{*-1}\mathbf{Q}_{xs} + \left(\mathbf{D}\mathbf{A}^* - \mathbf{Q}_{sx}\mathbf{A}^{-1}\right)\left(\mathbf{D}\mathbf{A}^* - \mathbf{Q}_{sx}\mathbf{A}^{-1}\right)^*\right)
\end{aligned} \tag{2.39}$$

Because the first and the second term of this result do not depend on $\mathbf{D}$, the result is minimised if:

$$\mathbf{D}\mathbf{A}^* - \mathbf{Q}_{sx}\mathbf{A}^{-1} = 0 \tag{2.40}$$

When this is worked out, finally, $\mathbf{D}$ is obtained:

$$\begin{aligned}
\mathbf{D}\mathbf{A}^* &= \mathbf{Q}_{sx}\mathbf{A}^{-1} \\
\mathbf{D} &= \mathbf{Q}_{sx}\mathbf{A}^{-1}\mathbf{A}^{*-1} = \mathbf{Q}_{sx}\left(\mathbf{A}^*\mathbf{A}\right)^{-1} = \mathbf{Q}_{sx}\mathbf{Q}_x^{-1}
\end{aligned} \tag{2.41}$$

where the fact is used that $\mathbf{A}^{-1}\mathbf{B}^{-1} = (\mathbf{BA})^{-1}$. When we go back to our channel model ($\mathbf{x} = \mathbf{Hs} + \boldsymbol{\nu}$) and assume that $\mathbf{Q}_s = E[\mathbf{ss}^*] = \sigma_s^2\mathbf{I}$ and $\mathbf{Q}_\nu = E[\boldsymbol{\nu}\boldsymbol{\nu}^*] = \sigma_\nu^2\mathbf{I}$ and that $\mathbf{Q}_x = E[\mathbf{xx}^*] = \mathbf{HQ}_s\mathbf{H}^* + \mathbf{Q}_\nu$ is invertible, then, with $\mathbf{Q}_{sx} = E[\mathbf{sx}^*] = E[\mathbf{s}(\mathbf{s}^*\mathbf{H}^* + \boldsymbol{\nu}^*)] = \mathbf{Q}_s\mathbf{H}^*$ ($\mathbf{s}$ and $\boldsymbol{\nu}$ are assumed independent), $\mathbf{D}$ becomes:

$$\mathbf{D} = \mathbf{Q}_s\mathbf{H}^*\left(\mathbf{HQ}_s\mathbf{H}^* + \mathbf{Q}_\nu\right)^{-1} = \sigma_s^2\mathbf{H}^*\left(\sigma_s^2\mathbf{HH}^* + \sigma_\nu^2\mathbf{I}\right)^{-1} = \mathbf{H}^*\left(\mathbf{HH}^* + \alpha\mathbf{I}\right)^{-1} \tag{2.42}$$

where $\alpha = \sigma_\nu^2/\sigma_s^2$. When we have a matrix of the form $\mathbf{A} + \mathbf{BCE}$, it is often useful to see what happens when applying the Matrix Inversion Lemma [27], which is given by:

$$\left(\mathbf{A} + \mathbf{BCE}\right)^{-1} = \mathbf{A}^{-1} - \mathbf{A}^{-1}\mathbf{B}\left(\mathbf{EA}^{-1}\mathbf{B} + \mathbf{C}^{-1}\right)\mathbf{EA}^{-1} \tag{2.43}$$

where, $\mathbf{A}$ and $\mathbf{C}$ are square invertible matrices. When using the Matrix Inversion Lemma and demanding that $\mathbf{Q}_s$ and $\mathbf{Q}_\nu$ must be invertible (this is true if $\sigma_s^2 > 0$ and $\sigma_\nu^2 > 0$), $\mathbf{D}$ can be rewritten as follows:

$$\begin{aligned}
\mathbf{D} &= \mathbf{Q}_s\mathbf{H}^*\left(\mathbf{Q}_\nu + \mathbf{HQ}_s\mathbf{H}^*\right)^{-1} = \mathbf{Q}_s\mathbf{H}^*\left(\mathbf{Q}_\nu^{-1} - \mathbf{Q}_\nu^{-1}\mathbf{H}\left(\mathbf{H}^*\mathbf{Q}_\nu^{-1}\mathbf{H} + \mathbf{Q}_s^{-1}\right)^{-1}\mathbf{H}^*\mathbf{Q}_\nu^{-1}\right) \\
&= \mathbf{Q}_s\mathbf{H}^*\mathbf{Q}_\nu^{-1} - \mathbf{Q}_s\mathbf{H}^*\mathbf{Q}_\nu^{-1}\mathbf{H}\left(\mathbf{H}^*\mathbf{Q}_\nu^{-1}\mathbf{H} + \mathbf{Q}_s^{-1}\right)^{-1}\mathbf{H}^*\mathbf{Q}_\nu^{-1} \\
&= \left(\mathbf{Q}_s\left(\mathbf{H}^*\mathbf{Q}_\nu^{-1}\mathbf{H} + \mathbf{Q}_s^{-1}\right) - \mathbf{Q}_s\mathbf{H}^*\mathbf{Q}_\nu^{-1}\mathbf{H}\right)\left(\mathbf{H}^*\mathbf{Q}_\nu^{-1}\mathbf{H} + \mathbf{Q}_s^{-1}\right)^{-1}\mathbf{H}^*\mathbf{Q}_\nu^{-1} \\
&= \left(\mathbf{H}^*\mathbf{Q}_\nu^{-1}\mathbf{H} + \mathbf{Q}_s^{-1}\right)^{-1}\mathbf{H}^*\mathbf{Q}_\nu^{-1} = \left(\frac{1}{\sigma_\nu^2}\mathbf{H}^*\mathbf{H} + \frac{1}{\sigma_s^2}\mathbf{I}\right)^{-1}\frac{1}{\sigma_\nu^2}\mathbf{H}^* = \left(\mathbf{H}^*\mathbf{H} + \alpha\mathbf{I}\right)^{-1}\mathbf{H}^*
\end{aligned} \tag{2.44}$$

where $\alpha = \sigma_\nu^2/\sigma_s^2 > 0$ because we have demanded that $\sigma_s^2 > 0$ and $\sigma_\nu^2 > 0$. However, a nice consequence of the last result is that if we substitute $\alpha = 0$ in it, we obtain the Zero Forcing solution.

So, to conclude, it can be said that in order to minimise the Mean Square Error (over $\mathbf{D}$), the processing at the receiver must be equal to:

$$\mathbf{D} = \left(\alpha\mathbf{I} + \mathbf{H}^*\mathbf{H}\right)^{-1}\mathbf{H}^*\ ,\ \alpha > 0 \tag{2.45}$$

where α is equal to $\sigma_\nu^2/\sigma_s^2 = N_t/\rho$. From Formula (2.45) it becomes clear that the ZF solution corresponds to an MMSE solution with $\alpha = 0$. On the other hand, the MMSE solution can be considered to be a ZF solution with:

$$\mathbf{H} \rightarrow \begin{bmatrix} \mathbf{H} \\ \sqrt{\alpha}\mathbf{I}_{N_t} \end{bmatrix} \text{ and } \mathbf{x} \rightarrow \begin{bmatrix} \mathbf{x} \\ 0 \end{bmatrix} \tag{2.46}$$

Notice that in practice it is hard to measure α, so, usually a fixed α that does not depend on the Signal-to-Noise Ratio is used instead.

2.4 Zero Forcing with Decision Feedback Decoding

The linear nulling approach as described in Section 2.2 is viable, but as will become clear from the results in Section 2.7, superior performance is obtained if non-linear techniques are used. One can

imagine that if somehow first the most reliable element of the transmitted vector **s** could be decoded and used to improve the decoding of the other elements of **s**, superior performance can be achieved. This is called *symbol cancellation* [60] and it exploits the timing synchronism inherent in the system model (the assumption of co-located transmitters makes this completely reasonable). Furthermore, linear nulling (i.e., ZF) or MMSE is used to perform detection. In other words, symbol cancellation is based on the subtraction of interference from already detected components of **s** from the receiver signal vector **x**. This results in a modified receiver vector in which, effectively, fewer interferers are present. Because this principle is somewhat analogous to decision feedback equalisation, it is called Decision Feedback Decoding (DFB).

When symbol cancellation is used, the order in which the components of **s** are detected becomes important to the overall performance of the system. To determine a good ordering of detection, the covariance matrix of the estimation error $\mathbf{s} - \mathbf{s}_{\text{est}}$ will be used. For ZF, this covariance matrix can be shown to be:

$$\begin{aligned} E\left[(\mathbf{s}-\mathbf{s}_{\text{est}})(\mathbf{s}-\mathbf{s}_{\text{est}})^*\right] &= E\left[\left(\mathbf{s}-\left(\mathbf{H}^*\mathbf{H}\right)^{-1}\mathbf{H}^*\mathbf{x}\right)\left(\mathbf{s}-\left(\mathbf{H}^*\mathbf{H}\right)^{-1}\mathbf{H}^*\mathbf{x}\right)^*\right] \\ &= E\left[\left(\mathbf{s}-\mathbf{s}-\left(\mathbf{H}^*\mathbf{H}\right)^{-1}\mathbf{H}^*\mathbf{v}\right)\left(\mathbf{s}-\mathbf{s}-\left(\mathbf{H}^*\mathbf{H}\right)^{-1}\mathbf{H}^*\mathbf{v}\right)^*\right] \\ &= E\left[\left(\mathbf{H}^*\mathbf{H}\right)^{-1}\mathbf{H}^*\mathbf{v}\mathbf{v}^*\mathbf{H}\left(\mathbf{H}^*\mathbf{H}\right)^{-1}\right] = \sigma_v^2\left(\mathbf{H}^*\mathbf{H}\right)^{-1} \hat{=} \sigma_v^2\mathbf{P} \end{aligned} \tag{2.47}$$

or, using the pseudo-inverse:

$$\mathbf{P} = \mathbf{H}^+\left(\mathbf{H}^+\right)^* \tag{2.48}$$

Let $(s_{\text{est}})_i$ be the i-th entry of $\mathbf{s}_{\text{est}}$, then, the "best" estimate, $(s_{\text{est}})_i$, is the one for which $\mathbf{P}_{ii}$ (i.e., the i-th diagonal element of **P**) is the smallest, because this is the estimate with the smallest error covariance. From Formula (2.48) it becomes clear that $\mathbf{P}_{ii}$ is equal to the squared length of the i-th row of the pseudo-inverse. So find the minimum squared length row of $\mathbf{H}^+$ is equivalent. Suppose that the order in the pseudo-inverse of **H** is arranged such that the row with the least squared length becomes the last row (the i-th row of $\mathbf{H}^+$ is permuted with the N_t-th row), then the N_t-th element of $\mathbf{s}_{\text{est}}$ can be independently decoded. Let $\hat{s}_{N_t}$ denote the decoded value, then this value can be used to improve the estimate of the remaining N_t–1 signals (i.e., symbol cancellation). If this procedure to find the "best" estimate is performed in a recursive way, the so-called Optimal Detection (OD) method as described in [60] is obtained. Here it is called the Decision Feedback Decoding algorithm with optimal detection. The recursive algorithm can be described as follows:

1. Compute $\mathbf{H}^+$;

2. Find the minimum squared length row of $\mathbf{H}^+$ and permute it to be the last row, permute the columns of **H** accordingly;

3. Form the estimate of the last component of **s**. In case of ZF: $\left(s_{\text{est}}\right)_{N_t} = \mathbf{H}_{N_t}^+\mathbf{x}$, where the transpose of $\mathbf{H}_{N_t}^+$ is said to be the N_t-th nulling vector [60];

4. Obtain $\hat{s}_{N_t}$ (via slicing) from $\left(s_{\text{est}}\right)_{N_t}$;

5. (While $N_t - 1 > 0$) go back to step 1, but now with:

$$\mathbf{H} \to \mathbf{H}^{(N_t-1)} = [\mathbf{h}_1 \quad \cdots \quad \mathbf{h}_{N_t-1}] \text{ , } \mathbf{x} \to \mathbf{x} - \mathbf{h}_{N_t}\hat{s}_{N_t} \text{ and } N_t \to N_t - 1.$$

Note that in case step 2 is skipped, the DFB algorithm is performed without optimal detection and the overall performance will be less, however, processing time is saved.

2.5 Minimum Mean Square Error with Decision Feedback Decoding

In order to perform Decision Feedback Decoding with Minimum Mean Square Error decoding, the DFB algorithm of Section 2.4 has to be adapted somewhat. Again, the covariance matrix of the estimation error $\mathbf{s} - \mathbf{s}_{\text{est}}$ will be used to determine a good ordering of detection. For MMSE, this covariance matrix can be shown to be:

$$\begin{aligned} E[(\mathbf{s}-\mathbf{s}_{\text{est}})(\mathbf{s}-\mathbf{s}_{\text{est}})^*] &= E[(\mathbf{s}-\mathbf{D}(\mathbf{Hs}+\mathbf{v}))(\mathbf{s}-\mathbf{D}(\mathbf{Hs}+\mathbf{v}))^*] \\ &= E[((\mathbf{I}-\mathbf{DH})\mathbf{s}-\mathbf{Dv})((\mathbf{I}-\mathbf{DH})\mathbf{s}-\mathbf{Dv})^*] \\ &= \sigma_s^2(\mathbf{I}-\mathbf{DH}-\mathbf{H}^*\mathbf{D}^*+\mathbf{DHH}^*\mathbf{D})+\sigma_v^2\mathbf{DD}^* \\ &= (\sigma_s^2\{\alpha\mathbf{I}+\mathbf{H}^*\mathbf{H}-\mathbf{H}^*\mathbf{H}-(\alpha\mathbf{I}+\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*\mathbf{H}(\alpha\mathbf{I}+\mathbf{H}^*\mathbf{H}) \\ &\quad +(\alpha\mathbf{I}+\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*\mathbf{HH}^*\mathbf{H}\}+\sigma_v^2(\alpha\mathbf{I}+\mathbf{H}^*\mathbf{H})^{-1}\mathbf{H}^*\mathbf{H})(\alpha\mathbf{I}+\mathbf{H}^*\mathbf{H})^{-1} \\ &= \sigma_v^2(\alpha\mathbf{I}+\mathbf{H}^*\mathbf{H})^{-1} \hat{=} \sigma_v^2\mathbf{P} \end{aligned} \tag{2.49}$$

Note that **P** is somewhat different from the case where ZF is used as detection. In order the do DFB based on the MMSE algorithm, the DFB algorithm is adapted and becomes:

1. Compute **D** (**P** is obtained while computing **D**);
2. Find the smallest diagonal entry of **P** and suppose this is the *i*-th entry. Permute the *i*-th column of **H** to be the last column and permute the rows of **D** accordingly;
3. Form the estimate of the "best" component of **s**: $(s_{\text{est}})_{N_t} = \mathbf{D}_{N_t}\mathbf{x}$, where $\mathbf{D}_{N_t}$ represents the last row of **D** and its transpose is the N_t-th nulling vector [60];
4. Obtain $\hat{s}_{N_t}$ (via slicing) from $(s_{\text{est}})_{N_t}$;
5. (While $N_t - 1 > 0$) go back to step 1, but now with:

$$\mathbf{H} \to \mathbf{H}^{(N_t-1)} = [\mathbf{h}_1 \quad \cdots \quad \mathbf{h}_{N_t-1}] \text{ , } \mathbf{x} \to \mathbf{x} - \mathbf{h}_{N_t}\hat{s}_{N_t} \text{ and } N_t \to N_t - 1.$$

The BER performance of DFB with ZF and DFB with MMSE will be compared in Section 2.7.

2.6 Maximum Likelihood Decoding

The only decoding method described in this report that is not based on calculation of matrix inverses is the Maximum Likelihood Decoding (MLD) algorithm. MLD is a method that compares the received signal with all possible transmitted signals and estimates **s** according to the Maximum

Likelihood principle. Suppose a matrix $\mathbf{C}$ gives all possibilities in $\mathbf{s}$ that could occur (the dimensions of $\mathbf{C}$ are $N_t \times K$, where $K = M^{N_t}$ and M represents the number of constellation points). Then, the receiver should store a matrix $\mathbf{Y}$ such that:

$$\mathbf{Y} = \mathbf{H} \cdot \mathbf{C} = [\mathbf{y}_1 \quad \cdots \quad \mathbf{y}_K] \tag{2.50}$$

At the receiver, the most likely transmitted signal is determined, as the one for which

$$\left\| \mathbf{x} - \mathbf{y}_j \right\|^2 \tag{2.51}$$

is minimal (with $1 \leq j \leq K$), i.e., the signal $\mathbf{s}_j$ that corresponds with the vector $\mathbf{y}_j$ which lays closed to the received vector is said to be the most likely signal to be transmitted. Thus, $\hat{\mathbf{s}}$ is chosen to be the j-th column of $\mathbf{C}$. This can be rewritten to the following formula where $\mathbf{s}_{\text{ml}}$ represents the maximum likelihood detection of the transmitted signal $\mathbf{s}$:

$$\hat{\mathbf{s}} = \mathbf{s}_{\text{ml}} = \arg \min_{\mathbf{s}_j \in \{\mathbf{s}_1, \ldots, \mathbf{s}_K\}} \left\| \mathbf{x} - \mathbf{H}\mathbf{s}_j \right\|^2 \tag{2.52}$$

Note that in the case of MLD, it is not required that $N_t \leq N_r$.

Another way to arrive at this result is saying that we want to find the vector $\mathbf{s}_j$ for which the probability $P(\mathbf{s}_j|\mathbf{x})$ is maximal (with $1 \leq j \leq K$). Note that according to [34], finding such a vector leads to the minimisation of the probability of error.

Using Bayes' rule, the probabilities may be expressed as:

$$P(\mathbf{s}_j|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{s}_j)P(\mathbf{s}_j)}{p(\mathbf{x})} \tag{2.53}$$

where $p(\mathbf{x}|\mathbf{s}_j)$ is the conditional probability density function of the observed vector given by $\mathbf{s}_j$, and $P(\mathbf{s}_j)$ is the probability of the j-th signal being transmitted. If the K signals are equally probable to be transmitted, then $P(\mathbf{s}_j) = 1/K$. Furthermore, the denominator in (2.53) is independent of which signal is transmitted. Consequently, the decision rule based on finding the signal that maximises $P(\mathbf{s}_j|\mathbf{x})$ is equivalent to finding the signal that maximises $p(\mathbf{x}|\mathbf{s}_j)$.

The (conditional) probability density function $p(\mathbf{x}|\mathbf{s}_j)$ is a multivariate normal distribution. The multivariate normal distribution of a real Gaussian random vector $\mathbf{x}'$ with a covariance matrix $\mathbf{Q}'$ is equal to [3] (where the $'$ denotes the fact that the elements of the vector/matrix are real):

$$p(\mathbf{x}') = \det(2\pi\mathbf{Q}')^{-1/2} \exp\left(-\frac{1}{2}(\mathbf{x}' - \boldsymbol{\mu}')^* \mathbf{Q}'^{-1}(\mathbf{x}' - \boldsymbol{\mu}') \right) \tag{2.54}$$

Following the same approach as in [47], this formula can be extended to the complex case. Here, a summary will be given. A complex random vector $\mathbf{x}$ is said to be Gaussian if the real random vector $\mathbf{x}'$:

$$\mathbf{x}' = \begin{bmatrix} \text{Re}(\mathbf{x}) \\ \text{Im}(\mathbf{x}) \end{bmatrix} \in \Re^{2N_r} \tag{2.55}$$

is Gaussian. We will say that a complex Gaussian random vector **x** is *special* if the covariance of the corresponding **x′** has the special structure

$$\mathbf{Q}' = E\left[(\mathbf{x}'-\boldsymbol{\mu}')(\mathbf{x}'-\boldsymbol{\mu}')^*\right] = \frac{1}{2}\begin{bmatrix} \mathrm{Re}(\mathbf{Q}) & -\mathrm{Im}(\mathbf{Q}) \\ \mathrm{Im}(\mathbf{Q}) & \mathrm{Re}(\mathbf{Q}) \end{bmatrix} \tag{2.56}$$

where **Q** represents the covariance matrix of the complex vector **x**. Note that the covariance matrix **Q** is Hermitian: $\mathbf{Q} = \mathbf{Q}^*$ (see Formula (2.35)). To transform the real multivariate normal distribution to the complex case, we need a number of properties that exist for the mapping:

$$\mathbf{z} \to \mathbf{z}' = \begin{bmatrix} \mathrm{Re}(\mathbf{z}) \\ \mathrm{Im}(\mathbf{z}) \end{bmatrix} \text{ and } \mathbf{A} \to \mathbf{A}' = \begin{bmatrix} \mathrm{Re}(\mathbf{A}) & -\mathrm{Im}(\mathbf{A}) \\ \mathrm{Im}(\mathbf{A}) & \mathrm{Re}(\mathbf{A}) \end{bmatrix} \tag{2.57}$$

These properties are:

$$\begin{aligned} &\mathbf{C} = \mathbf{AB} \Leftrightarrow \mathbf{C}' = \mathbf{A}'\mathbf{B}' \\ &\mathbf{C} = \mathbf{A}^{-1} \Leftrightarrow \mathbf{C}' = \mathbf{A}'^{-1} \\ &\det(\mathbf{A}') = \left|\det(\mathbf{A})\right|^2 = \det(\mathbf{A}\mathbf{A}^*) \\ &\mathbf{y} = \mathbf{A}\mathbf{x} \Leftrightarrow \mathbf{y}' = \mathbf{A}'\mathbf{x}' \\ &\mathrm{Re}(\mathbf{x}^*\mathbf{y}) = \mathbf{x}'^*\mathbf{y}' \end{aligned} \tag{2.58}$$

The first, fourth and fifth property are straightforward. The second property follows from the first and the fact that:

$$\mathbf{I}_{N_r} \Leftrightarrow \mathbf{I}' = \mathbf{I}_{2N_r} \tag{2.59}$$

The third property follows from:

$$\det(\mathbf{A}') = \det\left(\begin{bmatrix} \mathbf{I} & i\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\mathbf{A}'\begin{bmatrix} \mathbf{I} & -i\mathbf{I} \\ \mathbf{0} & \mathbf{I} \end{bmatrix}\right) = \det\left(\begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathrm{Im}(\mathbf{A}) & \mathbf{A}^* \end{bmatrix}\right) = \det(\mathbf{A})\det(\mathbf{A})^* \tag{2.60}$$

where, only in this case, the * stands for the conjugate of a vector/matrix (and not the transpose conjugate of a vector/matrix).

Using Formula (2.56), we can say that $\mathbf{Q}' = \mathbf{A}'/2$ and that $\mathbf{Q} = \mathbf{A}$. Now, to make the transformation more clear, we split the probability density function of the real multivariate normal distribution given by Formula (2.54) in two terms. The first term to transform is the determinant and the second is the exponent. The transformation of the determinant term is based on the third property of (2.58):

$$\det(2\pi\mathbf{Q}')^{-1/2} = \det(\pi\mathbf{A}')^{-1/2} = \left(\left|\det(\pi\mathbf{A})\right|^2\right)^{-1/2} = \det(\pi\mathbf{Q})^{-1} \tag{2.61}$$

where the last equality is obtained using the fact that **Q** is a nonnegative definite matrix (see Formula (2.36)). Furthermore, it is shown below Formula (2.36) that the eigenvalues of a nonnegative definite matrix are real and nonnegative. This again leads to the observation that the determinant of a nonnegative difinite matrix is also real and nonnegative, because the determinant is equal to the product of the eigenvalues [45]. This justifies the last equality of the formula above.

The second term of the probability density function given by Formula (2.54) (i.e., the exponent term), can be transformed to the complex case by using the second, fourth and fifth property of (2.58) and the substituations $\mathbf{a}' = \mathbf{x}' - \boldsymbol{\mu}'$ and $\mathbf{z}' = \mathbf{A}'^{-1}\mathbf{a}'$:

$$\begin{aligned}\exp\left(-\tfrac{1}{2}(\mathbf{x}'-\boldsymbol{\mu}')^{*}\mathbf{Q}'^{-1}(\mathbf{x}'-\boldsymbol{\mu}')\right) &= \exp\left(-\mathbf{a}'^{*}\mathbf{A}'^{-1}\mathbf{a}'\right)\\ &= \exp\left(-\mathbf{a}'^{*}\mathbf{z}'\right) = \exp\left(-\mathrm{Re}\left(\mathbf{a}^{*}\mathbf{z}\right)\right)\\ &= \exp\left(-\mathrm{Re}\left((\mathbf{x}-\boldsymbol{\mu})^{*}\mathbf{A}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)\right)\\ &= \exp\left(-(\mathbf{x}-\boldsymbol{\mu})^{*}\mathbf{Q}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right)\end{aligned} \tag{2.62}$$

where the last equality is obtained from the fact that, for a Hermitian matrix **Q** and for all complex vectors **z**, the number $\mathbf{z}^{*}\mathbf{Q}\mathbf{z}$ is real [45]. As a proof, we can compute $(\mathbf{z}^{*}\mathbf{Q}\mathbf{z})^{*}$. We expect to get the conjugate of the 1 by 1 matrix $\mathbf{z}^{*}\mathbf{Q}\mathbf{z}$, but instead, we get the same number back: $(\mathbf{z}^{*}\mathbf{Q}\mathbf{z})^{*} = \mathbf{z}^{*}\mathbf{Q}^{*}(\mathbf{z}^{*})^{*} = \mathbf{z}^{*}\mathbf{Q}\mathbf{z}$. So the number must be real.

Finally, if we combine the results, we get the complex form of the multivariate normal distribution:

$$p(\mathbf{x}) = \det(\pi\mathbf{Q})^{-1}\exp\left(-(\mathbf{x}-\boldsymbol{\mu})^{*}\mathbf{Q}^{-1}(\mathbf{x}-\boldsymbol{\mu})\right) \tag{2.63}$$

This leads to the following probability density function $p(\mathbf{x}|\mathbf{s}_j)$ for a specific **H**:

$$p(\mathbf{x}|\mathbf{H},\mathbf{s}_j) = \det(\pi\mathbf{Q})^{-1}\exp\left(-(\mathbf{x}-\mathbf{H}\mathbf{s}_j)^{*}\mathbf{Q}^{-1}(\mathbf{x}-\mathbf{H}\mathbf{s}_j)\right) \tag{2.64}$$

With **Q** equal to:

$$\begin{aligned}\mathbf{Q} &= E\left[(\mathbf{x}-\boldsymbol{\mu})(\mathbf{x}-\boldsymbol{\mu})^{*}\right]\\ &= E\left[(\mathbf{x}-\mathbf{H}\mathbf{s}_j)(\mathbf{x}-\mathbf{H}\mathbf{s}_j)^{*}\right] = E\left[\mathbf{v}\mathbf{v}^{*}\right] = \sigma_v^2\mathbf{I}_{N_r}\end{aligned} \tag{2.65}$$

this results in the conditional probability density function:

$$p(\mathbf{x}|\mathbf{H},\mathbf{s}_j) = \det(\pi\mathbf{Q})^{-1}\exp\left(-\frac{1}{\sigma_v^2}(\mathbf{x}-\mathbf{H}\mathbf{s}_j)^{*}(\mathbf{x}-\mathbf{H}\mathbf{s}_j)\right) \tag{2.66}$$

Consequently, finding the maximum of the conditional probability $P(\mathbf{s}_j|\mathbf{x})$ leads to:

$$\arg\max_{\mathbf{s}_j\in\{s_1,\ldots,s_K\}} p(\mathbf{x}|\mathbf{H},\mathbf{s}_j) = \arg\min_{\mathbf{s}_j\in\{\mathbf{s}_1,\ldots,\mathbf{s}_K\}}\left\|\mathbf{x}-\mathbf{H}\mathbf{s}_j\right\|^2 = \mathbf{s}_{\mathrm{ml}} \tag{2.67}$$

which equals Formula (2.52) and because finding the maximum of the conditional probability $P(\mathbf{s}_j|\mathbf{x})$ leads to the minimisation of the error probability [34], MLD is optimal in terms of BER performance. However, a major disadvantage is that the complexity of MLD is proportional to M^{N_t}, due to the fact that the size of **Y** grows exponentially with N_t (see Section 3.5).

Another way to show the superiority in BER performance of MLD over the other SDM techniques is by checking its diversity order. Here, the diversity order of a MLD system, with an antenna configuration of $(N_t,N_r) = (2,2)$ and with a BPSK modulation scheme, will be determined and conclusions about the diversity order will be drawn. For a general proof of the diversity order of a MLD system, one is referred to [51].

We will use the union upperbound that provides a representative estimate for the average probability of symbol errors for a particular modulation signal [36]:

$$P_s\left(\varepsilon\middle|\mathbf{H},\mathbf{s}_i\right)\leq \sum_{\substack{j=1\\ j\neq i}}^{K} Q\left(\frac{d_{ij}}{\sqrt{2N_0}}\right) \tag{2.68}$$

where $K = M^{N_t}$ with M representing the number of constellation points and d_{ij} is the Euclidean distance between the i-th and j-th signal point in the by the channel matrix adapted constellation.

Assuming a BSPK modulation scheme, there are four possible symbols to transmit:

$$\mathbf{s}_1 = \begin{pmatrix}\sqrt{E_b/2}\\ \sqrt{E_b/2}\end{pmatrix}, \mathbf{s}_2 = \begin{pmatrix}\sqrt{E_b/2}\\ -\sqrt{E_b/2}\end{pmatrix}, \mathbf{s}_3 = \begin{pmatrix}-\sqrt{E_b/2}\\ \sqrt{E_b/2}\end{pmatrix} \text{and } \mathbf{s}_4 = \begin{pmatrix}-\sqrt{E_b/2}\\ -\sqrt{E_b/2}\end{pmatrix} \tag{2.69}$$

Furthermore, suppose that the channel matrix is given by:

$$\mathbf{H} = \begin{pmatrix}h_{11} & h_{12}\\ h_{21} & h_{22}\end{pmatrix} \tag{2.70}$$

Now, we can determine the average probability of error on the condition that e.g. $\mathbf{s}_1$ is transmitted. In order to find $P_s\left(\varepsilon\middle|\mathbf{s}_1,\mathbf{H}\right)$, we need to calculate three distances, namely, d_{12}, d_{13} and d_{14}:

$$\begin{aligned}
d_{12} &= \left\|\mathbf{H}\mathbf{s}_1 - \mathbf{H}\mathbf{s}_2\right\| = \left\|\begin{pmatrix}2h_{12}\sqrt{E_b/2}\\ 2h_{22}\sqrt{E_b/2}\end{pmatrix}\right\| = \sqrt{2E_b}\sqrt{\left|h_{12}\right|^2+\left|h_{22}\right|^2}\\
d_{13} &= \left\|\mathbf{H}\mathbf{s}_1 - \mathbf{H}\mathbf{s}_3\right\| = \left\|\begin{pmatrix}2h_{11}\sqrt{E_b/2}\\ 2h_{21}\sqrt{E_b/2}\end{pmatrix}\right\| = \sqrt{2E_b}\sqrt{\left|h_{11}\right|^2+\left|h_{21}\right|^2}\\
d_{14} &= \left\|\mathbf{H}\mathbf{s}_1 - \mathbf{H}\mathbf{s}_4\right\| = \left\|\begin{pmatrix}2h_{11}\sqrt{E_b/2}+2h_{12}\sqrt{E_b/2}\\ 2h_{21}\sqrt{E_b/2}+2h_{22}\sqrt{E_b/2}\end{pmatrix}\right\|\\
&= \sqrt{2E_b}\sqrt{\left|h_{11}+h_{12}\right|^2+\left|h_{21}+h_{22}\right|^2} \leq \sqrt{2E_b}\sqrt{\left|h_{11}\right|^2+\left|h_{12}\right|^2+\left|h_{21}\right|^2+\left|h_{22}\right|^2}
\end{aligned} \tag{2.71}$$

When we fill in these results in the Q-function, we see a large resemblance to the diversity analysis of Section 2.2. The distances d_{12} and d_{13} involve a chi-square-distributed random variable with 2×2 degrees of freedom, whereas, the distance d_{14} involves a chi-square distribution with 2×4 degrees of freedom. We can conclude that the Q-terms that contain the distances d_{12} and d_{13} result in a second order of diversity and the Q-term of d_{14} leads to a fourth order of diversity. Because the different Q-terms are added according to (2.68), the Q-term that contains the distance d_{14} can be neglected for high SNR due to its higher order of diversity. Because a BSPK modulation scheme is used, we may use the diversity analysis performed in Section 2.2. Assuming that $p(\gamma_{12})$ and $p(\gamma_{13})$ are chi-squared probability density functions of variables with 2×2 degrees of freedom (L = 2), gives us the result:

$$\begin{aligned} P_s(\varepsilon|\mathbf{s}_1) &\le \int_0^\infty Q\left(\sqrt{2\gamma_{12}}\right)p(\gamma_{12})d\gamma_{12} + \int_0^\infty Q\left(\sqrt{2\gamma_{13}}\right)p(\gamma_{13})d\gamma_{13} \\ &= 2\cdot\left[\frac{1}{2}(1-\mu)\right]^2 \sum_{k=0}^{1}\binom{1+k}{k}\cdot\left[\frac{1}{2}(1+\mu)\right]^k \end{aligned} \tag{2.72}$$

where,

$$\gamma_{12} = \frac{1}{2}\frac{d_{12}^2}{2N_0} \text{ and } \gamma_{13} = \frac{1}{2}\frac{d_{13}^2}{2N_0} \tag{2.73}$$

because, the SNR per bit, γ_b, in case of two transmit antennas, is two times as small as γ_b from Formula (2.23).

Due to the assumption that the symbols given by (2.69) are equally likely to be transmitted and the fact that for a given **H** the distances between the symbols stay the same when a different symbol than $\mathbf{s}_1$ is transmitted, the final average probability of error becomes $P_s(\varepsilon) = P_s(\varepsilon|\mathbf{s}_1)$. So, a MLD system with two receive antennas has a diversity order of two. When we repeat this analysis for other antenna configurations (and other modulation schemes), we will find that the diversity order of a MLD system with N_r receive antennas is equal to N_r. This can be explained as follows: if we consider the general case with an **H** matrix with dimensions $N_r \times N_t$, the distances between two transmit vectors that only differ one symbol are the largest source for bit errors (e.g., d_{14} is neglected in the (2,2) case, because it is larger than d_{12} and d_{13}). So, for the general case, we find:

$$d_{12} = \|\mathbf{H}\mathbf{s}_1 - \mathbf{H}\mathbf{s}_2\| = \sqrt{\frac{4E_b}{N_t}}\sqrt{|h_{12}|^2 + |h_{22}|^2 + \ldots + |h_{N_r 2}|^2} \tag{2.74}$$

where the vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ are $N_t \times 1$ and given by,

$$\mathbf{s}_1 = \begin{pmatrix} \sqrt{E_b/N_t} \\ \sqrt{E_b/N_t} \\ \sqrt{E_b/N_t} \\ \vdots \\ \sqrt{E_b/N_t} \end{pmatrix} \text{ and } \mathbf{s}_1 = \begin{pmatrix} \sqrt{E_b/N_t} \\ -\sqrt{E_b/N_t} \\ \sqrt{E_b/N_t} \\ \vdots \\ \sqrt{E_b/N_t} \end{pmatrix} \tag{2.75}$$

We clearly see that d_{12} is chi-square-distributed with $2\times N_r$ degrees of freedom. Using the diversity analysis performed in Section 2.2 (in the same manner as is done for the (2,2) case), we find that the diversity order of a MLD system with N_r receive antennas is equal to N_r.

2.7 Simulation results

The SDM techniques, which are described in the former sections, are programmed in Matlab (see Appendix A) and some simulations are performed to obtain Bit Error Rate (BER) characteristics in order to compare the performance of the different SDM techniques. In Figure 2-3 the BERs for the different SDM techniques are depicted against E_b/N_0 per receiving antenna. An antenna configuration of $(N_t,N_r) = (2,2)$ is simulated. Furthermore, a BPSK modulation scheme is used and the data is transmitted without coding. Note that the signal model is based on the SNR (E_s/N_0) per

receive antenna. So, if E_b/N_0 per receiving antenna is given as input, this must be translated to E_s/N_0 per receive antenna. For an M-ary PSK modulation scheme or for M-ary QAM this is easily done using the following equation:

$$\frac{E_s}{N_0} = \frac{E_b}{N_0}\log_2 M \tag{2.76}$$

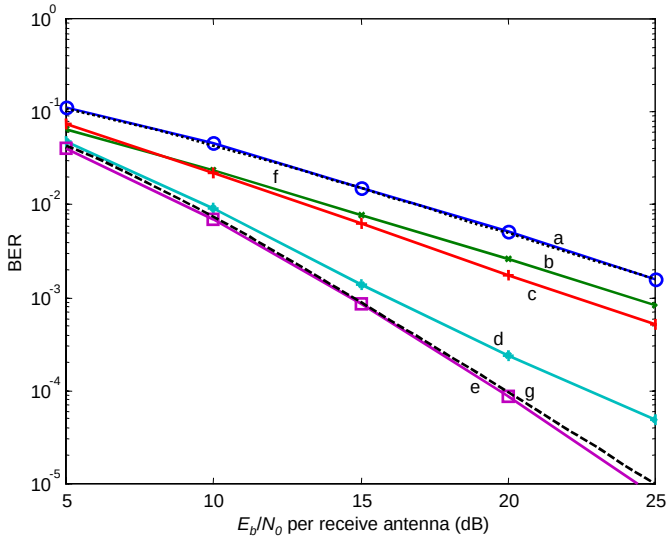

Figure 2-3: BER versus mean E_b/N_0 per receiving antenna for antenna configuration $(N_t,N_r) = (2,2)$, BPSK, no coding and with a) ZF, b) MMSE, c) ZF with DFB, d) MMSE with DFB, e) MLD, f) theoretical curve of ZF and g) upperbound of MLD.

From Figure 2-3 it can be seen that MLD has the best performance. Furthermore, it seems that the slope of the MLD method falls off with a second order diversity, whereas, the slope of the other methods tends to a curve with a first order of diversity. From the BER analysis given in Section 2.2 it can be deduced that the diversity order of the ZF algorithm for a system with two transmit and two receive antennas is equal to $N_r–N_t+1 = 2–2+1 = 1$. A first order diversity means that the BER becomes 10 times smaller if the SNR increases by 10 dB. The theoretical curve for ZF (see Formula (2.26)) is given by the dotted line (curve f) of Figure 2-3, but is hardly to see because the simulation of the ZF solution (curve a) gives almost the same result. This shows the correctness of the ZF simulation.

Furthermore, from Figure 2-3 it can be seen that the MLD curve (curve e) has a diversity of 2, because the BER decreases 10 times with a 5 dB increase in SNR. This is in agreement with the diversity analysis of Section 2.6, where it is said that the diversity order of a MLD system with 2 receive antennas is equal to 2. Despite the fact that the upperbound, for a MLD system with antenna configuration (2,2) given by Formula (2.72), is an upperbound for the *symbol* error rate, we will use it to verify the BER performance of this system. A MLD system that transmits BPSK symbols on two transmit antennas sends two bits per symbol (one bit on each transmit antenna). A symbol error occurs if one of these bits is decoded wrong or both bits are detected wrong. Assuming that these three possibilities occur equally likely, the average number of bit errors per symbol error is equal to 1/3 + 1/3 + 2/3 = 4/3. Because a symbol consists of 2 bits, the average bit error rate in case of a symbol error equals 2/3. Multiplying Formula (2.72) by 2/3 results in the dashed curve (curve g) of Figure 2-3. Thus, Formula (2.72), that has a diversity order of two, is indeed an upperbound for the (2,2) MLD case.

As observed before from Figure 2-3, the MLD method has the best performance. Its drawback, however, is that a large matrix **Y** needs to be calculated and stored (see Section 2.6). Especially in case of large constellation schemes and a large number of transmit antennas, this matrix becomes extremely large, because the size of **Y** grows exponentially with N_t. This results in an exponential growth of processing time and/or complexity in terms of increasing N_t (see Section 3.5).

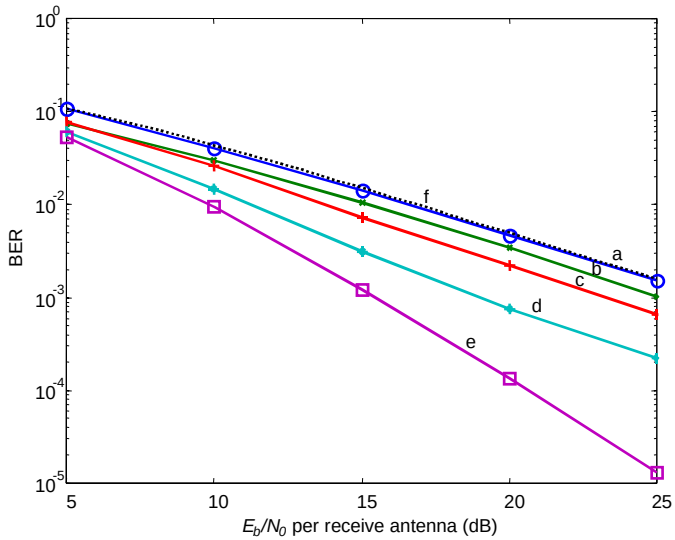

Figure 2-4: BER versus mean E_b/N_0 per receiving antenna for antenna configuration $(N_t,N_r) = (2,2)$, QPSK, no coding and with a) ZF, b) MMSE, c) ZF with DFB, d) MMSE with DFB, e) MLD and f) theoretical curve of ZF.

Figure 2-4 shows the same results as Figure 2-3, only instead of a BSPK modulation scheme, a QPSK modulation scheme is used. Curve f) shows the theoretical value, which again is obtained from [34]. Here, it is shown that the BER for a system with Maximal Ratio Combining with a QPSK modulation scheme and a Lth order of diversity is equal to:

$$P_e = \frac{1}{2}\left[1 - \frac{\mu}{\sqrt{2-\mu^2}} \sum_{k=0}^{L-1} \binom{2k}{k} \cdot \left(\frac{1-\mu^2}{4-2\mu^2}\right)^k\right] \tag{2.77}$$

where, by definition,

$$\mu = \sqrt{\frac{\gamma_c}{1+\gamma_c}} \tag{2.78}$$

In this formula γ_c represents the average received SNR per channel, thus, in case of multiple transmitting antennas γ_c is equal to E_s/N_0 divided by the number of transmit antennas N_t.

The largest difference between the results of Figure 2-3 and Figure 2-4 is the performance deterioration of the methods based on DFB in case of QPSK. Apparently, the DFB algorithm is performing less in case of more complex modulation schemes. It seems that in this case it is more difficult to find the “best estimate” in the DFB algorithm (see Section 2.4 and 2.5).

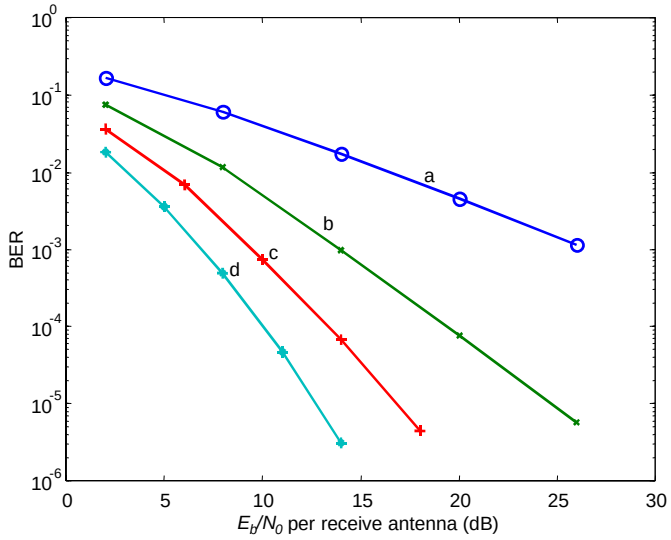

Figure 2-5: BER versus mean E_b/N_0 per receiving antenna for ZF, BPSK, no coding and for antenna configuration (N_t,N_r) equal to a) (2,2), b) (2,3), c) (2,4) and d) (2,5).

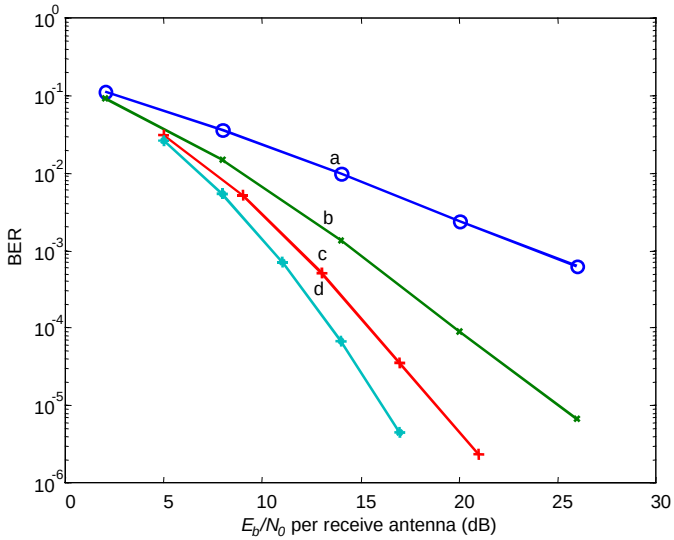

Figure 2-6: BER versus mean E_b/N_0 per receiving antenna for MLD, BPSK, no coding and for antenna configuration (N_t,N_r) equal to a) (1,1), b) (2,2), c) (3,3) and d) (4,4).

In order to check the diversity analysis, some simulations are performed with the ZF algorithm and with the MLD method, respectively, for different antenna configurations. The results are shown in Figure 2-5 and Figure 2-6. When we compare these figures, the difference in diversity between Zero Forcing and MLD becomes clear. A diversity order of one means that the BER decreases 10^1 times it the SNR increases by 10 dB. A diversity order of two implies a 10^2 times improvement of the BER with an SNR increase of 10 dB, and so on. From Figure 2-5 in can be seen that the diversity order of curve a), b), c) and d) is respectively equal to 1, 2, 3 and 4. Clearly, this diversity order satisfies the rule that the diversity order of a system with N_t transmit and N_r receive antennas based on ZF is equal to N_r–N_t+1 (as described in Section 2.2).

From Figure 2-6 it can be seen that the diversity order of a system based on the MLD technique is different from the diversity of ZF. The diversity of MLD is equal to the number of receive antennas

(N_r). So the diversity order of the curves a), b), c) and d) is respectively 1, 2, 3 and 4. It should be noted that the BER performance of a MLD system does not lose its diversity order if the number of transmitting antennas is increased, but the overall performance deteriorates (i.e., the curves of Figure 2-6 shift to the right when more transmit antennas are added).

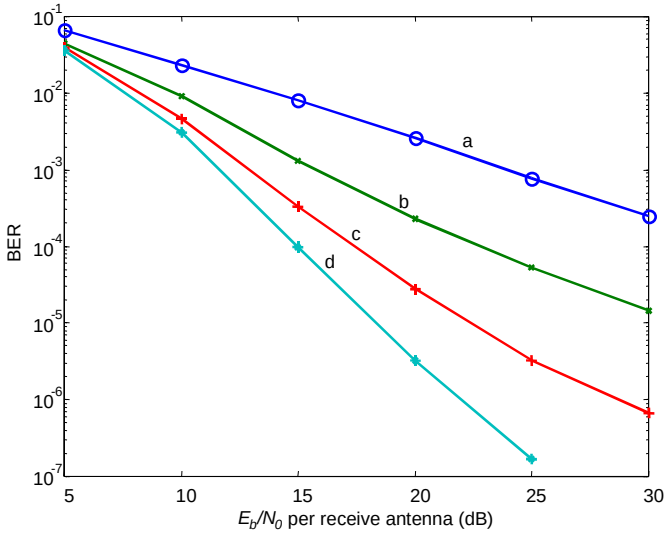

Figure 2-7: BER versus mean E_b/N_0 per receiving antenna for MMSE with DFB, BPSK, no coding and for antenna configurations (N_t,N_r): a) (1,1), b) (2,2), c) (3,3) and d) (4,4).

From Figure 2-3 it can be seen that the MMSE with DFB curve (curve d) follows the MLD curve (curve e) for low SNR values. For higher SNR, curve d is bending towards a curve with a first order of diversity and the difference in BER performance between MMSE with DFB and MLD increases. Therefore, it may be interesting to compare the result of Figure 2-6 with simulations of a system with MMSE and DFB and the same antenna configurations. These results are given in Figure 2-7. We will use an example to compare the results of Figure 2-6 and Figure 2-7. If we determine the SNR needed to achieve a BER of 10^{-4} for both systems with antenna configuration (N_t,N_r) = (4,4), then we see that the MLD system needs an SNR of approximately 13.5 dB, whereas the system with MMSE and DFB needs an SNR of 14.5 dB. So, the difference is approximately 1 dB. However, if we make the same comparison for a lower BER, then the difference in SNR increases due to the fact that all the curves of Figure 2-7 bend to a curve with first order of diversity for higher SNR values. So, in the range of a BER performance of 10^{-1} to 10^{-5}, MMSE with DFB has a slightly worse performance compared to MLD. But, the complexity of MMSE with DFB does not grow exponentially with the number of transmit antennas and, thus, especially for a larger number of transmit antennas, MMSE with DFB might be a good alternative.

Note that, when a more complex modulation scheme is chosen, the difference between MMSE with DFB and MLD increases (compare Figure 2-8 and Figure 2-9), because it seems that DFB is performing worse for more complex constellations.

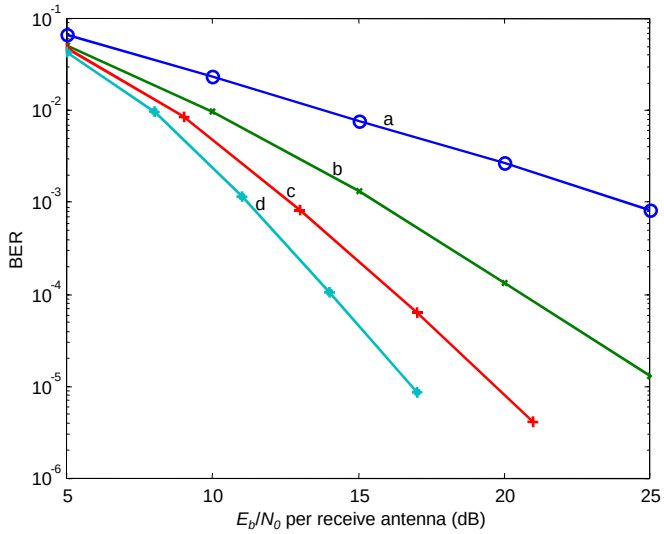

Figure 2-8: BER versus mean E_b/N_0 per receiving antenna for MLD, QPSK, no coding and for antenna configuration (N_t,N_r) equal to a) (1,1), b) (2,2), c) (3,3) and d) (4,4).

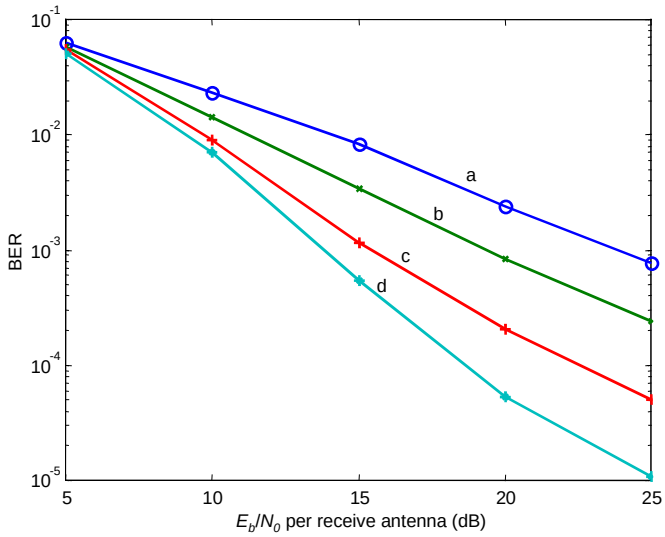

Figure 2-9: BER versus mean E_b/N_0 per receiving antenna for MMSE with DFB, QPSK, no coding and for antenna configurations (N_t,N_r): a) (1,1), b) (2,2), c) (3,3) and d) (4,4).

3 Complexity Analysis

Another way to compare the different SDM techniques is by determining the complexity of the different algorithms in terms of number of additions and multiplications. Such a complexity analysis is a good measure for the final complexity of the (receiver) hardware. However, it should be noticed that nothing is done to optimise the algorithms (for special hardware architectures), so, the results presented in this chapter are not binding.

The final result of the complexity analysis will be given in real additions (R_ADDs) and real multiplications(R_MULs). However, the signal model is complex and, thus, complex additions (C_ADDs) and complex multiplications (C_MULs) take place. Therefore, the complex calculations will be written in terms of real calculations. It is easy to see that one C_ADD consists of two R_ADDs (the real *and* the imaginary part of the two complex numbers are added). A complex multiplication can be rewritten in the following two ways [33]:

$$(a+ib)(c+id)=(ac-bd)+i(bc+ad) \tag{3.1}$$

$$(a+ib)(c+id)=(ac-bd)+i[(a+b)(c+d)-ac-bd] \tag{3.2}$$

The first option consists of 4 R_MULs (*ac*, *bd*, *bc* and *ad*) and 2 R_ADDs (*ac* – *bd* and *bc* + *ad*). The subtraction is counted as an addition and the addition before the *i* does not count because the real and imaginary parts are stored separately. The second option has only three real multiplications (*ac*, *bd*, (*a* + *b*)(*c* + *d*), plus five real additions. Compared with the first case, the total operations count is higher by two, but in some hardware architectures, multiplication is a more complex operation. In the rest of this chapter we will use the first option.

Furthermore, in the following sections more than once the complexity of a matrix multiplication needs to be determined. Therefore, the general case is considered here. Suppose two matrices **A** and **B** (real or complex) with dimensions $C \times D$ and $D \times E$ are multiplied:

$$\mathbf{A}_i \mathbf{b}_j = \sum_{k=1}^{D} a_{ik} b_{kj} \tag{3.3}$$

where $\mathbf{A}_i$ represents the *i*-th row of matrix **A**, $\mathbf{b}_j$ denotes the *j*-th column of **B** and a_{ik} and b_{kj} stand for the *k*-th element of this row and column, respectively. Thus, in order to obtain one element of the result matrix, D–1 additions and D multiplications need to be performed. The dimensions of the result matrix are $C \times E$ and, therefore, a total of $C(D–1)E$ additions and CDE multiplications (real or complex) are needed to multiply the two matrices **A** and **B**.

The complexity of the SDM techniques described in Chapter 2 can be divided in two parts: the complexity of the training phase and the complexity of the data phase. It is assumed that for a specific packet length the channel does not change. So, for each packet the channel matrix needs to be estimated only once and can be performed in the training phase. Furthermore, some channel matrix processing only depends on the channel matrix and, thus, also can be performed in the training phase. The operations that are required to process the received data are of course performed in the data phase. This processing needs to be carried out for each receiving vector **x**.

In this chapter we assume that the receiver already knows the channel matrix. Because the channel estimation does not depend on the SDM technique that is used it is a constant offset to the complexity results and, thus, the complexity of this part of the training is assumed known and will not be taken into account.

In the following sections the complexity of the Zero Forcing (ZF) algorithm, the Minimum Mean Square Error (MMSE) technique, Zero Forcing with Decision Feedback (DFB), Minimum Mean Square Error with Decision Feedback and Maximum Likelihood Decoding (MLD) will be determined.

3.1 Zero Forcing

As described in Section 2.2, the Zero Forcing technique is based on calculation of the pseudo-inverse of the channel matrix **H**. Because it is assumed that the channel matrix is constant for one packet, it is possible to calculate the pseudo-inverse in the training phase. In the data phase, the pseudo-inverse can be used during transmission of the whole packet to estimate the transmitted vectors **s**. In this section the complexity of the training phase and the data phase of the ZF algorithm will be determined.

3.1.1 The training phase

In the training phase, the pseudo-inverse of the channel matrix **H** is determined. For determining the complexity of the calculation of the pseudo-inverse we will use the equality that can be deduced from Section 2.2 and is given by:

$$\mathbf{H}^{+} = \left(\mathbf{H}^{*}\mathbf{H}\right)^{-1}\mathbf{H}^{*} \tag{3.4}$$

where, again, $^{+}$ stands for the pseudo-inverse of the specific matrix. The dimensions of $\mathbf{H}^{+}$, **H** and $\mathbf{H}^{*}$ are, respectively, $N_t \times N_r$, $N_r \times N_t$ and $N_t \times N_r$. Using the complexity analysis of a matrix product at the beginning of this chapter, the complexity of the product $\mathbf{H}^{*}\mathbf{H}$ can be found and is equal to $N_t^2(N_r-1)$ C_ADDs and $N_t^2 N_r$ C_MULs. The result is a square matrix with dimensions $N_t \times N_t$.

From this square matrix $\mathbf{H}^{*}\mathbf{H}$, the inverse needs to be determined. It is known that the direct inversion of a given square matrix **A** (with dimensions $N \times N$) has a complexity in the order of N^3 additions and N^3 multiplications in total [33]. So, inverting $\mathbf{H}^{*}\mathbf{H}$ has a complexity of N_t^3 C_ADDs and N_t^3 C_MULs.

Finally, the inverse of $\mathbf{H}^{*}\mathbf{H}$ (which is $N_t \times N_t$) is multiplied by $\mathbf{H}^{*}$. The complexity of this last multiplication is equal to $N_t(N_t-1)N_r$ C_ADDs and $N_t^2 N_r$ C_MULs. This leads to a total complexity of the training phase of $N_t^3 + N_t^2(N_r-1) + N_t(N_t-1)N_r$ C_ADDs and $N_t^3 + 2N_t^2 N_r$ C_MULs. Or in terms of real operations:

$$\begin{aligned} &4N_t^3 + 4N_t^2 N_r + 2N_t^2\left(N_r - 1\right) + 2N_t\left(N_t - 1\right)N_r \\ &\quad = 4N_t^3 + N_t^2\left(8N_r - 2\right) - 2N_t N_r \quad \text{R\_ADDs} \end{aligned} \tag{3.5}$$

and $4N_t^3 + 8N_t^2 N_r$ R_MULs.

3.1.2 The data phase

The data phase consists of a matrix-vector multiplication in order to obtain the estimates of the vector **s** and a slicing step to translate the estimated elements of **s** to the possible transmitted symbols. The matrix-vector multiplication can be found in Section 2.2 and is given by:

$$\mathbf{s}_{\text{est}} = \mathbf{H}^{+}\mathbf{x} \tag{3.6}$$

The complexity of this product is equal to $N_t(N_r-1)$ complex additions and N_tN_r complex multiplications.

The complexity of a slicer is minimal in terms of additions and/or multiplications. For the most modulation schemes only comparisons need to be made which have no or hardly any complexity in terms of additions and/or multiplications. Besides, the slicing step is equally complex for all demodulation schemes, and represents a constant offset to the complexity. Therefore, the complexity of the slicing step will not be taken into account. For clarity, however, the complexity of a BPSK slicer will be determined. In order to slice estimates of a BPSK modulation scheme, we only need to determine whether the real part of every element of **s** is less or larger than zero. If the real part of an element of **s** is larger than zero, most likely a 1 is transmitted, otherwise, –1 is transmitted. And if it is assumed that a comparison has equal complexity as a real addition, then, to slice N_t symbols, this would hold a complexity of N_t R_ADDs.

Summarising, the complexity of the data phase of the ZF algorithm is equal to $N_t(N_r-1)$ C_ADDs and N_tN_r C_MULs, or equivalently, $2N_tN_r + 2N_t(N_r-1)$ R_ADDs and $4N_tN_r$ R_MULs per transmitted vector **s** to decode, which is much less than the complexity of the training phase.

3.2 Minimum Mean Square Error

The complexity analysis of the Minimum Mean Square Error algorithm is almost equal to that of the Zero Forcing method described in Section 3.1. The data phase of the MMSE technique also consists of a matrix-vector product with the same dimensions and, thus, has equal complexity (see Section 3.1.2), namely, $2N_tN_r + 2N_t(N_r-1)$ R_ADDs and $4N_tN_r$ R_MULs for every transmitted vector **s** to decode.

In the training phase, the following processing matrix needs to be determined (see Section 2.3):

$$\mathbf{D} = \left(\alpha\mathbf{I} + \mathbf{H}^{*}\mathbf{H}\right)^{-1}\mathbf{H}^{*} \tag{3.7}$$

The calculation of the processing matrix has almost the same complexity as the determination of the pseudo-inverse in case of the ZF algorithm. Only a few extra additions need to be performed due to the addition of $\alpha\mathbf{I}$. Because α is real and $\mathbf{I}$ has dimensions $N_t \times N_t$, only N_t extra real additions need to be performed (i.e., the addition of α to the real part of the diagonal elements of $\mathbf{H}^{*}\mathbf{H}$). This leads to a total complexity of the training of $4N_t^3 + N_t^2(8N_r-2) - 2N_tN_r + N_t$ R_ADDs and $4N_t^3 + 8N_t^2N_r$ R_MULs.

3.3 Zero Forcing with Decision Feedback

The Decision Feedback algorithm (as described in Section 2.4) can also be divided in a training phase and a data phase. The training consists of determining the nulling vectors (step 1, 2 and the first part of step 5 of the algorithm described in Section 2.4), whereas the data phase consists of the DFB algorithm (step 3, 4 and the second part of step 5). For both phases, the complexity in terms of additions and multiplications will be analysed.

3.3.1 The training phase

In order to find the nulling vectors, an iterative algorithm that consists of three steps can be performed. First the steps are described and then the complexity will be determined:

1. Compute the pseudo-inverse of **H**;

2. Find the minimum squared length row of $\mathbf{H}^+$. This row is a nulling vector. Permute it to be the last row and permute the columns of **H** accordingly;

3. (While $N_t - 1 > 0$) go back to step 1, but now with:

$$\mathbf{H} \rightarrow \mathbf{H}^{(N_t-1)} = [\mathbf{h}_1 \quad \cdots \quad \mathbf{h}_{N_t-1}] \text{ and } N_t \rightarrow N_t - 1.$$

The complexity of step 1 (i.e., the calculation of the pseudo-inverse) is already determined in Subsection 3.1.1. For a matrix **H** with dimensions $N_r \times N_t$, this complexity is equal to $4N_t^3 + N_t^2(8N_r - 2) - 2N_tN_r$ R_ADDs and $4N_t^3 + 8N_t^2N_r$ R_MULs.

Step 2 consists of two tasks: calculation of the squared length of all rows of $\mathbf{H}^+$ and determination of the minimum squared length row. The squared length of the i-th row of the pseudo-inverse of the channel matrix can be derived by:

$$\begin{aligned} a_i &= \left|h_{i1}\right|^2 + \ldots + \left|h_{iN_r}\right|^2 \\ &= \left(\mathrm{Re}(h_{i1})\right)^2 + \left(\mathrm{Im}(h_{i1})\right)^2 + \ldots + \left(\mathrm{Re}(h_{iN_r})\right)^2 + \left(\mathrm{Im}(h_{iN_r})\right)^2 \end{aligned} \tag{3.8}$$

Clearly, $2N_r - 1$ real additions and $2N_r$ real multiplications have to be performed to calculate a_i. If $\mathbf{H}^+$ has N_t rows this leads to a total of $(2N_r - 1)N_t$ R_ADDs and $2N_rN_t$ R_MULs. However, from Section 2.4 it can be concluded that finding the minimum squared length row of **H** is equal to finding the minimum element $\mathbf{P}_{ii}$ on the diagonal of **P**. And due to the fact that **P** already is determined (for calculating the pseudo-inverse) no calculations need to be performed, but a_i can be set equal to $\mathbf{P}_{ii}$, where **P** is given by:

$$\mathbf{P} = \left(\mathbf{H}^*\mathbf{H}\right)^{-1} \tag{3.9}$$

In order to find the minimum of two numbers in hardware, the easiest thing to do is subtract the second number from the first and compare the result with zero. If the result is larger than zero, the second number is the smallest, otherwise the first number is the smallest. So, finding the minimum between two real numbers has the complexity of one real addition. To find the minimum of N_t values a, the following iterative process can be used:

a. $a_{\min} = a_1$, $i = 2$;

b. $a_{\min} = \min(a_{\min}, a_i)$;

c. while $i < N_t$ go to step b, with $i = i + 1$;

This iterative process has a complexity of $N_t - 1$ real additions. The permutations of step 2 are considered to have no complexity, the only thing that needs to be done is interchanging the memory pointers that respectively point to the two rows of $\mathbf{H}^+$ and the two columns of **H** that need to be permuted.

The third step of the algorithm to find the nulling vectors has hardly any complexity in terms of additions and/or multiplications. Note that the algorithm is repeated but now with $N_t = N_t - 1$ and, thus, the complexity of each iteration is reduced, because the dimensions of the matrices are reduced. To take along this reduction in complexity during the iteration, the total complexity can be written by using series. The total number of real additions becomes equal to:

$$\begin{aligned}
&4\sum_{n=1}^{N_t} n^3 + (8N_r - 2)\sum_{n=1}^{N_t} n^2 - 2N_r \sum_{n=1}^{N_t} n + \sum_{n=1}^{N_t} (n-1) \\
&= 4\sum_{n=1}^{N_t} n^3 + (8N_r - 2)\sum_{n=1}^{N_t} n^2 + (1-2N_r)\sum_{n=1}^{N_t} n - N_t \\
&= N_t^2 (N_t+1)^2 + (8N_r - 2)\frac{N_t(N_t+1)(2N_t+1)}{6} + (1-2N_r)\frac{N_t(N_t+1)}{2} - N_t \\
&= \frac{1}{6} N_t \left(6N_t^3 + 8N_t^2(1+2N_r) + 3N_t(1+6N_r) + 2N_r - 5\right)
\end{aligned} \tag{3.10}$$

The total number of real multiplications of the training phase of ZF with DFB is equal to:

$$\begin{aligned}
&4\sum_{n=1}^{N_t} n^3 + 8N_r \sum_{n=1}^{N_t} n^2 \\
&= N_t^2 (N_t+1)^2 + 8N_r \frac{N_t(N_t+1)(2N_t+1)}{6}
\end{aligned} \tag{3.11}$$

3.3.2 The data phase

During the data phase the nulling vectors are used to first estimate the best element of the transmitted vector **s**. The result is sliced to find the most likely transmitted constellation symbol and then, this symbol is used in the feedback loop in order to find the next best estimate. The following steps represent this iterative process:

1. Form the estimate of the best component (i.e., the last component due to permutation) of **s**. In case of ZF: $(s_{\text{est}})_{N_t} = \mathbf{H}_{N_t}^{+}\mathbf{x}$;

2. Obtain $\hat{s}_{N_t}$ (via slicing) from $(s_{\text{est}})_{N_t}$;

3. (While $N_t - 1 > 0$) go back to step 1, but now with:

   $\mathbf{x} \rightarrow \mathbf{x} - \mathbf{h}_{N_t}\hat{s}_{N_t}$ and $N_t \rightarrow N_t - 1$.

The complexity of the first step of this iterative algorithm equals N_r–1 C_ADDs and N_r C_MULs. Step 2 is assumed to have no complexity in terms of additions and multiplications as is explained in Subsection 3.1.2. Step 3 consists of a scalar-vector product and a vector subtraction. The scalar-vector product has a complexity that is equal to N_r C_MULs. The vector subtraction has, because subtraction is of equal complexity as addition, a complexity equal to N_r C_ADDs. These steps are performed N_t times.

In summary, it can be said that the complexity of the data phase of Zero Forcing with Decision Feedback equals $2N_t(4N_r-1)$ R_ADDs and $8N_tN_r$ R_MULs per transmitted vector **s**.

3.4 Minimum Mean Square Error with Decision Feedback

The complexity of Minimum Mean Square Error with Decision Feedback can be determined in the same way as it is done for Zero Forcing with DFB (see Section 3.3). Compared to ZF with DFB, there is a slight difference in the training phase in the way to obtain the nulling vectors. In case of MMSE, the iterative process of the training is given by:

1. Compute matrix $\mathbf{D} = \left(\alpha\mathbf{I} + \mathbf{H}^*\mathbf{H}\right)^{-1}\mathbf{H}^*$ (Note that $\mathbf{P} = \left(\alpha\mathbf{I} + \mathbf{H}^*\mathbf{H}\right)^{-1}\mathbf{H}^*$ is also obtained);

2. Find the smallest diagonal entry of **P** and suppose this is the *i*-th entry. Permute the *i*-th row of **D** to be the last row and permute the columns of **H** accordingly, the permuted row of **D** is a nulling vector;

3. (While $N_t - 1 > 0$) go back to step 1, but now with:

$$\mathbf{H} \rightarrow \mathbf{H}^{(N_t-1)} = [\mathbf{h}_1 \quad \cdots \quad \mathbf{h}_{N_t-1}] \text{ and } N_t \rightarrow N_t - 1.$$

Compared to ZF with DFB, the complexity of step 1 of the algorithm given in Subsection 3.3.1 is somewhat different. For a **H** matrix with dimensions $N_r \times N_t$ the complexity for determining matrix **D** is given in Section 3.2 and equals $4N_t^3 + N_t^2(8N_r-2) - 2N_tN_r + N_t$ R_ADDs and $4N_t^3 + 8N_t^2N_r$ R_MULs. Due to the iteration in the algorithm, this leads to a total complexity (including the complexity of finding the minimal diagonal element of **P**) of:

$$\begin{aligned}
&4\sum_{n=1}^{N_t} n^3 + (8N_r-2)\sum_{n=1}^{N_t} n^2 + (1-2N_r)\sum_{n=1}^{N_t} n + \sum_{n=1}^{N_t}(n-1) \\
&= 4\sum_{n=1}^{N_t} n^3 + (8N_r-2)\sum_{n=1}^{N_t} n^2 + 2(1-N_r)\sum_{n=1}^{N_t} n - N_t \\
&= N_t^2(N_t+1)^2 + (8N_r-2)\frac{N_t(N_t+1)(2N_t+1)}{6} + 2(1-N_r)\frac{N_t(N_t+1)}{2} - N_t \\
&= \frac{1}{3}N_t\left(3N_t^3 + 4N_t^2(1+2N_r) + 3N_t(1+3N_r) + N_r - 1\right) \text{ R_ADDs}
\end{aligned} \tag{3.12}$$

and

$$\begin{aligned}
&4\sum_{n=1}^{N_t} n^3 + 8N_r\sum_{n=1}^{N_t} n^2 \\
&= N_t^2(N_t+1)^2 + 8N_r\frac{N_t(N_t+1)(2N_t+1)}{6} \text{ R_MULs}
\end{aligned} \tag{3.13}$$

For the Minimum Mean Square Error algorithm with Decision Feedback, the data phase is exactly the same as the ZF technique with DFB, except that in this case the nulling vectors are rows of the processing matrix **D** in stead of rows of the pseudo-inverse of **H**. The last fact is irrelevant for the complexity, thus, the complexity of the data phase equals $2N_t(4N_r-1)$ R_ADDs and $8N_tN_r$ R_MULs per transmitted vector **s** (see Subsection 3.3.2).

3.5 Maximum Likelihood Decoding

The complexity of the Maximum Likelihood Decoding technique is largely dependent on the amount of memory that is available or that one is willing to use. We will compare two situations. Assume that in the first situation hardly any memory is available, but enough to store the products of the columns of **H** times the constellation points, thus, e.g. for BPSK it should be able to store ($\mathbf{H}_i$,$-\mathbf{H}_i$) for all $1 \le i \le N_t$. Note that it is not really necessary to store $-\mathbf{H}_i$, because it can easily be deduced from $\mathbf{H}_i$, so, actually two times less memory is required. Assume that in the other situation there is enough memory available to store all possibilities given by matrix $\mathbf{Y} = \mathbf{H}\cdot\mathbf{C}$ (see Section 2.6) in the memory during the training phase. In this case the overhead of recalculations like in the first situation is minimised, but matrix **Y** grows exponentially with the number of transmitting antennas (N_t) and, thus, a large memory may be required. In the following subsections the two situations will be compared.

3.5.1 Training phase (minimum amount of memory)

In the case that hardly any memory is used for Maximum Likelihood Decoding, a large part of the processing takes place in the data phase. The only processing needed in the training phase is the estimation of channel matrix **H** (which is assumed known) and the calculation of the product of the constellation values with the columns of **H**. If there are M constellation points, then the complexity of this product is MN_rN_t C_MULs or $2MN_rN_t$ R_ADDs and $4MN_rN_t$ R_MULs.

3.5.2 Data phase (minimum amount of memory)

As shown in Section 2.6 the maximum likelihood detection is given by:

$$\mathbf{s}_{\mathrm{ml}} = \arg \min_{\mathbf{s}_j \in \{\mathbf{s}_1,\ldots,\mathbf{s}_K\}} \left\| \mathbf{x} - \mathbf{H}\mathbf{s}_j \right\|^2 = \arg \min_{\mathbf{s}_j \in \{\mathbf{s}_1,\ldots,\mathbf{s}_K\}} \left\| \mathbf{x} - \sum_{i=1}^{N_t} \mathbf{H}_i \left(s_i\right)_j \right\|^2 \tag{3.14}$$

where $K = M^{N_t}$ and M represents the number of constellation points. Because $\mathbf{H}_i s_i$ is already calculated in the training phase for every constellation point, only complex additions need to be performed to find $\mathbf{x} - \mathbf{H}\mathbf{s}_j$. This calculation can be represented by a tree as shown in Figure 3-1 and its complexity is:

$$\sum_{i=1}^{N_t} M^i N_r = N_r M \frac{M^{N_t} - 1}{M - 1} \text{ C\_ADDs} \tag{3.15}$$

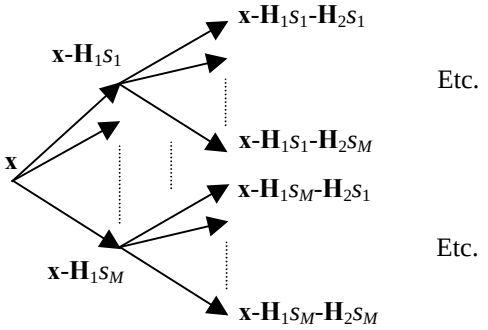

Figure 3-1: All possible vectors $\mathbf{x} - \mathbf{H}\mathbf{s}_j$.

The squared norm of the result can be calculated as follows:

$$\begin{aligned}\left\|\mathbf{x}-\mathbf{H}\mathbf{s}_j\right\|^2 &= \left|x_1-\mathbf{H}_1\mathbf{s}_j\right|^2+\ldots+\left|x_{N_r}-\mathbf{H}_{N_r}\mathbf{s}_j\right|^2 \\ &= \left(\mathrm{Re}\left(x_1-\mathbf{H}_1\mathbf{s}_j\right)\right)^2+\left(\mathrm{Im}\left(x_1-\mathbf{H}_1\mathbf{s}_j\right)\right)^2+\ldots+\left(\mathrm{Re}\left(x_{N_r}-\mathbf{H}_{N_r}\mathbf{s}_j\right)\right)^2 \\ &\quad +\left(\mathrm{Im}\left(x_{N_r}-\mathbf{H}_{N_r}\mathbf{s}_j\right)\right)^2\end{aligned} \tag{3.16}$$

This formula consists of $2N_r$ real multiplications and $2N_r$–1 real additions. The calculation of the squared norm described above needs to be performed K times (namely, for all K branches of the tree shown in Figure 3-1) and finally, the minimum of the K possible squared norms must be obtained which introduces K–1 extra real additions (see Subsection 3.3.1).

So, this yields a total complexity of $2KN_r$ R_MULs and

$$2N_rM\frac{K-1}{M-1}+2KN_r-1 \text{ R\_ADDs} \tag{3.17}$$

3.5.3 Training phase (maximal amount of memory)

If there is enough memory available to store the whole matrix $\mathbf{Y}$, this matrix can be determined in the training phase and used during the data phase. The complexity of the matrix-vector product $\mathbf{H}\mathbf{s}_j$ can be obtained using the tree of Figure 3-2 that shows all possible columns of $\mathbf{Y}$**:**

$$\mathbf{y}_j=\sum_{i=1}^{N_t}\mathbf{H}_i\left(s_i\right)_j \tag{3.18}$$

First the product of every column of $\mathbf{H}$ with every constellation point value should be determined. This has a total complexity of $2MN_rN_t$ R_ADDs and $4MN_rN_t$ R_MULs (see Subsection 3.5.1). After that, the additions as shown in the tree have to be performed. The complexity equals:

$$\sum_{i=2}^{N_t}M^iN_r=M^2N_r\frac{M^{N_t-1}-1}{M-1} \text{ C\_ADDs} \tag{3.19}$$

This gives a total complexity of $4MN_rN_t$ R_MULs and

$$2M^2 N_r \frac{M^{N_t-1}-1}{M-1} + 2MN_r N_t \text{ R\_ADDs} \tag{3.20}$$

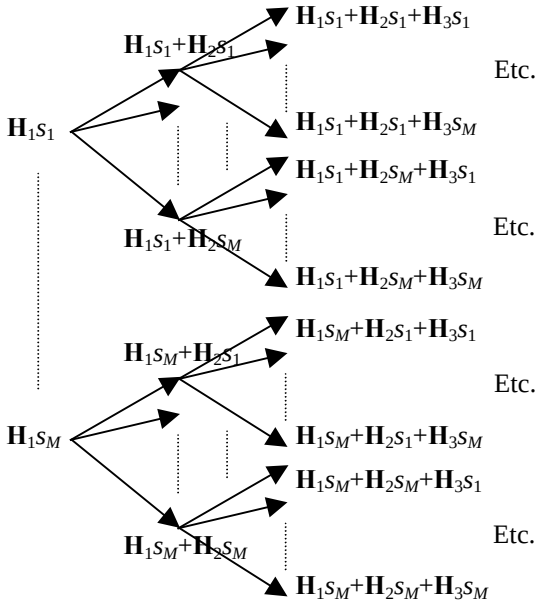

Figure 3-2: Tree notation of the matrix-vector product $\mathbf{Hs}_j$.

3.5.4 Data phase (maximal amount of memory)

During the data phase the vector subtraction $\mathbf{x} - \mathbf{Hs}_j$ and the squared norm of the result have to be determined for the K possible $\mathbf{s}$ vectors. The next step is to obtain the minimum of the squared norms. The vector subtraction $\mathbf{x} - \mathbf{Hs}_j$ is performed for $1 \le j \le K$ and has complexity equal to KN_r C_ADDs. Next, the norms of the K results have to be determined. This has a complexity of $K(2N_r - 1)$ R_ADDs and $2KN_r$ R_MULs. After that the minimum of the K (real) norms must be obtained which has a complexity equal to $K-1$ R_ADDs. So, this yields a total complexity of $4KN_r - 1$ R_ADDs and $2KN_r$ R_MULs. Note that these complexity figures increase linearly with the number of receiving antennas and they increase exponentially with the number of transmit antennas.

3.5.5 Less complex norm approximation

One of the significant terms in the complexity calculation for Maximum Likelihood Decoding is the determination of the norm of $\mathbf{x} - \mathbf{Hs}_j$. In Order to reduce the complexity, the following approximation will be used:

$$\begin{aligned} &\left\|\mathbf{x} - \mathbf{Hs}_j\right\| \approx \\ &\left|\mathrm{Re}\left(x_1 - \mathbf{H}_1\mathbf{s}_j\right)\right| + \left|\mathrm{Im}\left(x_1 - \mathbf{H}_1\mathbf{s}_j\right)\right| + \ldots + \left|\mathrm{Re}\left(x_{N_r} - \mathbf{H}_{N_r}\mathbf{s}_j\right)\right| + \left|\mathrm{Im}\left(x_{N_r} - \mathbf{H}_{N_r}\mathbf{s}_j\right)\right| \end{aligned} \tag{3.21}$$

It consists only of real additions (and no multiplications) making the MLD algorithm less complex. The drawback of MLD with the approximated norm is that the BER performance deteriorates by approximately 0.5 dB (compare the curves a and b of Figure 3-3 which show, respectively, the

BER performance of MLD with the normal norm definition and MLD with the approximation of the norm).

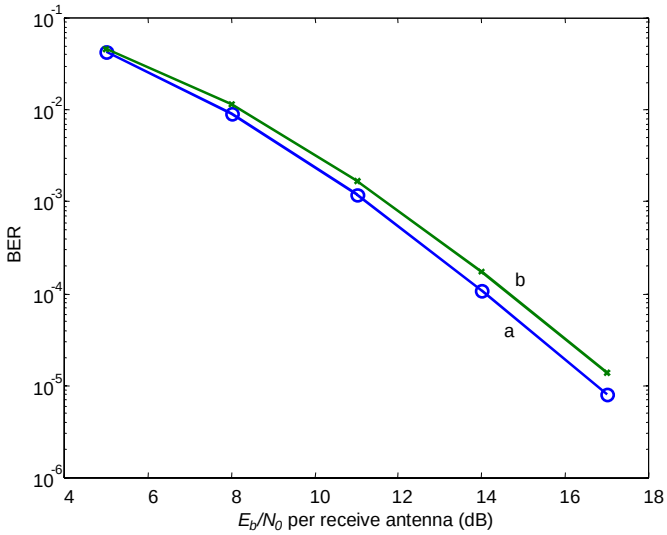

Figure 3-3: BER versus mean E_b/N_0 per receiving antenna for antenna configuration $(N_t,N_r) = (4,4)$, QPSK, no coding and a) MLD with regular norm definition and b) MLD with the approximation of the norm.

Now we will use this approximation of the norm in the situation of MLD with a minimum amount of memory. The approximation of Formula (3.21) does not make any changes to the training phase, so this complexity remains the same.

The data phase starts with the calculation of $\mathbf{x} - \mathbf{H}\mathbf{s}_j$ with complexity equal to (see Subsection 3.5.2):

$$\sum_{i=1}^{N_t} M^i N_r = N_r M \frac{M^{N_t} - 1}{M - 1} \text{ C\_ADDs} \tag{3.22}$$

After that, the approximation of the norm as defined by Formula (3.21) has to be calculated for the K possible $\mathbf{s}$ vectors. This process has a complexity of $K(2N_r-1)$ real additions. Finally, finding the minimum of the K (real) results has a complexity of $K-1$ R_ADDs.

This yields a total complexity of:

$$2N_r M \frac{K-1}{M-1} + 2KN_r - 1 \text{ R\_ADDs} \tag{3.23}$$

Clearly, this is a reduction in complexity compared to the results of Subsection 3.5.2.

3.6 Comparison

With the complexity analysis of the previous sections it is possible to compare the complexity of the different Space Division Multiplexing techniques. If we assume that the complexity of a real multiplier is 10 times higher than that of a real adder, then the complexity can be expressed in one

term. Of course the complexity difference between a multiplier and an adder depends on the hardware implementation, but a ratio of 10 is a reasonable one. If the complexity is expressed in one term, we can display the complexity of the training phase versus the complexity of the data phase. Furthermore, we can show the complexity for different antenna configurations. For a BPSK constellation scheme, the results are given in Figure 3-4 for antenna configurations (i,i) with $1 \leq i \leq 10$ (the complexity increases for increasing number of transmit and receive antennas).

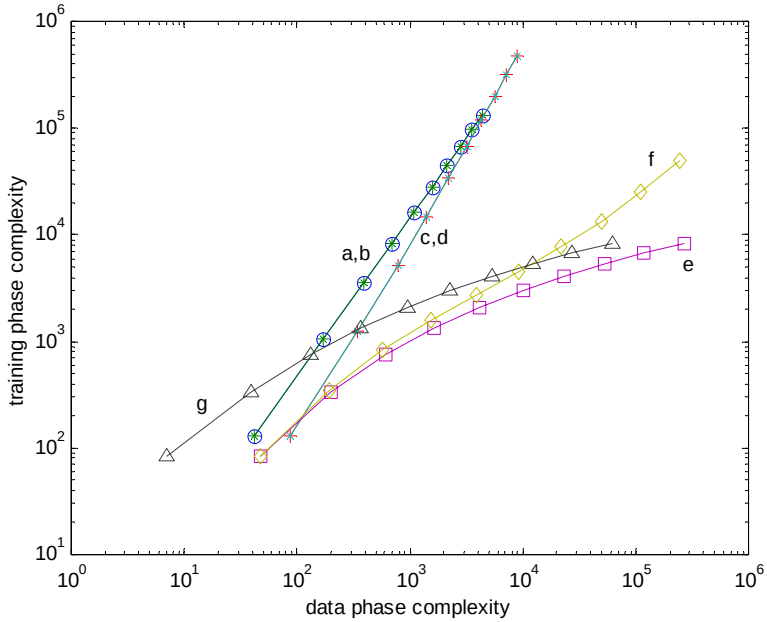

Figure 3-4: Complexity of the training phase versus the complexity of the data phase for BPSK, antenna configurations $(N_t,N_r) = (i,i)$ with $1 \leq i \leq 10$ for a) ZF, b) MMSE, c) ZF with DFB, d) MMSE with DFB, e) MLD with minimum amount of memory, f) MLD with maximal amount of memory and g) MLD with norm approximation and minimum amount of memory.

From Figure 3-4 it can be seen that the complexity of Zero Forcing (curve a) and Minimum Mean Square Error (curve b) are almost equal, this is also true for ZF with Decision Feedback (curve c) and MMSE with DFB (curve d). Comparing curve a with curve c shows that the complexity of the training phase and the data phase increases when DFB is introduced. So, in order to achieve the better BER performance of DFB, a more complex hardware architecture is required.

Furthermore, comparing curve e (Maximum Likelihood Decoding with minimum amount of memory) and curve f (MLD with maximum amount of memory) we could draw some useful conclusions. The complexity of the training phase of curve f increases exponentially, whereas the training complexity of curve e increases only with i^2, where $i = N_t = N_r$ (see Subsection 3.5.1). However, in case of MLD with maximum amount of memory (curve f), the benefit of the pre-calculations of al possible vectors $\mathbf{Hs}_j$ in the training phase can almost be neglected compared to the complexity of the data phase of curve e). This can be explained by the fact that the complexity of the data phase of both situations are proportional to K (see Subsections 3.5.2 and 3.5.4). So, in terms of complexity, MLD with minimum amount of memory is preferable.

Figure 3-4 also shows the complexity of MLD with a minimum amount of memory and with the norm approximation of Formula (3.21) (curve g). The complexity of the data phase of curve g is approximately 10 times less than that of curve e. A disadvantage is that the BER performance of MLD with the approximation of the norm is about 0.5 dB worse. However, even with this decrease in BER performance, it can be seen that for $i \leq 5$ the data phase of curve g is less complex than the

data phase of ZF and its performance is better (i-th order of diversity, whereas ZF only has a performance with a first order of diversity (if $N_t = N_r$)).

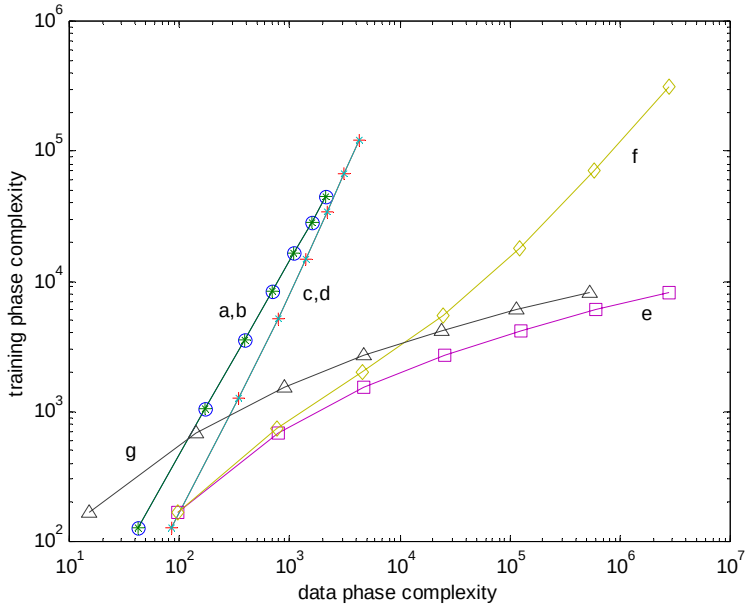

Figure 3-5: Complexity of the training phase versus the complexity of the data phase for QPSK, antenna configurations $(N_t,N_r) = (i,i)$ with $1 \leq i \leq 10$ for a) ZF, b) MMSE, c) ZF with DFB, d) MMSE with DFB, e) MLD with minimum amount of memory, f) MLD with maximal amount of memory and g) MLD with approximation of the norm.

Because the complexity of the MLD algorithms depends on the constellation size (M), it is useful to also compare the complexity for larger constellation schemes. In Figure 3-5 the result is given for a QPSK constellation scheme, with antenna configurations (i,i), where $1 \leq i \leq 7$ (the complexity increases for increasing number of transmit and receive antennas). Again, the complexity is expressed in one number by choosing the multiplications 10 times more complex than the additions and the training phase complexity is depicted versus the data phase complexity.

From Figure 3-5 it can be concluded that MLD with the norm approximation (curve g) is only preferable in data phase complexity if $i \leq 2$ with $i = N_t = N_r$, compared to $i \leq 5$ for BPSK. So, indeed, the complexity of the MLD techniques (strongly) depends on the size of the constellation scheme.

4 Orthogonal Frequency Division Multiplexing (OFDM)

One way to increase system capacity is by transmitting data parallel on different frequencies. In classical parallel data systems, the total signal frequency band is divided into *N non-overlapping* frequency subchannels. In such systems, there is sufficient guard space between adjacent subchannels to isolate them at the receiver with *N* demodulators, i.e., one per subchannel. This solution, however, is very bandwidth inefficient. A more efficient use of bandwidth can be obtained with a parallel system if the spectra of the individual subchannels are permitted to overlap, with specific orthogonality constraints imposed to facilitate separation of the subchannels at the receiver.

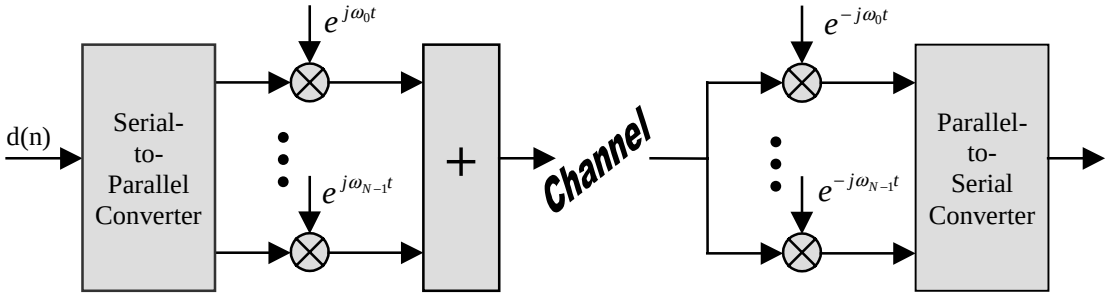

Figure 4-1: basic structure of a multicarrier system.

In the nearby future, more and more applications that operate on carrier frequencies in the order of several Giga-Hertz, like wireless LANs, will be based on multicarrier systems. Figure 4-1 shows the general structure of a multicarrier system [44]. An example of a multicarrier technique that operates with specific orthogonality constraints is Orthogonal Frequency Division Multiplexing (OFDM).

4.1 The principle

In OFDM the subcarrier pulse used for transmission is chosen to be rectangular. This has the advantage that the task of pulse forming and modulation can be performed by a simple Discrete Fourier Transform (DFT) which results in a remarkable reduction in equipment complexity (filters, modulators, etc.), as shown in [6] and [56]. In these references it is also shown that a multitone data signal is effectively the Fourier transform of the original data stream, and that a bank of coherent demodulators at the receiver is effectively an inverse Fourier transform. However, since definitions have been changes afterwards, it is now said, that a multitone data signal is effectively the inverse Fourier transform of the original data stream, and that a bank of demodulators is effectively a Fourier transform. Note that the (Inverse) Discrete Fourier Transform ((I)DFT) can be implemented very efficiently as a(n) (Inverse) Fast Fourier Transform ((I)FFT).

Consider the system shown in Figure 4-1. The input consists of a data symbol sequence d_n that can be represented as $a_n + jb_n$ (where a_n and b_n are real sequences representing the in-phase and quadrature components, respectively) and, using the Inverse Fourier Transform, the transmitted waveform can be represented as:

$$D(t) = \sum_{n=0}^{N-1} d_n e^{j\omega_n t} \tag{4.1}$$

However, for digital systems it is useful to consider the discrete case. In this case a Inverse Discrete Fourier Transform (IDFT) is performed at the transmitter on a vector $d = (d_0, d_1, \ldots, d_{N-1})$, the result is a vector $D = (D_0, D_1, \ldots, D_{N-1})$ with:

$$D_m = \sum_{n=0}^{N-1} d_n e^{j(2\pi mn/N)} = \sum_{n=0}^{N-1} d_n e^{j(2\pi f_n t_m)} \quad \text{with } m = 0,1,\ldots,N-1 \tag{4.2}$$

where

$$f_n = n\Delta f \;,\; \Delta f = \frac{1}{N\Delta t} \text{ and } t_m = m\Delta t \tag{4.3}$$

and $\Delta t = 1/f_s$ (where f_s is the symbol rate). Substituting $t = m\Delta t$ in (4.1) and using (4.3) shows that the sampled sequence $D(m\Delta t)$ is in fact the IDFT of the sequence d_n as shown in (4.2). Note that usually the whole OFDM symbol is upconverted to a specific carrier frequency of the RF band.

If we again look to the OFDM symbol time it can be noticed that the signalling interval T has been increased to $N\Delta t$, which makes the system less susceptible to delay spread impairments. Note that the act of truncating the rectangular signal to the interval $(0, N\Delta t)$ in the time domain imposes a $\sin(x)/x$ frequency response on each subchannel with zeros at multiples of $1/T$, which is the orthogonality principle of OFDM, as sketched in Figure 4-2.

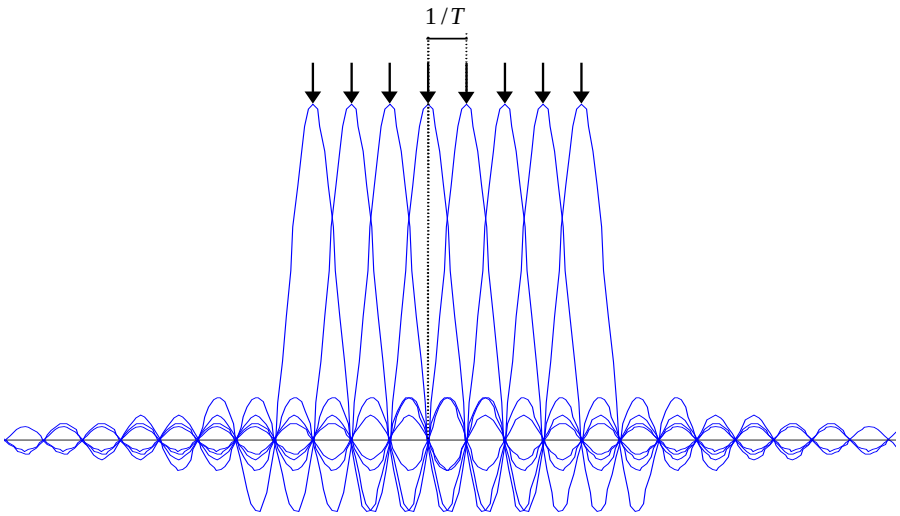

Figure 4-2: OFDM and the orthogonality principle.

At the receiver, the transmitted signals are recovered using the Discrete Fourier Transform. However, due to multipath distortion, the recovering of the transmitted signals brings some problems along (see Section 4.2).

4.2 Multipath Distortion

The reason why the information transmitted over the carriers can still be separated at the receiver is the so-called orthogonality relation giving OFDM its name. By using an IFFT for modulation, the spacing of the subcarriers is implicitly chosen in such a way, that at the frequencies where the

received signals are evaluated (indicated as arrows in Figure 4-2), all other signals are zero. In order for this orthogonality to be preserved the following must be true [44]:

1. The receiver and the transmitter must be perfectly synchronised. This means they both must assume exactly the same modulation frequency and the same time-scale for transmission (which usually is not the case).

2. The analogue components, part of transmitter and receiver, must be of very high quality.

3. There should be no multipath channel.

Unfortunately multipath distortion is (almost) unavoidable in radio communication systems and, thus, the received signal is affected. It is shown in [56] that the truncated subchannel sinusoids are delayed by differing amounts (i.e. channel delays), however, the distortion is mainly concentrated at the on-off transmissions of these waveforms. Hence a guard time and cyclic prefix (consisting of a modest increase in the signal duration) that are chosen longer than the maximal delay spread, will eliminate most interference among channels (Inter Carrier Interference (ICI)) and between adjacent transmission blocks (Inter Symbol Interference (ISI)). The smoothing of the on-off transitions, to avoid out of band radiation, can be implemented, e.g., by windowing each OFDM symbol by a raised cosine window (see Figure 4-3).

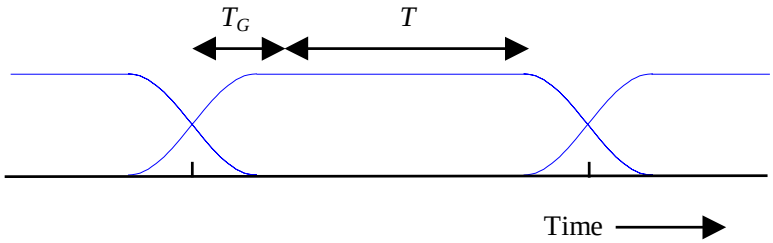

Figure 4-3: OFDM symbol structure showing guard time T_G and FFT interval T.

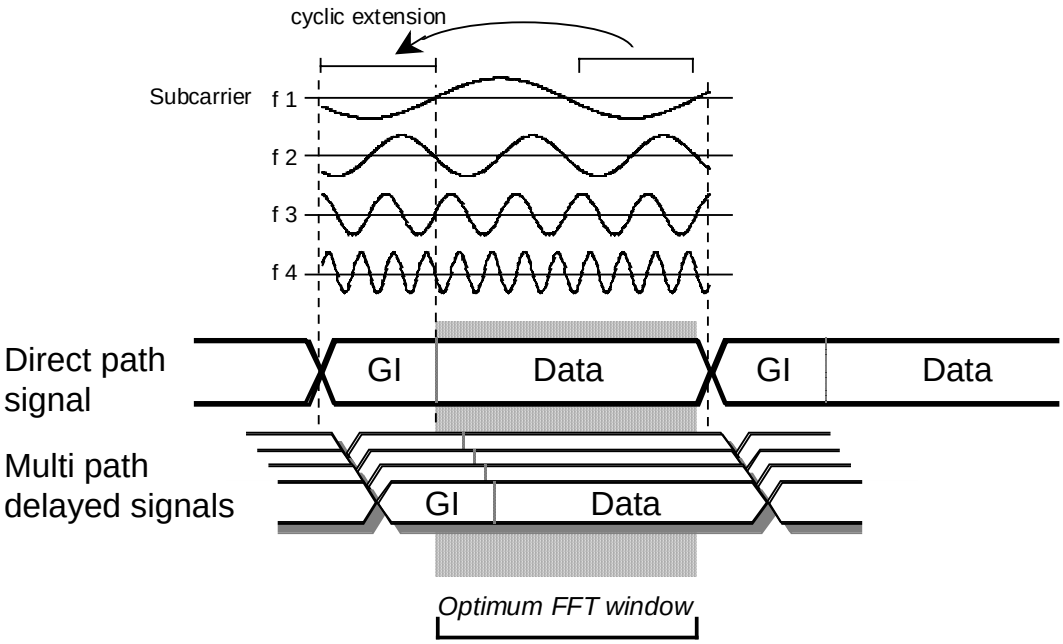

Figure 4-4: OFDM symbol with cyclic extension.

The validity of windowing each OFDM symbol by a raised cosine window can be explained by looking to the shapes of the subchannel sinusoids after the multipath-fading channel. An OFDM receiver uses only a part of this signal to calculate the FFT. In the FFT interval *T*, every subcarrier has an integer number of cycles, which ensures orthogonality. In the multipath-fading channel, the receiver-input signal will be a sum of delayed and scaled replicas of the transmitted subcarriers [52] (see Figure 4-4). Note that a sum of scaled and delayed sinusoids is again a sinusoide. So, as long as the Guard Interval (GI) time T_G is larger than the maximal channel delay there will still be an integer number of cycles within the FFT interval for each multipath component and all reflections of previous symbols are removed and the orthogonality is preserved. So, thanks to the cyclic prefix and guard time, the wideband multipath fading is experienced in OFDM as a set of narrowband fading subcarriers without ICI and ISI. The only remaining effect of multipath is a random phase and amplitude of each subcarrier. This effect can be minimised by adapting the subcarriers of the received signal with channel estimates per subcarrier, which can be obtained during a training phase. The only problem are the subcarriers in deep fades, in order to deal with these weak subcarriers that have a large probability to be decoded wrong, forward error correction across the subcarriers is applied.

4.3 Bandwidth Efficiency

From Figure 4-2 it becomes obvious that the spectrums of the subcarriers are not separated but overlap. Due to this overlapping, the bandwidth is much more efficiently used than in the classical multicarrier systems. Theoretically, *M*-ary digital modulation schemes using OFDM can achieve a bandwidth efficiency, defined as bit rate per unit bandwidth, of $\log_2 M$ bits/s/Hz [6]. Given the symbol rate of the serial data stream is $1/\Delta t$, the bit rate for a corresponding *M*-ary system is $R_b = \log_2 M / \Delta t$, however, in case a guard time is added, the effective bit rate is equal to:

$$R_b = \frac{T}{T_G + T} \log_2 M / \Delta t = \frac{N\Delta t}{T_G + N\Delta t} \log_2 M / \Delta t \tag{4.4}$$

The Nyquist bandwidth of a subchannel is Δf. Thus, the total bandwidth of the OFDM system is:

$$B = N\Delta f = \frac{1}{\Delta t} \tag{4.5}$$

Therefore, the bandwidth efficiency becomes:

$$\beta = \frac{R_b}{B} = \frac{T}{T_G + T} \log_2 M \tag{4.6}$$

For an optimal system without guard time (i.e., T_G = 0), the bandwidth efficiency becomes $\beta = \log_2 M$ bits/s/Hz. However, as follows from Section 4.2, the guard time is inserted in order to make the system more robust against multipath distortion, so a practical system will not achieve this optimal bandwidth efficiency. Furthermore, in reality, it is impossible to make an ideal band-pass filter, so in order to pass all subcarriers, the bandwidth *B* will be larger, which implies another decrease in bandwidth efficiency.

4.4 Main reasons for using OFDM

As mentioned before, one of the main reasons to use OFDM is its capability to deal with large delay spreads. In [53] the implementation complexity of an OFDM system is compared with a signal carrier system that is designed to deal with the same amount of delay spread. Here, a summary will be given.

The implementation complexity of a single carrier system is dominated by equalisation, which is necessary when the delay spread is larger than about 10% of the symbol duration. In an OFDM system, however, equalisation is not required, but instead, the complexity of an OFDM system is largely determined by the Fast Fourier Transform (FFT), which is used to demodulate the various subcarriers. From references [48] and [55], it can be learned that at least 8 feedforward and 8 feedback taps are required to handle a delay spread of 100 ns for a Gaussian Minimum Shift Keying (GMSK) modem at a data rate of 24 Mbps. To get the same delay spread performance as the 24 Mbps mode of the new IEEE 802.11 OFDM standard, which equals 250 ns using a 64-point FFT, the number of equaliser taps has to be increased to 8*250/100 = 20. Fortunately, for GMSK, only the real outputs of the complex multiplications are used, so each multiplier has to perform 2 real multiplications per sample (see Formula (3.1)). Hence, the number of real multiplications per second becomes $2 \cdot 20 \cdot 24 \cdot 10^6 = 960 \cdot 10^6$, where only the feedforward taps have been counted since the feedback taps have a much smaller implementation complexity. For the OFDM system, a 64-point FFT has to be processed every OFDM symbol duration which is 4 μs. With a radix-4 FFT algorithm, this requires 96 complex multiplications. One complex multiplication can be performed with four real multiplications (see Formula (3.1)) and, thus, the processing load is equal to $4 \cdot 96 / (4\ \mu s) = 96 \cdot 10^6$ real multiplications per second. So, in terms of multiplications per second, the OFDM system has a processing advantage of a factor of 10!

The other advantage of OFDM over single carrier systems with equalisers is that for the latter systems, the performance degrades abruptly if the delay spread exceeds the value for which the equalisers are designed. Because of error propagation, the raw bit error probability increases so quickly that introducing lower rate coding or a lower constellation size does not significantly improve the delay spread robustness. For OFDM, however, there are no such non-linear effects as error propagation, and coding and lower constellation sizes can be employed to provide fallback rates that are significantly more robust against delay spread. This is an important consideration, as it enhances the coverage area and avoids the situation that users in bad spots cannot get any connection at all.

4.5 OFDM Transceiver

The general block diagram of the baseband processing of an OFDM transceiver is shown in Figure 4-5 [52]. The diagram will be briefly explained following the data-stream from transmitter to receiver. In the transmitter, a convolutional encoder first encodes the binary input data. After interleaving, the binary values are mapped on Quadrature Amplitude Modulation (QAM) values. In order to correct the signal in the receiver for a possible phase drift, pilot carriers can be introduced, but in this report we assume optimal timing and, thus, no pilot carriers will be used. In the Serial to Parallel block, the serial QAM input symbol-stream is converted to a parallel stream with width equal to the number of subcarriers. These parallel symbols are modulated onto the subcarriers by applying the Inverse Fast Fourier Transform. Note that in order to get an output spectrum with a relative low out-of-band radiation, the size of the IFFT can be chosen larger than the number of subcarriers that is actually used to transmit the data. After the IFFT block, the parallel output is converted back to serial. To make the system robust to multipath propagation (see Section 4.2), a cyclic prefix is added. Further, windowing is applied to get a narrower output spectrum. After this

step, the digital output signals are converted to analog signals. These analog signals are then upconverted to the RF band, amplified and transmitted through an antenna.

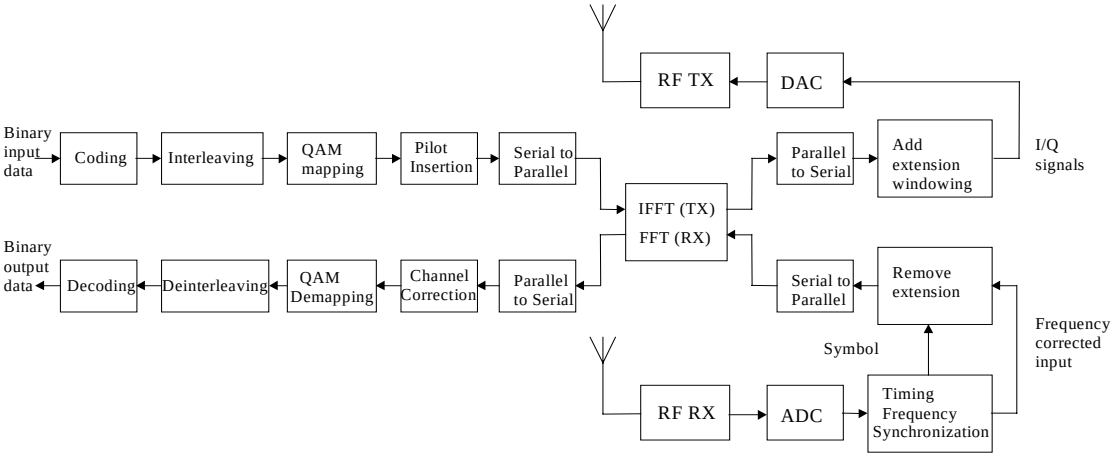

Figure 4-5: Block diagram of an OFDM transceiver.

The OFDM receiver basically performs the reverse operations of the transmitter, together with additional training tasks. First, the receiver has to estimate frequency offset and symbol timing, using special training symbols in a preamble. Then, it can do a Fast Fourier Transform for every symbol to recover the QAM values of all subcarriers. The training symbols and pilot subcarriers are used to correct for the channel response as well as the remaining phase drift. The QAM values are then demapped into binary values and de-interleaved, after which a Viterbi decoder can decode the information bits.

5 OFDM with SDM using Maximum Likelihood Decoding

In this chapter, a system will be described that combines Space Division Multiplexing (SDM) based on MLD with OFDM. First the system configuration will be proposed in such a way that the system can be used both for Space Division Multiplexing or Space Division Multple Access. After that it will be shown how the training, compared to that of the OFDM tranceiver of Section 4.5, needs to be adapted to distinguish the different transmitters at the receiver. After that the simulation parameters and the channel model will be given and before the simulation results are given different types of Maximum Likelihood Decoding algorithms will be discussed. It will be shown that the soft-input Viterbi decoder that is used to decode the convolutional code performs the best with a Maximum Likelihood Decoding algorithm that produces soft-output decisions. All these options are implemented in C++, simulations are performed for different MLD algorithms, different antenna configurations and a number of delay spreads and the results will be given and compared.

5.1 System configuration

The SDM algorithms as described in Chapter 2 are single carrier algorithms. In order to combine SDM with OFDM, a SDM algorithm has to be performed for each subcarrier. So, suppose the transmitter consists of N_t transmit antennas, then every subcarrier carries N_t datastreams. At the N_r receive antennas, the subcarrier information is separated by using FFTs. After that the N_r information symbols belonging to subcarrier i are routed to the i-th Multi-Antenna Processing Unit (MAPU) where the MLD algorithm is implemented to recover the transmitted data signals ($d_{1,i},...,d_{N_t,i}$). After that, the demapping, deinterleaving and the decoding is performed. This leads to the configuration of the receiver as shown in Figure 5-1 where N Multi-Antenna Processing Units (MAPUs) are introduced for the processing of the N subcarriers.

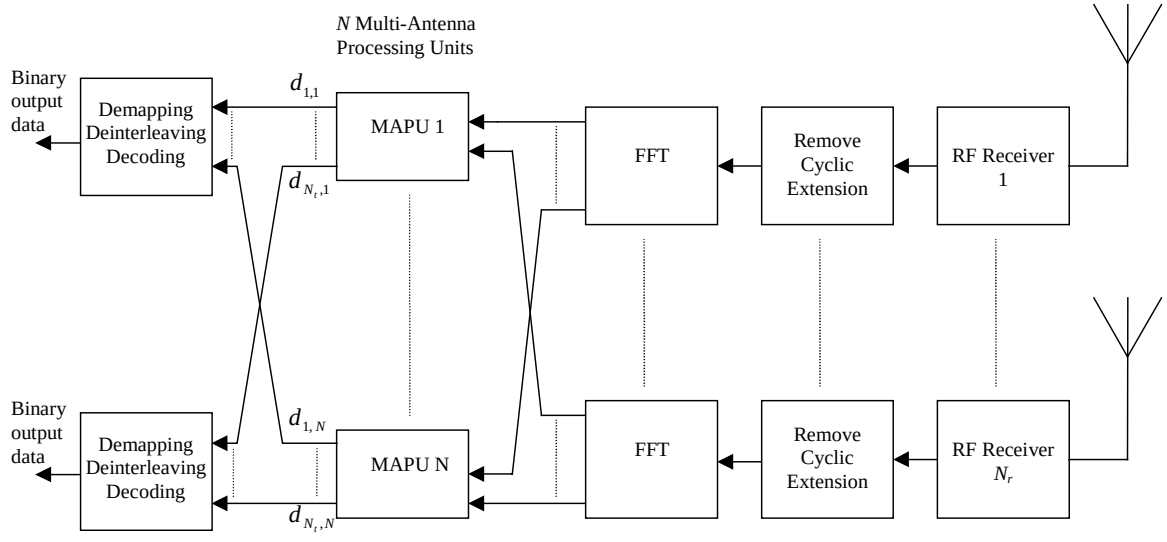

Figure 5-1: Multi-antenna receiver using Multi-Antenna Processing Unit (MAPU) with OFDM.

The transmitting part can consist of different users (in the case of Space Division Multiple Access) or of a multi-antenna transmitter (in the case of Space Division Multiplexing) or a combination of both, that is why the transmitting chains should be separable. This is represented schematically in Figure 5-2. The only disadvantage in case of SDMA is that the transmitters need to be

synchronised because only in case the receiver receives synchronised signals, the MLD algorithm can be properly executed.

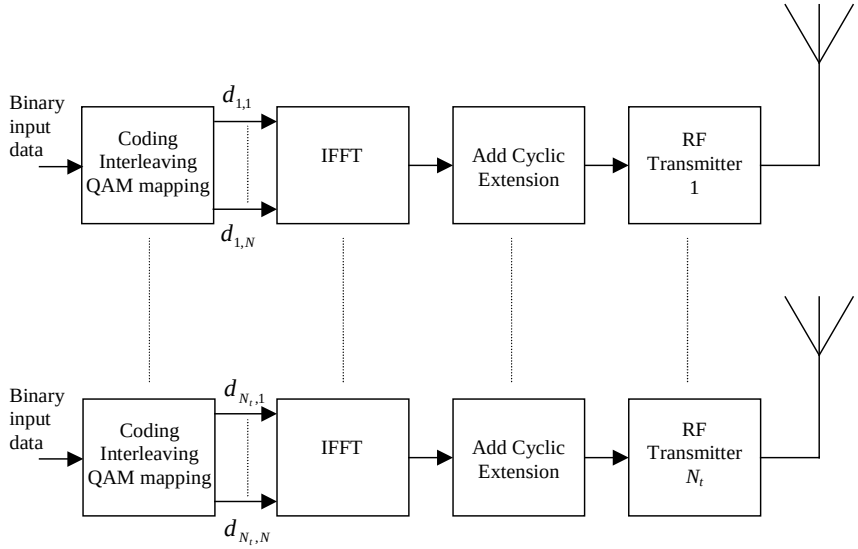

Figure 5-2: Multi-antenna transmitter(s) using OFDM.

OFDM has as advantage that, if N subcarriers are used, the symbol duration is N times longer. Thus, one MAPU is allowed to work N times slower than a MAPU for a single carrier system with comparable data rate as the OFDM system. Lower speed usually reduces complexity, thus, one MAPU in the OFDM system is not as complex as the MAPU in a single carrier system with comparable rate.

5.2 Training

In order to perform a SDM algorithm per subcarrier, the effect of the channel on each subcarrier has to be known at the receiver. Therefore, the channel effect per subcarrier has to be estimated. This can be done in the training phase since we assume that the channel is not changing during one packet. So, every OFDM packet is preceded by training symbols. The training symbols are designed in such a way that the peak-to-average power (PAP) ratio is approximately 3 dB, which is significantly lower than the PAP ratio of random OFDM data symbols. This guarantees the training degradation caused by non-linear amplifier distortion to be smaller than the distortion of the data symbols, which again guarantees that the channel is estimated more accurate than the accuracy of the general QAM signals that are received in the data phase. The result is a more accurate estimate of the transmitted signals compared to a system in which the training symbols have a larger PAP ratio. Note that one training symbol consists of N QAM values.

So, the channel is estimated per subcarrier. Actually, this is even necessary, because the bandwidth of the OFDM system is not narrow enough that we can treat the channel frequency characteristic flat across the whole OFDM band. However, we assume that the bandwidth per subchannel is narrow enough to be treated as flat. So, by estimating the channel per subcarrier the system becomes more robust against frequency selective fading, which of course is a big advantage.

If multiple transmit antennas are used, the training symbols for each transmit antenna must be orthogonal in order to distinguish the different transmitters at the receiver. For instance, if we have two transmit antennas and on the first one $\boldsymbol{S}\ \boldsymbol{S}$ and on the second one $\boldsymbol{S}\ \boldsymbol{-S}$ is transmitted (where $\boldsymbol{S}$ represents the OFDM training symbol), then every receiving antenna can distinguish both

transmit antennas since the signals they receive from both transmitters are orthogonal to each other. In this way, it is possible to estimate the channel from every transmit antenna to every receive antenna.

One possible method to make the training symbols orthogonal is using Walsh codes. We have chosen for a fixed training length of eight training symbols. An 8-bit Walsh code is used, that allows us to construct eight orthogonal training codes. With this code, a maximum of eight transmit antennas can be distinguished. If $\boldsymbol{S}$ is the training symbol of N QAM values, then the following eight orthogonal training sequences can be constructed using the 8-bit Walsh code:

$$\begin{array}{cccccccc}
\boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} \\
\boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} \\
\boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} \\
\boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} \\
\boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} \\
\boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} \\
\boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} \\
\boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S}
\end{array}$$

Note that a “power of two”-bit Walsh code can easily be obtained using the following algorithm: Start with the basic matrix:

$$\begin{array}{cc}
\boldsymbol{S} & \boldsymbol{S} \\
\boldsymbol{S} & -\boldsymbol{S}
\end{array}$$

Substitute for $\boldsymbol{S}$ the basic matrix. Now the 4-bit Walsh code is obtained:

$$\begin{array}{cccc}
\boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} & \boldsymbol{S} \\
\boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} \\
\boldsymbol{S} & \boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} \\
\boldsymbol{S} & -\boldsymbol{S} & -\boldsymbol{S} & \boldsymbol{S}
\end{array}$$

The 8-bit Walsh code follows after the next step.

The channel estimates can be obtained by multiplying the symbols received during the training period with the conjugated training sequences and dividing the result by the power of the training symbol. This can be explained by looking to only one subcarrier. Suppose, the OFDM symbol consists of only one subcarrier, then the training symbol $\boldsymbol{S}$ is one-dimensional and contains only one QAM value. Now, suppose we have a system comprising four transmit antennas, then, by choosing the first four 8-bit Walsh codes and denoting them by matrix **A**, the estimation of the channel matrix is obtained by:

$$\mathbf{H}_{\text{est}} = \frac{\mathbf{X}\mathbf{A}^*}{8\|\boldsymbol{S}\|^2} = \frac{1}{8\|\boldsymbol{S}\|^2}\left(\mathbf{H}\mathbf{A}\mathbf{A}^* + \mathbf{N}\mathbf{A}^*\right) = \mathbf{H} + \frac{1}{8\|\boldsymbol{S}\|^2}\mathbf{N}\mathbf{A}^* \tag{5.1}$$

where **X** is the received matrix of 8 time instants and **N** is the receiver noise matrix. Note that the channel estimation will be rather accurate, because after “decoding” the Walsh code, eight (almost identical) samples are averaged to get the channel estimate.

5.3 Simulation parameters

The system proposed in Figure 5-1 and Figure 5-2 has been programmed in C++, in order to do some simulations and tests. Furthermore, coding, interleaving and QAM mapping are implemented

for every transmitting chain and at the receiver, the decoding, de-interleaving and QAM demapping blocks are implemented after the MAPUs.

Table 5-1 lists the main parameters that are used for the simulation and are based on the draft OFDM standard in the IEEE 802.11 standard [52]. A key parameter, which largely determines the choice of the other parameters, is the guard interval of 800 ns. This guard interval provides robustness to root-mean-squared delay spreads up to several hundreds of nanoseconds, depending on the coding rate and modulation used. In practice, this means that the modulation is robust enough to be used in any indoor environment, including large factory buildings (see Section 5.4). It can also be used in outdoor environments, although directional antennas may be needed in this case to reduce the delay spread to an acceptable amount and to increase the range.

Table 5-1: Main Parameters based on the OFDM standard in IEEE 802.11.

Modulation	QPSK, 16-QAM, 64-QAM
Coding rate	1/2, 2/3, 3/4
Number of subcarriers	64
Number of subcarriers used	48
OFDM symbol duration	4 μs
Guard interval	800 ns
Subcarrier spacing	312.5 kHz
–3 dB Bandwidth	16.56 MHz
Channel spacing	20 MHz

In order to limit the relative amount of power and time spent on the guard time to 1 dB, the symbol duration was chosen to be 4 μs. This also determined the subcarrier spacing to be 312.5 kHz, which is the inverse of the symbol duration minus the guard time. By using 48 data subcarriers, uncoded data rates of 12 to 72 Mbps can be achieved by using variable modulation types from BPSK to 64-QAM.

In order to correct for subcarriers in deep fades, forward error correction across the subcarriers is used with variable coding rates, giving coded data rates from 6 up to 54 Mbps. Convolutional coding is used with the industry standard rate 1/2, constraint length 7 code with generator polynomials (133,171). Higher coding rates of 2/3 and 3/4 are obtained by puncturing the rate 1/2 code.

5.4 Indoor channel model

In order to obtain a reliable indoor channel model, it is useful to examine the types of multipath impairments that the receiver must combat. The multipath is the dominant impairment for the indoor wireless signal. The other impairment is thermal noise, which is established predominately by the Low Noise Amplifier (LNA) in the receiver front end.

From the references [19] and [52], it can be deduced that the multipath for indoor WLANs tends to fall off in time in an exponential way and it is often characterised by an exponentially-decayed Rayleigh fading [36]. The physical mechanism driving this indoor characteristic is mainly based on:

- the large number of reflectors within a typical building;
- the propagation loss characteristic, where longer propagation paths arriving linearly-later in time have logarithmically-weaker energy.

The Power Delay Profile (PDF) for the exponentially-decayed Rayleigh fading channel is shown in Figure 5-3. The power delay profile (PDF) measures the mean signal power relative to its dispersion across time. The mean power level establishes the variance of the corresponding Rayleigh components. The chief feature one should note about exponentially-decayed multipath is that on-average the strongest paths arrive earliest in time since propagation delay is proportional to distance travelled as shown in Figure 5-3. Radiated power drops off logarithmically with respect to distance. It could appear that weaker paths do precede the Channel Impulse Response (CIR) peak on a particular stochastic realisation, but on average there number is small, compared to the number of components which follow the CIR peak.

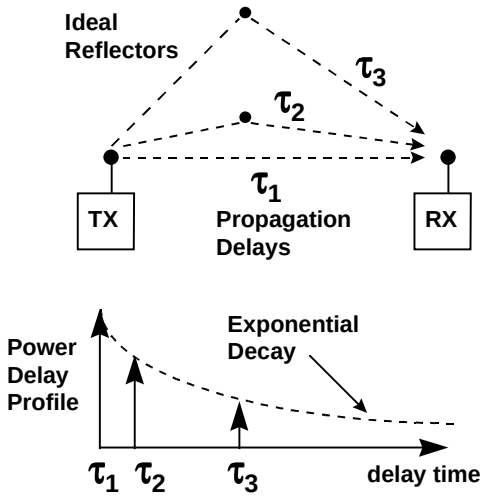

Figure 5-3: Exponentially-decayed Power Delay Profile (PDF).

Table 5-2: Measured delay spreads in frequency range of 1.8 to 2.4 GHz.

Remarks	Maximum Delay Spread [ns]	References
Large building (New York stock exchange)	120	[9]
Office building	95	[10]
Office building	150	[11]
Office building	56	[5]

Table 5-3: Measured delay spreads in frequency range of 4 to 6 GHz.

Remarks	Maximum Delay Spread [ns]	References
Large building (New York stock exchange)	120	[9]
Office building Meeting room (5m×5m) with metal walls Single room with stone walls	60 55 35	[16]
Office building	130	[11]
Indoor sports arena Factory Office building	120 125 65	[20]

The Root Mean Square (RMS) delay spread of the multipath can range from 20-50 ns for small-office and home-office environments, 50-100 nsec for commercial environments and 100-200 nsec for factory environments. These figures are obtained from [50] and the main results are given in Table 5-2, Table 5-3 and Table 5-4. Note that only the maximum delay spread is given, because many references only give the maximum delay spread. This value is only a few percent larger than the delay spread for which it can be guaranteed that the link works for at least 90% or 99% of all channels, which is of interest for designing wireless links.

Table 5-4: Measured delay spreads in frequency range of 800 MHz to 1.5 GHz.

Remarks	Maximum Delay Spread [ns]	References
Office building	50	[40]
Office building	56	[5]
Office building	43	[32]
Office building	58	[8]
Warehouse Factory	129 300	[35]

For exponentially-faded channels, the exponential decay constant is equal to the RMS delay spread. So, for simulations, it is convenient to use exponentially-decayed Rayleigh fading, because its characteristics are fully defined by one single parameter, namely, the RMS delay spread.

5.5 Coding

Special attention should be paid to coding. As said before, forward error correction across the subcarriers is used, in order to correct for subcarriers that are in deep fades. As forward error correction coding, convolutional coding with the industry standard rate 1/2, constraint length 7 code with generator polynomials (133,171) is chosen and implemented in the C++ simulation code. Higher coding rates of 2/3 and 3/4 are obtained by puncturing the rate 1/2 code. To decode this convolutional code, a Viterbi decoder is used. From [17] it is known that a Viterbi decoder performs better when soft-decisions are used as input. However, the MLD algorithm, as described in Section 2.6, returns hard-decision outputs. So, in order to improve performance, we need to adapt the MLD method in such a way that it will produce soft-decision outputs. Until now, we have found two ways, a simple and a complex one. These two options will be described theoretically in this section and the results of the hard- and soft-decision output MLD techniques will be compared in the next sections.

5.5.1 Simple MLD algorithm with soft-decision outputs

The MLD technique is modified to deliver not only the most likely transmitted symbol, but also reliability values, which are known as soft-decision outputs. The simple MLD algorithm with Soft-decision outputs is based on the intuition that the data transmitted over antennas that are in a deep fade (for a specific subcarrier) cannot be reliably received. In other words, if the column of **H** corresponding to a certain transmit antenna has a weak power then this transmit antenna is said to be in a deep fade. The soft-output decision of that particular element of **s** should be low. Thus, if s_i is the *i*-th element of **s**, where **s** is the output vector of the hard-decision MLD algorithm, and **H** is the estimated channel matrix for a given subcarrier, then soft-decision outputs can be produced according to:

$$L_i = c_i s_i \tag{5.2}$$

where,

$$c_i = \begin{cases} \dfrac{1}{N_r}\sqrt{\sum_{j=1}^{N_r}\left|h_{ji}\right|^2} & \text{if } \dfrac{1}{N_r}\sqrt{\sum_{j=1}^{N_r}\left|h_{ji}\right|^2} < 1 \\ 1 & \text{if } \dfrac{1}{N_r}\sqrt{\sum_{j=1}^{N_r}\left|h_{ji}\right|^2} \geq 1 \end{cases} \tag{5.3}$$

Note that this method produces soft-decision outputs per QAM symbol, i.e., per element of **s**. However, soft-decision outputs per bit most likely will give better performance.

5.5.2 Complex MLD algorithm with soft-decision outputs

Another way to find soft-decision outputs is by using the same approach as [17]. There, the log likelihood ratio is used as an indication for the reliability of a bit. If **x** denotes the received vector, b_l is the l-th bit to estimate, **H** is the estimated channel matrix for a given subcarrier and $\mathbf{s}_j$ is one of the possible transmitted vectors (with $1 \leq j \leq K$, where $K = M^{N_t}$ and M represents the number of constellation points), then the L-value of the estimated bit is:

$$L(b_l) = \ln\frac{P(b_l = +1|\mathbf{x})}{P(b_l = -1|\mathbf{x})} = \ln\frac{\sum_{\mathbf{s}_j|b_l=+1} P(\mathbf{s}_j|\mathbf{x})}{\sum_{\mathbf{s}_j|b_l=-1} P(\mathbf{s}_j|\mathbf{x})} = \ln\frac{\sum_{\mathbf{s}_j|b_l=+1} p(\mathbf{x}|\mathbf{s}_j)P(\mathbf{s}_j)}{\sum_{\mathbf{s}_j|b_l=+1} p(\mathbf{x}|\mathbf{s}_j)P(\mathbf{s}_j)} \tag{5.4}$$

Because the vectors $\mathbf{s}_j$ are equally likely to be transmitted, $P(\mathbf{s}_j)$ is equal for all vectors $\mathbf{s}_j$. Using Formula (2.66), we find the soft-output decisions:

$$\begin{aligned} L(b_l) &= \ln\frac{\sum_{\mathbf{s}_j|b_l=+1}\exp\left(-\dfrac{\left\|\mathbf{x}-\mathbf{H}\mathbf{s}_j\right\|^2}{\sigma_v^2}\right)}{\sum_{\mathbf{s}_j|b_l=-1}\exp\left(-\dfrac{\left\|\mathbf{x}-\mathbf{H}\mathbf{s}_j\right\|^2}{\sigma_v^2}\right)} \approx \ln\frac{\exp\left(\max_{\mathbf{s}_j|b_l=+1}-\dfrac{\left\|\mathbf{x}-\mathbf{H}\mathbf{s}_j\right\|^2}{\sigma_v^2}\right)}{\exp\left(\max_{\mathbf{s}_j|b_l=-1}-\dfrac{\left\|\mathbf{x}-\mathbf{H}\mathbf{s}_j\right\|^2}{\sigma_v^2}\right)} \\ &= \frac{1}{\sigma_v^2}\left(\min_{\mathbf{s}_j|b_l=-1}\left\|\mathbf{x}-\mathbf{H}\mathbf{s}_j\right\|^2 - \min_{\mathbf{s}_j|b_l=+1}\left\|\mathbf{x}-\mathbf{H}\mathbf{s}_j\right\|^2\right) \end{aligned} \tag{5.5}$$

Note that this type of decoding would involve an exhaustive search over all possible vectors $\mathbf{s}_j$ *per bit* to decode, making the decoding method even more complex than hard-decision output MLD which requires an exhaustive search per transmitted vector **s**. Furthermore, it might be useful in the simulations to determine the effect of the approximation in Formula (5.5).

If we look more closely to Formula (5.5), we see that, for (very) high SNR, the L-values could become (very) large. For a practical implementation, this is not very useful. We have chosen for an 8-bit quantiser to quantise the L-values in integer numbers from –128 to 127. If this quantiser is

used to quantise Formula (5.5), for high SNR, the output would mainly consist of –128 for $b_l = -1$ and 127 for $b_l = 1$, resulting in hard-decision outputs. Thus, for high SNR values, this implementation would achieve the same performance as MLD with hard-decision outputs. In order to avoid this problem, the L-values must be kept in the range of –128 to 127. This can be done by using a modified version of Formula (5.5). This modified version is found on a trail and error base and is given by:

$$L'(b_l) = \frac{25}{N_r}\left(\min_{\mathbf{s}_j | b_l = -1} \left\| \mathbf{x} - \mathbf{H}\mathbf{s}_j \right\|^2 - \min_{\mathbf{s}_j | b_l = +1} \left\| \mathbf{x} - \mathbf{H}\mathbf{s}_j \right\|^2 \right) \tag{5.6}$$

Furthermore, note that the Viterbi decoder sees a zero as a bit erasure, thus, if L' is quantised (by flooring it to an integer number) it could appear that, if $0 < L' < 1$, the quantised value of L' is equal to zero. To avoid this, the following adaptation is introduced:

$$\text{if } \left(0 < L'(b_l) < 1\right)\ L'(b_l) = 1; \tag{5.7}$$

After this, the quantised values can be obtained by flooring L'.

5.6 Simulation and results

As said before, OFDM is combined with SDM. The system, as described in the former sections, is simulated in C++ and in this section several simulation results will be given. The main goal is to compare the performance of systems with different antenna configurations. Therefore, the simulations are performed multiple times only with different number of transmitting and/or receiving antennas. All simulations are performed with a QPSK modulation scheme. The training length is chosen fixed at 8 OFDM symbols (see Section 5.2). We have implemented the SDM technique based on MLD because it has the best BER performance as can be concluded from Chapter 2 (we have not taken complexity into account). Furthermore, we will compare the performance results of the following MLD decoding algorithms (see Section 5.5): hard-decision output MLD (Subsection 5.6.1), simple MLD algorithm with soft-decision outputs (Subsection 5.6.2) and complex MLD algorithm with soft-decision outputs (Subsection 5.6.3).

5.6.1 Hard-decision output MLD

Figure 5-4 shows simulated Bit Error Ratios versus mean E_b/N_o for an Additive White Gaussian Noise (AWGN) channel and an antenna configuration (N_t,N_r) of (1,1) with hard-decision output MLD (see Section 2.6). For the curves a and b, no coding is used, thus, these results will also hold for the MLD algorithms with soft-decision outputs, because the Viterbi decoder is not activated. For curve d, a 1/2 rate convolutional code is simulated. Furthermore, a QPSK modulation scheme is used. The curves a and d are the results of simulations in which it is assumed that the channel is perfectly known at the receiver and in which no guard time is considered, i.e. $T_G = 0$ (see Section 4.3). Curve b is the result of the simulation where the training (consisting of 8 OFDM symbols) is used to estimate the channel and also a guard time of 800 ns is simulated. The dotted line represents the theoretical value of the BER in AWGN for QPSK and is given by [36]:

$$P_b = Q\left(\sqrt{\frac{2E_b}{N_0}} \right) \tag{5.8}$$

It can be concluded that, in case the channel is perfectly known at the receiver and no guard time is simulated, the simulation results perfectly match the theoretical curve. Furthermore, it can be noticed that in case channel estimation and guard time are simulated, the performance is approximately 1.5 dB worse (i.e. the difference between curve a and curve b). Finally, it can be concluded that when a rate 1/2 convolutional code is used, the BER performance increases (a lot) (compare curve b and d). This can be explained by the error correcting ability of the Viterbi decoder.

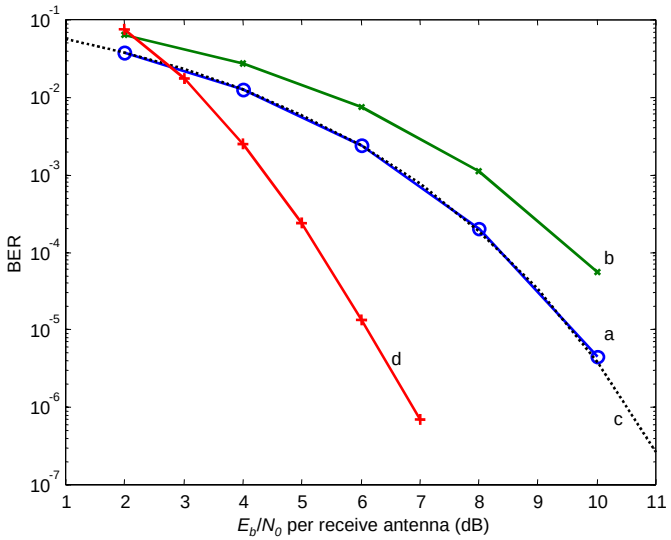

Figure 5-4: Bit error ratios versus E_b/N_0 per receive channel in AWGN for QPSK, 64 byte packets, no coding (except for curve d), antenna configuration $(N_t,N_r) = (1,1)$ and a) perfect knowledge of the channel and no guard time, b) channel estimation and a guard time of 800 ns, c) theoretical value: $P_b = Q(\sqrt{(2E_b/N_0)})$ and d) rate 1/2 convolutional code, perfect knowledge of the channel and without guard time.

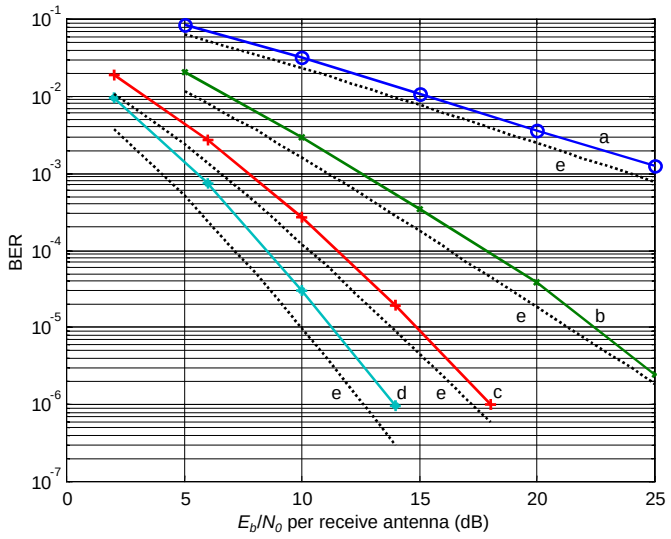

Figure 5-5: Bit Error Rate versus mean E_b/N_0 per receive antenna in a flat fading channel for QPSK, 64 byte packets, no coding, but with channel estimation and a guard time of 800 ns, with different antenna configurations (N_t,N_r): a) (1,1), b) (1,2), c) (1,3), d) (1,4) and e) theoretical curves (based on perfect knowledge of the channel and no guard time).

Figure 5-5 shows the Bit Error Ratios for a different number of receiving antennas and one transmit antenna versus the mean E_b/N_0 per receive antenna in a flat Rayleigh fading channel. The flat fading channel is a special case of the exponentially-decayed Rayleigh fading channel described in Section 5.4: only the first pulse of the Channel Impulse Response is chosen unequal to zero and it is assumed that this pulse does not introduce any delay (so $\tau_1 = 0$ ns in Figure 5-3). Furthermore, no coding is used and it is assumed that the receiver estimates the channel using training. Again, the training sequence has a length of 8 OFDM symbols. The guard time is chosen to be 800 ns. Curves a), b), c) and d), respectively, give the simulation results for the antenna configurations (1,1), (1,2), (1,3) and (1,4), where the first number represents the number of transmitting antennas (N_t) and the second number represents the number of receiving antennas (N_r). If curves a), b), c) and d) are considered, clearly the diversity (see Section 2.7) increases when the number of receiving antennas is increased.

The theoretical curves e) are based on perfect knowledge of the channel at the receiver and no guard time. These curves can be obtained by using Formula (2.77) with the diversity order L equal to the number of receiving antennas (see the diversity analysis of Section 2.6). In order to illustrate the correctness of using this formula, we need to show that MRC is a special case of MLD, i.e., MLD with antenna configuration $(1,N_r)$. MRC is a technique in which the received signals (from the different receiving antennas) are added, each weighted by the complex conjugate of the corresponding channel element and this sum is divided by the power of the channel vector. In vector notation this sum equals $\mathbf{h}^*\mathbf{x}/\|\mathbf{h}\|^2$. After that, the most likely transmitted signal is sought. Mathematically, this can be represented by:

$$s_{\text{MRC}} = \min_{s_j \in \{s_1,\ldots,s_M\}} \left\| \frac{1}{\|\mathbf{h}\|^2} \mathbf{h}^* \mathbf{x} - s_j \right\|^2 \tag{5.9}$$

where M corresponds to the number of constellation points, $\mathbf{x}$ is the received vector and $\mathbf{h}$ is the channel vector. Rewriting this formula leads to:

$$\begin{aligned} s_{\text{MRC}} &= \arg \min_{s_j \in \{s_1,\ldots,s_M\}} \left\| \frac{1}{\|\mathbf{h}\|^2} \mathbf{h}^* \mathbf{x} - s_j \right\|^2 \\ &= \arg \min_{s_j \in \{s_1,\ldots,s_M\}} \left(\frac{1}{\|\mathbf{h}\|^4} \left(\mathbf{h}^* \mathbf{x}\right)^* \mathbf{h}^* \mathbf{x} + \left|s_j\right|^2 - 2\,\text{Re}\left(\frac{1}{\|\mathbf{h}\|^2} \left(\mathbf{h}^* \mathbf{x}\right)^* s_j \right) \right) \\ &= \arg \min_{s_j \in \{s_1,\ldots,s_M\}} \left(\frac{1}{\|\mathbf{h}\|^2} \|\mathbf{x}\|^2 + \left|s_j\right|^2 - 2\,\text{Re}\left(\frac{1}{\|\mathbf{h}\|^2} \mathbf{x}^* \mathbf{h} \cdot s_j \right) \right) \end{aligned} \tag{5.10}$$

On the other hand, MLD with an antenna configuration of $(1,N_r)$ can be rewritten as follows:

$$
\begin{aligned}
s_{\text{ml}} &= \arg \min_{s_j \in \{s_1,\ldots,s_M\}} \left\| \mathbf{x} - \mathbf{h}s_j \right\|^2 \\
&= \arg \min_{s_j \in \{s_1,\ldots,s_M\}} \left(\|\mathbf{x}\|^2 + \left(\mathbf{h}s_j{}^*\right)\left(\mathbf{h}s_j\right) - 2\,\mathrm{Re}\left(\mathbf{x}^*\mathbf{h} \cdot s_j\right)\right) \\
&= \arg \min_{s_j \in \{s_1,\ldots,s_M\}} \left(\|\mathbf{x}\|^2 + \|\mathbf{h}\|^2 |s_j|^2 - 2\,\mathrm{Re}\left(\mathbf{x}^*\mathbf{h} \cdot s_j\right)\right) \\
&= \arg \min_{s_j \in \{s_1,\ldots,s_M\}} \left(\frac{1}{\|\mathbf{h}\|^2}\|\mathbf{x}\|^2 + |s_j|^2 - 2\,\mathrm{Re}\left(\frac{1}{\|\mathbf{h}\|^2}\mathbf{x}^*\mathbf{h} \cdot s_j \right)\right)
\end{aligned}
\tag{5.11}
$$

where $\|\mathbf{h}\|^2$ is real, positive and constant for a specific **h**, which justifies the last equality. Comparing both results shows the fact that MRC and MLD with antenna configuration $(1,N_r)$ indeed give the same result. When we compare the theoretical curves of Figure 5-5 with the simulation results, again, it can be noticed that the BER performance deteriorates approximately 1.5 dB, when the channel is estimated (i.e., not perfectly known) and a guard time is simulated.

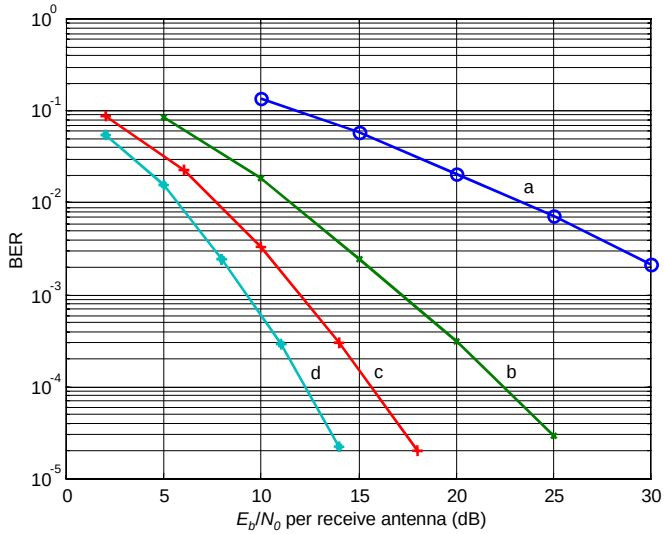

Figure 5-6: Bit Error Rate versus mean E_b/N_0 per receive antenna in a Rayleigh fading channel for QPSK, 64 byte packets, no coding, but with channel estimation and a guard time of 800 ns, with different antenna configurations (N_t,N_r): a) (2,1), b) (2,2), c) (2,3) and d) (2,4).

The results that are depicted in Figure 5-6, are the results of almost the same simulation as the previous one. The only difference is that in this case two transmit antennas are used. When comparing Figure 5-5 and Figure 5-6, it can be concluded that introducing an extra transmitting antenna, does not change the diversity order in case of the SDM technique MLD. However, in the case of 2, 3 and 4 receiving antennas, approximately a 4 dB increase in E_b/N_0 is needed to achieve the same BER performance if N_t is upgraded from 1 to 2. So this is the cost for increasing the capacity. If we change the antenna configuration from $(N_t,N_r)=(1,1)$ to $(N_t,N_r)=(2,1)$ even a 7.5 dB increase in E_b/N_0 is needed. This can be explained with Figure 5-7. Suppose that the antenna configuration is $(N_t,N_r)=(2,1)$ and that the components of the channel matrix (in this case: channel vector) are pointing in (almost) the same direction (see Figure 5-7b). Then it is difficult for the decoder to successfully decode the following two transmitted signals (in case of BPSK): (1,–1) and (–1,1) (See Figure 5-7c), which influences the BER performance in a negative way.

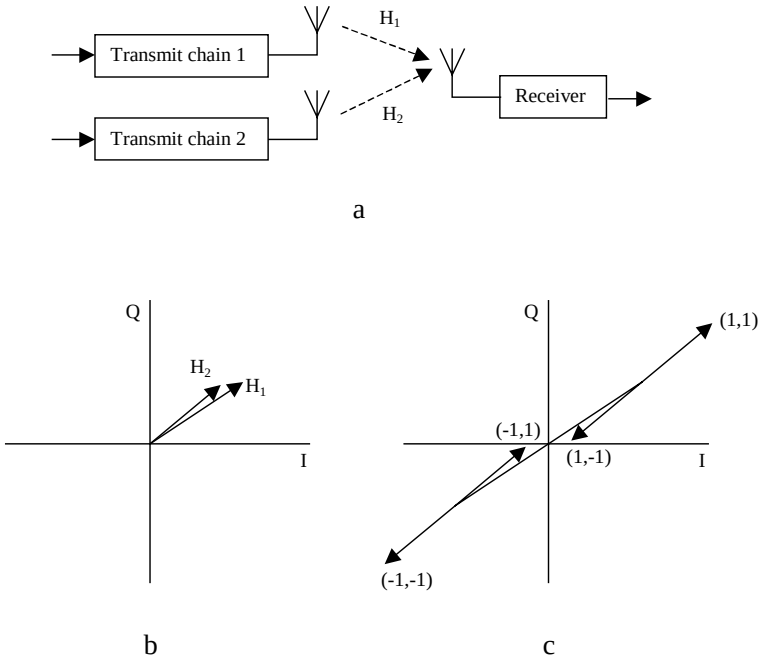

Figure 5-7: Example of system with (N_t,N_r) = (2,1) in which decoding could go wrong, especially for transmitted signals (–1,1) and (1,–1).

The most interesting conclusion is that if the antenna configuration is changed from (1,1) (see Figure 5-5) to (2,2) (see Figure 5-6), the BER performance becomes better *and* the capacity increases (in case of MLD)! So, by introducing an extra transmit *and* an extra receive antenna, the BER performance *and* the capacity increase!

For the next simulations, the indoor channel model with exponential decaying power profile, described in Section 5.4, is used. With this model, it is possible to look at the performance of the system under different delay spreads and with different antenna configurations. Figure 5-8 shows the Packet Error Rate versus the mean E_b/N_0 per receive antenna with the indoor channel model for different delay spreads. The simulation is performed with an antenna configuration equal to (N_t,N_r) = (1,1) and the data is coded with a rate 1/2 convolutional code. Note that in this case we always use a guard time, because otherwise the system is not able to deal with delay spread.

The delay spread is normalised, because the performance of the system under delay spread depends on the length of the guard time interval. If the guard time is chosen larger, the system can handle linearly higher delay spreads. So, in order to make the data of the simulations with delay spread independent of the guard time, the rms value of the delay spread is divided by the guard time length. From Figure 5-8 it can be concluded that at first the PER performance increases when the delay spread increases, but if the normalised delay spread exceeds 1/3, the performance starts to deteriorate again. This can be explained as follows. A small normalised delay spread is equivalent to a small ratio of OFDM signal bandwidth and coherence bandwidth. In such a situation, the channel frequency response is relatively flat within the OFDM signal bandwidth, so if there is a deep fade, all of the subcarriers are significantly attenuated. In the case of a large normalised delay spread, a fade only affects a few adjacent subcarriers. There can be several fades within the OFDM signal bandwidth, with relatively strong subcarriers in between the fades. As a result, the average signal power is much more constant over several channels than in the case of small delay spreads. The coding benefits from this by using the stronger subcarriers to compensate for the attenuated subcarriers. Finally, if the normalised delay spread becomes larger than 1/3, the performance goes down, because in this case the paths with a large delay cannot be resolved with the guard time and will appear as Inter-Symbol Interference (ISI).

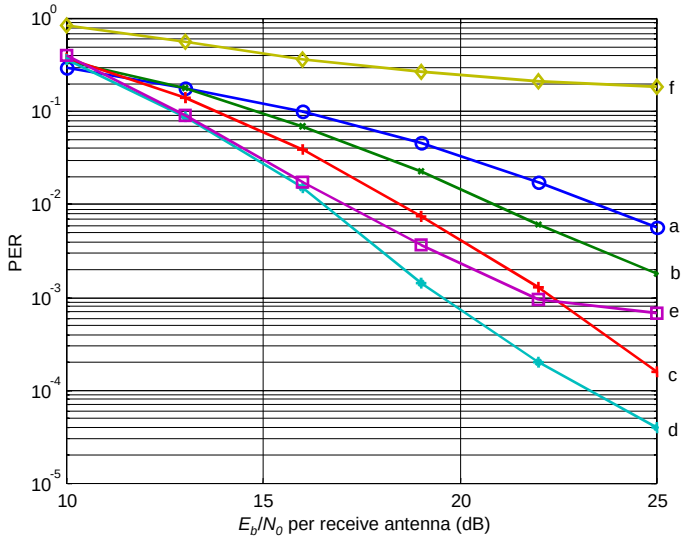

Figure 5-8: Packet Error Ratio versus mean E_b/N_0 per receive antenna in a channel with an exponential decaying power profile for QPSK, 64 byte packets, hard-decision MLD, rate 1/2 convolutional coding, antenna configuration (N_t,N_r) = (1,1), normalised delay spread τ_{rms}/T_G = a) 1/48, b) 1/24, c) 1/12, d) 1/6, e) 1/3, f) 2/3.

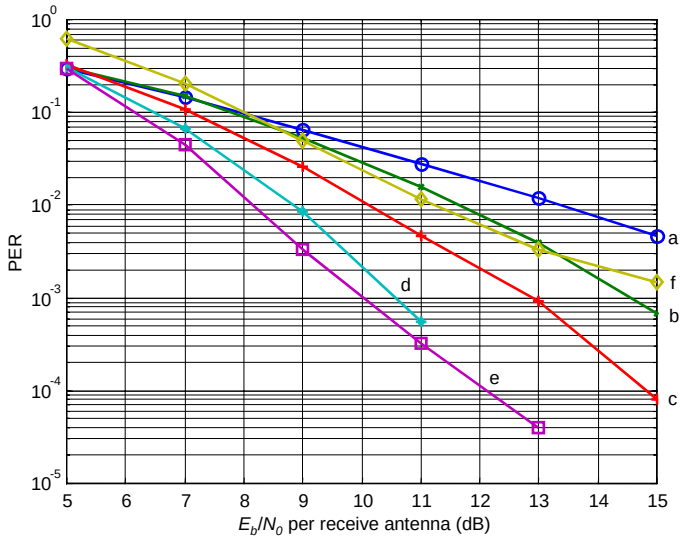

Figure 5-9: Packet Error Ratio versus mean E_b/N_0 per receive antenna in a channel with an exponential decaying power profile for QPSK, 64 byte packets, hard-decision MLD, rate 1/2 convolutional coding, antenna configuration (N_t,N_r) = (1,2), normalised delayspread τ_{rms}/T_G = a) 1/48, b) 1/24, c) 1/12, d) 1/6, e) 1/3, f) 2/3.

Figure 5-9 displays the results of a simulation that is almost the same as the previous one, only the antenna configuration is changed to (N_t,N_r) = (1,2). In Section 2.7 it is found that the diversity increases by one if an extra receiving antenna is introduced. When Figure 5-8 and Figure 5-9 are compared, it becomes clear that introducing an extra receive antenna makes the system also more robust against delay spread. This can be explained by the fact that a system with SDM becomes more robust against Gaussian noise in case of introducing an extra receive antenna (the order of diversity increases). Apparently, the delayed OFDM symbols have more or less that same

characteristics as Gaussian noise and, thus, a system with an extra receiving antenna is more robust against delay spread.

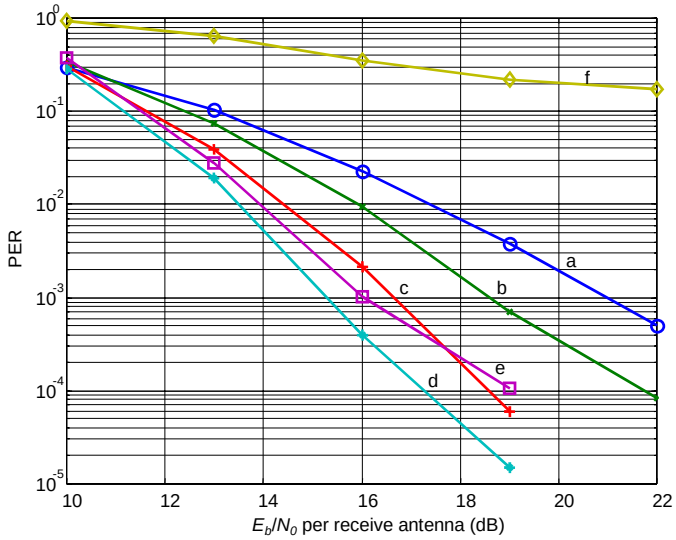

Figure 5-10: Packet Error Ratio versus mean E_b/N_0 per receive antenna in a channel with an exponential decaying power profile for QPSK, 64 byte packets, hard-decision MLD, rate 1/2 convolutional coding, antenna configuration (N_t,N_r) = (2,2), normalised delayspread τ_{rms}/T_G = a) 1/48, b) 1/24, c) 1/12, d) 1/6, e) 1/3, f) 2/3.

Figure 5-10 shows the results of almost the same simulation as the previous two, only in this case the antenna configuration is (N_t,N_r) = (2,2). Again, it can be seen that the diversity order is increased in comparison with Figure 5-8. However, when Figure 5-9 and Figure 5-10 are compared, it can be noticed that, when an extra transmit antenna is introduced, more SNR per receive antenna is needed to achieve the same PER. This is easily explained, namely, now two parallel streams of data are received, both having half of the power of the single data stream of the previous case. Thus, in order to separate these two streams and in order to get the same PER, a higher SNR per receive antenna is needed. Furthermore, comparing Figure 5-9 and Figure 5-10 leads to the conclusion that introducing an extra transmit antenna makes the system less robust against delay spread. This can be explained by the fact that an extra transmitter is seen as an extra "interferer" while decoding one of the two streams of data. Clearly, a system is less robust against delay spread if more interferers are present.

From Figure 5-8, Figure 5-9 and Figure 5-10 it can be noticed that, in case of a large delay spread, the PER appears to have a floor, even for a high SNR value. This PER floor is the result of the increasing influence of ISI in case the delay spread goes up. Furthermore, it seems that this PER floor is dependent of the antenna configuration.

Figure 5-11 shows the PER floor versus the normalised delay spread for different antenna configurations. The simulations are performed with a high SNR value of 80 dB. Curve a gives the PER floor in case of antenna configuration (1,1) and no coding. The other curves are the result of simulations in which the data is coded with a rate 1/2 convolutional code. If curve a and curve b are compared, the effect of coding becomes clear. Furthermore, when we look at curves b, c and d, (again) the conclusion can be drawn that an OFDM system becomes more robust against delay spread when extra receiving antennas are introduced. However, if an extra transmit antenna is introduced, the system becomes less robust against delay spread which can be explained by the fact that, due to this extra transmitting antenna, an extra "interferer" is introduced and, thus, the performance goes down. But, a system with a (2,2) antenna configuration (curve e) still performs

better for low delay spreads than a system with antenna configuration (1,1) (curve b). So, increasing the capacity, by going from an antenna configuration (1,1) to (2,2), does not have a negative influence on the delay performance! Although this looks promising for designing a system that is robust enough for a maximum delay spread of 300 ns (see Section 5.4), we should note that the delay spread performance deteriorates when more complex constellation schemes like, 16-QAM and 64-QAM, are used [53].

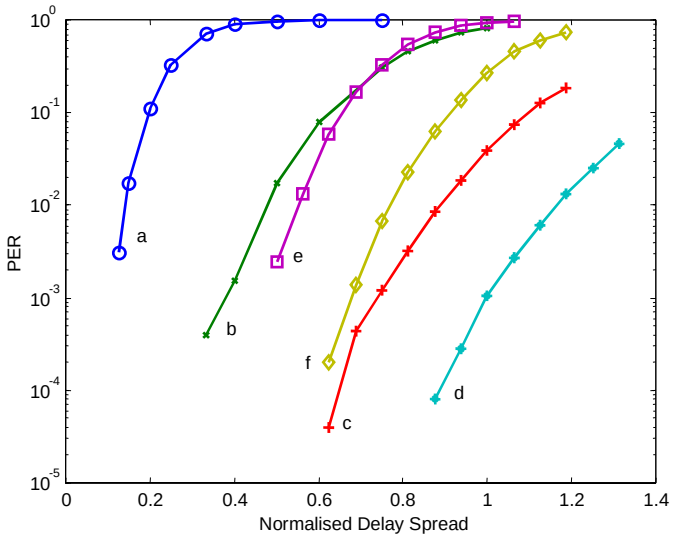

Figure 5-11: Irreducible packet error ratios versus normalised delay spread τ_{rms}/T_G for 64 byte packets, QPSK, hard-decision MLD, rate 0.5 code (except for curve a) and antenna configuration (N_t,N_r): a) (1,1), no coding, b) (1,1), c) (1,2), d) (1,3), e) (2,2), f) (2,3).

5.6.2 Simple MLD algorithm with soft-decision outputs

In this subsection the simulation results are given of the MLD algorithm which produces soft-decision outputs according to the power of the columns of **H** as described in Subsection 5.5.1. For the performance results under the condition of Additive White Gaussian Noise (AWGN), we can simply refer to the results in AWGN of hard-decision output MLD given in Subsection 5.6.1 because for AWGN, the channel is equal to **H** = 1. So, the power of the channel is equal to 1, which leads to equal outputs as MLD with hard-decision outputs. However, when the indoor channel model with exponential decaying power profile, described in Section 5.4, is used, the power of the channel columns is variable, leading to soft-decision outputs with the method of Subsection 5.5.1. So, in this case we may expect a performance improvement.

Figure 5-12 shows the Packet Error Rate versus the mean E_b/N_0 per receive antenna for different delay spreads when the indoor channel model is used. The antenna configuration equals (N_t,N_r) = (1,1) and a rate 1/2 convolutional code is used for coding the data. Again, the delay spread is normalised, because the performance of the system under delay spread depends on the length of the guard time. That is, the system can handle linearly higher delay spreads when the guard time is increased. Therefore, the rms value of the delay spread (to normalise it) is divided by the guard time length.

It is interesting to compare these results with the performance of the hard-decision output MLD algorithm with the same configuration (see Figure 5-8). Clearly, the expected performance improvement is true. What is also interesting is the increasing improvement, when the delay spread

increases. Comparing the curves a of Figure 5-8 and Figure 5-12, we see an improvement of the SNR of approximately 2 dB in the latter case. But when we compare the curves d, this improvement increases to about 4.5 dB. So, it can be concluded that the simple MLD algorithm with soft-decision outputs is more robust against delay spread than hard-decision output MLD, which also can be seen from the lower PER floor of curve f of Figure 5-12 compared to that of Figure 5-8. This can be explained by the fact that the Viterbi decoder performs better when receiving soft-decision values as input, or in other words, the coding benefits even better from the stronger subcarriers in between the fades (due to selective fading, see Subsection 5.6.1). Thus, it can handle higher delay spreads, while achieving the same PER performance.

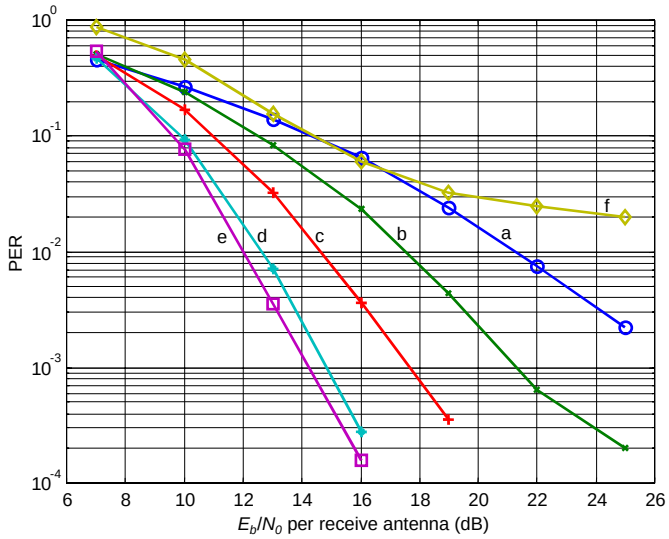

Figure 5-12: Packet Error Ratio versus mean E_b/N_0 per receive antenna in a channel with an exponential decaying power profile for QPSK, 64 byte packets, simple MLD algorithm with soft-decision outputs, rate 1/2 convolutional coding, antenna configuration $(N_t,N_r) = (1,1)$, normalised delayspread τ_{rms}/T_G = a) 1/48, b) 1/24, c) 1/12, d) 1/6, e) 1/3, f) 2/3.

Figure 5-13 and Figure 5-14 show the results of the same simulation as that of Figure 5-12, only now with antenna configuration $(N_t,N_r) = (1,2)$, respectively, $(N_t,N_r) = (2,2)$. Comparing these results with the results of the simulations with hard-decision output MLD (Figure 5-9 and Figure 5-10, respectively) confirms the earlier drawn conclusions. Namely that the simple MLD algorithm with soft-decision outputs performs better in combination with soft-decision input Viterbi decoding than hard-decision output MLD and also that the former one is more robust against delay spread.

However, when we take a closer look to for instance Figure 5-10 and Figure 5-14, we can see that the performance improvement deteriorates in case of larger antenna configurations. E.g., the curves a of Figure 5-10 and Figure 5-14 show an improvement of only 0.75 dB, whereas the improvement in the $(N_t,N_r) = (1,1)$ case is shown to be 2 dB. Also the PER floor is not increasing that much anymore for larger antenna configurations. An explanation for this is the fact that the chance that a transmit antenna is in a fade for all receive antennas deteriorates when larger antenna configurations are used, i.e., the columns of **H** have a less variable power for larger matrices of **H**. Thus, for (very) large antenna configurations the performance of the simple MLD algorithm with soft-decision outputs tends towards the performance of hard-decision output MLD.

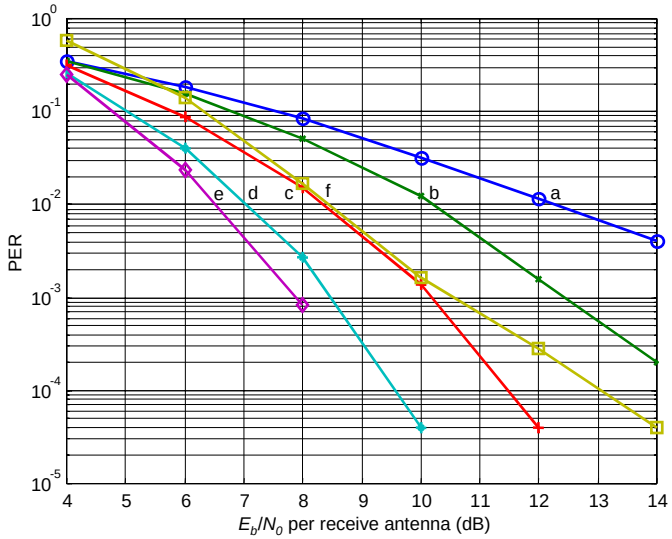

Figure 5-13: Packet Error Ratio versus mean E_b/N_0 per receive antenna in a channel with an exponential decaying power profile for QPSK, 64 byte packets, simple MLD algorithm with soft-decision outputs, rate 1/2 convolutional coding, antenna configuration (N_t,N_r) = (1,2), normalised delayspread τ_{rms}/T_G = a) 1/48, b) 1/24, c) 1/12, d) 1/6, e) 1/3, f) 2/3.

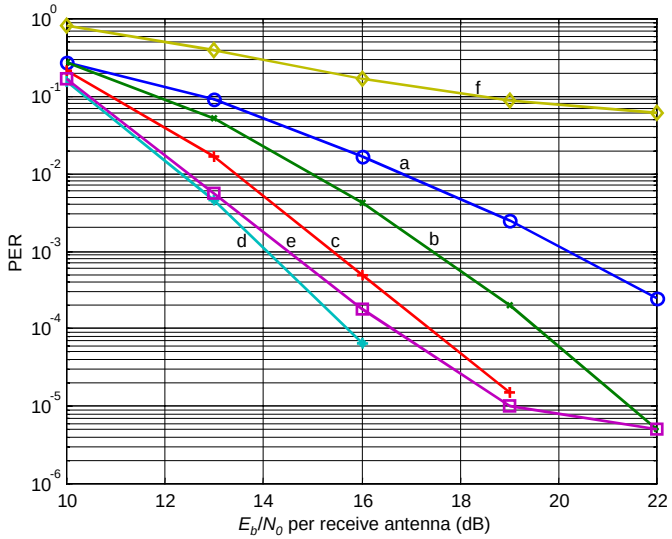

Figure 5-14: Packet Error Ratio versus mean E_b/N_0 per receive antenna in a channel with an exponential decaying power profile for QPSK, 64 byte packets, simple MLD algorithm with soft-decision outputs, rate 1/2 convolutional coding, antenna configuration (N_t,N_r) = (2,2), normalised delayspread τ_{rms}/T_G = a) 1/48, b) 1/24, c) 1/12, d) 1/6, e) 1/3, f) 2/3.

This conclusion can also be drawn when we compare the irreducible packet error ratios of the simple MLD method that produces soft-decision outputs (Figure 5-15) with the irreducible packet error ratios of Figure 5-11. From the latter figure, it can clearly be seen that, for normalised delay spread < 0.7, the (N_t,N_r) = (2,2) situation (curve e) has a lower irreducible PER floor than the (N_t,N_r) = (1,1) case with coding (curve b). Comparing the curves a and d from Figure 5-15, it can be seen that here, this crossing point (if it exists) must has a normalised delay spread lower than 0.5. So, the performance difference between the (N_t,N_r) = (1,1) situations is larger than between the (N_t,N_r) = (2,2) situations, which indeed confirms the conclusion above.

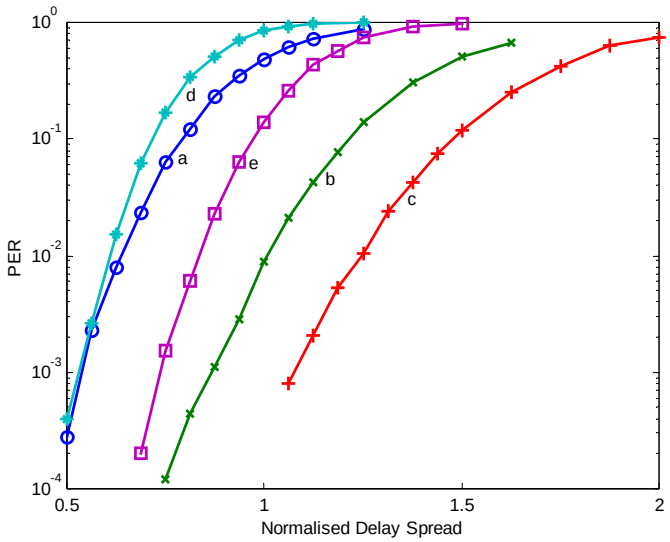

Figure 5-15: Irreducible packet error ratios versus normalised delay spread τ_{rms}/T_G for 64 byte packets, QPSK, simple MLD algorithm with soft-decision outputs, rate 0.5 code and antenna configuration (N_t,N_r): a) (1,1), b) (1,2), c) (1,3), d) (2,2), e) (2,3).

5.6.3 Complex MLD algorithm with soft-decision outputs

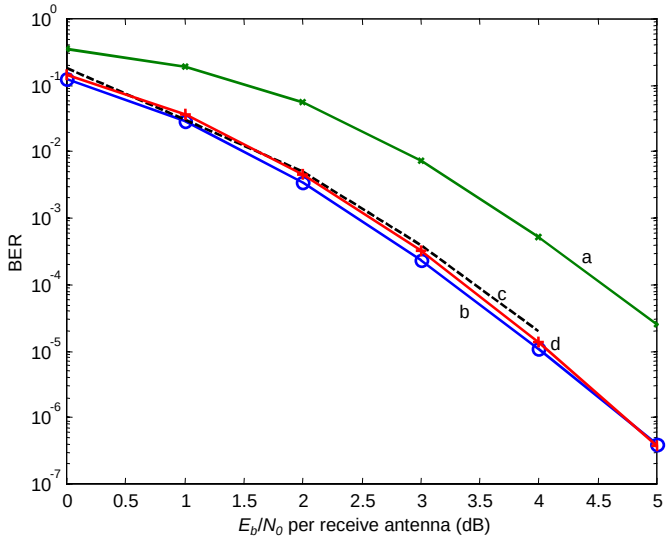

Figure 5-16: Bit error ratios versus E_b/N_0 per receive channel in AWGN for QPSK, 64 byte packets, complex MLD algorithm with soft-decision outputs, rate 1/2 convolutional code, antenna configuration (N_t,N_r) = (1,1) with a) channel estimation and a guard time of 800 ns, b) perfect knowledge of the channel and no guard time and c) results from [57] and d) same curve as b, only for 512 byte packets.

The results of the Maximum Likelihood Decoding algorithm, producing soft-decision outputs according to the log-likelihood ratio, will be discussed in this subsection. For this case, it is interesting to look at the performance in Additive White Gaussian Noise, because even for a

channel **H** = 1, the complex MLD algorithm with soft-decision outputs is able to produce soft-decision values.

In Figure 5-16 the Bit Error Ratio is depicted against E_b/N_0 per receive antenna for AWGN. Curve a shows the result for rate 1/2 convolutional code, antenna configuration (N_t,N_r) = (1,1), with channel estimation and a guard time. Curve b shows the same result only now the receiver has perfect knowledge of the channel and no guard time is implemented. We (again) see a difference of about 1.5 dB between the curves, because by estimating instead of knowing the channel the system performance deteriorates and also introducing a guard time has a negative influence on the performance. However, note that in case of delay spread a guard time is introduced to make the system more robust against delay spread. Curve c of Figure 5-16 is obtained from [57] and shows the results of a 64-state rate 1/2 convolutional code in AWGN for a 'regular' OFDM transmitting system without guard time and with perfect knowledge by the receiver of the channel. Because our convolutional code is also a 64-state code, the results can be compared. Clearly, our system performs somewhat better. This can be explained by the fact that we use relative short packets of 64 bytes and in order to use a Viterbi decoder, every coded packet has to end with 8 bits of zero. In case of short packets, this is some kind of coding overhead the Viterbi decoder benefits from. E.g., by using packets of 512 bytes (see curve d), the performance slightly deteriorates compared to curve b.

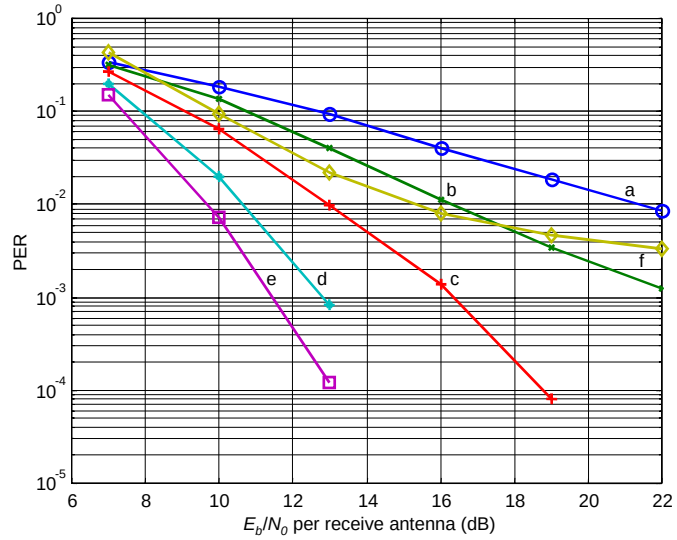

Figure 5-17: Packet Error Ratio versus mean E_b/N_0 per receive antenna in a channel with an exponential decaying power profile for QPSK, 64 byte packets, complex MLD algorithm with soft-decision outputs, rate 1/2 convolutional coding, antenna configuration (N_t,N_r) = (1,1), normalised delayspread τ_{rms}/T_G = a) 1/48, b) 1/24, c) 1/12, d) 1/6, e) 1/3, f) 2/3.

Figure 5-17, Figure 5-18 and Figure 5-19 show the PER performance under different delay spreads, for the antenna configurations (N_t,N_r) = (1,1), (N_t,N_r) = (1,2) and (N_t,N_r) = (2,2), respectively. It might be interesting to check the performance improvement compared to hard-decision output MLD, e.g. for antenna configuration (N_t,N_r) = (1,1). Comparing the curves a of Figure 5-8 and Figure 5-17 we see an improvement of 2 dB. This is equal to the improvement of the simple MLD algorithm with soft-decision outputs (see Subsection 5.6.2). However, the improvement for curve d is approximately 6.5 dB, which is larger than that of the simple MLD algorithm with soft-decision outputs. Clearly, the MLD algorithm that produces soft-decision outputs according to the log-likelihood ratio is more robust for higher delay spreads.

In contrast to the simple MLD algorithm with soft-decision outputs, for the complex MLD with soft-decision outputs we see a clear performance improvement for larger antenna configurations (compare Figure 5-9, Figure 5-13 & Figure 5-18 and Figure 5-10, Figure 5-14 & Figure 5-19). Thus, also for large antenna configurations, this method keeps performing better that hard-decision output MLD. However, the main drawback of this method is that it requires a full search *per bit* to decode (see Subsection 5.5.2), whereas, hard-decision output MLD requires a full search per transmitted vector.

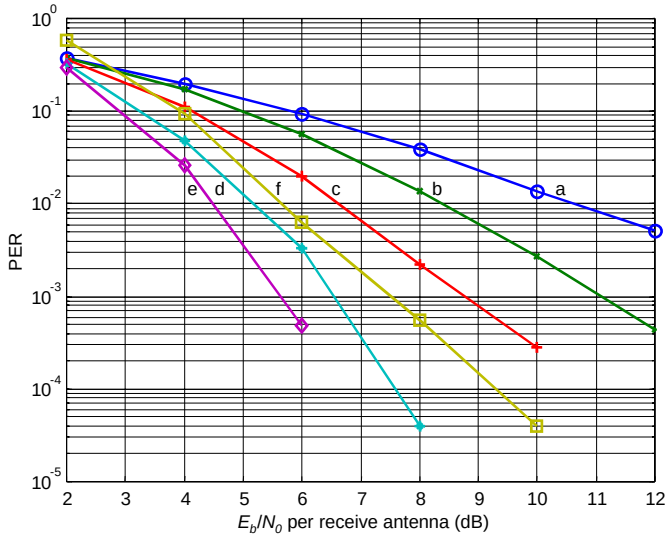

Figure 5-18: Packet Error Ratio versus mean E_b/N_0 per receive antenna in a channel with an exponential decaying power profile for QPSK, 64 byte packets, complex MLD algorithm with soft-decision outputs, rate 1/2 convolutional coding, antenna configuration (N_t,N_r) = (1,2), normalised delayspread τ_{rms}/T_G = a) 1/48, b) 1/24, c) 1/12, d) 1/6, e) 1/3, f) 2/3.

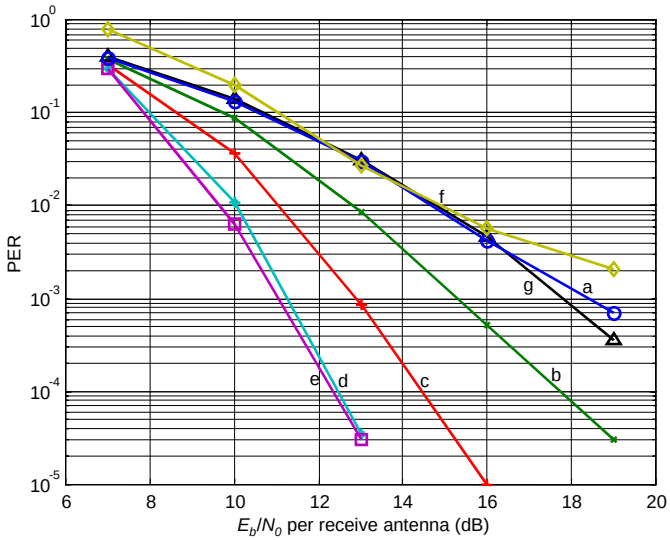

Figure 5-19: Packet Error Ratio versus mean E_b/N_0 per receive antenna in a channel with an exponential decaying power profile for QPSK, 64 byte packets, complex MLD algorithm with soft-decision outputs, rate 1/2 convolutional coding, antenna configuration (N_t,N_r) = (2,2), normalised delayspread τ_{rms}/T_G = a) 1/48, b) 1/24, c) 1/12, d) 1/6, e) 1/3, f) 2/3 and g) 1/48 with exact soft-decision output MLD according to the beginning of Formula (5.5).

Figure 5-20 shows the PER floor versus the normalised delay spread for different antenna configurations and for a high SNR value of 80 dB. Comparing Figure 5-11, Figure 5-15 and Figure 5-20, we clearly see that the complex MLD algorithm with soft-decision outputs is the most robust against delay spread impairments. So, overall it can be concluded that this algorithm has the best PER performance and the best performance under delay spread.

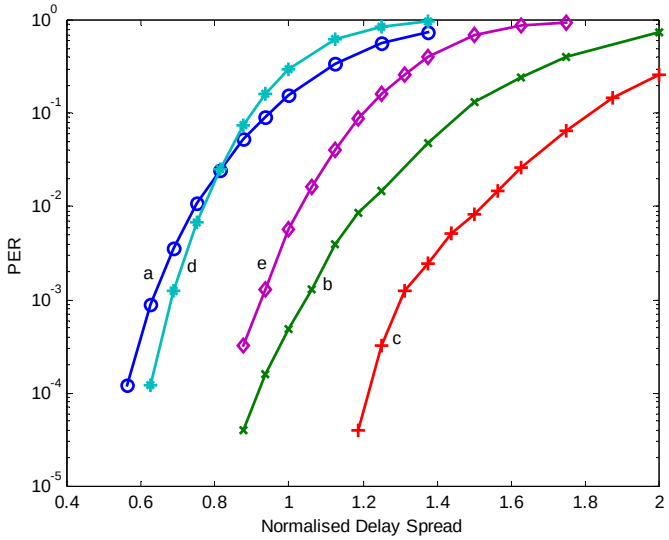

Figure 5-20: Irreducible packet error ratios versus normalised delay spread τ_{rms}/T_G for 64 byte packets, QPSK, complex MLD algorithm with soft-decision outputs, rate 0.5 code and antenna configuration (N_t,N_r): a) (1,1), b) (1,2), c) (1,3), d) (2,2), e) (2,3).

6 Conclusions and future work

6.1 Conclusions

The main goals in developing new wireless communication systems are increasing the bit rate and increasing system capacities. In literature ([15] and [37]), it has been shown that the multipath wireless channel is capable of enormous capacities, provided that the multipath scattering is sufficiently rich and is properly exploited. The Space Division Multiplexing (SDM) techniques proposed in this report are promising solutions to properly exploit the multipath. Basically, these techniques transmit different signals simultaneously on different transmit antennas. While these parallel streams of data are mixed-up in the air, the transmitted signals still can be recovered at the receiver.

The SDM techniques proposed in this report are called Zero Forcing (ZF), Minimum Mean Square Error (MMSE), ZF with Decision Feedback Decoding (DFB), MMSE with DFB and Maximum Likelihood Decoding (MLD) and are listed here in increasing performance. It can be shown that the diversity order of ZF, in case of Rayleigh fading, equals the number of receive antennas minus the number of transmit antennas plus one [58], whereas, in this thesis, the diversity order of MLD is shown to be equal to the number of receive antennas. So, the Space Division Multiplexing techniques not only provide an increase in capacity but also an improvement of the Bit Error Rate (BER) performance due to diversity.

The major disadvantage of the MLD method is that its complexity grows exponentially with the number of transmit antennas. But it is shown that for a reasonable number of transmit antennas (less than 6 for BPSK) and a somewhat adapted and slightly less performing MLD algorithm the complexity is comparable to or even less than the other methods.

To obtain sufficient delay spread robustness, the best performing SDM technique, namely, MLD is applied to the spectrum efficient transmission technique called Orthogonal Frequency Division Multiplexing (OFDM). The delay spread channel is modelled by exponentially-decayed Rayleigh fading. From the results of Monte Carlo simulations, it can be concluded that, firstly, introducing one extra transmit antenna does not change the diversity order of MLD, but the overall BER performance and the delay spread performance deteriorate, because this extra transmit antenna can be seen as an extra “interferer”. Secondly, introducing one extra transmit and one extra receive antenna not only increases the bit rate but also the BER performance improves due to diversity, while the delay spread performance stays more or less the same. Thirdly, introducing only one extra receive antenna makes the system more robust against delay spread and the BER performance is increased.

When a forward error correction code, like a convolutional code, is used and decoded at the receiver by, e.g., a Viterbi decoder, even better BER performances and delay spread performances are achieved. If this Viterbi decoder is able to decode soft-decision values and the MLD algorithm is adapted in such a way that it delivers soft-decision values, according to the log-likelihood ratio, as output, superior BER performance and delay spread performance is obtained.

The overall conclusion is that Space Division Multiplexing techniques, especially the Maximum Likelihood Decoding algorithm, are a promising solution for achieving spectrum efficient transmission of data in a rich scattered environment. Because these techniques exploit the multipath scattering properly, the data rate and the Signal-to-Noise performance can be increased.

6.2 Future work

As can be concluded from this thesis, Space Division Multiplexing is a promising technique to increase bit rates and system capacities. However, there is still a lot of theoretical and practical work to be done. Concerning the theoretical work, firstly, it would be of great use if exact representations or tight upperbounds of the Bit Error Rate performance of the SDM techniques, discussed in this report, are found. Secondly, the SDM techniques are all based on methods that also appear in equalisation, remote sensing and image processing. So, it might be interesting to define global algorithms, from which the methods of these different areas can be deduced. It should be noted that in this report, only a one way communication link is described Thus, in order to design a realistic (bi-directional) physical layer for a communication system like a wireless LAN, still a lot of work needs to be done. As a solution, we are thinking of a Time Division Duplex structure.

Considering more practical work, firstly, a lot of effort should be spent on trying to optimise the SDM algorithms in order to reduce their complexity. Usually, a trade-off is involved here, because in order to find a less complex algorithm, most likely a penalty has to be paid in performance. Secondly, in this report it is shown that the BER performance improves if soft-decision values are used as input for the soft-decision input Viterbi decoder. This states an interesting question, namely, may we see SDM as a code (the ‘inner’ code) and is it possible to combine this in such a way with a convolutional code (or another ‘outer’ code), that a form of turbo coding is achieved? Of course, again, it should be tried to keep the complexity as low as possible. Thirdly, because a practical system with SDM has a very high bit rate, the amount of overhead compared to the packet duration increases. This makes the total system more inefficient than current wireless communication systems. Therefore, in the future a more efficient transmission protocol must be found to deal with this problem, e.g., by making the packet size larger, or reducing the overhead, or both.

In the nearby future, also practical work needs to be done, e.g., to check the assumptions we have made about the channel. This can be done by using some kind of channel sounder to measure the channel. If the channel indeed appears to be rich scattered (which seems logical) and if less complex algorithms have been found for SDM, a prototype could be build in order to do practical measurements on the system and prove the spectrum efficiency of Space Division Multiplexing.

7 Acknowledgements

At the end of this report I would like to express my gratitude to several people. First of all I want to thank my supervisors Dr. Ir. R. van Nee and Dr. Ir. G. Awater of Bell Labs, Lucent Technologies, Nieuwegein, for their constructive advise, co-ordination and review. Without their effort this work would not have reached its present form.

Furthermore, I want to thank Prof. G. Brussaard and Dr. Ir. P.F.M. Smulders from the Eindhoven University of Technology for their supervision and constructive review and for arranging the possibility to do my thesis project in a company.

Finally, I want to thank Lucent Technologies, Nieuwegein for allowing me to work is such a stimulating environment, for offering me a project from which I have learned a lot and for their sponsorship.

Allert van Zelst
Nieuwegein, September 1999

References

[1] Agrawal D. and V. Tarokh, A. Naguib, N. Seshadri,
"Space-time coded OFDM for high data-rate wireless communication over wideband channels",
48th IEEE Vehicular Technology Conference 1998, vol. 3, 1998, pp. 2232-2236.

[2] Alamouti S.M.,
"A simple transmit diversity technique for wireless communications",
IEEE Journal on Selected Areas in Communications, vol. 16, October 1998, no. 8, pp. 1451-1458.

[3] Anderson T.W.,
"An Introduction to Multivariate Statistical Analysis",
Second edition, New York, John Wiley & Sons, 1984.

[4] Bulumulla S.B. and S.A. Kassam, S.S. Venkatesh,
"An adaptive diversity receiver for OFDM in fading channels",
IEEE International Conference on Communications 1998, vol. 3, 1998, pp. 1325-1329.

[5] Bultitude R.J.C. and S.A. Mahmoud, W.A. Sullivan,
"A Comparison of Indoor Radio Propagation Characteristics at 910 MHz and 1.75 GHz",
IEEE Journal on Selected Areas in Communications, vol. 7, January 1989, no. 1, pp. 20-30.

[6] Cimini L.J., Jr.,
"Analysis and Simulation of a Digital Mobile Channel Using Orthogonal Frequency Division Multiplexing",
IEEE Trans. on Communications, vol. com-33, July 1985, no. 7, p. 665-675.

[7] Cimini L.J., Jr. and N.R. Sollenberger,
"OFDM with Diversity and Coding for Advanced Cellular Internet Services",
Global Telecommunications Conference 1997, vol. 1, 1997, pp. 305-309.

[8] Davies R. and J.P. McGeehan,
"Propagation Measurements at 1.7 GHz for Microcellular Urban Communications",
Electronics Letters, vol. 26, July 5, 1990, no. 14, pp. 1053-1055.

[9] Devasirvatham D.M.J.,
"Multi-Frequency Propagation Measurements and Models in a Large Metropolitan Commercial Building for Personal Communications",
IEEE PIMRC'91, London, September 23-25, 1991, pp. 98-103.

[10] Devasirvatham D.M.J.,
"Time Delay Spread Measurements at 850 MHz and 1.7 GHz inside a Metropolitan Office Building",
Electronics Letters, February 2, 1989, pp. 194-196.

[11] Devasirvatham D.M.J., and M.J. Krain, D.A. Rappaport,
"Radio Propagation Measurements at 850 MHz, 1.7 GHz and 4 Ghz inside two dissimilar office buildings",
Electronics Letters, vol. 26, March 29, 1990, no. 7, pp. 445-447.

[12] ETSI home page,
http://www.etsi.org/, Sept. 1999.

[13] FCC,
"Amendment of the Commission's Rules to Provide for Operation of Unlicensed NII Devices in the 5-GHz Frequency Range",
memorandum option and order, ET Docket No. 96-102, June 24, 1998.

[14] Foschini, G.J.,
"Layered Space-Time Architecture for Wireless Communication in a Fading Environment When Using Multiple Antennas",
Bell Laboratories Technical Journal, vol. 1, autumn 1996, no. 2, pp. 41-59.

[15] Foschini G.J. and M.J. Gans,
"On Limits of Wireless Communications in a Fading Environment When Using Multiple Antennas",
Wireless Personal Communications, Volume 6, March 1998, no. 3, pp. 311-335.

[16] Hafesi P. and D. Wedge, M. Beach, M. Lawton,
"Propagation Measurements at 5.2 GHz in Commercial and Domestic Environments",
IEEE PIMRC '97, Helsinki, September 1-4, 1997, pp. 509-513.

[17] Hagenauer J. and P. Hoeher,
"A Viterbi Algorithm with Soft-Decision Outputs and its Applications",
Proc. IEEE Globecom 1989, Dallas, Texas, Nov. 1989, pp. 47.1.1-47.1.7.

[18] Hamazumi H. and Y. Ito, H. Miyazawa,
"Performance of Frequency Domain Sub-band Diversity Combination Technique for Wide-band Mobile Radio Reception: An Application to OFDM (Orthogonal Frequency Division Multiplexing) Reception",
Electronics and Communications in Japan, Part 1, vol. 81, 1998, no. 8, pp. 32-41. Translated from Denshi Joho Tsushin Gakkai Ronbunshi, vol. J80-B-II, June 1997, no. 6, pp. 466-474.

[19] Hashemi H.,
"The Indoor Propagation Channel",
Proceedings of the IEEE, vol. 81, July 1993, no. 7, pp. 943-968.

[20] Hawbaker D.A. and T.S. Rappaport,
"Indoor Wideband Radiowave Propagation Measurements at 1.3 GHz and 4.0 GHz",
Electronics Letters, vol. 26, October 11, 1990, no. 21, pp. 1800-1802.

[21] Hill A.,
"Wireless Andrew: Bringing mobile computing to a university community of 10,000",
IEEE Spectrum, vol. 36, June 1999, no. 6, pp. 49-53.

[22] Horn R.A. and C.R. Johnson,
"Matrix Analysis",
Cambridge, Cambridge University Press, 1985.

[23] IEEE 802.11,
"IEEE Standard for Wireless LAN, Medium Access Control (MAC) and Physical Layer (PHY) specifications",
Nov. 1997.

[24] IEEE 802.11 Web URL,
http://grouper.ieee.org/groups/802/11/,
Sept. 1999.

[25] IEEE,
'Draft Supplement to Standard for Telecommunications and Information Exchange Between Systems – LAN/MAN Specific Requirements – Part 11: Wireless MAC and PHY Specifications: High Speed Physical Layer in the 5-GHz band",
P802.11a/D5.0, April 1999.

[26] Joonsuk Kim and L.J. Cimini, Jr, J.C. Chuang,
"Coding strategies for OFDM with antenna diversity for high-bit-rate mobile data applications",
48th IEEE Vehicular Technology Conference 1998, vol. 2, 1998, pp. 763-767.

[27] Kailath T. and A.H. Sayed, B. Hassibi,
"State Space Estimation",
New Jersey, Prentice-Hall, expected in Nov. 1999.

[28] Kopmeiners, R.,
"Analytical Evaluation of Imperfect Diversity Branches Applied to Polarization Diversity Measurements at 1800 MHz",
IEEE First Symp. on Communications and Vehicular Technology in the Benelux, 1993,
pp. 3.3.1-3.3.6.

[29] MMAC, homepage,
http://www.arib.or.jp/mmac/e/index.htm, Sept. 1999

[30] Matsumoto Y. and M. Umehira,
"Subchannel Space-Combining Transmission Diversity Incorporating OFDM",
ETSI EP BRAN#7, January 30, 1998.

[31] Naguib A.F. and V. Tarokh, N. Seshadri and A.R. Calderbank,
"A Space-Time Coding Modem for High-Data-Rate Wireless Communications",
IEEE Journal on Selected Areas in Communications, vol. 16, Oct. 1999, no. 8,
pp. 1459-1478.

[32] Pahlavan K. and S.J. Howard,
"Frequency Domain Measurements of Indoor Radio Channels",
Electronics Letters, vol. 25, November 23, 1989, no. 24, pp. 1645-1647.

[33] Press W.H. and S.A. Teukolsky, W.T. Vetterling, B.P. Flannery,
"Numerical Recipes in C, The Art of Scientific Computing",
Second Edition, Cambridge, Cambridge University Press, 1992.

[34] Proakis J.G.,
"Digital Communications",
Third Edition, New York, McGraw-Hill, 1995, McGraw-Hill Series in Electrical and Computer Engineering.

[35] Rappaport T.S. and C.D. McGillem,
"Characterization of UHF Multipath Radio Channels in Factory Buildings",
IEEE Transactions on Antennas and Propagation, vol. 37, August 1989, no. 8,
pp. 1058-1069.

[36] Rappaport T.S.,
"Wireless Communications, Principles and Practice",
New Jersey, Prentice-Hall, 1996.

[37] Raleigh G.G. and J.M. Cioffi,
"Spatio-temporal coding for wireless communication",
IEEE Trans. on Communications, vol. 46, March 1998, no. 3, pp. 357-366.

[38] Raleigh G.G. and T. Boros,
"Joint space-time parameter estimation for wireless communication channels",
IEEE Transactions on Signal Processing, Vol. 46, May 1998, no. 5, pp. 1333-1343.

[39] Roy Y. and J. Chouinard, S.A. Mahmoud,
"Selection Diversity Combining with Multiple Antennas for MM-Wave Indoor Wireless Channels",
IEEE Journal on Sel. Areas in Communications, vol. 14, May 1996, no. 4, pp. 674-682.

[40] Saleh A.A.M. and R.A. Valenzuela,
"A Statistical Model for Indoor Multipath Propagation",
IEEE Journal on Selected Areas in Communications, vol. SAC-5, February 1987, no. 2, pp. 128-137.

[41] Salz J.,
"Digital Transmission Over Cross-Coupled Linear Channels",
AT&T Technical Journal, vol. 64, July-August 1985, no. 6, pp. 1147-1159.

[42] Z. Sayeed and S.A. Kassam,
"Orthogonal Frequency Division Multiplexing with Coding and Transmitter Antenna Diversity",
Proc. of SPIE – The Int. Society for Optical Engineering, vol. 2601, Oct. 23-25, 1995, pp. 169-176.

[43] Shiann-Shiun Jeng, G.T. Okamoto, Guanghan Xu, W.J. Vogel, Hsin-Piao Lin,
"Experimental evaluation of smart antenna system performance for wireless communications",
IEEE Transactions on Antennas and Propagation, vol. 46, June 1998, no. 6, pp. 749-757.

[44] Speth M.,
"OFDM Receivers for Broadband-Transmission",
http://www.ert.rwth-aachen.de/Projekte/Theo/OFDM/www_ofdm.html, April 1999.

[45] Strang G.,
"Linear Algebra and its Applications",
Third Edition, San Diego, Harcourt Brace Jovanovich, Publishers, 1988.

[46] Takanashi H. and R. van Nee,
"Merged Physical Layer Specification for the 5-GHz Band",
IEEE P802.11-98/72-r1, March 1998.

[47] Telatar I.E.,
"Capacity of Multi-antenna Gaussian Channels",
Lucent Technologies, Internal report, BL011217-950615-07TM, June 1995.

[48] Tellado-Mourelo J. and E.K.Wesel, J.M.Cioffi,
"Adaptive DFE for GMSK in Indoor Radio Channels",
IEEE Trans. On Sel. Areas in Comm., vol. 14, April 1996, no. 3, pp. 492-501.

[49] Vahid T. and N. Seshadri, A.R. Calderbank,
"Space-Time Codes for High Data Rate Wireless Communication: Performance Criterion and Code Construction",
IEEE Transactions on Information Theory, vol. 44, March 1998, no. 3, pp. 744-756.

[50] Van Nee R.,
"Delay Spread Requirements for Wireless Networks in the 2.4 GHz and 5 GHz Bands",
doc.:IEEE P802.11-97/125, November 1997.

[51] Van Nee R.,
"Maximum Likelihood Decoding in a Space Division Multiplex System",
Submitted to IEEE VTC 2000-Spring, Japan, May 2000.

[52] Van Nee R. and G. Awater, Masahiro Morikura, Hitoshi Takanashi, M. Webster, K. Helford,
"New High Rate Wireless LAN standards",
IEEE Communications Magazine, to be published in December 1999.

[53] Van Nee R. and R. Prasad,
"OFDM for Mobile Multimedia Communications",
Boston, Artech House, to appear December 1999.

[54] Vook F.M. and K.L. Baum,
"Adaptive Antennas for OFDM",
48th IEEE Vehicular Technology Conference 1998, vol. 1, 1998, pp. 606-610.

[55] Wales S.W.,
"Modulation and Equalization Techniques for HIPERLAN",
Proceedings of PIMRC/WCN, The Hague, September 21-23, 1994, pp. 959-963.

[56] Weinstein S.B. and P.M. Ebert,
"Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform",
IEEE Trans. on Communication Technology, vol. com-19, October 1971, no. 5, pp. 628-634.

[57] Wesel R.D. and J.M. Coiffi,
"Fundamentals of Coding for Broadcast OFDM",
In proceedings of 29th Asilomar Conf. on Signals, Systems, and Computers (ASILOMAR-29 '95), 1996, pp. 2-6.

[58] Winters J.H. and J. Salz, R.D. Gitlin,
"The inpact of antenna diversity on the capacity of wireless communication systems",
IEEE Trans. on Commun., vol. 42, Feb./Mar./Apr. 1994, no. 2, pp. 1740-1751.

[59] Wittneben A.,
"A new bandwidth efficient transmit antenna modulation diversity scheme for linear digital modulation",
Proc. 1993 IEEE Int. Communications Conf., Geneve, Switzerland, June 1993,
pp. 1630-1634.

[60] Wolniansky P.W. and G.J. Foschini, G.D. Golden, R.A. Valenzuela,
"V-Blast: An Architecture for Realizing Very High Data Rates Over the Rich-Scattering Wireless Channel",
1998 URSI International Symposium on Signals, Systems, and Electronics, ISSSE 98, Pisa, 29 Sept. - 2 Oct. 1998, pp. 295-300.

[61] Ye Li and J.C. Chuang, N.R. Sollenberger,
"Transmitter Diversity for OFDM Systems and Its Impact on High-Rate Data Wireless Networks",
IEEE Journal on Sel. Areas in Communications, vol. 17, July 1999, no. 7, pp. 1233-1243.

Appendix A

Zero Forcing algorithm

```
function
   ber = ZF(nt,nr,snr,numsnr,packetlength,experiments,limit,numphases)
% simulates multiple antenna system, with pseudo inverse.
% function call:
   %ZF(nt,nr,snr,numsnr,packetlength,experiments,limit,numphases)
% where:
%  nt = #TX antennas;
%  nr = #RX antennas;
%  snr = SNR array (EbN0 per RX antenna);
%  numsnr = #SNR values in SNR array;
%  limit = if bit errors > limit then stop;
%  numphases = #phases of constellation; 2 for BPSK and 4 for QPSK
% returns mean bit error rate.

numbitspersymbol=log(numphases)/log(2);

ps = zeros(size(snr));
outage=0;
snrlow = 1;
expnr=0;

while (expnr<experiments) & (ps(numsnr)<limit),
   if (ps(snrlow)>limit-1)
      snrlow=snrlow+1;
   end;
   expnr=expnr+1;
   dat=floor(numphases*rand(nt,packetlength));  % make random data
   datest=0*dat;

   H=(randn(nr,nt)+j*randn(nr,nt))/sqrt(2);  % Make random channel matrix
   PinvH = pinv(H);
   N=(randn(nr,packetlength)+j*randn(nr,packetlength))/sqrt(2);

   % Transmitter
   x=exp(j*(dat*2*pi/numphases));
   rsig = H*x;

   % Receiver
   for s = snrlow:numsnr,
      exparray(s)=expnr;
      alpha = nt/(10^((snr(s)+10*log10(numbitspersymbol))/10));
         % alpha of noise, corrected to get
         % Eb/N0 curve in stead of Es/N0 curve
      r=rsig+sqrt(alpha)*N;   % r = received array
      estimate = PinvH*r;

      datest = round(angle(estimate)/(2*pi/numphases));
         % Estimate nt phases
      datest = rem(datest + numphases,numphases);

      errors=dat-datest;      % calculate the BER
      errors=errors(errors~=0);
      errors=min(abs(errors),numphases-abs(errors));
      ps(s)=ps(s)+sum(errors);
   end; %snr loop
end; % experiments loop
for s = 1:numsnr,
   bertemp(s) = ps(s)/(exparray(s)*packetlength*nt*numbitspersymbol);
end;
ber = bertemp;
```

Minimum Mean Square Error algorithm

```
function
   ber = LMS(nt,nr,snr,numsnr,packetlength,experiments,limit,numphases)
% simulates multiple antenna system, with LMS.
% function call:
   %LMS(nt,nr,snr,numsnr,packetlength,experiments,limit,numphases)
% where:
%  nt = #TX antennas;
%  nr = #RX antennas;
%  snr = SNR array (SNR per RX antenna);
%  numsnr = #SNR values in SNR array;
%  limit = if bit errors > limit then stop;
%  numphases = #phases of constellation; 2 for BPSK and 4 for QPSK
% returns mean bit error rate.

numbitspersymbol=log(numphases)/log(2);

ps = zeros(size(snr));
snrlow = 1;
outage=0;
expnr=0;

while (expnr<experiments) & (ps(numsnr)<limit),
   if (ps(snrlow)>limit-1)
      snrlow=snrlow+1;
   end;
   expnr=expnr+1;
   dat=floor(numphases*rand(nt,packetlength));  % make random data
   datest=0*dat;

   H=(randn(nr,nt)+j*randn(nr,nt))/sqrt(2);  % Make random channel matrix
   N=(randn(nr,packetlength)+j*randn(nr,packetlength))/sqrt(2);

   % Transmitter
   x=exp(j*(dat*2*pi/numphases));
   rsig = H*x;

   % Receiver
   for s = snrlow:numsnr,
      exparray(s)=expnr;
      alpha = nt/(10^((snr(s)+10*log10(numbitspersymbol))/10));
         % alpha of noise, corrected to get
         % Eb/N0 curve in stead of Es/N0 curve
      r=rsig+sqrt(alpha)*N;   % r = received array

      Htemp=[H;sqrt(alpha)*eye(nt)];
      r=[r;zeros(nt,packetlength)];

      estimate = pinv(Htemp)*r;

      datest = round(angle(estimate)/(2*pi/numphases));
         % Estimate nt phases
      datest = rem(datest+numphases,numphases);

      errors=dat-datest;      % calculate BER
      errors=errors(errors~=0);
      errors=min(abs(errors),numphases-abs(errors));
      ps(s)=ps(s)+sum(errors);
   end; %snr loop
end; % experiments loop
for s = 1:numsnr,
   bertemp(s) = ps(s)/(exparray(s)*packetlength*nt*numbitspersymbol);
end;
ber = bertemp;
```

Zero Forcing with Decision Feedback

```
function
  ber=ZFwithOD(nt,nr,snr,numsnr,packetlength,experiments,limit,numphases)
% simulates multiple antenna system, with pseudo inverse, DFD and optimal
% detection
% function call:
   % ZFwithOD(nt,nr,snr,numsnr,packetlength,experiments,limit,numphases)
% where:
%  nt = #TX antennas;
%  nr = #RX antennas;
%  snr = SNR array (SNR per RX antenna);
%  numsnr = #SNR values in SNR array;
%  limit = if bit errors > limit then stop;
%  numphases = #phases of constellation; 2 for BPSK and 4 for QPSK
% returns mean bit error rate.

numbitspersymbol=log(numphases)/log(2);

ps = zeros(size(snr));
snrlow = 1;
outage=0;
expnr=0;

while (expnr<experiments) & (ps(numsnr)<limit),
   if (ps(snrlow)>limit-1)
      snrlow=snrlow+1;
   end;
   expnr=expnr+1;
   dat=floor(numphases*rand(nt,packetlength));  % make random data
   datest=0*dat;

   H=(randn(nr,nt)+j*randn(nr,nt))/sqrt(2);  % Make random channel matrix
   N=(randn(nr,packetlength)+j*randn(nr,packetlength))/sqrt(2);

   % Transmitter
   x=exp(j*(dat*2*pi/numphases));
   rsig=H*x;

   % Receiver
   for s = snrlow:numsnr,
      exparray(s)=expnr;
      alpha = nt/(10^((snr(s)+10*log10(numbitspersymbol))/10))
         % alpha of noise, corrected to obtain Eb/N0
      r=rsig+sqrt(alpha)*N;   % r = received array

      estimate = 0*dat;
      Htemp = H;
      dattemp = dat;
      for i = nt:-1:1,
         % iteration for finding the next column with the smallest power
         G = pinv(Htemp);
         [Y,k] = min(sum(abs(G).^2,2));

         PER = eye(i); PER(k,k) = 0; PER(i,i) = 0; %create a matrix to
         PER(k,i) = 1; PER(i, k) = 1;              %permute H and pinv(H)
         dattemp(1:i,:) = PER * dattemp(1:i,:);
         G = PER * G;
         Htemp = Htemp * PER;

         esttemp = G(i,:) * r;
         datest(i,:)=round(angle(esttemp)/(2*pi/numphases));
            % Estimate nt phases
         datest(i,:)=rem(datest(i,:)+numphases,numphases);
         esttemp = exp(j*2*pi*datest(i,:)/numphases);
```

```
         r = r - Htemp(:,i)*esttemp;
         Htemp = Htemp(:,1:i-1);
      end;

      errors=dattemp-datest;     % calculate BER
      errors=errors(errors~=0);
      errors=min(abs(errors),numphases-abs(errors));
      ps(s)=ps(s)+sum(errors);
   end; % snr loop
end; % experiments loop
for s = 1:numsnr,
   bertemp(s) = ps(s)/(exparray(s)*packetlength*nt*numbitspersymbol);
end;
ber = bertemp;
```

Minimum Mean Square Error algorithm with Decision Feedback

```
function
 ber=LMSwithOD(nt,nr,snr,numsnr,packetlength,experiments,limit,numphases)
% simulates multiple antenna system, with LMS, DFD and optimal detection
% function call:
   % LMSwithOD(nt,nr,snr,numsnr,packetlength,experiments,limit,numphases)
% where:
%  nt = #TX antennas;
%  nr = #RX antennas;
%  snr = SNR array (SNR per RX antenna);
%  numsnr = #SNR values in SNR array;
%  limit = if bit errors > limit then stop;
%  numphases = #phases of constellation; 2 for BPSK and 4 for QPSK
% returns mean bit error rate.

numbitspersymbol=log(numphases)/log(2);

snrlow=1;
ps=zeros(size(snr));
outage=0;
expnr=0;

while (expnr<experiments) & (ps(numsnr)<limit),
   if (ps(snrlow)>limit-1)
      snrlow=snrlow+1;
   end;
   expnr=expnr+1;
   dat=floor(numphases*rand(nt,packetlength));  % make random data
   datest=0*dat;

   H=(randn(nr,nt)+j*randn(nr,nt))/sqrt(2);  % Make random channel matrix
   N=(randn(nr,packetlength)+j*randn(nr,packetlength))/sqrt(2);

   % Transmitter
   x=exp(j*(dat*2*pi/numphases));
   rsig = H*x;

   % Receiver
   for s = snrlow:numsnr,
      exparray(s)=expnr;
      alpha = nt/(10^((snr(s)+10*log10(numbitspersymbol))/10));
         % alpha of noise, corrected to obtain Eb/N0
      r=rsig+sqrt(alpha)*N;   % r = received array

      Htemp=[H;sqrt(alpha)*eye(nt)];
      r=[r;zeros(nt,packetlength)];

      estimate = 0*dat;
      dattemp = dat;
      for i = nt:-1:1,
```

```
         % iteration for finding the next column with the smallest power
         G = pinv(Htemp);
         [Y,k] = min(sum(abs(G).^2,2));

         PER = eye(i); PER(k,k) = 0; PER(i,i) = 0; %create a matrix to
         PER(k,i) = 1; PER(i, k) = 1;              %permute H and pinv(H)
         dattemp(1:i,:) = PER * dattemp(1:i,:);
         G = PER * G;
         Htemp = Htemp * PER;

         esttemp = G(i,:) * r;
         datest(i,:)=round(angle(esttemp)/(2*pi/numphases));
            % Estimate nt phases
         datest(i,:)=rem(datest(i,:)+numphases,numphases);
         esttemp = exp(j*2*pi*datest(i,:)/numphases);

         r = r - Htemp(:,i)*esttemp;
         Htemp = Htemp(:,1:i-1);
      end;

      errors=dattemp-datest;        % calculate BER
      errors=errors(errors~=0);
      errors=min(abs(errors),numphases-abs(errors));
      ps(s)=ps(s)+sum(errors);
   end; % snr loop
end; % experiments loop
for s = 1:numsnr,
   bertemp(s) = ps(s)/(exparray(s)*packetlength*nt*numbitspersymbol);
end;
ber = bertemp;
```

Maximum Likelihood Decoding algorithm

```
function
   ber =
meamld(nt,nr,snr,numsnr,packetlength,experiments,limit,numphases)
% simulates multiple antenna system, using a maximum likelihood decoder
% function call:
   % meamld(nt,nr,snr,numsnr,packetlength,experiments,limit,numphases)
% where:
%  nt = #TX antennas;
%  nr = #RX antennas;
%  snr = SNR array (SNR per RX antenna);
%  numsnr = #SNR values in SNR array;
%  limit = if bit errors > limit then stop;
%  numphases = #phases of constellation; 2 for BPSK and 4 for QPSK
% returns mean bit error rate.

numbitspersymbol=log(numphases)/log(2);

snrlow=1;
ps=zeros(size(snr));

for i = 1:numphases,
   Constel(i)=exp(j*2*pi*(i-1)/numphases);
end;

% Main loop: various SNR values
Carray = zeros(nt,numphases^nt); % determine all possible constellations
for i = 1:nt,                    % on nt transmit antennas
   for k = 1:numphases^nt,
  Carray(i,k) = Constel(floor(mod((k-1)/(numphases^(i-1)),numphases))+1);
```

```
   end;
end;

outage=0;
expnr=0;

while (expnr<experiments) & (ps(numsnr)<limit),
   if (ps(snrlow)>limit-1)
      snrlow=snrlow+1;
   end;
   expnr=expnr+1;
   dat=floor(numphases*rand(nt,packetlength));  % make random data
   datest=0*dat;

   H=(randn(nr,nt)+j*randn(nr,nt))/sqrt(2);  % Make random channel matrix
   Darray = H*Carray;          % determine all possible receiver vectors
   N=(randn(nr,packetlength)+j*randn(nr,packetlength))/sqrt(2);

   % Transmitter
   x=exp(j*(dat*2*pi/numphases));
   rsig = H*x;

   % Start receiver
   for s = snrlow:numsnr,
      exparray(s) = expnr;
      alpha = nt/(10^((snr(s)+10*log10(numbitspersymbol))/10));
         % alpha of noise, corrected to obtain Eb/N0
      r=rsig+sqrt(alpha)*N;   % r = received array
      for i = 1:packetlength,
         ri=r(:,i);
         Rarray = ri(:,ones(numphases^nt,1));
         [Y,Imin] = min(sum((abs(Rarray - Darray)).^2,1));
            % find the max. likelihood estimate
         estimate(:,i) = Carray(:,Imin);
      end;

      datest=round(angle(estimate)/(2*pi/numphases)); %Estimate nt phases
      datest=rem(datest+numphases,numphases);

      errors=dat-datest;          % calculate BER
      errors=errors(errors~=0);
      errors=min(abs(errors),numphases-abs(errors));
      ps(s)=ps(s)+sum(errors);
   end; % snr loop
end; % experiments loop
for s = 1:numsnr,
   bertemp(s) = ps(s)/(exparray(s)*packetlength*nt*numbitspersymbol);
end;
ber = bertemp;
```