ITERATIVE DECODING OF SPACE-TIME-FREQUENCY BLOCK CODED MIMO CONCATENATED WITH LDPC CODES

by

Philip Rudolph Botha

Submitted in partial fulfilment of the requirements for the degree

Master of Engineering (Electronic Engineering)

in the

Department of Electrical, Electronic and Computer Engineering
Faculty of Engineering, Built Environment and Information Technology
UNIVERSITY OF PRETORIA

April 2013

SUMMARY

ITERATIVE DECODING OF SPACE-TIME-FREQUENCY BLOCK CODED MIMO CONCATENATED WITH LDPC CODES

by

Philip Rudolph Botha

Supervisor(s):	Prof B.T. Maharaj
Department:	Electrical, Electronic and Computer Engineering
University:	University of Pretoria
Degree:	Master of Engineering (Electronic Engineering)
Keywords:	MIMO, LDPC, Sphere Decoder, a priori, Zero Forcing, Minimum Mean Squared Error, Decision Feedback, Turbo, TAST.

In this dissertation the aim was to investigate the usage of algorithms found in computer science and apply suitable algorithms to the problem of decoding multiple-input multiple-output (MIMO) space-time-frequency block coded signals. It was found that the sphere decoder is a specific implementation of the A* tree search algorithm that is well known in computer science. Based on this knowledge, the sphere decoder was extended to include a priori information in the maximum a posteriori probability (MAP) joint decoding of the STFC block coded MIMO signals. The added complexity the addition of a priori information has on the sphere decoder was investigated and compared to the sphere decoder without a priori information. To mitigate the potential additional complexity several algorithms that determine the order in which the symbols are decoded were investigated. Three new algorithms incorporating a priori information were developed and compared with two existing algorithms. The existing algorithms compared against are sorting based on the norms of the channel matrix columns and the sorted QR decomposition.

Additionally, the zero forcing (ZF) and minimum mean squared error (MMSE) decoders

with and without decision feedback (DF) were also extended to include a priori information. The developed method of incorporating a priori information was compared to an existing algorithm based on receive vector translation (RVT). The limitation of RVT to quadrature phase shift keying (QPSK) and binary shift keying (BPSK) constellations was also shown in its derivation. The impact of the various symbol sorting algorithms initially developed for the sphere decoder on these decoders was also investigated. The developed a priori decoders operate in the log domain and as such accept a priori information in log-likelihood ratios (LLRs). In order to output LLRs to the forward error correcting (FEC) code, use of the max-log approximation, occasionally referred to as hard-to-soft decoding, was made.

In order to test the developed decoders, an iterative turbo decoder structure was used together with an LDPC decoder to decode threaded algebraic space-time (TAST) codes in a Rayleigh faded MIMO channel. Two variables that have the greatest impact on the performance of the turbo decoder were identified: the hard limit value of the LLRs to the LDPC decoder and the number of independently faded bits in the LDPC code.

OPSOMMING

ITERATIEWE DEKODERING VAN RUIMTE-TYD-FREKWENSIE BLOK GEKODEERDE VIVU SAAMGEVOEG MET LDPC KODES

deur

Philip Rudolph Botha

Supervisor(s):	Prof B.T. Maharaj
Departement:	Elektriese, Elektroniese en Rekenaar-Ingenieurswese
Universiteit:	Universiteit van Pretoria
Graad:	Magister in Ingenieurswese (Elektroniese Ingenieurswese)
Sleutel woorde:	VIVU, LDPC, Sfeerdekodeerder, aprioriese, nulforserende, minimum gemiddelde kwadraatfout, beslissingsterugvoer, turbo, gestruktureerde algebraïese ruimtetyd.

In hierdie dissertasie was die doel om die moontlike gebruik te ondersoek van algoritmes wat in rekenaarwetenskap gebruik word en van die algoritmes toe te pas om veelvuldige-inset-veelvuldige-uitset (VIVU) ruimte-tyd-frekwensie (RTF) gekodeerde seine te dekodeer. Die sfeer dekodeerder (SD) was geïdentifiseer as ’n spesifieke implementasie van die welbekende A* algoritme wat in rekenaarwetenskap gebruik word vir boom soektogte. Met hierdie kennis was die sfeer dekodeerder uitgebrei om aprioriese informasie te benut om maksimale aposterioriese waarskynlikheidsdekodering (MAW) van VIVU RTF seine te verrig. Die effek wat die insluiting van aprioriese informasie op die hoeveelheid berekenings wat deur die SD vereis word, is ondersoek. Drie algoritmes wat gebruik maak van aprioriese informasie was ontwikkel in ’n poging om die moontlike toename in die vereiste berekeninge teen te werk. Die drie algoritmes is vergelyk met twee bestaande algoritmes: herrangskikking gebaseer op die norme van die kolomme van die kanaal matriks en die gerangskikte QR matriksdekomposisie.

Verder, die nulforserende (NF) en die minimum gemiddelde kwadraatfout (MGKF) dekodeerders, met en sonder beslissingsterugvoer, was ook uitgebrei om gebruik te maak van aprioriese informasie. Die ontwikkelde metode om aprioriese informasie in te sluit by hierdie dekodeerders was vergelyk met ’n bestaande algoritme gebaseer op ontvangde vektortranslasie (OVT). In die afleiding van OVT, word die beperking van OVT tot kwadratuur faseskuifsleuteling en binêre faseskuifsleuteling konstellasies, bewys.

Die impak van die verskeie rangskikkingsalgoritmes, wat vir die SD ontwikkel is, op die doeltreffendheid van die NF en MMKF dekodeerders, was ook ondersoek. Die ontwikkelde aprioriese dekodeers funksioneer almal in die logaritmiese domein en maak as sodanig gebruik van log-waarskynlikheidverhoudings (LWV) as insette.

Ten einde die doeltreffendheid van die verskillende dekodeerders te toets, was ’n iteratiewe turbo dekodeerder struktuur, te same met ’n lae-digtheid pariteit check (LDPC) kode, van gebruik gemaak om gestruktureerde algebraïese ruimtetyd (SART) kodes iteratief te dekodeer in ’n Rayleigh vervaagde VIVU kanaal.

Twee veranderlikes wat die grootste impak op die werksverrigting van die turbo dekodeerder het, is geïdentifeer: die harde limiet waarde van die LMWs na die LDPC dekodeerder, en die aantal onafhanklike vervaagde busse in die LDPC kode.

I dedicate this work to my loving family and friends, for their unending support and encouragement during this study. Thanks.

ACKNOWLEDGEMENTS

I would like to thank the following people and institutions.

- First and foremost, my appreciation and gratitude for my parents and their never ending support and encouragement.
- I am deeply grateful to my supervisor, Prof. B.T. Maharaj for his support, guidance and patience during this study.
- I would also like to thank my engineering friends D.J. Louw, D.J. Basilio, T. Prinsloo, N. van der Neut, A. Swiatko and K. Dhuness for their insights and support during this study.
- Credit goes to the University of Pretoria and the Sentech Chair in Broadband Wireless Communication for the financial sponsorship of my Masters degree.

LIST OF ABBREVIATIONS

Abbreviation	Description
AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
BP	Belief Propagation
BPSK	Binary Phase Shift Keying
CIR	Channel Impulse Response
CSE	Channel State Estimation
CSI	Channel State Information
DF	Decision Feedback
EXIT	Extrinsic information Transfer
FEC	Forward Error Correction
FFT	Fast Fourier Transform
GSM	Global System for Mobile
HNN	Hopfield Neural Network
IIR	Inifinite Impulse Response
ISI	Inter-Symbol Interference
LDPC	Low Density Parity Check
LLR	Log-likelihood Ratio
LMS	Least Mean Square
LMSE	Least Means Square Estimation
LOS	Line of Sight
MAP	Maximum apriori Probability
MIMO	Multiple-input Multiple-output
ML	Maximum Likelihood
MMSE	Minimum Mean Squared Error
MRC	Maximum Ratio Combining
MRT	Maximum Ratio Transmission
OFDM	Orthogonal Frequency Division Multiplexing
pdf	Probability density Function
PDP	Power Delay Profile

Abbreviation	Description
PEP	Pairwise Error Probability
PSD	Power Spectral Density
QPSK	Quadrature Phase Shift Keying
RMS	Root Mean Squared
RVT	Receive Vector Translation
SD	Sphere Decoder
SISO	Single Input Single Output
SQRD	Sorted QR Decomposition
STF	Space Time Frequency
STFC	Space Time Frequency Code
TAST	Threaded Algebraic Space Time Code
WSSUS	Wide-Sense Stationary Uncorrelated Scattering
ZF	Zero Forcing

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

Multiple-input multiple-output (MIMO) systems potentially offer a significant increase in the capacity and reliability of wireless systems as compared to traditional single-input single-output (SISO) systems [1]. Special codes that allow MIMO systems to make full use of the available capacity and diversity in the MIMO channel, called space time frequency codes (STFC), have been developed [2].

Just as forward error correction (FEC) codes have enabled SISO systems to achieve capacity at the expense of increased computational complexity, so have STFCs done the same for MIMO systems. Used together, they form a serially concatenated code, with the FEC code used as the outer code and the STFC used as the inner code [3]. Concatenated codes can be iteratively decoded to yield improved performance [4, 5], and is the basis for Turbo codes [6]. It is therefore expected that iterative decoding of the serially FEC concatenated STFCs will yield a similar improvement. To perform iterative decoding with the STFC, the decoder for the STFC must be able to accept a priori information.

In this dissertation well known algorithms such as zero forcing (ZF), minimum mean squared error (MMSE) and sphere decoder (SD) that have previously been developed to decode the MIMO STFCs, are described.

The decoders are then modified to include a priori information such that they may be employed as part of an iterative decoder structure for the iterative decoding of serially FEC concatenated STFCs. Low density parity check (LDPC) codes are used as the FEC codes

and STFCs based on threaded algebraic space time (TAST) codes are used to test the iterative decoders.

1.1 BACKGROUND

The background of some of the key parts in this dissertation is discussed in this section. The discussion includes MIMO STFCs, MIMO STFC decoders and LDPC codes.

1.1.1 MIMO and STF codes

MIMO systems potentially offer two main advantages over a traditional SISO system: increased capacity and increased communication reliability. In a MIMO system the theoretical channel capacity scales linearly with the minimum of the number of transmit antennas or receive antennas [1]. The ability of the MIMO system to improve reliability in fading environments, scales with the product of the number of transmit and receive antennas [1, 7].

Various coding schemes have been proposed that enable a MIMO system to make use of the available diversity and enhanced capacity that a MIMO channel provides. The first class of codes, the Alamouti [8] space time code only made use of diversity and does not make full use of the additional capacity of the MIMO channel. The Alamouti code is a specific type of orthogonal code, and Tarokh et al. [9] generalise the construction of orthogonal codes by means of orthogonal design.

The codes based on orthogonal design achieve a lower capacity than that of a SISO channel, though with higher reliability. For three or four transmit antennas, there exist orthogonal codes that achieve a capacity of three quarters that of a SISO system. For instances where there are more than four antennas these codes achieve only half the capacity of a SISO system [9, 10]. Lo [10] presents a technique called *maximal ratio transmission* that achieves optimal communication that does, however, require channel state information (CSI) at the transmitter.

Another class of codes that achieve full diversity and capacity without requiring CSI at the

transmitter is the family of codes based on threading of component codes [11]. These codes use Diophantine numbers to separate the threads in order to achieve full rate and diversity. The component codes are generally codes based on algebraic constellation rotations [12]. Liu et al. [13] show that constellation rotation is not only applicable to MIMO systems but can be applied to orthogonal frequency division multiplexing (OFDM) systems as well as to improve OFDM's natural exploitation of the diversity in a multipath environment.

Thus, by appropriately placing the encoded channel symbols from the STFC on the available transmit antennas, time slots and frequency subcarriers, a code extracting space, time and frequency diversity can be constructed. The aim of the placement is to minimise the probability that all of the encoded symbols experience a deep fade.

1.1.2 MIMO STF decoders

The TAST codes used in this dissertation require joint decoding at the receiver for optimal performance. This is in contrast to the codes based on orthogonal design. Several well known algorithms exist and have been used to perform the joint decoding. The most elementary is the ZF decoder [1]. This decoder simply attempts to invert the MIMO channel and STFC to arrive at the transmitted symbols. This is similar to least means square estimation (LMSE). Several methods for performing the LMSE are known. The simplist is using the Penrose-Moore pseudo inverse, other methods involve triangular decompositions such as the LU, Cholesky and QR decompositions to using the singular value decomposition [14].

Triangular decompositions have the advantage of easily permitting the use of decision feedback (DF) that has significant performance benefits. As will be shown in this dissertation, triangular decompositions also permit the incorporation of a priori information in the decision step of the back substitution decoders.

In Wubben et al. [15] the use of QR decomposition to implement MMSE decoding is also performed. This allows the same advantage of decision feedback and a priori information to be employed at the decoder.

Triangular decompositions also have the added benefit of transforming the decoding problem into a tree structure. This allows for efficient searches for the optimal solution. This decoder, exploiting the tree structure, is generally known as the SD. This decoder was first proposed as a lattice based decoder for space time codes by Damen et al. [16]. Safar et al. [17] give a general description of the sphere decoder with a flow chart for easy implementation.

The metric the sphere decoder evaluates is monotonously increasing [18, p. 165] and is the foundation upon which the sphere decoder is based. The addition of a priori information, as shown in chapter three of this dissertation, causes the metric to no longer be monotonously increasing. The standard method of overcoming this *limit* is to employ a list based sphere decoder [19]. In the list based soft-input sphere decoder a list of all the successive candidate solutions are kept in a list. Once the sphere decoder has finished, entry in the list of vector with the lowest MAP metric when a priori information is included, is used as the decoded value. This results in suboptimal performance. Wang and Giannakis [20] present an alternative approach, henceforth referred to as receive vector translation (RVT). In chapter three this method is shown to be limited to binary phase shift keying (BPSK) and quadrature phase shift keying signal constellations. Studer et al. [21] describe a SD that performs MAP decoding incorporating a priori information by computing and adding bias terms to the distance metrics such that they are always positive and monotonously increasing.

1.1.3 LDPC codes

Gallager [22] first discovered LDPC codes in 1963 but were considered too computationally intensive to implement at the time. The codes were subsequently rediscovered by MacKay and Neal in 1996 [23] and later extended by them to encompass arbitrary Galois fields in Mackay and Davey [24]. The decoding of LDPC codes are generally performed using a message passing decoder such as Pearl's belief propagation (BP) [25]. A fast Fourier transform (FFT) based decoder was developed by Richardson and Urbanke [26]. This decoder reduces the complexity of the LDPC decoder. A more efficient FFT implementation and an extension to higher order Galois fields was then described by Mackay and Davey [24]. An implementation of the FFT decoding algorithm was done in the log domain by Song and

Cruz [27].

Several methods for constructing LDPC codes exist. Some of the best known methods are that of density evolution [26] and extrinsic information transfer (EXIT) chart as used by Bennatan and Burshtein [28]. There are also designs that permit efficient encoding via shift-registers such as Quasi-cyclic LDPC codes [27, 29]. Additionally many parity check matrixes and their performance for various LDPC codes can be found at *David MacKay's Gallager code resources* [30].

1.2 MOTIVATION AND OBJECTIVE OF THE DISSERTATION

The objective of this dissertation is to develop iterative decoders for MIMO communication systems to decode STF encoded MIMO transmissions. These iterative decoders use information obtained from an FEC code such as an LDPC code with which the MIMO STFC is concatenated. The iterative decoders should be able to decode a signal with an arbitrary signal constellation.

The motivation behind the development of iterative decoders is to improve the performance of currently used non-iterative STFC decoders, such as ZF, MMSE and the SD by including extrinsic information in the decoding process. The extrinsic information can be obtained from a priori information about the transmission or, as is the case in this dissertation, from the FEC code with which the STFC is concatenated.

Though longer length FEC and STFCs can be used to improve the performance of current decoders, these come at the expense of significant additional latency, specifically for block codes, and computation complexity. Since most MIMO STFC communications systems are concatenated with an FEC outer code, an iterative decoder would not require any major structural changes to the communication system in order to improve the performance of the system.

1.3 AUTHOR'S CONTRIBUTION

In this section some of the contributions made by the author in this dissertation are presented. The contributions include a new representation of the triply selective MIMO fading channel presented by Xiao et al. [31], an investigation in the diversity achieved by an LDPC code, iterative decoding of MIMO STF systems and the effect of symbol order on the performance of the iterative decoders.

1.3.1 Alternative representation of a triply selective MIMO fading channel

An alternative channel implementation model was derived in chapter two in order to improve upon the model proposed by Xiao et al. [31]. This model offers some simulation benefits such as extensions to arbitrary fading distributions such as Nakagami and Rician fading. It also potentially offers reduced computational requirements as compared to the triply selective fading channel developed by Xiao et al. [31]. The proposed model is shown mathematically to be capable of fully modelling the channel by Xiao et al. [31].

1.3.2 Diversity achieved by an LDPC code

Louw [32] alluded that LDPC codes by themselves can extract diversity, though at the cost of decreased throughput and increased latency. LDPC codes typically require the reception of the entire encoded block before decoding can commence, in contrast to convolutional codes. This introduces latency proportional to the length of the encoded block. Furthermore, FEC functions by introducing redundant information, thus decreasing the rate at which information is transmitted. In chapter three the performance of LDPC codes in Rayleigh fading for varying orders of diversity is shown thereby demonstrating the ability of LDPC codes to extract diversity.

1.3.3 Iterative decoding of MIMO STF systems

Iterative decoding of MIMO STFC is shown for the ZF and MMSE decoders incorporating DF. The derivation of the RVT method described, but not derived, by Wang and Giannakis [20] is also given. The derivation of the RVT method also highlights the limitations of the method, that being limited to signal constellation having a linear mapping to the bits associated with the channel symbols.

In chapter three a log-domain based sphere decoder incorporating a priori information without excessively increasing the computations required is described. The implementation of Safar et al. [17] is used as a basis. The complexity increase due to the addition of a priori information on the sphere decoder was also analysed. It was found that, with proper sorting of the symbol decoding order, addition of a priori information has a negligible impact on the computations required by the decoder. In certain cases there is a decrease in the required computations.

Lastly, the energy function of the Hopfield neural network (HNN) was used as a decoder by Hopfield and Tank [33]. This was extended by Myburgh [34] to the decoding of very long multipath channels. In this dissertation it is shown how the HNN may be used to decode the MIMO STFC.

1.3.4 Symbol sorting

The impact of changing the decoding order of the symbols was also investigated. It was shown that the ZF-DF algorithm can significantly benefit from a symbol order based on the a priori information.

The ability of symbol order sorting in reducing the complexity of the iterative sphere decoder was also investigated. To this end a modification of the sorted QR decomposition (SQRD) was done so as to incorporate a priori information. It is shown in chapter six that this significantly reduces the computations required by the iterative sphere decoder.

1.4 PUBLICATIONS

During the course of this study various articles were co-authored based on the research presented in this dissertation.

1.4.1 Conference articles

The conference papers published or submitted for review are:

1. A paper titled: "A reduced complexity soft-input soft-output MIMO detector combining a sphere decoder with a Hopfield neural network" authored by D.J. Louw, P.R. Botha and B.T. Maharaj was presented at the IEEE Wireless Communications and Networking Conference (WCNC) 2010 held in Sydney, Australia in April 2010.

2. A paper titled "Achievable diversity limits in a quantized MIMO-OFDM linear precoded system" authored by P.R. Botha, D.J. Louw and B.T. Maharaj was presented at the International Symposium on Wireless and Pervasive Computing (ISWPC) 2010 held in Modena, Italy in May 2010.

3. A paper titled "A low complexity soft-input soft-output MIMO detector which combines a sphere decoder with a Hopfield network" authored by D.J. Louw, P.R. Botha and B.T. Maharaj was presented at the Mediterranean Electrotechnical Conference (MELECON) 2010 held in Valletta, Malta in April 2010. The paper came joint second in the region eight IEEE student paper competition.

4. A paper titled "Turbo STFC decoding with the zero forcing decoder" authored by P.R. Botha and B.T. Maharaj was presented at IEEE Africon 2011 held in Livingstone, Zambia in September 2011.

5. A paper titled "Log domain turbo STFC decoding with the zero forcing and MMSE decoders" authored by P.R. Botha and B.T. Maharaj was presented at the Southern Africa Telecommunication Networks and Applications Conference (SATNAC) 2011 held in East-London, South Africa in September 2011.

6. A paper titled "Iterative zero-forcing MIMO decoder with symbol sorting" was submitted for review to the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC) to be held in London, United Kingdom in September 2013.

1.4.2 Journal articles

As part of this dissertation the following paper was submitted for review:

1. "Log-domain iterative sphere decoder with symbol sorting" authored by P.R. Botha and B.T. Maharaj to the European Association for Signal Processing (EURASIP) Journal on Wireless Communication and Networking.

1.5 OUTLINE OF THE DISSERTATION

This dissertation is comprised of eight chapters. In addition to chapter one, the next seven chapters are organised as follows:

1. Chapter two provides an introduction to the wireless MIMO fading channel. This is to understand the physical effect the channel has on the transmitted signal and how some of the important aspects are modelled. In this chapter various physical aspects of the wireless MIMO channel are discussed. These include noise, fading, Doppler, correlation and multipath. An alternative implementation of the triply selective fading MIMO channel of Xiao et al. [31] is also provided and compared.

2. In chapter three the maximum a priori probability (MAP) decoding criterion is derived and shown to be slightly different when the noise is modelled as complex Gaussian noise versus when it is modelled as real Gaussian noise when a priori information is included.

   Furthermore, various decoders that aim to achieve MAP decoding such as the classical ZF and MMSE decoders are described. A method for including a priori information in

these decoders is presented. An alternative method for including a priori in an arbitrary decoder based on receive vector translation proposed by Wang and Giannakis [20] is discussed and shown to be limited to signal constellations that have a linear mapping from channel symbols to their binary representation, such as the BPSK and QPSK signal constellations.

Additionally the classical sphere decoder is discussed and extended to include a priori information. It is theorised that the extension of the sphere decoder will increase the computations required by the decoder. To ameliorate the increase in complexity, various metrics determining the order in which the symbols are decoded are presented.

Decoding using the energy function of the HNN is also presented. It is shown that the MAP decoding corresponds to a constrained optimisation using the HNN.

Lastly, in the chapter a SD-HNN decoder developed in collaboration with Mr. D.J. Louw is presented. The aim of the decoder is to reduce the complexity of the soft-output sphere decoder.

3. FEC codes and their use in communication systems is discussed in chapter four. The chapter focuses on the linear block codes, specifically sparse graph codes such as LDPC codes. In the chapter the decoding of linear block codes using message passing is described by presenting the decoding operation as a marginalisation operation using factor graphs.

   Lastly the performance of the LDPC codes in additive white Gaussian noise (AWGN) and Rayleigh fading used in this dissertation is given. It is shown that LDPC codes can exploit diversity in a communication system.

4. Chapter five introduces STFCs used in MIMO systems. Their characteristics such as coding gain and diversity are discussed. Two main types of STFCs are presented: orthogonal codes and linear precodes. The focus is on linear precodes using algebraic rotations. It is shown how STFCs based on layer of components codes can achieve full diversity and throughput. Lastly, the performance of the STFCs used in this disserta-

tion is given.

5. In chapter six the overall communication system structure is described. The block diagrams of the transmitter and receiver are provided. The mathematical model of the received signal incorporating the STFC is described. Finally the effect that limiting of the log-likelihood ratios (LLR) from the output of the STFC decoder has on the performance of the iterative decoder is presented. Additionally the iterative reduction in the bit error rate (BER) is illustrated.

6. The overall results of the iterative decoder comparing the various STFC decoders is shown in chapter seven. The results show that the RVT method outperforms the method of including a priori information presented in this dissertation, except for the case of the ZF decoder employing DF. It is also shown that the ZF based decoders perform worse with an increase in the STFC length, whilst an improvement is expected. This is not the case for the MMSE decoders. It is shown that the iterative decoded MMSE-DF decoder performs close to the sphere decoder. It is also shown that the ZF decoder cannot fully exploit all the diversity in the channel and that LDPC codes provide diversity gain if sufficient in length.

   The performance of the iterative sphere decoder is also presented. It is shown that in contrast to the ZF decoder, it is able to make full use of the diversity in the channel. It is shown that the addition of a priori information adds significant complexity to the decoder. The sorting strategies proposed are shown to mitigate the added complexity and indeed show a decrease as compared to the classical sphere decoder.

   Lastly the performance of the SD-HNN decoder jointly developed with Mr. D.J. Louw [35] is presented. It is shown that by using iterative decoding the SD-HNN is able to match the performance of the classical sphere decoder.

7. Chapter eight ends this research work by providing a summary of the key conclusions obtained from this dissertation. The chapter finishes by proposing a few future research possibilities.

CHAPTER 2

CHANNEL

In this chapter a brief overview of the wireless channel is given. Channel effects such as AWGN, multipath fading, correlation and frequency selectivity, amongst others, are described and modelled mathematically. Two methods of modelling the classical SISO multipath channel propagation model, the stochastic covariance based method used by Xiao et al. [31] and an alternative direct method, incorporating some of these physical channel effects are described and then extended to model a MIMO channel. Lastly some of the advantages and disadvantages of the two models presented are shown.

The described channel models permit the simulation of signal transmissions of a MIMO system that can be used to test and compare the effectiveness of various FEC and STF coding and decoding schemes used in this dissertation.

2.1 PHYSICAL EFFECTS OF THE WIRELESS CHANNEL

In this section various physical effects encountered in the wireless channel, such as noise (of which thermal noise is the main noise source used in this dissertation), multipath fading, frequency selective fading and Doppler spread are discussed.

2.1.1 Noise

The sum of all the signals present in a communications system that are independent of the transmitted signal can be considered as noise. All communication systems experience signal corruption due to the effects of noise inherently present in all communication systems. The sources of these signals are diverse and may include effects such as the thermal vibration of atoms (often referred to as thermal noise), shot noise, flicker noise, burst noise and avalanche noise [1, 36]. In communication systems a distinction is generally made between noise and interference, distortion, jamming and other signals deliberately added from other transmitters.

In this dissertation only the effects of thermal noise will be considered. Thermal noise is generally modelled as having a flat power spectral density (PSD) and a Gaussian probability density function (pdf). Thermal noise is considered additive, thus it is modelled by adding a zero mean Gaussian distributed signal to the received signal at the receiver [37, 38, 39, 40]. Hence the name AWGN. The pdf of the AWGN signal is given by:

$$p\left(\eta\left(t\right)\right) = \frac{1}{\sqrt{2\pi\sigma_{\eta}^{2}}} \exp\left(-\frac{\eta^{2}\left(t\right)}{2\sigma_{\eta}^{2}}\right), \tag{2.1}$$

where $\eta(t)$ is the noise signal and σ_{η}^{2} is the power of the noise signal.

In systems using complex modulation schemes, such as QPSK, noise is added to both the in-phase and quadrature part of the signal. The noise in complex modulation schemes is generally modelled as a complex symmetric Gaussian signal given by:

$$\eta(t) = \eta_{\Re}(t) + j\eta_{\Im}, \tag{2.2}$$

where $\eta_{\Re}(t)$ and $\eta_{\Im}$ are real independent zero mean Gaussian processes with the same pdf of eqn. 2.1. The PSD of AWGN is considered flat for frequencies up to approximately 10^{12} Hz, which is the frequency range of interest for most wireless communications systems [37]. The amplitude of the double sided PSD of the AWGN is denoted as $N_0/2$, with:

$$N_0 = k_B T_e, \tag{2.3}$$

where N_0 is the amplitude of the single sided noise power spectral density, k_B is Boltzmann's constant and T_e is the environmental temperature in Kelvin [1].

In this dissertation the AWGN that is added to the signal will be scaled in order to achieve a particular E_b/N_0 ratio, where E_b is the energy contained in an information bit.

2.1.2 Fading

Fading is the random fluctuations in the attenuation of the transmitted signal as a function of time, geography and frequency and may be modelled as a random process. Fading as a result of geography is generally termed *large scale fading*. Fading due to destructive interference is typically referred to as *multipath fading*.

2.1.2.1 Large scale fading

The effects of large scale fading are effected by the presence of prominent terrain features between the transmitter and receiver. These features can include hills, buildings and forests. A receiver in such an environment is referred to as being "shadowed" by these features [38]. This result in the transmitted signal being severely attenuated.

Due to the size of the obstacles involved a receiver that is "shadowed" by one of these obstacles may be required to move large distances, several meters, in order to exit the "shadowed" area. This is in contrast to small scale fading, as is the case with multipath fading, where the receiver only has to walk a distance on the order of the wavelength λ to experience a higher signal amplitude.

2.1.2.2 Multipath fading

Wireless communication is achieved when a transmitted radio signal is successfully received. The transmitted signal can reach the receiver via a number of propagation paths. These can include the line of sight (LOS) path as well as indirect paths where the signal is reflected, diffracted or scattered towards the receiver by objects in the surrounding geographical area

[38, 39]. Numerous independent paths exist in the channel due to reflections, diffraction and scatterings.

Each of the propagation paths is associated with an amplitude, delay, direction of departure from the transmitter and direction of arrival at the receiver. Most importantly, they also have different phases [1]. Due to the differing phases, when these paths recombine at the receiver they may experience either constructive or destructive interference.

The multiple paths join up at the receiver where they can, depending on their phases, interfere constructively or destructively with each other. Movement of the transmitter, receiver or interfering objects causes changes in the phases of the paths and thus can change the interference at the receiver. Thus, when there is motion in the channel, the amplitude of the received signal changes with time. Movement on the order of $\lambda/2$ can change constructive into destructive interference but also destructive into constructive interference. At 2 GHz, the required motion is on the order of 10 cm [1], hence the reason this type of fading is commonly referred to as small scale fading.

In addition to the interference of the various paths with each other, each path also experiences its own changes in amplitude and phase. Figure 2.1 shows a simplistic model of the LOS and multipath components between a transmitter and receiver. In the figure the LOS path is shown as the direct path between the transmitter and receiver, with two additional paths. One of the additional paths is a reflection off of a building, the other off of a mountain. Notice that each reflection has its own multiple paths as well.

In this dissertation only small scale fading effects are considered and will henceforth simply be referred to as fading.

2.1.3 Types of small scale fading

The characteristic of the fading experienced by a signal in a channel is dependant on a variety of factors. These factors include, inter alia, the signal bandwidth, channel delay spread and Doppler spread. The various types of fading due to the various factors can be grouped

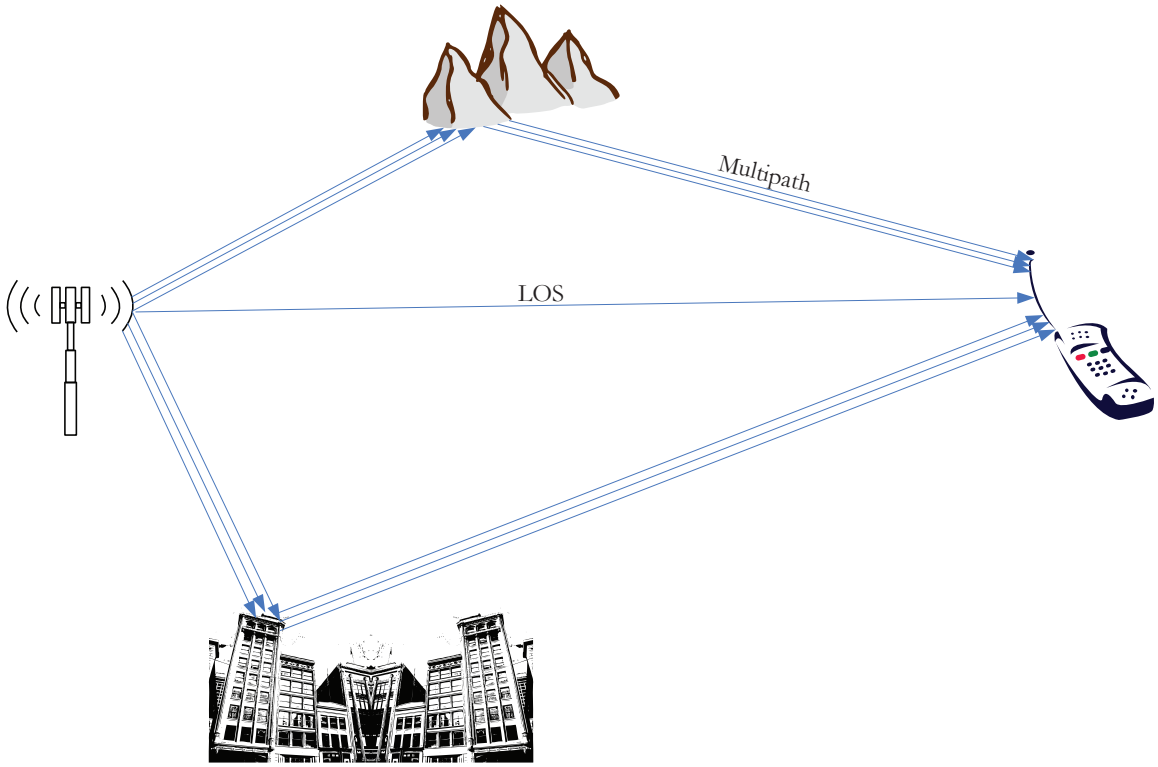

Figure 2.1: Simplistic representation of the LOS and multipath components.

in four distinct categories [37] namely flat fading, frequency selective fading, fast fading, slow fading (to which block fading can be added). Each of these types are summarised below.

2.1.3.1 Flat Fading

Where the channel has constant average gain and a linear phase response over a larger bandwidth than the bandwidth of the signal propagating through the channel, then the signal experience flat fading. The conditions that have to be met for flat fading are:

$$B_W \ll B_C \tag{2.4}$$

$$\frac{1}{B_W} \gg \tau_D, \tag{2.5}$$

where B_W is the bandwidth of the signal, B_C is the coherence bandwidth and τ_D is the root mean squared (RMS) delay spread of the channel.

In flat fading channels the spectral characteristics of the signal are preserved, and only the received power of the signal is altered. Since $\frac{1}{B_W} \ll \tau_D$, the flat fading channel can be modelled by a single discrete multipath tap [37, 38]. Thus the channel does not introduce inter-symbol interference.

2.1.3.2 Frequency selective fading

Frequency selective fading occurs when the spectral response of the channel is no longer flat over the bandwidth of interest. In frequency selective channels multiple versions of the transmitted signal, that are delayed on the order of multiples of the symbol period T_{sym}, are received at the receiver. This results in inter-symbol interference (ISI).

Frequency selective fading is characterised by signal propagating through a channel with a gain that is fairly constant [39]. The phase response of the channel is only linear for a portion of the bandwidth of the signal. A channel is frequency selective when:

$$\frac{1}{B_W} < \tau_D \tag{2.6}$$

$$B_W > B_C. \tag{2.7}$$

Additionally, frequency components separated by more than B_C can be expected to undergo uncorrelated fading [38]. These independent frequency components can be used to achieve diversity [41].

2.1.3.3 Fast Fading

In the case where the channel impulse response (CIR) undergoes significant changes during a symbol period, the fading is said to be fast fading. The conditions characterising fast fading are [39]:

$$\frac{1}{B_W} > T_C \tag{2.8}$$

$$B_W < B_D, \tag{2.9}$$

where T_C is the coherence time of the channel and B_D is the maximum Doppler frequency (see 2.1.4.1 for the discussion of Doppler).

Doppler spread gives rise to frequency dispersion of the signal in the frequency domain and in turn causes time distortions on the transmitted sequence. This can cause inter-carrier interference in OFDM systems [42] and introduce error floors [43].

2.1.3.4 Slow Fading

In slow fading the CIR changes slowly over time, much slower than the transmitted signal. This type of channel can be assumed to be static over a certain period of time, larger than $\frac{1}{B_W}$ [39]. The conditions for slow fading are [39]:

$$\frac{1}{B_W} \ll T_C \tag{2.10}$$

$$B_W \gg B_D. \tag{2.11}$$

2.1.3.5 Block Fading

In block fading the channel is assumed to be quasi constant over a transmission block, hence the name block fading [44, sec. 4.3]. The assumption of block fading reduces the complexity of the simulated channel since the fading parameters are updated on a block basis instead of a symbol basis. This can occur if the coherence time (see section 2.1.4.4) is larger than the transmitted block.

2.1.4 Doppler

The Doppler effect is the change in frequency for a wave for an observer when there is a relative motion between the observer and the source of the wave. In this section the frequency shift due to the relative motion between source and observer is described, the Doppler PSD, as well as Doppler spreading of the received frequency and the resultant coherence bandwidth.

2.1.4.1 Doppler frequency shift

The relative motion between the receivers, interfering objects and transmitters gives rise to a frequency modulation of the signal. This frequency modulation is due the Doppler effect. The severity of the shift in carrier frequency caused by the Doppler effect is proportional to the relative velocities between the objects.

The shift in carrier frequency due to a moving transmitter is given by:

$$f_{d,i}(t) = -\frac{v_{t,r}(t)}{\lambda_0}, \tag{2.12}$$

where $f_{d,i}$ is the frequency shift for the i^{th} signal path, $v_{t,r}$ is the relative velocity of the transmitter with respect to the receiver and λ_0 the original wavelength of the transmitted signal. The equation only holds for when $\|v_{t,r}\| \lll c$, where c is the speed of light [45]. For a stationary transmitter eqn. 2.12 becomes:

$$f_{d,i}(t) = -\frac{v_r(t)}{\lambda_0}\cos\left(\theta_{A,i}(t)\right), \tag{2.13}$$

where $v_r(t)$ is the speed of the receiver and $\theta_{A,i}(t)$ is the angle of arrival at the receiver of the i^{th} path. The result of time varying Doppler shift is termed Doppler spread, see section 2.1.4.3. In eqns. 2.12 and 2.13 the effects of the motion of the scattering and reflecting objects have been ignored, which may be done if the speeds of the scatters and reflectors are small compared to that of the receiver and transmitter and will henceforth also be ignored in this dissertation.

2.1.4.2 Doppler Power Spectral Density

In the multipath fading environment (see section 2.1.2.2) the signal experiences fading to the interference of various paths with differing phases. These paths can have differing angles of arrival, departure and reflection. In the case where the scatterers, transmitters or receivers are in motion, there will be an associated Doppler frequency shift, as described in section 2.1.4.1.

Due to the differing angles of arrival and velocities in the channel, amongst other factors, the Doppler shifts of each path will vary (see eqn. 2.13). The sum of these Doppler shifts

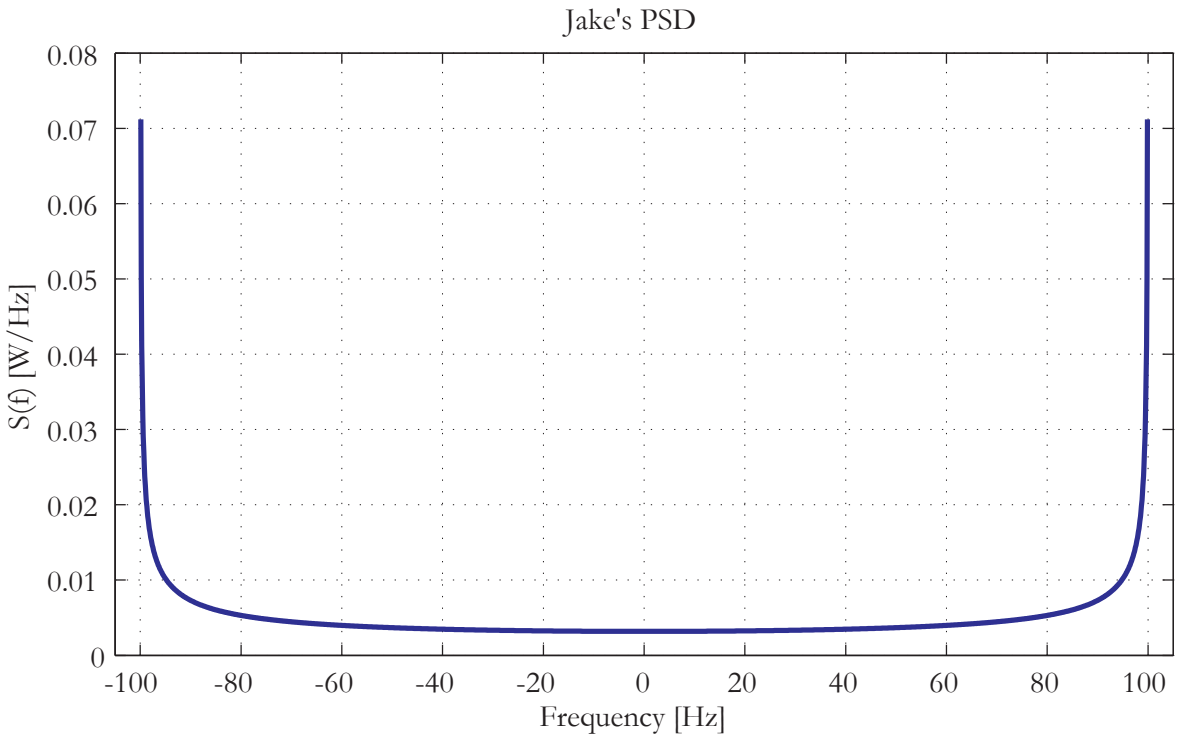

Figure 2.2: Plot of Jake's PSD (eqn. 2.14) for an f_D of a 100 Hz and $\sigma_0^2 = 1$.

gives rise to the Doppler PSD [46]. Three commonly used PSDs are used [46, 47]. The first being the ubiquitous Jake's spectra, the remaining two are Gaussian spectra with different parameters.

Jake's PSD is defined as [46, 48]:

$$S_D(f) = \begin{cases} \frac{\sigma_0^2}{\pi f_D \sqrt{1-\left(\frac{f}{f_D}\right)^2}}, & |f| \leq f_D, \\ 0, & |f| > f_D, \end{cases} \tag{2.14}$$

Figure 2.2 shows a plot of eqn. 2.14 for a maximum Doppler frequency of 100 Hz, and $\sigma_0 = 1$.

The Gaussian PSD is given as [46, eqn. 3.11]:

$$S_D(f) = \frac{\sigma_0^2}{f_c}\sqrt{\frac{\ln 2}{\pi}} e^{-\ln 2\left(\frac{f}{f_c}\right)^2}, \tag{2.15}$$

where f_c is the 3dB cut-off frequency. Figure 2.3 shows a plot of the Gaussian PSD for an

$B_D = 100$ Hz and $\sigma_0 = 1$. Notice that the majority of the Jake's PSD is situated close to B_D, whereas for the Gaussian PSD it is centred at zero.

In the COST 207 models [47], the Gaussian PSD is used to define two PSDs, referred to as Gauss I and Gauss II [47, 46]. Gauss I PSD being defined as:

$$S_D(f) = \frac{50}{\sqrt{2\pi}3B_D} \exp\left[-\frac{(f+0.8B_D)^2}{2(0.05B_D)^2}\right] + \frac{50}{10\sqrt{2\pi}3B_D} \exp\left[-\frac{(f-0.4B_D)^2}{2(0.1B_D)^2}\right], \tag{2.16}$$

and Gauss II PSD being defined as:

$$S_D(f) = \frac{10^{1.5}}{\sqrt{2\pi}\left(\sqrt{10}+0.15B_D\right)} \exp\left[-\frac{(f-0.7B_D)^2}{2(0.1B_D)^2}\right] + \frac{1}{\sqrt{2\pi}\left(\sqrt{10}+0.15B_D\right)} \exp\left[-\frac{(f+0.4B_D)^2}{2(0.15B_D)^2}\right]. \tag{2.17}$$

Figures 2.4 and 2.5 are plots of the Gauss I and Gauss II PSDs represented by eqns. 2.16 and 2.17 respectively. Jake's PSD is typically used to model the first few multipath components, followed by the Gauss I PSD and finally the Gauss II PSD [46].

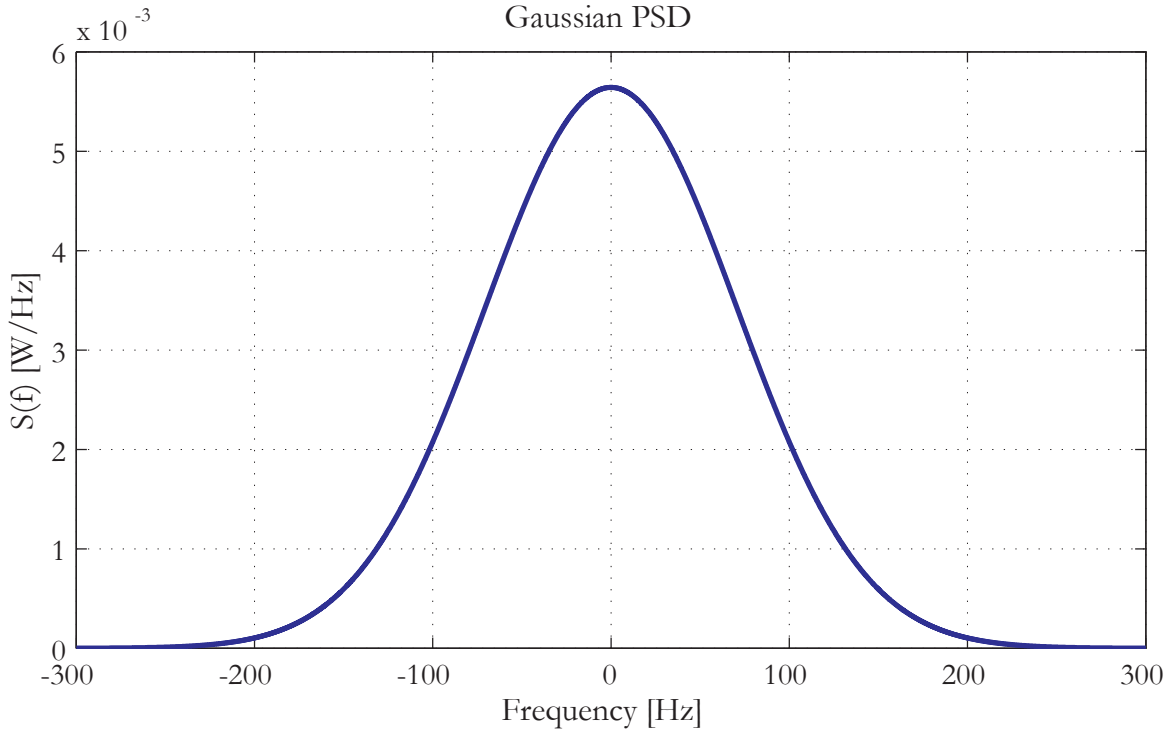

Figure 2.3: Plot of the Gaussian PSD (eqn. 2.15) for a B_D of a 100 Hz, $f_c = \sqrt{\ln 2} B_D$ and $\sigma_0^2 = 1$.

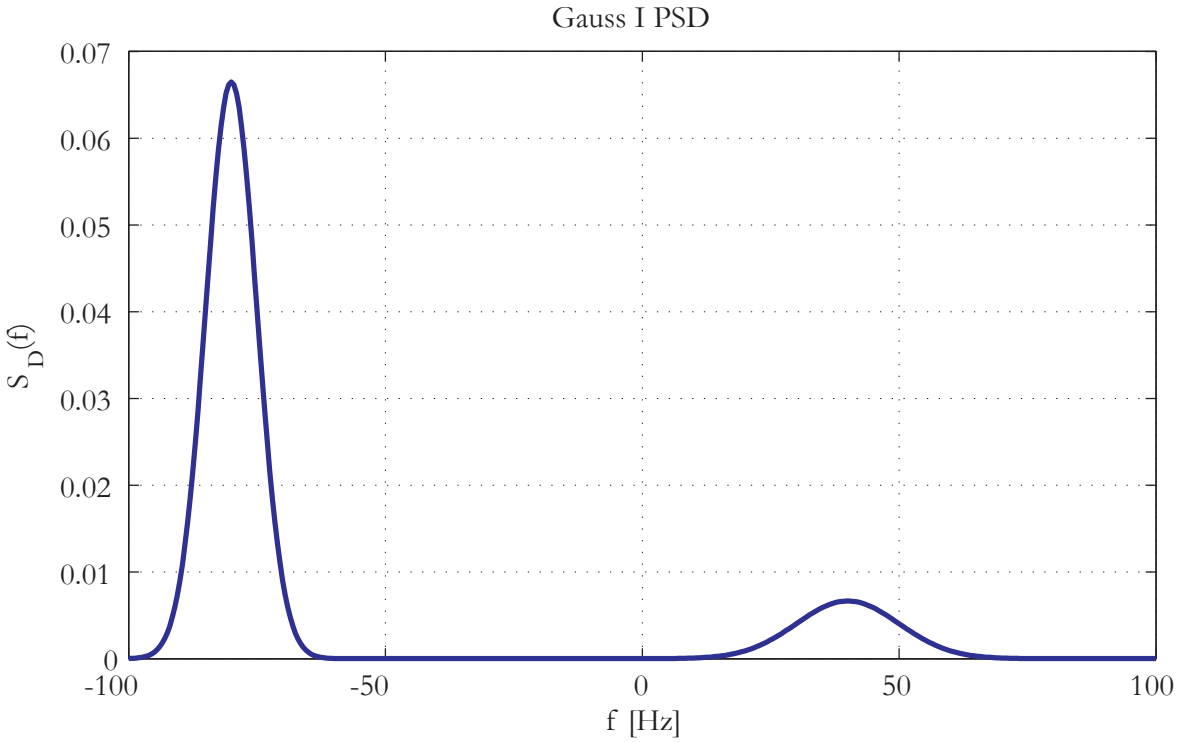

Figure 2.4: Plot of the Gauss I PSD (eqn. 2.16) for a B_D of a 100 Hz.

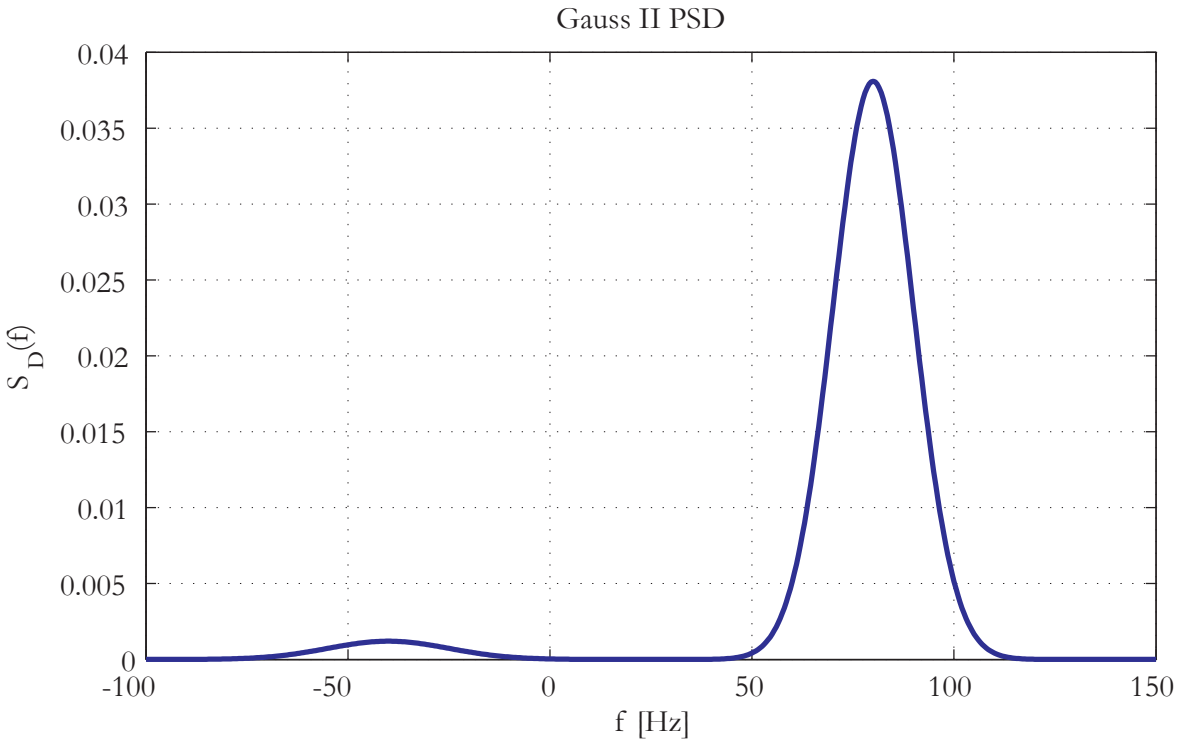

Figure 2.5: Plot of the Gauss II PSD (eqn. 2.17) for a B_D of a 100 Hz.

2.1.4.3 Doppler Spread

Doppler spread is a measure of the spectral broadening that occurs due to the time-variant nature of the channel [37, 38, 39]. It is a measure of the single sided spectral width of a received single given that the input was a single tone carrier signal [39, 46].

Pätzold [46] defines the Doppler spread in terms of the second moment of the Doppler PSD:

$$B_D = \sqrt{\frac{\int_{-\infty}^{\infty} (f - B_{D_{ave}})^2 S_D(f) df}{\int_{-\infty}^{\infty} S_D(f) df}}, \tag{2.18}$$

where $B_{D_{ave}}$ is the average Doppler frequency shift and is given by:

$$B_{D_{ave}} = \frac{\int_{-\infty}^{\infty} f S_D(f) df}{\int_{-\infty}^{\infty} S_D(f) df}, \tag{2.19}$$

and $S_D(f)$ is the Doppler PSD. In the case where the PSD is symmetrical, $B_{D_{ave}} = 0$ and on average there is not a frequency offset.

2.1.4.4 Coherence time

The coherence time is a measure of the time interval that a channel can be considered time invariant. It is given as being inversely proportional to the maximum Doppler spread B_D [39]:

$$T_C = \frac{\zeta}{B_D}, \tag{2.20}$$

where ζ is a constant and is dependant on the amplitude correlation and typically varies from 1 to $9/(16\pi)$ for correlation values between 0.9 and 0.5.

Pätzold [46] defines the coherence time as the time difference that fulfils the following condition:

$$\left|r_{\mu\mu}(T_C)\right| = \frac{1}{2}\left|r_{\mu\mu}(0)\right|, \tag{2.21}$$

where $r_{\mu\mu}(\tau')$ is the time correlation function defined as:

$$r_{\mu\mu}(\tau') = \int_{-\infty}^{\infty} S_D(f) e^{j2\pi f\tau'} df \tag{2.22}$$

$$= \mathcal{F}^{-1}\{S_D(f)\}, \tag{2.23}$$

where $S_D(f)$ is the PSD of the Doppler Spread and $\mathcal{F}^{-1}$ is the inverse Fourier transform.

In block fading the coherence time is assumed to be larger than the length of the block that is undergoing fading.

2.2 MULTIPATH CHANNEL PROPAGATION MODEL

In fig. 2.1 an example of a multipath channel is illustrated. In this section a model based on combining the multiple distinct paths in a channel is described. The COST 207 [47] models for example combine up to twelve discrete paths, with different delays, attenuation and Doppler spectra, to describe a channel.

In a multipath channel, the received signal $r(t)$ is given, in the absence of noise, by the convolution of the transmitted signal $s(t)$ with the channel $h(t,\tau)$ [39]:

$$r(t) = s(t) \circledast h(t,\tau), \tag{2.24}$$

where $\circledast$ denotes the continuous-time convolution function. The time variant nature of the channel originates from the relative motion between the transmitter and receiver. In eqn. 2.24 τ represents, for fixed t, the channel multipath delay.

The passband channel impulse response can be given in terms of the baseband signal [39, 40]:

$$h(t,\tau) = \Re\left\{h_b(t,\tau)\exp\left[j2\pi f_c t\right]\right\}, \tag{2.25}$$

where $h_b(t,\tau)$ is the equivalent time-variant baseband channel impulse response and f_c is the carrier frequency. In the case of a discrete channel with $L+1$ discrete multipath components, such as will be assumed in this dissertation, the baseband channel response can be written as

[39]:

$$h_b(t,\tau) = \sum_{i=0}^{L} \beta_i(t,\tau)\delta(\tau - \tau_i(t)) \exp\left[\jmath(2\pi f_c \tau_i(t) + \theta_i(t,\tau))\right], \tag{2.26}$$

where $\beta_i(t,\tau)$ is the instantaneous amplitude and $\tau_i(t)$ the time delay of the i^{th} multipath component of the channel. The instantaneous phase shift of the i^{th} multipath component due to its instantaneous time delay $\tau_i(t)$ is represented by the $2\pi f_c(\tau_i(t)$ term in the exponent of eqn. 2.26 while other phase shift alterations in the $\theta_i(t,\tau)$ term. Staphorst [37] lumped these phase shift components together as:

$$\phi_i(t,\tau) = 2\pi f_c \tau_i(t) + \theta_i(t,\tau). \tag{2.27}$$

Assuming that the time delay $\tau_i(t)$ is invariant with respect to time, eqn. 2.26 can be simplified to:

$$h_b(t,\tau) = \sum_{i=0}^{L} \beta_i(t)\delta(\tau - \tau_i) \exp\left[\jmath\phi_i(t)\right], \tag{2.28}$$

$\beta_i(t)$ and $\phi_i(t)$ are the time varying amplitude and phase characteristics of the i^{th} multipath component with fixed time delay of τ_i.

Further assuming that the channel is wide sense stationary [40] over a small time interval, eqn. 2.28 can be further simplified to [37, 39]:

$$h_b(\tau) = \sum_{i=0}^{L} E\left[\beta_i(t)\right] \delta(\tau - \tau_i) \exp(\jmath E\left[\phi_i(t)\right]), \tag{2.29}$$

where $E[\cdot]$ is the expectation of a random variable. The normalised instantaneous amplitude and phase of the fading $\alpha_i(t)$ can be given as:

$$|\alpha_i(t)| = \frac{\beta_i(t)}{E[\beta_i]} \tag{2.30}$$

$$\angle\alpha_i(t) = \phi_i(t). \tag{2.31}$$

2.2.1 Power Delay Profiles

One common manner of defining a multipath channel, is by using Power Delay Profiles (PDP). PDPs are a measure of the power in a channel as a function of the delay τ. In the

COST 207 models [47], a study for the Global System for Mobile Communications (GSM), various channel soundings were used to obtain PDPs of different terrains to be used in simulations. The PDP of a time varying discrete multipath channel is given by [38, 39]:

$$P(t,\tau) = |h_b(t,\tau)|^2 = \sum_{i=0}^{L} |\beta_i(t)|^2 \delta(\tau - \tau_i). \tag{2.32}$$

Taking the expectation of eqn. 2.32, with respect to time, yields the following time invariant version:

$$P(\tau) = |h_b(\tau)|^2 = \sum_{i=0}^{L} P(\tau_i)\delta(\tau - \tau_i)|^2, \tag{2.33}$$

where $P(\tau_i)$ is the power in the i^{th} tap and from eqns. 2.28 and 2.32 is given as:

$$P(\tau_i) = E[\beta_i^2]. \tag{2.34}$$

In MIMO channels, each path between a transmit and receive antenna can be modelled as a SISO channel and in SISO channels the PDP, together with the Doppler frequency and Doppler PSD, are the major parameters in defining the channel. An example of a PDP used in modelling for cellular channels in an urban environment is shown fig. 2.6.

Two important characteristics of a PDP are the *delay spread* and the *coherence bandwidth* of the PDP.

2.2.1.1 Delay Spread

The delay spread τ_D of a channel is a measure of how much multipath components there are in the channel. It is also used to calculate the coherence bandwidth of the channel, a quantity that is important in determining the type of fading.

There are three commonly used metric used to characterise the delay spread of the channel, the simplest being the time difference between the first received multipath component and the last received multipath component. Sklar [38] and Rappaport [39] refer to the latter as the maximum excess delay:

$$\tau_{max} = \tau_L - \tau_0, \tag{2.35}$$

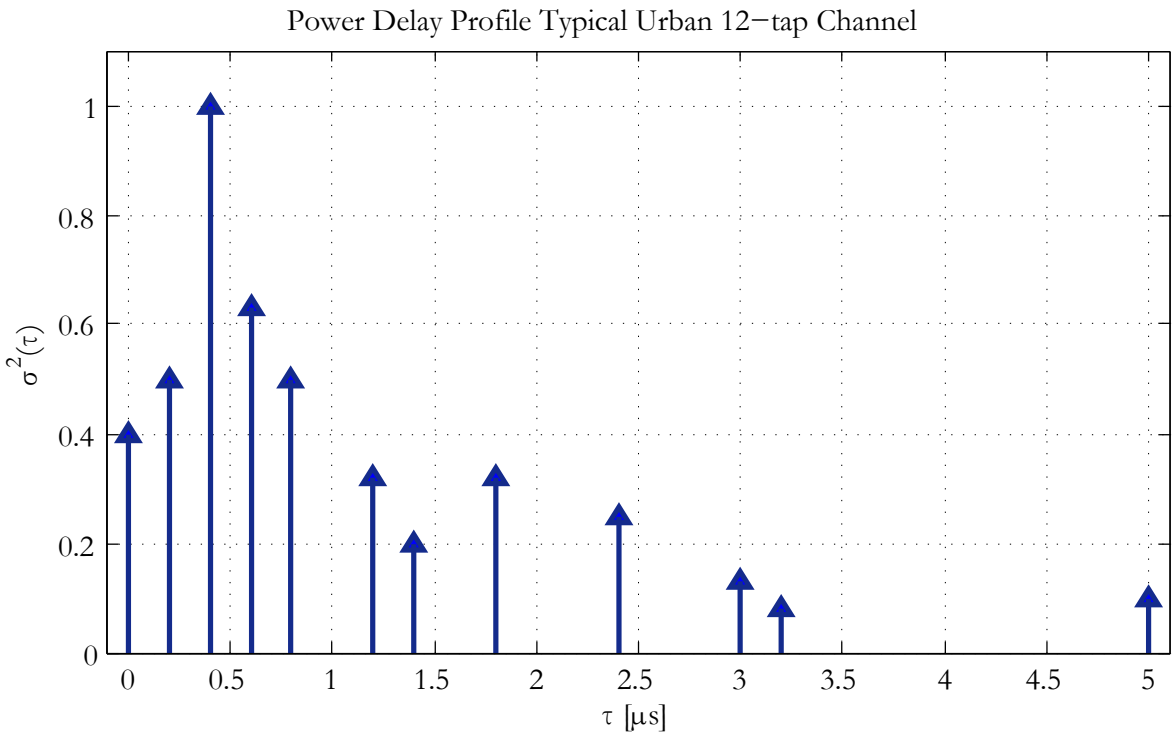

Figure 2.6: Power Delay Profile of the COST 207 Typical Urban Twelve Tap Channel.

where τ_L is assumed to be the last multipath component whose expected power is above an arbitrary cut-off point P_{drop} and τ_0 is the time delay of the first multipath component. Typical values for P_{drop} is -30 dB.

The second metric is that of calculating the first moment of the PDP $P(\tau)$ [46]:

$$\tau_{ave} = \frac{\int_{-\infty}^{\infty} \tau P(\tau) d\tau}{\int_{-\infty}^{\infty} P(\tau) d\tau}, \tag{2.36}$$

which in the case of a discrete channel simply becomes [38, 39]:

$$\tau_{ave} = \frac{\sum_{i=0}^{L} \tau_i P(\tau_i)}{\sum_{i=0}^{L} P(\tau_i)}. \tag{2.37}$$

The last measure is the RMS delay spread which is taken as the second moment of the PDP

$P(\tau)$ [46]:

$$\tau_{RMS} = \sqrt{\frac{\int_{-\infty}^{\infty}(\tau - \tau_{ave})^2 P(\tau d\tau}{\int_{-\infty}^{\infty} P(\tau) d\tau}}, \tag{2.38}$$

which in the case of a discrete channel becomes [38, 39]:

$$\tau_{RMS} = \sqrt{\frac{\sum_{i=0}^{L}(\tau_i - \tau_{ave})^2 P(\tau_i)}{\sum_{i=0}^{L} P(\tau_i)}}. \tag{2.39}$$

The quantity in the denominators of eqns. 2.36, 2.37, 2.38 and 2.39 is generally taken to be equal to one. That is the PDP of the channel is normalised to have a power of one, so that it does not change the energy content of the received signal [31].

Channels with a delay spread

$$\tau_{rms} < 0.2 T_{sym}, \tag{2.40}$$

can generally be adequately modelled as a frequency non-selective channel as the majority of frequency components of the transmitted signal undergo the same variations [46].

2.2.1.2 Coherence Bandwidth

The coherence bandwidth of a signal B_C is a measure of the range of frequencies over which a channel can be considered flat. Thus, frequencies separated by more than B_C are expected to vary significantly in amplitude. B_C is inversely related to the time delay spread τ_D of the channel [38, 39]:

$$B_C = \frac{1}{\phi \tau_D}, \tag{2.41}$$

where ϕ is a constant dependent on the correlation between the two frequency components. For a correlation factor of 0.9 ϕ is often taken as equal to 50, and for a correlation factor of 0.5, is taken as equal to 5 [37].

Pätzold [46] takes the coherence bandwidth as the separation in frequency required to have a difference of 3dB, i.e. a ratio of a half in frequency correlation function. Thus, B_C is the

frequency separation required to fulfil the following [46]:

$$\left|r_{\tau\tau}(B_C)\right| = \frac{1}{2}\left|r_{\tau\tau}(0)\right|, \tag{2.42}$$

where the frequency correlation function $r_{\tau\tau}(f')$ is defined as:

$$r_{\tau\tau}(f') = \int_{-\infty}^{\infty} P(\tau)e^{-j2\pi f'\tau}d\tau \tag{2.43}$$

$$= \mathcal{F}\{P(\tau)\}, \tag{2.44}$$

where $\mathcal{F}$ denotes the Fourier transform. The ratio of B_C to signal bandwidth of the transmitted signal determines whether the signal will experience frequency-selective fading or flat fading, as described in section 2.1.3.

2.3 SISO CHANNEL

MIMO channel models often assume that the individual paths between the transmit and receive antennas are independent SISO channels before introducing the required correlation between antennas [1]. In this section, the component SISO channel model of the MIMO channel model is described.

Xiao et al. [31], propose a triply selective fading MIMO channel that can accommodate the inter tap correlations that occur due to fractionally spaced channel taps. The model is a generalisation of the SISO wide sense stationary uncorrelated scattering (WSSUS) multipath channel, proposed by Hoeher [49], to incorporate the spacial selectivity of a MIMO channel. This was done using the Kronecker model [50, 51].

The triply selective fading model of Xiao et al. [31] was analysed by Mietzner and Hoeher [52], and they came to the conclusion that there are several shortcomings in said model. However, in a subsequent paper Mietzner et al. [53] conclude that if the assumptions made by Xiao et al. [31] are assumed to hold, then the model is valid. It was also stated that these assumptions are sufficiently accurate for a wide range of practical scenarios.

A modification is proposed to the model of the WSSUS multipath channel proposed by Hoeher [49]. This change, although not addressing the shortcomings identified by Mietzner [52],

can result in a substantial reduction in computing complexity and in an increased flexibility of the model. In section 2.4 the proposed WSSUS channel model is then extended to MIMO in a similar manner to that of Xiao et al. [31] and will be called the *direct method*.

From the description of the two methods, it will be evident that the direct method may offer a reduction in computational complexity, by reducing the amount of independent Rayleigh flat fading processes required as shown in section 2.6. Furthermore the increase in the flexibility of direct method over that of the covariance method, which allows for the simulation of channels having taps with different Doppler PSDs will also be shown.

The method of Xiao et al. [31], which is referred to as the *covariance method*, and the proposed direct method are compared in simulation. These simulations will empirically show that they are equivalent. In the next section the covariance method is described for the SISO channel case followed by a description of the direct method for the SISO case.

Consider a SISO channel link between two antennas. Let ρ_T and ρ_R be the time invariant impulse response of the transmit and receive filters respectively. Furthermore, let $g(t,\tau)$ be the response of the physical channel between the two antennas at time t due to an impulse at time $t-\tau$. The overall impulse response of the channel $h(t,\tau)$ can then be given as:

$$h(t,\tau) = \rho_T \circledast g(t,\tau) \circledast \rho_R, \tag{2.45}$$

where $\circledast$ denotes the convolution operation. Let $s(t)$ represent the complex data symbol to be sent over the channel, $y(t)$ the received signal and $z(t)$ the filtered AWGN. The received signal can then be expressed as:

$$y(t) = s(t) \circledast h(t,\tau) + z(t), \tag{2.46}$$

where $z(t)$ is given by:

$$z(t) = \rho_R \circledast n(t), \tag{2.47}$$

with $n(t)$ being AWGN. As mentioned by Xiao et al. [31] if

$$R_{\rho_R\rho_R}(kT_s) = 0, \quad k \neq 0, \tag{2.48}$$

then the filtered AWGN $z(t)$ is still white sample to sample.

Assuming that at the receiver the sampling period (T_s) is equal to the symbol period (T_{sym}), then the received signal may be given as:

$$y[k] = \sum_{l=-\infty}^{\infty} s[k-l]h[k,l] + z[t], \tag{2.49}$$

where $y[k]$, $s[k]$, $z[k]$ and $h[k,l]$ are the sampled versions of the received signal, transmit symbol, channel impulse response and noise respectively. The sampled $h[k,l]$ is a non-causal time varying infinite impulse response (IIR) filter.

In practice, the length of the impulse response is truncated such that only ϵ of the energy is lost, where ϵ is an arbitrary number that is generally chosen as small as practical. The result of the truncation of the impulse response is a finite impulse response (FIR) filter with $l \in [-L_1; L_2]$, with L_1 and L_2 chosen such that $(L_1 + L_2)$ is minimised and ϵ is an acceptable quantity. In this dissertation it will also be assumed that the channel PDP is discrete and that the channel is of the following form:

$$g(t,\tau) = \sum_{I} \sigma_i \delta(t-\tau_i)\Phi_i(t), \tag{2.50}$$

where I is the number of discrete paths, σ_i^2 is the power of the i^{th} discrete path and $\Phi_i(t)$ is the zero mean instantaneous fading value for the i^{th} discrete path.

In this dissertation two methods are described to generate the sampled channel $h[k,l]$ for use in simulations. These two methods are the covariance method and the direct method, as mentioned previously.

2.3.1 Generating the channel, covariance method

The sampled channel $h[k,l]$ can be generated from its covariance matrix for the case of Rayleigh fading, as by Xiao et al. [31] and Hoeher [49]. Thus the name *covariance method*. The covariance matrix, of size $A \times A$, is given by:

$$\mathbf{C}_{ISI} = \mathbf{E}\left[(\mathbf{h} - \mathbf{E}[\mathbf{h}])(h - \mathbf{E}[\mathbf{h}])^{\mathcal{H}}\right], \tag{2.51}$$

where $(\cdot)^{\mathcal{H}}$ denotes the conjugate transpose or Hermitian operator. It is also assumed that $\mathbf{E}[\mathbf{h}] = 0$ since $g(t,\tau)$ is assumed to be a zero mean complex Gaussian process. The entries

of the covariance matrix, $g(t, \tau)$ being discrete, are given as [31]:

$$C_{ISI}(l_1, l_2) = \sum_{i=1}^{L} \sigma_i^2 R_{\rho_T \rho_R}(l_1 T_s - \tau_i) R^*_{\rho_T \rho_R}(l_2 T_s - \tau_i), \tag{2.52}$$

where $(\cdot)^*$ denotes the conjugate operator.

From the covariance matrix, the resultant channel coefficients can be generated as:

$$\mathbf{h}[k] = \mathbf{C}_{ISI}^{1/2} \cdot \Upsilon[k], \tag{2.53}$$

where $(\cdot)^{1/2}$ denotes the matrix square root function and $\Upsilon[k]$ is a column vector, length A, of uncorrelated complex Rayleigh flat fading.

2.3.2 Direct method of generating the channel

Instead of generating the channel coefficients by correlating a vector of uncorrelated zero mean complex Rayleigh flat fading, the coefficients can be generated by performing the convolution of:

$$h(t, \tau) = \rho_T \circledast \rho_R \circledast g(t, \tau) \tag{2.54}$$

and then sampling the resultant $h(t, \tau)$ to yield $h[k, l]$. The convolution of ρ_T and ρ_R is equivalent to the pulse shape $R_{\rho_T \rho_R}$. Assuming that the PDP is discrete, then the convolution of $g(t, \tau)$ and $R_{\rho_T \rho_R}$ will be the superposition of $IR_{\rho_T \rho_R}$ pulses scaled by σ_i and delayed by τ_i. Thus, the oversampled channel coefficients can be calculated as:

$$h[k, l] = \sum_{I} R_{\rho_R \rho_T}(lTs - \tau_l) g(kTs, \tau_i). \tag{2.55}$$

Since the pulse shape, path powers and path delays are constant, the summation in eqn. 2.55 can be written as the product of a constant matrix, size $A \times I$, and a column vector, length I, of fades:

$$\mathbf{h}[k] = \mathbf{C}_S \Phi[k], \tag{2.56}$$

where the entries of $\mathbf{C}_S$ are given by:

$$C_S(l, i) = \sigma_i R_{\rho_T \rho_R}(lT_s - \tau_i). \tag{2.57}$$

From a statistical point of view, the two methods are equivalent and indeed it is possible to show that the $\mathbf{C}_{ISI}$ can be calculated from $\mathbf{C}_S$ as follows:

$$\mathbf{C}_{ISI} = \mathbf{C}_S \mathbf{C}_S^{\mathcal{H}}. \tag{2.58}$$

The advantages and disadvantages of the two methods, as compared with each other, for the full MIMO channel are described in section 2.6.

2.4 EXTENSION FROM A SISO CHANNEL TO A MIMO CHANNEL

The MIMO channel model of Xiao et al. [31] models the channel between the receive and transmit antennas as consisting of individual SISO channels between each transmitter and receiver antenna. The individual SISO channel paths is illustrated in fig. 2.7.

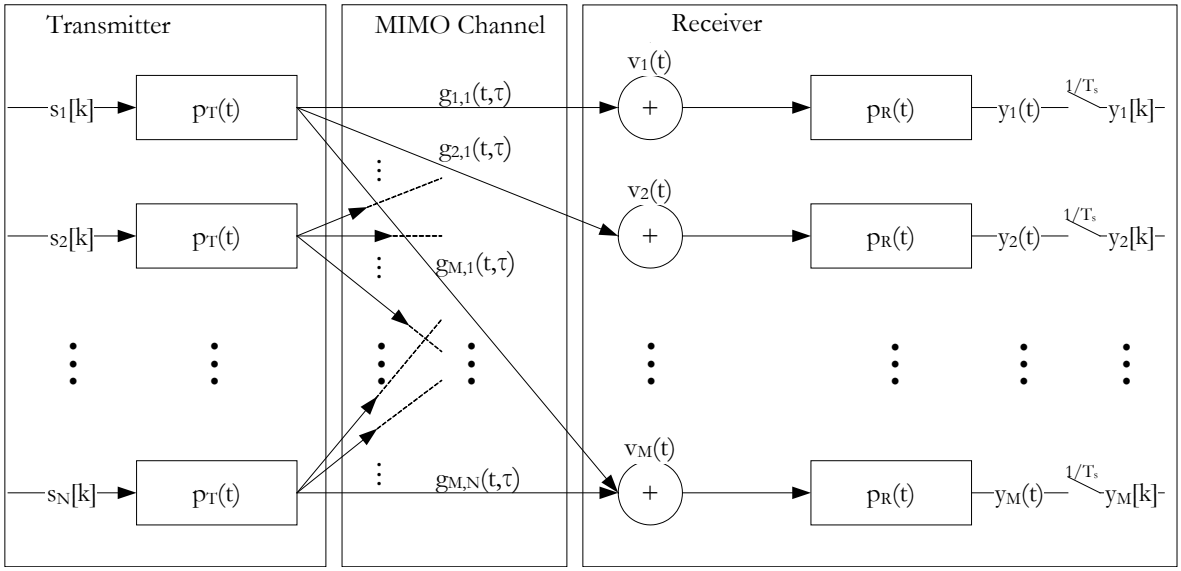

Figure 2.7: MIMO communications model.

This allows the extension of the direct method to MIMO by using the Kronecker product as by Xiao et al. [31]. The MIMO channel is thus given by:

$$\mathbf{H}[k] = \mathbf{C}_M \Phi[k], \tag{2.59}$$

with Φ being a column vector of $N_{TX}N_{RX}I$ independent flat Rayleigh fading and $\mathbf{C}_M$ be-

ing:

$$\mathbf{C}_M = \mathbf{\Psi}_{RX}^{1/2} \otimes \mathbf{\Psi}_{TX}^{1/2} \otimes \mathbf{C}_S, \tag{2.60}$$

where $\mathbf{\Psi}_{TX}$ and $\mathbf{\Psi}_{RX}$ are the transmitter and receiver correlation matrices respectively and N_{TX} and N_{RX} are the number of receive and transmit antennas. In the above eqn. $\otimes$ represents the Kronecker product.

This channel has the exact same covariance matrix, and is therefore statistically the same, as the channel proposed by Xiao et al. [31]. This can be shown by computing the covariance matrix of $\mathbf{H}$ and showing that it is identical to that of Xiao et al. [31]:

$$\begin{aligned}
\mathbf{C}_h &= \mathbf{E}\left[\mathbf{H}\mathbf{H}^{\mathcal{H}}\right] && (2.61)\\
&= \mathbf{E}\left[(\mathbf{C}_M\Phi)(\mathbf{C}_M\Phi)^{\mathcal{H}}\right] && (2.62)\\
&= \mathbf{E}\left[\mathbf{C}_M\Phi\Phi^{\mathcal{H}}\mathbf{C}_M^{\mathcal{H}}\right] && (2.63)\\
&= \mathbf{E}\left[[\mathbf{C}_M\mathbf{C}_M^{\mathcal{H}}\right] && (2.64)\\
&= \mathbf{E}\left[\left(\mathbf{\Psi}^{1/2}\otimes\mathbf{C}_S\right)\left(\mathbf{\Psi}^{1/2}\otimes\mathbf{C}_S\right)^{\mathcal{H}}\right] && (2.65)\\
&= \mathbf{E}\left[\left(\mathbf{\Psi}^{1/2}\left(\mathbf{\Psi}^{1/2}\right)^{\mathcal{H}}\right)\otimes\left(\mathbf{C}_S\mathbf{C}_S^{\mathcal{H}}\right)\right] && (2.66)\\
&= \mathbf{E}\left[\Psi\otimes\mathbf{C}_{ISI}\right] && (2.67)\\
&= \mathbf{E}\left[\Psi_{RX}\otimes\Psi_{TX}\otimes\mathbf{C}_{ISI}\right]. && (2.68)
\end{aligned}$$

2.5 SIMULATION RESULTS

Both the covariance and direct methods were simulated for both the SISO and MIMO cases. The simulation parameters are given in tab. 2.1. For the MIMO simulations a 2×2 channel with a correlation factor of $r = 0.25$ was used for the receiver and transmitter.

Table 2.1: Simulation Parameters

FFT Size	σ^2	$\tau\ (T_S)$
128	[1/3, 1/3, 1/3]	[0, 1.5 , 1.9]

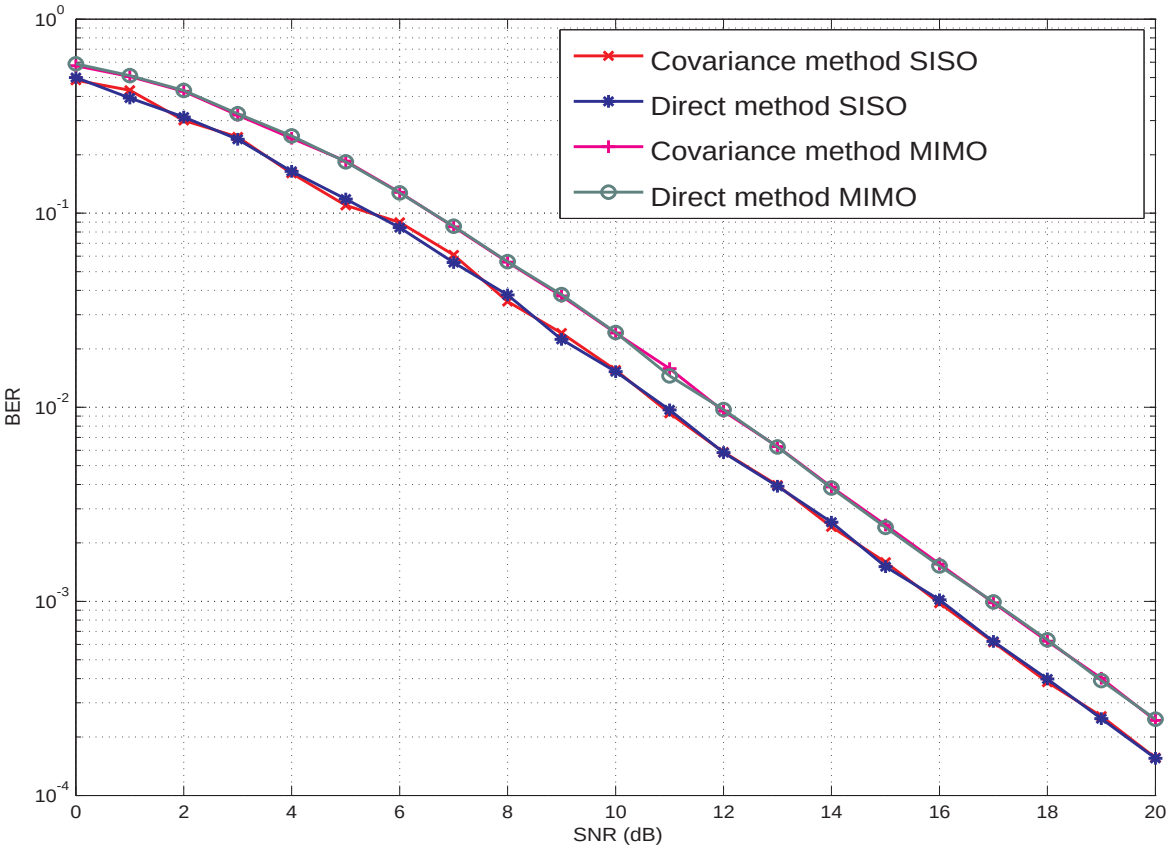

Figure 2.8: The comparison of the simulated BER of an elementary OFDM system on the two channel realisation methods for SISO and MIMO.

From fig. 2.8, one can observe that the covariance method and the direct method yield the same BER performance in frequency selective Rayleigh fading channels. The results were obtained using a simplistic OFDM system and perfect channel state information was assumed. For the MIMO channel a zero forcing equaliser was used. The exact same noise and data vectors were used for the comparison of the two methods, only the fading vectors differed.

2.6 COMPARISON OF THE TWO CHANNEL MODELS

The advantage of the direct method over the covariance method (see sections 2.3.1 and 2.3.2) lies in the fact that the length of the vector of fades is generally reduced. The direct method only has as many fading processes as there are taps in the PDP. Channels that have few taps

in their PDPs, yet having a large delay spread, will thus experience a significant reduction in complexity upon employing the direct method. This is due to the fact that the covariance method will, in such channels, require *at least* $\tau_D f_s$ fading processes, where τ_D is the delay spread and f_s is the sampling frequency. In general, the covariance method requires, assuming $\tau_i f_s \in \mathbb{N}_0,\ \forall i$, $\tau_I f_s N_{TX} N_{RX}$ independent Rayleigh flat fading processes, while the direct method requires at most $I N_{TX} N_{RX}$ processes. Thus the reduction in complexity is, at least, given by:

$$\frac{I N_{TX} N_{RX}}{\tau_I f_s N_{TX} N_{RX}} = \frac{I}{\tau_I f_s}. \tag{2.69}$$

The reduction in complexity may be greater depending on $R_{\rho_T \rho_R}$ and $\tau_i f_s$. For example, compare two channels with a symmetrical $R_{\rho_T \rho_R}$ having $L_1 + L_2 + 1 = 11$, one two-tap channel $\tau_D = 10 T_s$ and a second channel with thirty delay elements. The direct method requires two fading processes to simulate this channel, while the covariance method requires approximately twenty. The direct method would therefore be approximately ten times faster. For the second channel, the direct method requires thirty fading processes, while the covariance method would still only require twenty. Essentially, the direct method simulates the given channel, while the covariance matrix simulates a statistically equivalent channel.

An additional advantage of the direct method is that it is able to simulate fading processes having differing Doppler PSDs. This is of use when simulating, for example, COST models [47]. Simulating a COST model, having different Doppler spectra, can be done by simply ordering the fading vector $\mathbf{\Phi}$ appropriately. Thus, if the i^{th} path has a Jakes spectrum, then Φ_i must be generated from a Rayleigh distribution having a Jakes spectrum. If the i^{th} path has a Gaussian spectrum, then Φ_i must be generated from a Rayleigh distribution having a Gaussian spectrum with appropriate parameters.

In summery, in this chapter the physical aspects, such as noise, Doppler spread and fading, of a typical wireless channels was described. The use of PDPs in describing a multipath channel was detailed and used to generate a discrete sampled channel realisation for use in a SISO channel. Two methods for generating the discrete sampled channel values, the covariance and direct method as described in section 2.3.1 and 2.3.2 respectively, were introduced. The

SISO channel was then extended to a MIMO model using the Kronecker model. It was shown that the two methods of generating the channel coefficients, the covariance and direct method, are mathematically identical for zero mean i.i.d fading. Finally, some of the advantages and disadvantages that the covariance and direct method of generating channel values have, with respect to each other, were presented.

CHAPTER 3

MIMO MAP DETECTION

In this chapter various methods for determining the most likely (ML) sent data given the received signal, **y**, and the channel, **H**, are discussed. The algorithms discussed are the ZF, ZF-DF, MMSE, MMSE-DF algorithm, the SD and the HNN. The aim of this dissertation is to extend these algorithms to allow for iterative decoding of MIMO STFCs using extrinsic information, such as may be supplied by the concatenation of the STFC with an FEC outer code (see chapter 6).

The above mentioned algorithms are commonly used to determine the channel data symbols **x** that were sent. A general metric for MAP that includes a priori information (that is information about the sent data that is known beforehand) is derived in section 3.1. It is shown that the metric for the case where the noise is complex AWGN differs slightly from the case where the noise is real AWGN. It is also shown that where no a priori information is used, or provided, the metrics are identical.

Since FEC is generally employed after the decoder and it is known that up to a 2 dB improvement can be obtained from using soft input FEC decoders [40] it is advantageous to calculate soft output values from the MIMO decoder. In this chapter a *hard-to-soft* conversion is described in section 3.2 using the max-log approximation in order to obtain soft outputs.

In section 3.3 ZF decoder using triangular decompositions is presented, followed by extending the ZF decoder to a MMSE decoder in section 3.4. The ZF and MMSE decoder is then

extended to include DF in section 3.5. In section 3.6 a method using a priori information in the ZF and MMSE decoders (with or without DF) is presented.

The classical sphere decoder algorithm is presented in section 3.7.1 and extended to include a priori information based on the observation that the sphere decoder is a tree search algorithm.

In section 3.8 various symbol ordering strategies are proposed. It is known that for the ZF type decoder the symbol decoding order can have an impact on the performance of the decoder and that for the sphere decoder it can reduce the computations performed [15]. The proposed alternative sorting algorithms may offer improved BER performance for the ZF and MMSE decoders and reduced decoding computations for the sphere decoder. The ordering strategies use a combination of the a priori and channel information to determine the symbol decoding order. The results of the various sorting methods can be seen in section 7.1.5 and 7.2.

The HNN of Hopfield and Tank [33] was extended to be able to decode MIMO STFCs in section 3.9. It is then shown how to use the bias input of the HNN to include a priori information.

Additionally an alternative method of including a priori information into a decoder is given in section 3.10 and shown to be limited to signal constellations that have a linear mapping from channel symbols to data symbols. This method, proposed by Wang and Giannakis [20], is based on translating the received signal vector based on the a priori information.

Lastly, in section 3.11 a reduced complexity SD is presented that uses the HNN to calculate the counter hypothesis (see section 3.2) instead of using the SD to perform the calculation.

3.1 MAP MIMO DETECTION

In this section the MAP detection criteria is derived. This is effectively the computation of the ML metric given a prior distribution (other than the uniform distribution). The derivation

makes use Bayes' rule [40] to derive a metric for the calculation of the most likely sent data given the received signal, the channel, the a priori information and the fact that the noise is assumed to be Gaussian. The starting point of the derivation is the expression of the received MIMO signal, which is then used as a basis for the calculation of the MAP metric.

The received signal, $\mathbf{y}$, of a MIMO system can be described as:

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n}, \tag{3.1}$$

where $\mathbf{H}$ is the equivalent channel matrix, $\mathbf{x}$ is the column vector of transmitted channel symbols and $\mathbf{n}$ is the column vector of AWGN. This takes into consideration the channel as well as any linear pre-coding and layering that was done on the transmitted sequence.

The probability that a particular data sequence was transmitted given the received signal and the channel through which it was transmitted can be expressed as:

$$P(\mathbf{x}|\mathbf{y}, \mathbf{H}). \tag{3.2}$$

Using Bayes' rule, eqn. 3.2 can be expanded as:

$$P(\mathbf{x}|\mathbf{y}, \mathbf{H}) = P(\mathbf{y}|\mathbf{x}, \mathbf{H})P(\mathbf{x}). \tag{3.3}$$

The term $P(\mathbf{x})$ represent the a priori information about the transmitted data vector. Frequently if no a priori information is available, it is assumed that all data vectors are equiprobable and the term ignored. If, however a priori information is available, it can be represented in LLR form (see section 3.1.1) that can simplify the calculation of the a priori term as will be evident in section 3.1.2.

3.1.1 A priori information and log likelihood Ratios

A priori information is often expressed as a vector, λ^a, of LLR values. The a priori information λ^a, represented in LLR form, is given by:

$$\lambda_i^a = \ln\left(\frac{P(b_i = +1)}{P(b_i = -1)}\right), \tag{3.4}$$

where λ_i^a is the a priori information for the i^{th} bit, b_i.

If given the a priori LLR, λ_i^a, of a bit, the a priori probabilities of that bit can be calculated as follows:

$$P(b_i = +1) = \frac{1}{1 + \exp(\lambda_i^a)}, \tag{3.5}$$

$$P(b_i = -1) = \frac{\exp(\lambda_i^a)}{1 + \lambda_i^a}. \tag{3.6}$$

The a priori probability of $\mathbf{x}$ can be calculated as:

$$P(\mathbf{x}) = \prod_i P(x_i) \tag{3.7}$$

$$= \prod_{i:b_i=+1} P(b_i = +1) \prod_{i:b_i=-1} P(b_i = -1), \tag{3.8}$$

where b_i is the i^{th} bit. BPSK signalling is assumed for the bits.

Using eqns. 3.5 and 3.6 the a priori probability of $\mathbf{x}$ can be written in terms of LLRs as:

$$P(\mathbf{x}) = \prod_i \frac{\exp\left[\frac{1}{2}(1 + b_i)\lambda_i^a\right]}{1 + \exp(\lambda_i^a)}. \tag{3.9}$$

Equation 3.9 will be used in eqn. 3.26 to simplify the calculation of the contribution of the a priori information.

3.1.2 Real AWGN MAP Metric

Frequently only real AWGN is used in simulations of communication systems as often only BPSK systems are simulated. Should QPSK be simulated it can easily be done via a complex-to-real matrix transform, such as eqn. 3.38 [54][55, p. 229]. When performing hard decision decoding without a priori information there is no difference between the MAP metric for real and complex AWGN. However, there is a slight difference in the MAP metric once a priori information is introduced. This is evident in the difference between eqns. 3.33 and 3.37.

In the event that the noise is zero mean AWGN, the conditional probability $P(\mathbf{y}|\mathbf{x},\mathbf{H})$ is given by:

$$P(\mathbf{y}|\mathbf{x},\mathbf{H}) = \frac{1}{\sqrt{(2\pi)^{N_R}\det(\mathbf{\Sigma})}}\exp\left[-\frac{1}{2}(\mathbf{y}-\mathbf{H}\mathbf{x})^{\mathcal{T}}\mathbf{\Sigma}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{x})\right], \tag{3.10}$$

where $\mathbf{\Sigma}$ denotes the covariance matrix of $(\mathbf{y}-\mathbf{H}\mathbf{x})$ given by:

$$\mathbf{\Sigma} = E\left[(\mathbf{y}-\mathbf{H}\mathbf{x})(\mathbf{y}-\mathbf{H}\mathbf{x})^{\mathcal{T}}\right]. \tag{3.11}$$

Since the noise is Gaussian and i.i.d., eqn. 3.11 can be written as:

$$\mathbf{\Sigma} = \sigma_n^2\mathbf{I}_{N_R}, \tag{3.12}$$

where σ_n^2 is the variance of the noise. Consequently $\det(\mathbf{\Sigma})$ and $\mathbf{\Sigma}^{-1}$ can be written as:

$$\det(\mathbf{\Sigma}) = \det(\sigma_n^2\mathbf{I}_{N_R}) \tag{3.13}$$

$$= \sigma_n^{2N_R}, \tag{3.14}$$

$$\mathbf{\Sigma}^{-1} = \sigma_n^2\mathbf{I}_{N_R}^{-1} \tag{3.15}$$

$$= \frac{1}{\sigma_n^2}\mathbf{I}_{N_R}. \tag{3.16}$$

Substituting eqn. 3.14 and eqn. 3.16 into eqn. 3.2 yields the following:

$$P(\mathbf{y}|\mathbf{x},\mathbf{H}) = \frac{1}{\sigma_n^{N_R}\sqrt{(2\pi)^{N_R}}}\exp\left[-\frac{(\mathbf{y}-\mathbf{H}\mathbf{x})^{\mathcal{T}}\mathbf{I}_{\mathbf{N_R}}(\mathbf{y}-\mathbf{H}\mathbf{x})}{2\sigma_n^2}\right] \tag{3.17}$$

$$= \frac{1}{\sigma_n^{N_R}\sqrt{(2\pi)^{N_R}}}\exp\left[-\frac{(\mathbf{y}-\mathbf{H}\mathbf{x})^{\mathcal{T}}(\mathbf{y}-\mathbf{H}\mathbf{x})}{2\sigma_n^2}\right]. \tag{3.18}$$

Making note of the fact that for real valued column vectors:

$$\mathbf{x}^{\mathcal{T}}\mathbf{x} = \|\mathbf{x}\|^2, \tag{3.19}$$

eqn. 3.18 can be written as:

$$P(\mathbf{y}|\mathbf{x},\mathbf{H}) = \frac{1}{\sigma_n^{N_R}\sqrt{(2\pi)^{N_R}}}\exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}\right]. \tag{3.20}$$

Substituting eqn. 3.9 and eqn. 3.20 into eqn. 3.3 results in the following:

$$P(\mathbf{x}|\mathbf{y},\mathbf{H}) = P(\mathbf{y}|\mathbf{x},\mathbf{H})P(\mathbf{x}) \tag{3.21}$$

$$= \frac{1}{\sigma_n^{N_R}\sqrt{(2\pi)^{N_R}}} \exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}\right] \prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}. \tag{3.22}$$

The MAP solution, $\hat{\mathbf{x}}$, is the $\mathbf{x}$ vector that maximises probability:

$$\hat{\mathbf{x}} = \arg\max_{\mathbf{x}} \{P(\mathbf{y}|\mathbf{x},\mathbf{H})P(\mathbf{x})\}. \tag{3.23}$$

Since all probabilities are less than one, minimising the negative of the logarithm of the conditional probability is equivalent to maximising the conditional probability:

$$\mathbf{x}_{MAP} = \arg\max_{\mathbf{x}} \{P(\mathbf{y}|\mathbf{x},\mathbf{H})P(\mathbf{x})\} \tag{3.24}$$

$$= \arg\min_{\mathbf{x}} \{-\ln\left(P(\mathbf{y}|\mathbf{x},\mathbf{H})P(\mathbf{x})\right)\} \tag{3.25}$$

$$= \arg\min_{\mathbf{x}} \left\{-\ln\left(\frac{1}{\sigma_n^{N_R}\sqrt{(2\pi)^{N_R}}} \exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}\right] \prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}\right)\right\}. \tag{3.26}$$

It is possible to simplify eqn. 3.26 by ignoring terms that are not dependent on $\mathbf{x}$. This includes the scaling terms such as:

$$\frac{1}{\sigma_n^{N_R}\sqrt{(2\pi)^{N_R}}}. \tag{3.27}$$

Simplifying eqn. 3.26 and ignoring constant terms and common factors:

$$\mathbf{x}_{MAP} = \arg\min_{\mathbf{x}} \left\{ \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2} - \frac{1}{2}\sum_i (1 + b_i)\lambda_i^a + \sum_i \ln\left[1 + \exp(\lambda_i^a)\right] \right\} \tag{3.28}$$

$$= \arg\min_{\mathbf{x}} \left\{ \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2} - \frac{1}{2}\sum_i (1 + b_i)\lambda_i^a \right\} \tag{3.29}$$

$$= \arg\min_{\mathbf{x}} \left\{ \frac{\|\mathbf{y} + \mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2} - \frac{1}{2}\sum_i b_i\lambda_i^a - \frac{1}{2}\sum_i \lambda_i^a \right\} \tag{3.30}$$

$$= \arg\min_{\mathbf{x}} \left\{ \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2} - \frac{1}{2}\sum_i b_i\lambda_i^a \right\} \tag{3.31}$$

$$= \arg\min_{\mathbf{x}} \left\{ \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2} - \frac{1}{2}\mathbf{b} \cdot \boldsymbol{\lambda}^a \right\} \tag{3.32}$$

$$= \arg\min_{\mathbf{x}} \left\{ \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\sigma_n^2} - \mathbf{b} \cdot \boldsymbol{\lambda}^a \right\}, \tag{3.33}$$

where 3.33 is the final simplified equation for obtaining the MAP solution for the case where the noise is real AWGN.

3.1.3 Complex AWGN MAP Metric

In the previous subsection the MAP metric for real AWGN was derived. It will be shown that the metric for complex AWGN, where a priori information is included, differs marginally from the real AWGN case. This is of importance when the system employs both quadrature and in-phase components that must be decoded.

In the case where the noise is complex AWGN and circularly symmetrical, the Gaussian probability distribution differs slightly. In this case the conditional probability is given by [56][55, p. 226]:

$$P(\mathbf{y}|\mathbf{x}, \mathbf{H}) = \frac{1}{(\pi)^{N_R} \det(\boldsymbol{\Sigma})} \exp\left[-(\mathbf{y} - \mathbf{H}\mathbf{x})^{\mathcal{H}} \boldsymbol{\Sigma}^{-1} (\mathbf{y} - \mathbf{H}\mathbf{x})\right], \tag{3.34}$$

where Σ denotes the covariance matrix of $(\mathbf{y} - \mathbf{H}\mathbf{x})$ given by:

$$\boldsymbol{\Sigma} = E\left[(\mathbf{y} - \mathbf{H}\mathbf{x})(\mathbf{y} - \mathbf{H}\mathbf{x})^{\mathcal{H}}\right]. \tag{3.35}$$

Proceeding in a similar fashion as in section 3.1.2 with substitutions and simplifications, the

following is obtained:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\sigma_n^2} - \frac{1}{2}\sum_i (1 + b_i)\lambda_i^a + \sum_i \ln\left[1 + \exp(\lambda_i^a)\right] \right\} \tag{3.36}$$

$$= \arg\min_{\mathbf{x}} \left\{ \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\sigma_n^2} - \frac{1}{2}\mathbf{b} \cdot \boldsymbol{\lambda}^a \right\}, \tag{3.37}$$

where 3.37 is the final simplified equation for obtaining the MAP solution for when the noise is complex AWGN. It should be noticed that the contribution of the a priori information in 3.37 has been reduced by a factor of two, as compared to the case where the noise was real as in eqn. 3.33.

It is also possible to use the following transformation of the complex valued MIMO system for determining the MAP solution [54, 55]:

$$\begin{bmatrix} \Re\{\mathbf{y}\} \\ \Im\{\mathbf{y}\} \end{bmatrix} = \begin{bmatrix} \Re\{\mathbf{H}\} & -\Im\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} & \Re\{\mathbf{H}\} \end{bmatrix} \begin{bmatrix} \Re\{\mathbf{x}\} \\ \Im\{\mathbf{x}\} \end{bmatrix} + \begin{bmatrix} \Re\{\mathbf{n}\} \\ \Im\{\mathbf{n}\} \end{bmatrix}, \tag{3.38}$$

which will be referred to as the *complex-to-real* transformation. This transformation of the complex valued MIMO system into a real valued MIMO system can alternatively be used, in conjunction with eqn. 3.10, to derive the MAP solution to the complex AWGN case.

3.2 SOFT OUTPUT MAP DETECTION

In section 3.1 the assumption was made that only the vector $\mathbf{x}$ that was most likely transmitted was of importance. This is referred to as *hard output* decoding. It is well known that certain FEC algorithms can make use of a priori information as well, algorithms such as convolutional and LDPC codes, to name but a few. In these cases, it is useful to not only provide the MAP solution, but also the probabilities or likelihood ratios of the symbols or bits of the MAP solution.

In this section the MAP log likelihood ratios for each bit is derived. As in section 3.1 this is done for both real and complex AWGN.

3.2.1 Soft output calculation in real AWGN

The a posteriori bit LLRs, λ^p, are given by:

$$\lambda_i^p = \ln\left(\frac{P(b_i = +1|\mathbf{y}, \mathbf{H})}{P(b_i = -1|\mathbf{y}, \mathbf{H})}\right), \tag{3.39}$$

where BPSK signalling per bit has been assumed. The probability that a bit equal one, $b_i = 1$, can be obtained by summing the probabilities of all possible transmitted vectors where that bit equals one:

$$P(b_i = +1) = \sum_{\mathbf{x}\in\mathbf{X}_i^{b=+1}} P(\mathbf{x}|\mathbf{y}, \mathbf{H}), \tag{3.40}$$

where $\mathbf{X}_i^{b=s}$ denotes the set of all possible transmit vectors with $b_i = s$. Similarly:

$$P(b_i = -1) = \sum_{\mathbf{x}\in\mathbf{X}_i^{b=-1}} P(\mathbf{x}|\mathbf{y}, \mathbf{H}). \tag{3.41}$$

Substituting eqn. 3.40 and eqn. 3.41 into eqn. 3.39 then yields:

$$\lambda_i^p = \ln\left(\frac{\sum_{\mathbf{x}\in\mathbf{X}_i^{b=+1}} P(\mathbf{x}|\mathbf{y}, \mathbf{H})}{\sum_{\mathbf{x}\in\mathbf{X}_i^{b=-1}} P(\mathbf{x}|\mathbf{y}, \mathbf{H})}\right), \tag{3.42}$$

where $P(\mathbf{x}|\mathbf{y}, \mathbf{H})$ has already been calculated by means of Bayes' Theorem in eqn. 3.22. Thus,

substituting eqn. 3.22 into eqn. 3.42 yields:

$$\lambda_i^p = \ln\left(\frac{\displaystyle\sum_{\mathbf{x}\in\mathbf{X}_i^{b=+1}} \frac{1}{\sigma_n^{N_R}\sqrt{(2\pi)^{N_R}}} \exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}\right]\prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}}{\displaystyle\sum_{\mathbf{x}\in\mathbf{X}_i^{b=-1}} \frac{1}{\sigma_n^{N_R}\sqrt{(2\pi)^{N_R}}} \exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}\right]\prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}}\right) \tag{3.43}$$

$$= \ln\left(\frac{\displaystyle\sum_{\mathbf{x}\in\mathbf{X}_i^{b=+1}} \exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}\right]\prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}}{\displaystyle\sum_{\mathbf{x}\in\mathbf{X}_i^{b=-1}} \exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}\right]\prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}}\right) \tag{3.44}$$

$$= \ln\left(\sum_{\mathbf{x}\in\mathbf{X}_i^{b=+1}} \exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}\right]\prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}\right) -$$
$$\ln\left(\sum_{\mathbf{x}\in\mathbf{X}_i^{b=-1}} \exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}\right]\prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}\right) \tag{3.45}$$

$$= \ln\left(\sum_{\mathbf{x}\in\mathbf{X}_i^{b=+1}} \exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2} + \ln\left(\prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}\right)\right]\right) -$$
$$\ln\left(\sum_{\mathbf{x}\in\mathbf{X}_i^{b=-1}} \exp\left[-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2} + \ln\left(\prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}\right)\right]\right). \tag{3.46}$$

From eqn. 3.46 it is evident that calculating the a posteriori LLRs will require a summation over the entire transmit vector space. This would then have to be repeated separately for each bit's LLR value. This is clearly very computationally intensive.

Fortunately the calculation in eqn. 3.46 may be simplified by making use of the max-log approximation [57, eqn. 4.48]:

$$\ln\left(\sum_i e^{x_i}\right) \approx \arg\max_i\{x_i\}, \tag{3.47}$$

thus yielding:

$$\lambda_i^p \approx \underset{\mathbf{x}\in\mathbf{X}_i^{b=+1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}+\ln\left(\prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}\right)\right\}- \underset{\mathbf{x}\in\mathbf{X}_i^{b=-1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}+\ln\left(\prod_i \frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}\right)\right\} \tag{3.48}$$

$$= \underset{\mathbf{x}\in\mathbf{X}_i^{b=+1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}+\sum_i \ln\left(\frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}\right)\right\}- \underset{\mathbf{x}\in\mathbf{X}_i^{b=-1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}+\sum_i \ln\left(\frac{\exp\left[\frac{1}{2}(1+b_i)\lambda_i^a\right]}{1+\exp(\lambda_i^a)}\right)\right\} \tag{3.49}$$

$$= \underset{\mathbf{x}\in\mathbf{X}_i^{b=+1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}+\sum_i \frac{1}{2}(1+b_i)\lambda_i^a-\sum_i(1+\exp(\lambda_i^a))\right\}- \underset{\mathbf{x}\in\mathbf{X}_i^{b=-1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}+\sum_i \frac{1}{2}(1+b_i)\lambda_i^a-\sum_i(1+\exp(\lambda_i^a))\right\} \tag{3.50}$$

$$= \underset{\mathbf{x}\in\mathbf{X}_i^{b=+1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}+\frac{1}{2}\mathbf{b}\cdot\boldsymbol{\lambda}^a+\sum_i\left(\frac{1}{2}\lambda_i^a-1+\exp(\lambda_i^a)\right)\right\}- \underset{\mathbf{x}\in\mathbf{X}_i^{b=-1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}+\frac{1}{2}\mathbf{b}\cdot\boldsymbol{\lambda}^a+\sum_i\left(\frac{1}{2}\lambda_i^a-1+\exp(\lambda_i^a)\right)\right\}. \tag{3.51}$$

In eqn. 3.51 the term:

$$\sum_i\left(\frac{1}{2}\lambda_i^a-1+\exp(\lambda_i^a)\right) \tag{3.52}$$

does not depend on the transmitted vector, thus it can be placed outside of the maximisation terms. Since it is common to both maximisations, though with different signs, it will cancel out. This results in:

$$\lambda_i^p \approx \underset{\mathbf{x}\in\mathbf{X}_i^{b=+1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}+\frac{1}{2}\mathbf{b}\cdot\boldsymbol{\lambda}^a)\right\}- \underset{\mathbf{x}\in\mathbf{X}_i^{b=-1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2}+\frac{1}{2}\mathbf{b}\cdot\boldsymbol{\lambda}^a\right\} \tag{3.53}$$

$$= \frac{1}{2}\left(\underset{\mathbf{x}\in\mathbf{X}_i^{b=+1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2}+\mathbf{b}\cdot\boldsymbol{\lambda}^a\right\}- \underset{\mathbf{x}\in\mathbf{X}_i^{b=-1}}{\arg\max}\left\{-\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2}+\mathbf{b}\cdot\boldsymbol{\lambda}^a\right\}\right). \tag{3.54}$$

It is often more useful to express eqn. 3.54 as a minimisation instead of a maximisation:

$$\lambda_i^p \approx \frac{1}{2}\left(\arg\min_{\mathbf{x}\in\mathbf{X}_i^{b=-1}}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2}-\mathbf{b}\cdot\boldsymbol{\lambda}^a\right\}-\arg\min_{\mathbf{x}\in\mathbf{X}_i^{b=+1}}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2}-\mathbf{b}\cdot\boldsymbol{\lambda}^a\right\}\right). \tag{3.55}$$

The problem of determining the a posteriori LLRs of the received bits has thus been simplified to finding the MAP solution, $\hat{\mathbf{x}}$, and then finding a counter hypothesis $\mathbf{x}_\mathbf{i}'$ for each bit b_i:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2}-\mathbf{b}\cdot\boldsymbol{\lambda}^a\right\} \tag{3.56}$$

$$\mathbf{x}_i' = \arg\min_{\mathbf{x}\in\mathbf{X}_i^{b=-\hat{b}}}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2}-\mathbf{b}\cdot\boldsymbol{\lambda}^a\right\}. \tag{3.57}$$

The max-log MAP LLR λ_i^p value is then calculated as:

$$\lambda_i^p \approx \frac{\hat{b}_i}{2}\left[\left(\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}_i'\|^2}{\sigma_n^2}-\mathbf{b}_i'\cdot\boldsymbol{\lambda}^a\right)-\left(\frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{x}}\|^2}{\sigma_n^2}-\hat{\mathbf{b}}\cdot\boldsymbol{\lambda}^a\right)\right]. \tag{3.58}$$

Thus, the magnitude of the max-log MAP LLR soft output is the difference between the MAP metric and the MAP metric for the counter hypothesis. In the next section the MAP LLR metric for complex Gaussian noise is calculated and is shown to be similar.

3.2.2 Soft output calculation in complex AWGN

The a posteriori LLR for when the noise is circularly complex Gaussian i.i.d. can be calculated in a similar manner as was done in section 3.2.1 for the case where the noise was real AWGN. The difference being that the conditional probability:

$$P(\mathbf{y}|\mathbf{x},\mathbf{H}) = \frac{1}{(\pi)^{N_R}\det(\boldsymbol{\Sigma})}\exp\left[-(\mathbf{y}-\mathbf{H}\mathbf{x})^{\mathcal{H}}\boldsymbol{\Sigma}^{-1}(\mathbf{y}-\mathbf{H}\mathbf{x})\right], \tag{3.59}$$

is used instead. This results in the a posteriori LLRs being given by:

$$\lambda_i^p \approx \arg\min_{\mathbf{x}\in\mathbf{X}_i^{b=-1}}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2}-\frac{1}{2}\mathbf{b}\cdot\boldsymbol{\lambda}^a\right\}-\arg\min_{\mathbf{x}\in\mathbf{X}_i^{b=+1}}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2}-\frac{1}{2}\mathbf{b}\cdot\boldsymbol{\lambda}^a\right\}. \tag{3.60}$$

Equation 3.60 can then be rewritten in terms of the MAP solution $\hat{\mathbf{x}}$ and the MAP counter hypothesis $\mathbf{x}'_i$ as:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}} \left\{ \frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2} - \frac{1}{2}\mathbf{b}\cdot\boldsymbol{\lambda}^a \right\} \tag{3.61}$$

$$\mathbf{x}'_i = \arg\min_{\mathbf{x}\in\mathbf{X}_i^{b=-\hat{b}}} \left\{ \frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{\sigma_n^2} - \frac{1}{2}\mathbf{b}\cdot\boldsymbol{\lambda}^a \right\}. \tag{3.62}$$

The final max-log MAP LLR λ_i^p value is then calculated as:

$$\lambda_i^p \approx \hat{b}_i \left[\left(\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}'_i\|^2}{\sigma_n^2} - \frac{1}{2}\mathbf{b}'_i\cdot\boldsymbol{\lambda}^a \right) - \left(\frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{x}}\|^2}{\sigma_n^2} - \frac{1}{2}\hat{\mathbf{b}}\cdot\boldsymbol{\lambda}^a \right) \right]. \tag{3.63}$$

It should be noted that the complex-to-real transformation eqn. 3.38 could also have been used to derive an expression for the a posteriori LLRs. Using eqn. 3.38 in conjunction with eqn. 3.58 will also result in a correct result. However, there may be some computational performance penalties associated with the transformation due to the resultant larger matrix sizes.

3.3 ZERO FORCING DECODER

In sections 3.1 and 3.2 the max-log MAP LLR calculation required the calculation of the MAP solution and an alternative hypothesis.

One method of performing the actual calculation of the MAP solution is that of the ZF decoder. The ZF decoder corresponds to the least mean squares (LMS) estimate where the metric being minimised is equivalent to the MAP metric.

In this section it is shown how the ZF decoder can be implemented using an algorithm based on a triangular matrix decomposition. This allows the addition of DF and a priori information as will be shown later in this chapter.

The ZF decoder at its simplist multiplies the received signal by the inverse of the channel in an attempt to estimate the most likely sent sequence. Multiplying eqn. 3.1 by the inverse of

the channel matrix, $\mathbf{H}^{-1}$ yields the ZF solution:

$$\mathbf{H}^{-1}\mathbf{y} = \mathbf{H}^{-1}\mathbf{H}\mathbf{x} + \mathbf{H}^{-1}\mathbf{n} \tag{3.64}$$

$$= \mathbf{x} + \mathbf{H}^{-1}\mathbf{n}. \tag{3.65}$$

In the event that $\mathbf{H}$ is non invertible the Moore-Penrose pseudo inverse can also be used. The Moore-Penrose pseudo inverse, denoted by $()^{+}$, being defined as [14]:

$$\mathbf{H}^{+} = (\mathbf{H}^{\mathcal{H}}\mathbf{H})^{-1}\mathbf{H}^{\mathcal{H}}. \tag{3.66}$$

Alternatively, the QR decomposition of $\mathbf{H}$ may be used to obtain the ZF solution [14]. First the QR decomposition is used to transform eqn. 3.1 into an upper triangular system:

$$\mathbf{Q}\mathbf{R} = \mathbf{H}, \tag{3.67}$$

$$\therefore \mathbf{Q}^{\mathcal{H}}\mathbf{y} = \mathbf{R}\mathbf{x} + \mathbf{Q}^{\mathcal{H}}\mathbf{n}, \tag{3.68}$$

where use of the fact that $\mathbf{Q}^{-1} = \mathbf{Q}^{\mathcal{H}}$ has been made. Furthermore, since $\mathbf{Q}$ has orthonormal columns, it is assumed that $\mathbf{Q}^{\mathcal{H}}\mathbf{n}$ is still AWGN.

Equation 3.68 has therefore been reduced to an upper triangular system, which is very easy to solve with back substitution, thus $\mathbf{x}$ can now be decoded on a symbol-by-symbol basis starting with the last symbol.

The rest of this section will show how the classical back substitution is performed and how symbol-by-symbol decoding is used to implement decision DF as well as used to introduce a priori information into the decoder. The extension to MMSE decoding will also be shown.

The classical ZF decoder uses simple back substitution to solve the upper triangular set system in 3.68. Thus, what will be termed the *classical ZF* estimate and is given by:

$$\hat{x}_n = \frac{y'_n - \sum_{i=0}^{n-1} \hat{x}_{n-i} R_{n,n-i}}{R_{n,n}}, \tag{3.69}$$

where y'_n is the n^{th} element in the column vector given by $\mathbf{Q}^{\mathcal{H}}\mathbf{y}$. To obtain the final ZF output, a search for the closest matching transmit symbol is performed:

$$\hat{s}_n = \arg\min_{x_n \in \mathbb{X}} ||\hat{x}_n - x_n||^2, \tag{3.70}$$

with $\hat{s}_n$ the symbol number corresponding to channel symbol x_n and $\mathbb{X}$ representing the set of all channel symbols, i.e. the signal constellation.

Alternatively, by combining eqn. 3.69 and eqn. 3.70 the following may be obtained:

$$\hat{s}_n = \arg\min_{x_n \in \mathbf{X}} \left\{ \left\| y'_n - \sum_{i=0}^{n-1} \hat{x}_{n-i} R_{n,n-i} - R_{n,n} x_n \right\|^2 \right\}. \tag{3.71}$$

3.4 MINIMUM MEAN SQUARED ERROR DECODER

In section 3.3 one method of solving for the transmitted sequence was to pre-multiply the received signal by the pseudo inverse $\mathbf{H}^+$. The MMSE solution is to find a weighting matrix $\mathbf{W}'$ that minimises the following [58]:

$$\mathbf{W}' = \arg\min_{\mathbf{W}} \left\{ (\mathbf{W}\mathbf{y} - \mathbf{x})(\mathbf{W}\mathbf{y} - \mathbf{x})^{\mathcal{H}} \right\}, \tag{3.72}$$

and to pre-multiply the received vector by the weighting matrix to obtain:

$$\hat{\mathbf{x}} = \mathbf{W}'\mathbf{y}. \tag{3.73}$$

It is possible to solve $\mathbf{W}'$ analytically thereby obtaining:

$$\mathbf{W}' = \left[\mathbf{H}^{\mathcal{H}}\mathbf{H} + \frac{\sigma_x^2}{\sigma_n^2} \mathbf{I}_{N_T} \right]^{-1} \mathbf{H}^{\mathcal{H}}, \tag{3.74}$$

where σ_x^2 is the received signal power and $\mathbf{I}_{N_T}$ is an $N_T \times N_T$ identity matrix.

Alternative, use of the QR decomposition can also be made. In this case the QR decomposition of an augmented $\mathbf{H}$, denoted by $\mathbf{H}_{aug}$, is done where [59]:

$$\mathbf{H}_{aug} = \begin{bmatrix} \mathbf{H} \\ \frac{\sigma_n}{\sigma_x} \mathbf{I}_{N_T} \end{bmatrix}. \tag{3.75}$$

That is eqn. 3.1 is transformed into

$$\mathbf{Q}\mathbf{R} = \mathbf{H}_{aug}, \tag{3.76}$$

$$\therefore \mathbf{Q}_1^{\mathcal{H}}\mathbf{y} = \mathbf{R}_1\mathbf{x} + \mathbf{Q}_1^{\mathcal{H}}\mathbf{n}, \tag{3.77}$$

where $\mathbf{Q}_1$ is the upper left $N_R \times N_T$ part of $\mathbf{Q}$ and $\mathbf{R}_1$ is the upper triangular part of $\mathbf{R}$. The transform:

$$\mathbf{A} = \mathbf{QR} = [\mathbf{Q}_1, \mathbf{Q}_2] \begin{bmatrix} \mathbf{R}_1 \\ \mathbf{0} \end{bmatrix}, \tag{3.78}$$

is referred to as the *thin QR factorisation of* $\mathbf{A}$ [14]. From eqn. 3.77 the same methods used in the ZF decoder to perform the back substitution may be used as the system has been reduced to a similar upper triangular system. This includes the use of decision feedback.

3.5 DECISION FEEDBACK

In decision feedback the previous most likely decoded symbols are used in the back substitution operation of the ZF decoder in the calculation of the subsequent symbols. This is effectively the same as successive interference cancellation.

Decision feedback can be introduced into the ZF and MMSE decoders by using the channel symbols, x, of the previously decoded symbols, $\hat{s}$, in the summations of equations 3.69 and 3.71:

$$\hat{s}_n = \arg\min_{x_n \in \mathbf{X}} \left\{ \left\| y'_n - \sum_{i=0}^{n-1} \Omega(\hat{s}_{n-i}) R_{n,n-i} - R_{n,n} x_n \right\|^2 \right\}. \tag{3.79}$$

where $\Omega(s_n)$ is the function mapping the symbol number s_n to channel symbol x_n.

It is evident that there is not a large performance penalty in including DF in the back substitution step and, as is shown in section 7.1, can offer significant performance improvements.

3.6 INCLUDING A PRIORI INFORMATION IN ZF AND MMSE DECODERS

Using eqn. 3.71 a priori information can be introduced into the back substitution step of the ZF and MMSE decoders, with or without DF, where the current most likely symbol is being

determined:

$$\hat{s}_n = \arg\min_{x_n \in \mathbf{X}} \left\{ \left\| \hat{y}_n - R_{n,n} x_n \right\|^2 - \mathbf{b}_i \cdot \boldsymbol{\lambda}_i^a \right\}, \tag{3.80}$$

where $\hat{y}_n$ is given by:

$$\hat{y}_n = y'_n - \sum_{i=0}^{n-1} \hat{x}_{n-i} R_{n-i,n}, \tag{3.81}$$

$\mathbf{b}_i$ is the vector of bits corresponding to channel symbol x_n and $\boldsymbol{\lambda}_i^a$ is the vector of a priori LLRs associated with the corresponding bits in $\mathbf{b}_i$.

Thus, the symbol-by-symbol decoding nature of the triangular decomposition decoders permits the calculation of the MAP for each symbol at the step where it is being decoded. This step-by-step approach is suboptimal and the performance is greatly dependent on the order in which the symbols are decoded, as shown in section 7.1.5.

The SD is an example of a decoder that can be optimal, though has historically posed certain problems in including a priori information. It is presented in the next section together with an extension to perform optimal MAP decoding.

3.7 SPHERE DECODER

The sphere decoder can be used to efficiently compute the optimal MAP metric, in contrast to the ZF and MMSE which do not necessarily arrive at the optimal solution. In this section the classical sphere decoder algorithm is described. It is then shown how a sphere decoder that incorporates a priori information can be implemented, based on the observation that the sphere decoder is a tree search algorithm.

3.7.1 Classical sphere

The sphere decoder by Safar et al. [17] is essentially a tree search algorithm. It is a variation of the A* tree search algorithm [60, 61]. In the classical sphere decoder the A* algorithm is used to solve the following system:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x} \in \mathbb{X}} \left\{ \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2)}{\sigma_n^2} \right\} \tag{3.82}$$

in order to obtain the MAP transmitted channel symbols and, thus, transmitted data. This is essentially the same as eqns. 3.33 and 3.37 without the a priori information and is the same equation that the classical ZF and MMSE decoders attempt to solve. For use with the sphere decoder, eqn. 3.82 is transformed into a triangular system. In Safar et al. [17], as with most other formulations, the Cholesky decomposition [14]:

$$\mathbf{R}^{\mathcal{H}}\mathbf{R} = \mathbf{A}, \tag{3.83}$$

where $\mathbf{R}$ is upper triangular, is used. Equation 3.82 is then transformed into the triangular equation:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}} \left\{ (\mathbf{x} - \hat{\mathbf{x}})^{\mathcal{H}} \mathbf{R}^{\mathcal{H}}\mathbf{R} (\mathbf{x} - \hat{\mathbf{x}}) \right\}, \tag{3.84}$$

where $\hat{\mathbf{x}}$ is the zero-forcing solution, and $\mathbf{R}^{\mathcal{H}}\mathbf{R} = \mathbf{H}^{\mathcal{H}}\mathbf{H}$ and σ_n^2 has been omitted as it is a constant and does not affect the result. The zero-forcing solution, $\hat{\mathbf{x}}$ can be obtained using, for example, the Moore-Penrose Pseudo inverse:

$$\mathbf{A}^{\dagger} = (\mathbf{A}^{\mathcal{H}}\mathbf{A})^{-1}\mathbf{A}^{\mathcal{H}}, \tag{3.85}$$

as:

$$\hat{\mathbf{x}} = \mathbf{H}^{\dagger}\mathbf{y}. \tag{3.86}$$

Alternatively the QR decomposition, SV decomposition, LU decomposition or even the Cholesky decomposition can be used to compute the zero-forcing solution. Equation 3.82 can also be transformed into a triangular system using the QR decomposition:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}} \left\{ \|\mathbf{Q}^{\mathcal{H}}\mathbf{y} - \mathbf{R}\mathbf{x}\|^2 \right\}, \tag{3.87}$$

where $\mathbf{QR} = \mathbf{H}$. This formulation will be used henceforth.

The tree structure arrises from the observation that $x \in \mathbf{X}$, i.e. the channel symbol x is part of a finite set $\mathbf{X}$ and that when performing the back substitution, the value of the current symbol is dependent on the previous symbols. Thus, each node in the tree represents a combination of symbols up to that level in the tree and similarly up to that level in the back substitution. Child nodes represent the different symbol choices that can be made from their parent nodes. The child nodes of a parent node therefore have a shared history of symbols used in the back

substitution. In the tree the leaf nodes represent all the possible combinations of transmitted symbols.

In eqns. 3.84 and 3.87 it is important to note that when the equations are being evaluated, the evaluations can be calculated on a *symbol-by-symbol* basis, by traversing the tree, and are *monotonously* increasing:

$$T_{n-1} \geq T_n, \tag{3.88}$$

where it should be noted that the tree is traversed from node N upwards in an upper triangular system. Thus, as soon as the metric for a branch, T_n in the tree exceeds the current lowest metric, $T_{\min}$, it will *always* exceed that metric and therefore the branch can be pruned from the solution space. In such cases, the A* algorithm is the optimal tree-traversal algorithm "..in the sense of never expanding a node that can be skipped by some other algorithm having access to the same heuristic information that A* uses.."[60].

The pruning is implemented in the `SYMLIST()` function, see section 3.7.3.2, and the `UPDATE()`, see section 3.7.3.3. In the `SYMLIST()` the symbols in the current level that are less than the pruning metric are calculated. In the function `UPDATE()` all the levels are stepped trough and the symbols that still meet the pruning metric are updated.

In the next section it will be shown that should a priori information be incorporated into the MAP solution, then the branch metric is no longer monotonously increasing. Thus, a branch can no longer be pruned from the tree once its metric exceeds the current minimum.

3.7.2 A priori sphere decoder

The classical sphere decoder derives much of its decoding efficiency from the observation that the MAP metric without a priori information is monotonously increasing when calculated on a symbol-by-symbol basis from a triangular system. This permits the pruning of branches once the branch metric exceed the current best metric. This is no longer the case when a priori information is included. An alternative pruning criterion has to be developed to ensure efficient and optimal MAP decoding.

The MAP solution including a priori information is given by:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}} \left\{ \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{\sigma_n^2} - \frac{\boldsymbol{\lambda} \cdot \mathbf{b}}{2} \right\}, \tag{3.89}$$

where $\boldsymbol{\lambda}$ is the LLR of the a priori information and $\mathbf{b}$ is the bitwise data vector in a BPSK representation of $\mathbf{x}$.

Equation 3.89 can be partly transformed into a triangular system as in section 3.7.1:

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}} \left\{ \frac{\|\mathbf{Q}^{\mathcal{H}}\mathbf{y} - \mathbf{R}\mathbf{x}\|^2}{\sigma_n^2} - \frac{\boldsymbol{\lambda} \cdot \mathbf{b}}{2} \right\}, \tag{3.90}$$

where it should be noted that σ_n^2 can no longer be omitted from the calculations. Whilst eqn. 3.90 can still be represented as a tree structure, it is evident that the equation is no longer monotonously increasing. In the event that the current branch metric, T_n exceeds the minimum metric, $T_{\min}$ it is still possible that the remaining a priori information can result in the branch metric being lower than the current minimum. Therefore it is no longer possible to prune the branch the moment the branch metric exceeds the current minimum metric as was the case with the classical sphere decoder.

3.7.2.1 Pruning metric for the a priori sphere decoder

As shown, in section 3.7.2, it is necessary to modify the pruning metric to incorporate the a priori information in $\boldsymbol{\lambda}$. The pruning metric now only prunes a branch if the current branch metric exceeds the current minimum *and* the remaining a priori information is insufficient to change this. Thus the branch is pruned when:

$$T_n - \frac{1}{2}\sum_{i=n}^{N} \boldsymbol{\lambda}_i > T_{\min}, \tag{3.91}$$

where $\boldsymbol{\lambda}_i$ is the a priori information associated with the channel symbol x_i. Whilst the new pruning metric allows the sphere decoder to incorporate the a priori information, it potentially delays the pruning of branches potentially increasing the average number of computations. However, it is possible to re-arrange the tree such that the branches are pruned earlier as is discussed in the next section.

3.7.3 Algorithm details

The a priori sphere decoder is based on the implementation by [17]. It has been modified to use the QR decomposition instead of the Cholesky decomposition. The flow diagram for the a priori sphere decoder is shown in figure 3.1 on page 60. In the flow diagram three functions called `PRE-PROCESS()`, `SYMLIST()` and `UPDATE()` are used. Their specific details are described in this subsection.

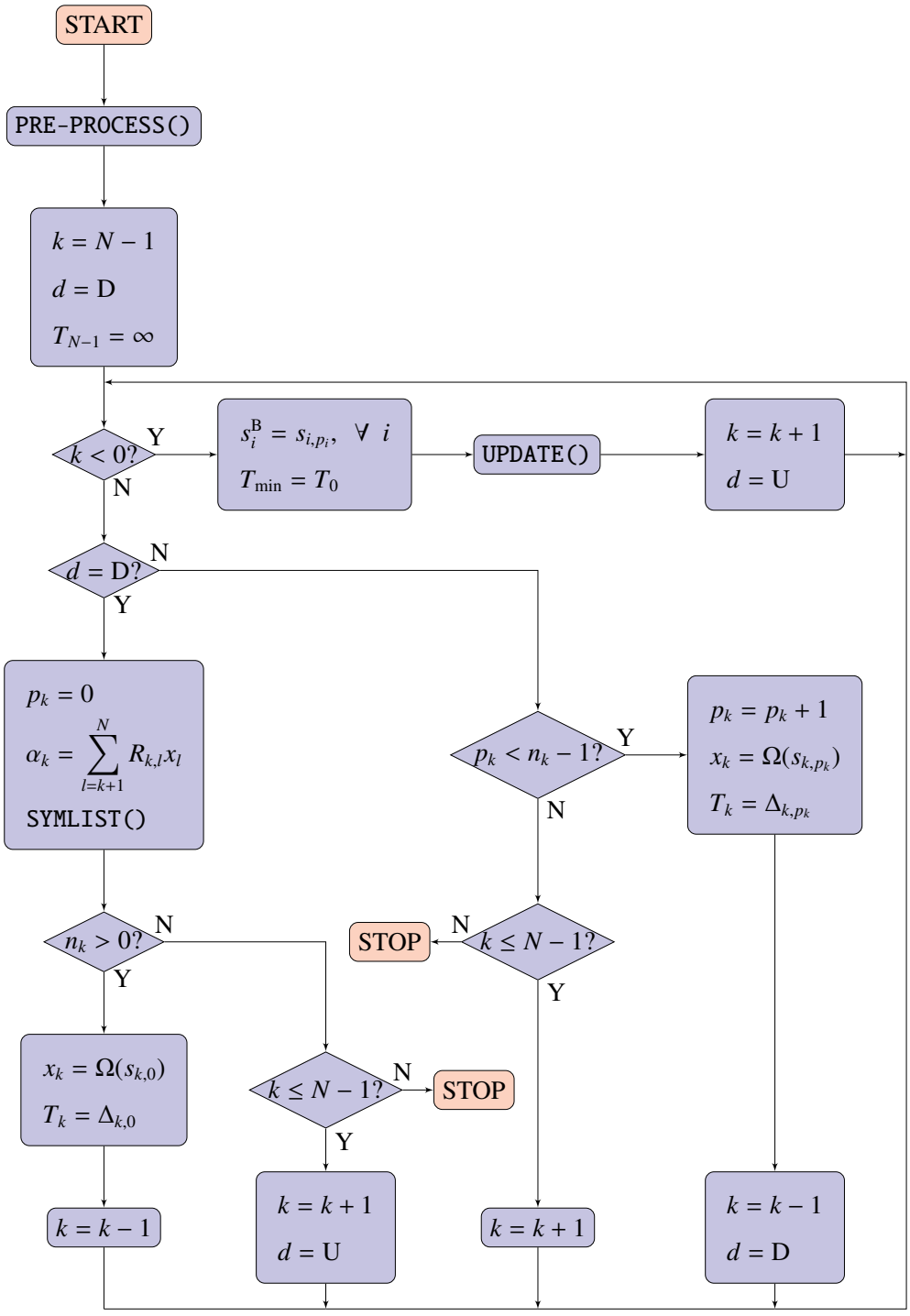

Figure 3.1: Flow diagram of the tree a priori search algorithm based on the QR decomposition.

3.7.3.1 PRE-PROCESS()

The `Pre-process` block is responsible for calculating the best symbol order to obtain the permutation matrix $\mathbf{P}$, the QR decomposition of $\mathbf{H}$ as well as the calculation of the cumulative LLRs λ^c:

for $i = 0$, to $N - 1$ **do**

$\lambda_i^c = \lambda_{i-1}^c + \sum_{j=0}^{N_b-1} |\lambda_{i \cdot N_b + j}|$

end for

$\mathbf{P} \leftarrow$ Optimal symbol order for decoding

$\mathbf{Q}, \mathbf{R} \leftarrow \text{qr}(\mathbf{H}\mathbf{P}^{-1})$

$\mathbf{y} \leftarrow \mathbf{Q}^{\mathcal{H}}\mathbf{y}$

where $\lambda_{-1} = 0$.

3.7.3.2 SYMLIST()

The `SYMLIST()` function calculates the possible symbols for the level and calculates their metrics. The symbols are then sorted in ascending order according to the metric. The variable n_k is then assigned the number of symbols that are smaller than the pruning metric:

for $i = 0$, to $Ns - 1$ **do**

$$\beta_i = \sum_{j=0}^{N_b-1} \lambda_{j+N_b k} \hat{b}_j^i$$

$$\Delta_i^k = \frac{||y_k - \alpha_k - R_{k,k}\Omega(i)||^2}{\sigma_n^2} - \frac{1}{2}\beta_i + \Delta_{p_{k+1}}^{k+1}$$

end for

`sort`(Δ^k, s^k)

$$n_k = \arg\max_i \left\{ \Delta_i^k < T_{\min} + \frac{1}{2}\lambda_k^c \right\}$$

$T_k = \Delta_0^k$

where $\hat{b}_j^i$ denotes the j^{th} bit represented by symbol i. The `sort()` function sort the symbols ascendantly according to their Δ_i^k metrics and stores the symbol order in s_i^k. The number of

symbols that are less than the pruning metric is stored in n_k.

The `SYMLIST()` function is the function where the vast majority of the calculations are performed. It is also called at each node in the tree. Thus, the number of times the `SYMLIST()` function is called is used as a performance metric in evaluating the relative performance of the sphere decoder with regard to various symbol ordering strategies. This allows the metric to be independent of the specific implementation architecture and programming.

In section 7.2 the number of times this function is called is used as a basis for comparing the effectiveness of the various symbols sorting methods proposed in section 3.8.

3.7.3.3 `UPDATE()`

The `UPDATE()` function uses the newly found minimum metric T_{min} to prune the tree. It iterates through the levels and updates the n_k value for each level by only keeping the symbols whose metrics are less than the new pruning metric:

for $k = 0$, to $N - 1$ **do**

$$n_k = \arg\max_j \left\{ \Delta_j^k < T_{\min} + \frac{1}{2}\lambda_k^c \right\}$$

end for

The next section, section 3.8, presents various metrics that can be used to determine the order in which the symbols are traversed in the tree. In section 7.2 it is shown that some of these metrics can be used to mitigate the increased computational complexity of the SD when a priori information is introduced.

3.8 RE-ORDERING OF SYMBOLS

It was previously mentioned, in sections 3.3 and 3.4, that the order in which the symbols are decoded has a significant effect on the BER performance of the ZF type decoders. In addition it was also mentioned that the order in which the SD traverses the tree can have an effect on the total number of nodes visited, see section 3.7. In this section various metrics

according to which the symbols can be ordered are presented. These are based on the a priori information and the channel matrix.

The decoders that depend on back substitution show that, in a manner similar to successive interference cancellation [15], starting the back substitution from the symbols with the highest SNR yields improved error rates. This is particularly evident in decoders employing decision feedback.

In the case of the sphere decoder it is known that having the diagonal of $\mathbf{R}$ ordered in ascending order, i.e. $|r_{n+1,n+1}| \geq |r_{n,n}|$, can speed up the sphere decoder [21]. Similarly the a priori sphere decoder can be sped up by ordering the symbols in the same manner.

Decoders using a priori information add an additional criterion that can be included in the determination of the symbol decoding order, the a priori information associated with each symbol. Three approaches to symbol ordering have been investigated in this dissertation: one based solely on the a priori information $\boldsymbol{\lambda}$, one based solely on the channel information $\mathbf{H}$ and the third based on a combination of a priori and the channel $\mathbf{H}$ information.

3.8.1 Re-ordering by a priori information

In the sphere decoder, the reasoning behind ordering the tree based on the a priori information, is the idea of reducing the impact of the remaining a priori information on the classical pruning metric. Thus, ideally the tree should be traversed in the *descending* order of λ_n, where λ_n is:

$$\lambda_n = \sum_{i=0}^{N_b-1} |\lambda_i|, \tag{3.92}$$

i.e. the sum of the absolute values of the LLRs of the bits of symbol n. In this manner the sum of absolute values of the LLRs diminishes rapidly as the tree is traversed by the decoder. Since the system being decoded has been transformed into an upper triangular system, this means that the channel symbols and their respective columns in $\mathbf{H}$ must be sorted based on the *ascending* values of λ_n, the total a priori information for symbol x_n.

For the case of the ZF and MMSE decoders, starting the decoding with the symbols that has

the most a priori information leads to a higher probability that the initials symbols will be decoded correctly. Since the decoding of subsequent symbols are dependant on the values of the preceding symbols, correctly decoding the latter reduces error in the decoding of the former.

3.8.2 Re-ordering by a priori information and channel information

It is known that performing the triangular transformation in such a way that the diagonal elements, of the upper triangular system, are increasing results in a speedup of the sphere decoder [21]. A sub-optimal approximation is to sort the column of $\mathbf{H}$ according to their Frobenius norms [14]. This corresponds to the sum of the *powers* of the individual paths for each symbol. The symbol that was transmitted through the strongest channel, i.e. highest power, should be the symbol with the highest SNR.

In an attempt to combine the ordering of the symbols with the norms of the columns of $\mathbf{H}$ with their ordering with respect to their a priori information, the following metric C_n, inspired by eqn. 3.90, is proposed:

$$C_n = \frac{1}{\sigma_n^2} \sum_{i=0}^{N_{RX}-1} |H_{i,n}|^2 + \frac{1}{2}\lambda_n. \tag{3.93}$$

The symbols are then ordered in ascending order according C_n. Equation 3.93 scales the contribution of the channel by the noise power. The first term is essentially an indicator of the SNR of symbol n. Thus, the MAP equation is used to determine the proportion of the contribution of the channel and a priori information respectively in determining the order of the symbols.

3.8.3 Sorted QR Decomposition

The sorted QR decomposition proposed in Wubben et al. [15] attempts to order the diagonal elements of $\mathbf{R}$ in increasing orders of magnitude. The algorithm is:

$\mathbf{R} = \mathbf{0}$, $\mathbf{Q} = \mathbf{H}$ & $\mathbf{P} = \mathbf{I}_{N_T}$

for $n = 0$ to $N_T - 1$ **do**

$k_n = \arg\min_{j=n,\cdots,N_T} |\mathbf{q}_j|^2$
exchange columns n and k_n in $\mathbf{Q}$, $\mathbf{R}$ and $\mathbf{P}$
$R_{n,n} = |\mathbf{q}_n|$
$\mathbf{q}_n = \mathbf{q}_n / R_{n,n}$
for $j = n + 1$, to $N_T - 1$ **do**
$R_{n,j} = \mathbf{q}_n^{\mathcal{H}} \mathbf{q}_j$
$\mathbf{q}_j = \mathbf{q}_j - R_{n,j}\mathbf{q}_n$
end for
end for

where $\mathbf{q}_n$ is the n^{th} column of matrix $\mathbf{Q}$. $\mathbf{P}$ is the permutation matrix by which the columns of $\mathbf{H}$ and the rows of $\mathbf{x}$ have been permuted.

The diagonal of $\mathbf{R}$ represents the channel power associated with the symbols. The other terms in the matrix represent the interference of the other symbols on the current symbol. Thus, ideally the diagonal elements should be larger than the interference terms. Ensuring that the symbols with the largest diagonal elements also have the least amount of interference terms should ensure that those symbols are more likely to be correctly decoded. The probability of correctly decoding subsequent symbols is also increased if the previous symbols were correctly decoded.

3.8.4 Modified Sorted QR Decomposition

The standard SQRD does not make use of information other than the channel matrix $\mathbf{H}$. The algorithm was therefore modified to include a priori information in its sorting stage, and is thus given as:

$\mathbf{R} = \mathbf{0}$, $\mathbf{Q} = \mathbf{H}$ & $\mathbf{P} = \mathbf{I}_{N_T}$
for $n = 0$ to $N_T - 1$ **do**
$\lambda_k^s = \sum_{k=0}^{N_b} |\lambda_{nN_b+k}|$
end for
for $n = 0$ to $N_T - 1$ **do**

$$k_n = \arg\min_{j=n,\cdots,N_T} \left\{ \frac{\|\mathbf{q}_j\|^2}{\sigma_n^2} + \frac{1}{2}\lambda_j^s \right\}$$

exchange columns n and k_n in $\mathbf{Q}$, $\mathbf{R}$, $\mathbf{P}$ and $\boldsymbol{\lambda}^s$

$R_{n,n} = |\mathbf{q}_n|$

$\mathbf{q}_n = \mathbf{q}_n / R_{n,n}$

for $j = n + 1$, to $N_T - 1$ **do**

$R_{n,j} = \mathbf{q}_n^{\mathcal{H}} \mathbf{q}_j$

$\mathbf{q}_j = \mathbf{q}_j - R_{n,j}\mathbf{q}_n$

end for

end for

where σ_n^2 is the noise variance and N_b is the number of bits per symbol. This sorting strategy is similar to the suboptimal $\mathbf{H}$ & $\boldsymbol{\lambda}$ strategy, except that the sorted QR decomposition is used as it provides optimal ordering when based solely on channel information. This symbol sorting method ensures that the MAP symbol, including a priori information, is back substituted in the decision feedback decoders.

The effect of the various sorting metric on the decoder performance can be seen in sections 7.1.5 and 7.2 where significant performance benefits are evident. These sorting metrics may also be of use in the HNN of the following section (section 3.9) if sequential updating is used instead of batch updating of the neurons. This was, however, not investigated as part of this dissertation.

3.9 HOPFIELD NEURAL NETWORK EQUALISATION

Hopfield and Tank [33] proposed a method of using a HNN to perform channel equalisation. Myburgh [34] showed the advantages of using HNN in extremely long channels at reduced complexity. In this section it is shown how to use the energy function of the HNN to perform MIMO STFC decoding. The performance of the HNN was found to be inferior to the ZF decoder. It was therefore used in the SD-HNN decoder of section 3.11 to calculate the counter hypothesis to provide soft outputs to the SD.

3.9.1 Introduction to Hopfield Neural Networks

Hopfield neural networks (HNN), consist of fully interconnected neurons, as shown in figure 3.2, These neurons compute an output signal, $v_i(t)$, from an input signal, $u_i(t)$ as:

$$v_i(t) = g[u_i(t)], \tag{3.94}$$

where i is an indice denoting the neuron, and $g(x)$ is a smooth monotonically increasing sigmoid function with the following requirements:

$$\begin{aligned} \lim_{x\to-\infty} g(x) &= 0 \\ g(0) &= \frac{1}{2} \\ \lim_{x\to\infty} g(x) &= 1. \end{aligned} \tag{3.95}$$

The Hopfield Neural Network follows the following equation [33]:

$$\frac{du_i(t)}{dt} = -\frac{u_i(t)}{\tau} + \sum_{j=1}^{N} T_{ij}v_j(t) + I_i, \tag{3.96}$$

where τ is a constant that can be set to unity, I_i is a constant bias added to the input of the neuron, T_{ij} is the interconnection weight between neurons i and j, and N is the number of neurons. The interconnection weights form a symmetrical connectivity matrix, $\mathbf{T}$, with zeros on the diagonal. This is because the HNN does not have the same neuron connected to itself. Thus:

$$\begin{aligned} T_{ij} &= T_{ji} \\ T_{ii} &= 0, \ \forall\ i. \end{aligned} \tag{3.97}$$

Hopfield has shown in [33] that when eqn. 3.97 are satisfied, and when the width of $g(x)$ is narrow, the high-gain limit, the stable states of the network are the local minima of:

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}v_i v_j - \sum_{i=1}^{N} v_i I_i \tag{3.98}$$

$$= -\frac{1}{2}\mathbf{v}\mathbf{T}\mathbf{v}^{\mathcal{T}} - \mathbf{v}\mathbf{I}^{\mathcal{T}}. \tag{3.99}$$

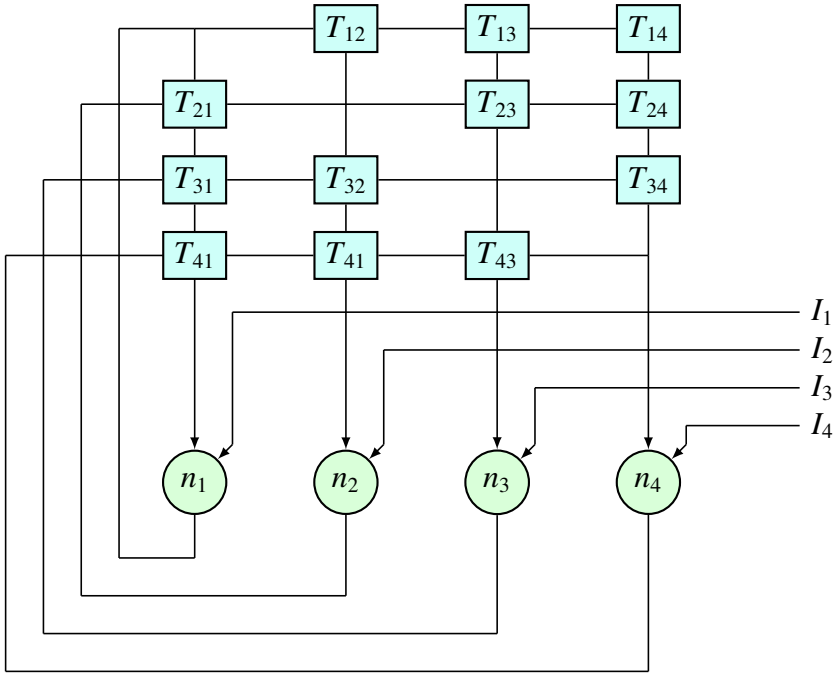

Figure 3.2: Diagram of a four node Hopfield Neural Network.

Thus, the HNN can be used to minimise functions that can be written in the form of the energy function of (3.98). Thus, to perform the decoding of the MIMO channel with the HNN, the decoding problem has to be written in a form similar to that of the energy function. The mapping of the MIMO MAP decoding problem to the energy function of the HNN is presented in the next subsection.

3.9.2 Equalisation with Hopfield Neural Networks

MAP MIMO STFC decoding using the HNN can be performed by reformulating the MAP metric in terms of the energy function of the HNN. This can be done by expanding the MAP metric and comparing the terms to those of the energy function as will be shown.

The MAP problem, ignoring a priori information, is given by:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ (\mathbf{y} - \mathbf{H}\mathbf{x})^{\mathcal{T}} (\mathbf{y} - \mathbf{H}\mathbf{x}) \right\}, \tag{3.100}$$

where it is assumed that real signals are used. Equation 3.100 can be expanded as:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ \mathbf{y}^{\mathcal{T}}\mathbf{y} - \mathbf{y}^{\mathcal{T}}\mathbf{H}\mathbf{x} - \mathbf{x}^{\mathcal{T}}\mathbf{H}^{\mathcal{T}}\mathbf{y} + \mathbf{x}^{\mathcal{T}}\mathbf{H}^{\mathcal{T}}\mathbf{H}\mathbf{x} \right\}. \tag{3.101}$$

As each term in 3.101 is a scaler, and since the first term is independent of $\mathbf{x}$ it can be simplified to:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ -2\mathbf{y}^{\mathcal{T}}\mathbf{H}\mathbf{x} + \mathbf{x}^{\mathcal{T}}\mathbf{H}^{\mathcal{T}}\mathbf{H}\mathbf{x} \right\}. \tag{3.102}$$

Comparing the terms in eqn. 3.99 and eqn. 3.102 one can observe that the term $\mathbf{H}^{\mathcal{T}}\mathbf{H}$ will be associated with the connection matrix $\mathbf{T}$. The only problem being that the diagonal of $\mathbf{H}^{\mathcal{T}}\mathbf{H}$ is not zero as is required of $\mathbf{T}$.

Hopfield and Tank [33] addressed this issue by adding a constraint term that is equal to zero for all valid solutions to the problem and positive otherwise. This term is given by:

$$C_t = -\sum_{i=0}^{N-1} \mathbf{H}_i^{\mathcal{T}}\mathbf{H}_i(x_i^2 - 1), \tag{3.103}$$

where $\mathbf{H}_i$ denotes the i^{th} column in matrix $\mathbf{H}$. Next, it will be shown that simply using $-2\mathbf{H}^{\mathcal{T}}\mathbf{H}$ and setting to zero the diagonal of the resultant matrix for $\mathbf{T}$, corresponds to the constrained HNN.

Using the following relation:

$$\mathbf{T} = -2\mathbf{H}^{\mathcal{T}}\mathbf{H} \tag{3.104}$$

$$\text{diag}(\mathbf{T}) = \mathbf{0}, \tag{3.105}$$

and adding the constraint term, C_t in eqn. 3.103, eqn. 3.102 can be written as:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ -2\mathbf{y}^{\mathcal{T}}\mathbf{H}\mathbf{x} - \frac{1}{2}\mathbf{x}^{\mathcal{T}}\mathbf{T}\mathbf{x} + \sum_{i=0}^{N-1} \mathbf{H}_i^{\mathcal{T}}\mathbf{H}_i x_i^2 - \sum_{i=0}^{N-1} \mathbf{H}_i^{\mathcal{T}}\mathbf{H}_i(x_i^2 - 1) \right\}, \tag{3.106}$$

where the second term accounts for the missing diagonal elements of $\mathbf{H}^{\mathcal{T}}\mathbf{H}$ in $\mathbf{T}$. Expanding the constraint term results in:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ -2\mathbf{y}^{\mathcal{T}}\mathbf{H}\mathbf{x} - \frac{1}{2}\mathbf{x}^{\mathcal{T}}\mathbf{T}\mathbf{x} + \sum_{i=0}^{N-1} \mathbf{H}_i^{\mathcal{T}}\mathbf{H}_i x_i^2 \right. \tag{3.107}$$

$$\left. - \sum_{i=0}^{N-1} \mathbf{H}_i^{\mathcal{T}}\mathbf{H}_i x_i^2 + \sum_{i=0}^{N-1} \mathbf{H}_i^{\mathcal{T}}\mathbf{H}_i \right\}, \tag{3.108}$$

where the third and fourth terms cancel each other and the last term is a constant that does not change the result. From this it is evident that the following relations are obtained for

solving the MIMO decoding problem with the HNN:

$$\mathbf{T} = -2\mathbf{H}^{\mathcal{T}}\mathbf{H} \tag{3.109}$$

$$\mathbf{I} = 2\mathbf{y}^{\mathcal{T}}\mathbf{H}. \tag{3.110}$$

Thus, by setting the connectivity matrix $\mathbf{T}$ and the node bias vector $\mathbf{I}$ to the results of eqn. 3.109 and eqn. 3.110 and running the HNN, will cause the nodes of the HNN to converge to the MAP solution.

3.9.3 Annealing the Hopfield Neural Network

It is known that annealing greatly improves the performance of the HNN [62]. In annealing the gain of the neurons is slowly increased from a low starting point. In addition the contribution from the bias input $\mathbf{I}$ is also scaled by a factor that is a function of the iteration number [63, 64].

In addition a *momentum* term is included that provides a memory effect for the neurons. That is, a scaled version of the previous neuron output is used in the calculation of the current value [62]:

$$\mathbf{u}_{i+1} = \alpha(i)\mathbf{u}_i + \mathbf{T}\mathbf{v}_i + \gamma(i)\mathbf{I} \tag{3.111}$$

where $\alpha(i)$ and $\gamma(i)$ are time variant functions that, together with the gain $\beta(i)$ had the form:

$$\alpha(i) = c_0 + c_1 \exp(c_2 i + c_3), \tag{3.112}$$

where c_i are constants. These constants were obtained using a genetic optimisation algorithm to minimise the average MSE $|\mathbf{y} - \mathbf{H}\mathbf{x}|$ over a large E_B/N_0 range.

For use in the soft-to-hard conversion, the initial states were loaded with the hard decoded output, except for the current bit being soft-computed which was always set to the opposite to its hard decoded value.

3.9.4 Including a priori information in the Hopfield Neural Network

It is trivial to include a priori information in the HNN with the RVT method if the arc tan function is used as the sigmoid function. It is, however, possible to avoid performing the additional computations involved in calculating the RVT by reformulating the HNN decoder to include a priori information.

The MAP decoding problem (see section 3.1) for the HNN including a priori information is given by:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ (\mathbf{y} - \mathbf{H}\mathbf{x})^{\mathcal{T}} (\mathbf{y} - \mathbf{H}\mathbf{x}) - \sigma_n^2 \mathbf{b} \cdot \boldsymbol{\lambda} \right\}, \tag{3.113}$$

where the equation has been multiplied by σ_n^2 and real valued signals are assumed. Furthermore, assuming that BPSK signalling is used, the following is obtained:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ (\mathbf{y} - \mathbf{H}\mathbf{x})^{\mathcal{T}} (\mathbf{y} - \mathbf{H}\mathbf{x}) - \sigma_n^2 \mathbf{x} \cdot \boldsymbol{\lambda} \right\}. \tag{3.114}$$

Following a similar approach to section 3.9.2 the following is obtained:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \left\{ (-2\mathbf{y}^{\mathcal{T}}\mathbf{H} - \sigma_n^2 \boldsymbol{\lambda})\mathbf{x} - \frac{1}{2}\mathbf{x}^{\mathcal{T}}\mathbf{T}\mathbf{x} + \sum_{i=0}^{N-1} \mathbf{H}_i^{\mathcal{T}} \mathbf{H}_i \right\}, \tag{3.115}$$

where the third term may be ignored in the minimisation. Thus, the following is obtained:

$$\mathbf{T} = -2\mathbf{H}^{\mathcal{T}}\mathbf{H} \tag{3.116}$$

$$\mathbf{I} = 2\mathbf{y}^{\mathcal{T}}\mathbf{H} + \sigma_n^2 \boldsymbol{\lambda}, \tag{3.117}$$

where it is evident that only the neuron bias vector has changed in order to incorporate the a priori information.

3.10 RECEIVE VECTOR TRANSLATION

RVT was developed as an attempt to use the sphere decoder with a priori information [20]. The sphere decoder relies on the fact that the metric it computes is monotonously increasing.

Once this metric exceeds the search radius, the current branch being searched is pruned and the other nodes in the tree are searched instead. Adding a priori information, such as was done in eqn. 3.80, causes the metric to no longer be monotonously increasing. This can cause an increase in the number of computations required.

An attempt was made by Wang and Giannakis [20] to add the a priori information to the decoder by adding a translation vector $\tilde{\mathbf{y}}$ to the receive signal $\mathbf{y}$. Thus the decoder metric becomes:

$$\hat{\mathbf{s}} = \arg\min_{\mathbf{x}} \left\{ \frac{\|\mathbf{y} + \tilde{\mathbf{y}} - \mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2} \right\}, \tag{3.118}$$

where the denominator may be left out. The following derivation for the value of $\tilde{\mathbf{y}}$ was published by Botha and Maharaj [65].

The translation vector, $\tilde{\mathbf{y}}$, can be obtained by equating eqn. 3.118 with the classical a priori MAP metric:

$$\frac{\|\mathbf{y} + \tilde{\mathbf{y}} - \mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2} = \frac{\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2}{2\sigma_n^2} - \frac{1}{2}\mathbf{b} \cdot \boldsymbol{\lambda}^a. \tag{3.119}$$

The term in the numerator may be expanded as:

$$\|\mathbf{y} + \tilde{\mathbf{y}} - \mathbf{H}\mathbf{x}\|^2 = (\mathbf{y} + \tilde{\mathbf{y}} - \mathbf{H}\mathbf{x})^{\mathcal{H}}(\mathbf{y} + \tilde{\mathbf{y}} - \mathbf{H}\mathbf{x}). \tag{3.120}$$

Performing the same operation on the right hand side and removing common terms yields the following:

$$\tilde{\mathbf{y}}^{\mathcal{H}}\mathbf{y} + \tilde{\mathbf{y}}^{\mathcal{H}}\tilde{\mathbf{y}} - \tilde{\mathbf{y}}^{\mathcal{H}}\mathbf{H}\mathbf{x} - \mathbf{x}^{\mathcal{H}}\mathbf{H}^{\mathcal{H}}\tilde{\mathbf{y}} = -\sigma_n^2\mathbf{b}^{\mathcal{T}}\boldsymbol{\lambda}^a. \tag{3.121}$$

The terms $\tilde{\mathbf{y}}^{\mathcal{H}}\mathbf{y}$ and $\tilde{\mathbf{y}}^{\mathcal{H}}\tilde{\mathbf{y}}$ may be ignored as they are constants and neglecting them will only cause a constant difference in the final metrics. The argument, $\mathbf{x}$, that minimises the metric will be the same. The decoded signal will still be the same, thus yielding:

$$\sigma_n^2\mathbf{b}^{\mathcal{T}}\boldsymbol{\lambda}^a = \tilde{\mathbf{y}}^{\mathcal{H}}\mathbf{H}\mathbf{x} + \mathbf{x}^{\mathcal{H}}\mathbf{H}^{\mathcal{H}}\tilde{\mathbf{y}} \tag{3.122}$$

$$= (\mathbf{x}^{\mathcal{H}}\mathbf{H}^{\mathcal{H}}\tilde{\mathbf{y}})^{\mathcal{H}} + \mathbf{x}^{\mathcal{H}}\mathbf{H}^{\mathcal{H}}\tilde{\mathbf{y}}. \tag{3.123}$$

Since both sides of 3.123 are scalar the following is obtained:

$$\sigma_n^2\mathbf{b}^{\mathcal{T}}\boldsymbol{\lambda}^a = (\mathbf{x}^{\mathcal{H}}\mathbf{H}^{\mathcal{H}}\tilde{\mathbf{y}})^* + \mathbf{x}^{\mathcal{H}}\mathbf{H}^{\mathcal{H}}\tilde{\mathbf{y}}. \tag{3.124}$$

Further, assuming that $\mathbf{x}$, $\mathbf{H}$ and $\tilde{\mathbf{y}}$ are real, the following is obtained:

$$\sigma_n^2 \mathbf{b}^{\mathcal{T}} \boldsymbol{\lambda}^a = (\mathbf{x}^{\mathcal{T}} \mathbf{H}^{\mathcal{T}} \tilde{\mathbf{y}}) + \mathbf{x}^{\mathcal{T}} \mathbf{H}^{\mathcal{T}} \tilde{\mathbf{y}} \tag{3.125}$$

$$= 2\mathbf{x}^{\mathcal{T}} \mathbf{H}^{\mathcal{T}} \tilde{\mathbf{y}}. \tag{3.126}$$

Assuming that BPSK signalling is used then eqn. 3.126 can be simplified to:

$$\sigma_n^2 \boldsymbol{\lambda}^a = 2\mathbf{H}^{\mathcal{T}} \tilde{\mathbf{y}}, \tag{3.127}$$

which may be solved for $\tilde{\mathbf{y}}$ by inverting $\mathbf{H}^{\mathcal{T}}$, using the pseudo inverse, QR decomposition or Singular Value Decomposition (SVD) [14]. Since the mapping from bits to channel symbols is generally non-linear, it will be difficult to extend this method to higher modulation schemes. This method is also limited to real valued systems, though using the complex-to-real conversion complex systems may also be decoded [54]:

$$\begin{bmatrix} \Re\{\mathbf{y}\} \\ \Im\{\mathbf{y}\} \end{bmatrix} = \begin{bmatrix} \Re\{\mathbf{H}\} & -\Im\{\mathbf{H}\} \\ \Im\{\mathbf{H}\} & \Re\{\mathbf{H}\} \end{bmatrix} \begin{bmatrix} \Re\{\mathbf{x}\} \\ \Im\{\mathbf{x}\} \end{bmatrix}. \tag{3.128}$$

The RVT method can be used to add a priori information to all decoders that only use the channel matrix $\mathbf{H}$ and the received vector $\mathbf{y}$. This method can therefore also be used to include a priori information in the ZF and MMSE decoders, with or without DF, the sphere decoder and the HNN. As mentioned the method does have the limitation of requiring a linear mapping from channel symbols to binary information. This limitation implies that RVT is limited to BPSK signal constellations; though it is possible to represent some QPSK signal constellations as BPSK signal constellations.

3.10.1 Using RVT for the ZF type decoders

As mentioned the RVT method can be used to include a priori information in most decoders; however, this may include additional complexity. This additional complexity arrises from the fact that an additional system of linear equations, that yields the value of $\tilde{\mathbf{y}}$, need to be solved. Additionally, for complex valued systems, the required usage of the complex-to-real matrix transform imposes additional computational requirements. These additional requirements are quantified in the rest of this section. This comparison was published by Botha and Maharaj [65].

3.10.1.1 Real valued systems

In the receive vector translation method, for the case of BPSK signalling and real valued noise and channel matrices and vectors, the difference in complexity is mainly due to the fact that the translated vector, $\tilde{\mathbf{y}}$, must be calculated in addition to the transmitted symbols. Using the QR decomposition method to solve eqn. 3.127 for $\tilde{\mathbf{y}}$, the receive vector translation effectively requires two QR decompositions. The QR decomposition requires approximately $\frac{10}{3}n^3$ [66, 67] flops with the back substitution requiring an additional n^2 flops. The overall computations for RVT is approximately $\frac{20}{3}n^3 + 2n^2$ flops.

The symbol-by-symbol method only requires a single QR decomposition and back substitution, thus it only requires half the computations of the receive vector translation method, and can thus be expected to be twice as fast for real valued systems.

3.10.1.2 Complex valued systems

In the case of complex valued systems, use of the complex-to-real transform is required for the receive vector translation. This quadruples the space required to store the system of equations. Furthermore, since the order of the QR decomposition (using reflectors [67]) is $O(n^3)$ and since the resultant real valued matrix is twice the size of the original complex valued matrix, it can be expected that the real QR decomposition will be eight times slower than the complex valued decomposition in terms of flops required.

However, complex mathematics requires more computations than real maths. The calculation of the product of two real numbers requires only a single flop. The number of flops required for the multiplication of two complex numbers is six flops, four multiplies and two additions. The complex QR decomposition requires approximately $\frac{16}{3}n^3$ flops [66], and is thus approximately 1.8 times slower than the equivalently sized real QR decomposition. Using the complex-to-real transform and performing the QR decomposition is thus approximately four times slower than directly computing the complex QR decomposition. This is due to the doubling of the matrix size when performing the complex-to-real conversion having a greater impact that the additional complexity required by complex mathematics.

It should also be kept in mind that the complex number back substitution will be slower than the real version. Real valued back substitution requires approx $\frac{n^2}{2}$ multiplications and $\frac{n^2}{2}$ additions for a total of n^2 flops [67]. Since complex multiplication requires six flops and complex addition requires two flops per operation, the total flops for complex number back substitution is approximately $\frac{6}{2}n^2 + n^2 = 4n^2$.

Comparing RVT to the symbol-by-symbol method shows that the ratio of flops required as the size of the system tends to infinity goes to 10. The symbol-by-symbol method is therefore up to ten times faster.

Figure 3.3 is a plot of the ratio of the approximate flops required for the two methods assuming complex systems. In real valued system the ratio will be a constant factor of two. The figure also shows that the limit, as n tends to infinity, approaches ten.

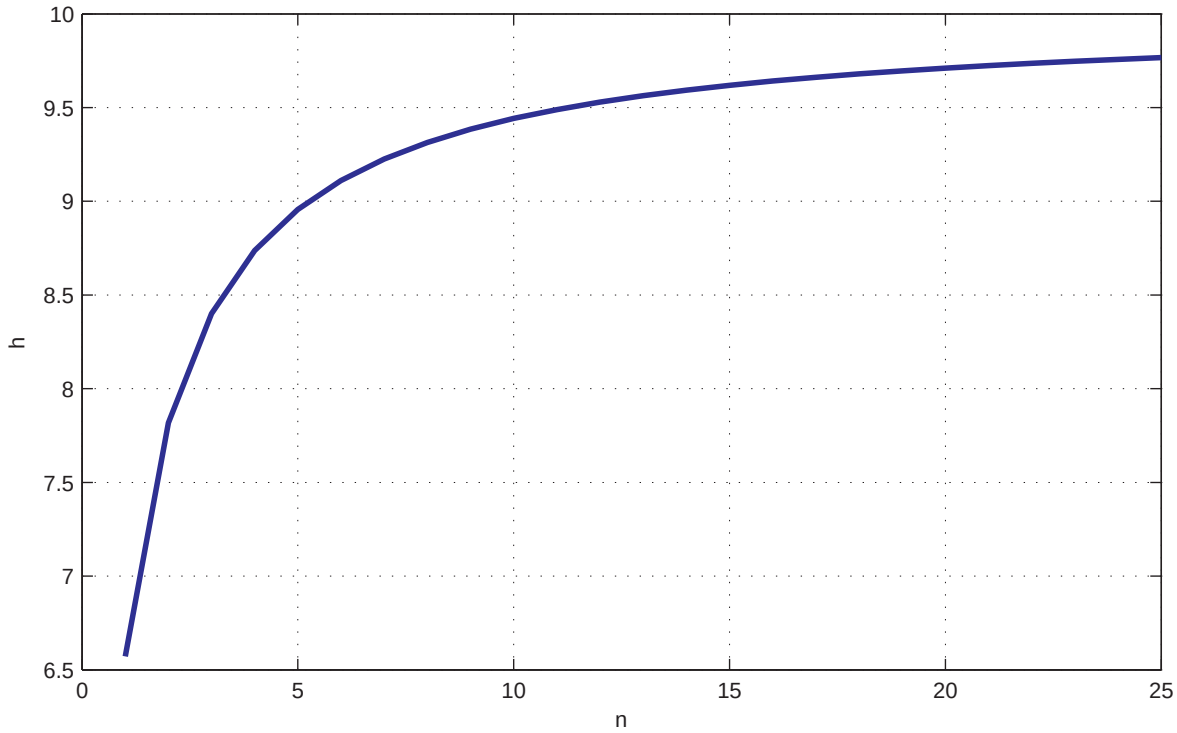

Figure 3.3: Plot of the ratio of the approximate flops required for the Vector Translation Method and the BM method (η) for different sized complex matrices (n).

3.11 REDUCED COMPLEXITY SOFT OUTPUT SPHERE DECODER

The HNN, together with RVT, was used in a turbo structure to reduce the complexity of the classical soft output sphere decoder [63, 64]. The soft output sphere decoder that is used to decode N_s symbols containing a total of N_b bits, needs to run the tree search algorithm $N_b + 1$ times. The iteration is to obtain the hard-output values of the bits. The subsequent iterations are to obtain soft-output values based on the MAP solution *given* that the bit, for which the soft value is being computed, has the opposite value as the hard output MAP solution.

Each soft-output tree search does not necessarily have the same complexity. One of the reductions that can be made is apparent in the tree structure itself. By calculating the soft-output values in reverse order, not all of the levels in the tree needs to be traversed for all of the symbols. Only the levels up to the level containing the current bit need be traversed. Nevertheless, it still involves many iterations.

Wang and Giannakis [20] proposed to simply use bit-inversion where instead of the MAP solution, where the bit has the opposite value to the hard-output value, the MAP metric for the anti-hypothesis is simply calculated using the opposite value of the bit. This is computationally efficient, as no additional decoding steps are performed, although it leads to soft values that are over optimistic. However, when higher order modulation schemes are used it is no longer as straightforward because there are many possible symbols with the bit value being opposite to the hard-output value.

Whilst the reduction in complexity is substantial, there is a corresponding decrease in the accuracy of the soft-outputs. This has the effect of increasing the BER at the output of a FEC employing soft decision decoding.

The HNN, whilst not providing the best BER performance, is computationally less complex than the sphere decoder and should perform better than simple *bit-flipping* [63, 64]. However, with the QR decomposition based sphere decoder it is trivial to perform the hard-to-soft calculation using a ZF-DF decoder. Additionally the tree structure can be exploited to reduce the required complexity even further by not having to traverse the entire tree for each bit. The

ZF-DF decoder also gives better results than the HNN.

In summary, in this chapter the MAP decoding metric incorporating a priori information was derived. It was shown that the metric differs slightly when the noise is real and when the noise is complex when a priori information is used. The calculation of the MAP metric, eqn. 3.63, was then performed using the ZF and MMSE decoders. It was shown that by using a triangular transform a priori information and DF can be implemented for these decoders. Furthermore, it was shown how the SD can be used to perform the decoding and how the SD can be extended to use a priori information was presented and described.

A competing method for including a priori information in the decoding problem based on receive vector translation was also described and shown to be limited to signal constellations having a linear mapping between channel symbols and binary data symbols.

Finally, different strategies for sorting the symbol decoding order was detailed. These strategies use a combination of a priori and channel information to sort the symbols. For the back substitution decoder these leads to performance gains whilst for the SD it leads to reduced computations.

CHAPTER 4

FORWARD ERROR CORRECTING CODES

In the previous chapters, methods for decoding of MIMO STFCs were presented. It was shown how these decoders can be adapted to make use of a priori information to perform MAP decoding. In this dissertation the a priori information provided to the MIMO STFC decoders is obtained from the outer FEC code with which the MIMO STFCs are concatenated with (see fig. 6.1, page 116). Together the STFC MIMO decoder and the FEC decoder form a turbo decoding structure (see fig. 6.2, page 117) that permits iterative turbo decoding.

This chapter starts with an introduction to the fundamentals of coding theory, describing how coding theory aids in achieving reliable communication. FEC codes, specifically linear block codes, that allow for the detection and correction of bit errors at the receiver, are discussed. The decoding of linear block codes is described as a marginalisation problem that can be more efficiently solved using factorisation and the message passing decoder.

LDPC codes, a type of linear block code, are described and the BER performance of the LDPC codes used in this dissertation is given. Furthermore, the ability of an LDPC code to make use of diversity in fading channels is briefly investigated. Finally, the effect of hard limiting the soft values to the LDPC decoder on the BER performance is investigated.

4.1 FUNDAMENTALS OF CODING THEORY

In a communications system information is transferred between two or more entities. The medium through which the information is transferred between the entities is called the *channel*. Were the channel ideal, no loss or corruption of information would occur during the transfer through the channel. Unfortunately most, if not all, channels deviate from the ideal channel and thus change the information transmitted through them. In digital communication systems these changes normally manifest themselves as bit errors at the receiver. For a reliable communication system it is desirable to be able to, at least, detect these errors or, ideally, to be able to correct these bit errors. FEC codes enable the detection and correction of erroneously received data at the receiver.

FEC codes are able to detect and correct bit errors at the receiver by adding redundant information to the original information that can be used to identify and correct received errors. The *rate*, r, of an FEC code is used as a metric to quantify the amount of redundancy added to the information. This measure is given by:

$$r = k/n, \tag{4.1}$$

where k is the number of original information bits and n is the number of original information bits together with redundant data bits sent.

There is an upper-bound on the maximum rate of a code given a certain transmission rate, R, and the capacity of the channel, C. This bound is known as the *Noisy Channel Coding Theorem* or alternatively as *Shannon's Theorem* [68]:

Theorem 4.1. *[Noisy Channel Coding Theorem] There exists channel codes that make it possible to achieve reliable communication, with as small an error probability as desired, if the transmission rate, R, is less than the channel capacity, C. If $R > C$, there is no method to make the probability of error tend to zero with any code.*

One such category of channel codes is that of FEC codes. FEC codes add redundant information at the receiver that allows the receiver to identify and correct a number of errors. FECs differ primarily in the manner in which they compute and add the redundant information, i.e.

the manner in which they *encode* the data. The two main categories into which FEC codes fall are: convolutional codes and block codes [69].

Convolutional codes encodes streams of continuous data on a symbol-by-symbol basis, whilst block codes operate on discrete *blocks* of data. In practise only finite sequences of data are transmitted, implying that convolutional codes must be terminated at a certain time. This enables convolutional codes to be represented as block codes, albeit as rather large block codes.

In classical coding theory, a very important measurement is the Hamming distance. The Hamming distance of a code is a useful measure that characterises the error correcting ability of a code. It is defined as follows [69, def. 1.11]:

Definition 4.1 (Hamming distance and Hamming weight)**.** *Let $u, v \in \mathbb{F}^n$. The Hamming weight of a word u, denoted by $w_H(u)$, is equal to the number of non-zero symbols in u, i.e., the cardinality of the support set. The Hamming distance of a pair (u, v), denoted by $d_H(u, v)$, is the number of positions in which u differs from v. $d_H(u, v) = d_H(u - v, 0) = w_H(u - v), d_H(u, v) = d_H(v, u)$ and $d_H(u, v) \geq 0$, with equality if and only if $u = v$. Also, $d_H(\cdot, \cdot)$ satisfies the triangle inequality:*

$$d_H(u, v) \leq d_H(u, t) + d_H(t, v), \tag{4.2}$$

for any triple $u, v, t \in \mathbb{F}^n$. In other words, $d(\cdot, \cdot)$ is a true distance in the mathematical sense.

In bounded distance decoding, a linear block code can correct up to $\frac{d_{min}-1}{2}$ errors [70, p. 98] or alternatively detect up to $d_{min} - 1$ errors, where d_{min} is the minimum Hamming distance between any two codewords. It should be noted that not all codeword pairs have a Hamming distance equal to the minimum Hamming distance. Some pairs have a Hamming distance much larger than the minimum and are thus able to correct or detect more errors. For decoders that are able to exploit this property the distribution of the Hamming distance become an important factor. Richardson and Urbanke [69, p. 12] show that bounded distance decoding algorithms can not achieve capacity.

Block codes will be used in this dissertation specifically the class of *linear block* codes.

4.2 LINEAR BLOCK CODES

The definition of a linear code is given by Lin [71, p. 34] as:

Definition 4.2 (Linear Block Code)**.** *A set of 2^k n-tuples is called a linear code if and only if it is a subspace of the vector space $\mathbb{F}_q^n$ of all n-tuples.*

Definition 4.2 of a linear code implies that a set of 2^k codewords, or code vectors, can be described by a set of k linearly independent basis vectors that may be arranged into a $k \times n$ matrix:

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_k \end{bmatrix} = \begin{bmatrix} g_{01} & g_{02} & g_{03} & \cdots & g_{0n} \\ g_{11} & g_{12} & g_{13} & \cdots & g_{1n} \\ \vdots & \vdots & \vdots & & \vdots \\ g_{k1} & g_{k2} & g_{k3} & \cdots & g_{kn} \end{bmatrix}, \tag{4.3}$$

where $\mathbf{G}$ is referred to as the *generator matrix*. The generator matrix provides a simple method for obtaining the codeword for a particular message, $\mathbf{s}$:

$$\mathbf{x} = \mathbf{G}^{\mathcal{T}}\mathbf{s}, \tag{4.4}$$

where $\mathbf{s}$ is a column vector of length k. Thus $\mathbf{G}$ is a mapping of $\mathbf{s}$ onto $\mathbf{x}$. This mapping is not unique as a different set of basis vectors may give rise to a different mapping. However, the performance of the code is not affected by the specific mapping used. A particular form of generator matrix is that which results in a *systematic code*, where the first k bits of the codeword is a direct mapping of the original information bits [69]. This has two benefits:

1. After decoding, the information bits are trivially extracted; and

2. In high noise environments where the decoding operation introduces additional errors, the information can still be extracted with greater reliability than with a non-systematic code.

The generator matrix representation of a systematic code has the form:

$$\mathbf{G}_s = [\mathbf{I}_k | \mathbf{P}_{k\times(n-k)}], \tag{4.5}$$

where $\mathbf{I}_k$ is a $k \times k$ identity matrix and $\mathbf{P}$ is a $k \times (n-k)$ parity matrix.

From Definition 4.2 linear codes are characterised by two properties:

1. Any combination of two codewords results in a new codeword; and

2. Multiplying any codeword by an element in $\mathbb{F}_q$ produces a new codeword.

Using these two properties, a non-systematic generator matrix can usually be transformed into a systematic generator matrix through elementary matrix row operations.

These two properties also imply that the all-zero codeword is a valid codeword in a linear code. As the all zero code word has a Hamming weight of zero, $w_H((0)) = 0$, the Hamming distance, d_H, for a linear code, C, can easily be obtained. The Hamming distance is given by [69]:

$$d_{min}(C) = \min_{\mathbf{x}\in C, \mathbf{x}\neq\mathbf{x}_0} \{w_H(\mathbf{x} - \mathbf{x}_0)\} \tag{4.6}$$

$$= \min_{\mathbf{x}\in C, \mathbf{x}\neq\mathbf{0}} \{w_H(\mathbf{x} - \mathbf{0})\} \tag{4.7}$$

$$= \min_{\mathbf{x}\in C, \mathbf{x}\neq\mathbf{0}} \{w_H(\mathbf{x})\} \tag{4.8}$$

For the decoding of a linear block code it is useful to define the dual code, $C^{\perp}$ consisting of all the codeword in $\mathbb{F}_q^n$ that are orthogonal to C. In linear block codes the generator matrix represents the entire ensemble of codewords, it is sufficient to ensure that all of the codewords of $C^{\perp}$ are orthogonal to all of the rows in $\mathbf{G}$. Let $\mathbf{H}$ be the generator matrix, of size $(n-k)\times n$ representing the entire ensemble of codewords in $\mathbf{C}^{\perp}$ then:

$$\mathbf{G}\mathbf{H}^{\mathcal{T}} = 0. \tag{4.9}$$

The matrix $\mathbf{H}$ is also referred to as the *parity check* matrix of the code C. Due to the orthogonality of C and $C^{\perp}$, and therefore the orthogonality between $\mathbf{G}$ and $\mathbf{H}$, multiplying any codeword, $\mathbf{x}$, in C by $\mathbf{H}$ yields:

$$\mathbf{H}\mathbf{y} = \mathbf{0}. \tag{4.10}$$

This relationship between the codewords of C and $\mathbf{H}$ can be used to check that a received codeword is correct. This relationship also plays an integral part in the decoding of LDPC codes. Similarly to $\mathbf{G}$ not being uniquely defined for a code C, likewise $\mathbf{H}$ is not uniquely

defined as any set of basis vectors for $C^{\perp}$ may be used. Some decoding techniques may, however not perform equally well with all versions of $\mathbf{H}$ [69, p. 74].

The orthogonality between the parity check matrix and the generator matrix can be used to perform decoding of the linear block code via what is referred to as *syndrome decoding* [72, p. 10].

4.2.1 Syndrome decoding of linear block codes

Given a code word in C, $\mathbf{y}$ corrupted by a noise vector $\mathbf{e}$:

$$\mathbf{r} = \mathbf{y} + \mathbf{e} \tag{4.11}$$

$$= \mathbf{G}^{\mathcal{T}}\mathbf{x}, \tag{4.12}$$

where $\mathbf{r}$ is the received signal. The received syndrome $\mathbf{z}$ is then given by [72, p. 12]:

$$\mathbf{z} = \mathbf{Hr} \tag{4.13}$$

$$= \mathbf{H}(\mathbf{y} + \mathbf{e}) \tag{4.14}$$

$$= \mathbf{Hy} + \mathbf{He} \tag{4.15}$$

$$= \mathbf{HG}^{\mathcal{T}}\mathbf{x} + \mathbf{He} \tag{4.16}$$

$$= \mathbf{He}. \tag{4.17}$$

Using, for example, a lookup table to obtain the estimate of the error vector $\hat{\mathbf{e}}$ from the syndrome $\mathbf{z}$, the final decoded results $\hat{\mathbf{x}}$ is then given by:

$$\hat{\mathbf{x}} = \mathbf{r} - \hat{\mathbf{e}}. \tag{4.18}$$

The size of the lookup table is dependant on the number of different possible syndromes, which is given as q^{n-k} [73, p. 35].

In the subsection it is shown how convolutional codes can also be represented as a linear block code [69].

4.2.2 Convolutional codes as linear block codes

It is interesting to note that convolutional codes can be expressed in block form as:

$$\mathbf{x} = \mathbf{u}\mathbf{G}_{\infty}, \tag{4.19}$$

where:

$$\mathbf{G}_{\infty} = \begin{bmatrix} \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \mathbf{g}_v & & & \\ & \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \mathbf{g}_v & & \\ & & \mathbf{g}_0 & \mathbf{g}_1 & \cdots & \mathbf{g}_v & \\ & & & \cdots & \cdots & \cdots & \cdots \end{bmatrix}. \tag{4.20}$$

The disadvantage of this approach is that the resultant size of the generator matrix, and associated parity check matrix, increases with an increase in the length of the sequence encoded. These codes do have an advantage in that the encoding can be done via a shift-register type operation. Felström and Zigangirov [74] describe an LDPC code that is based on a convolutional block code. Some of the advantages of the LDPC convolutional code are lower encoding and decoding implementation complexity than normal LDPC block codes [74]. MacKay [72, p. 581] notes that turbo-codes are a special case of LDPC codes and could therefore be decoded in a similar manner using message passing (see section 4.4).

The following section formulates the decoding problem of a linear block code as a marginalisation (see subsection 4.3.1) operation. This is important as for large redundancy code words, the lookup table used for syndrome decoding may be impractical due to its size.

4.3 DECODING USING MARGINALISATION

An alternative to syndrome decoding of the encoded received message is based on the marginalisation of a function. This section starts by giving a short introduction to the marginalisation of a multivariate probability distribution, then formulates the decoding operation as a marginalisation operation. It is then shown how marginals may be calculated using factorisation. Finally, the section ends by showing how factor graphs are used to perform the marginalisation.

4.3.1 Marginalisation

Given a joint multivariate probability distribution:

$$P(x_0, x_2, \ldots, x_{N-1}), \tag{4.21}$$

the probability distribution of a single variable can be calculated by *marginalisation*. The marginal of the probability distribution with respect to a variable is given by:

$$P(x_i) = \int_{\tilde{x}_i} P(x_0, x_2, \ldots, x_{N-1}) \mathrm{d}\tilde{x}_i, \tag{4.22}$$

where $\tilde{x}_i$ denotes all variables *except* x_i. It is evident that the probability distribution of a symbol in a code can be obtained by marginalisation, if the joint probability distribution for all the symbols in a code is available.

4.3.2 Decoding as a marginalisation

In order to perform decoding via marginalisation the decoding problem must first be described in terms of a marginal. After the decoding problem has been stated as a marginal, efficient techniques for performing the marginalisation can then be used such as message passing [69, p. 54].

Let $\mathbf{s}$ be the column vector of data symbols to be encoded such that $s_i \in \mathbb{F}_q \forall i$ and $\mathbf{c} \in C$, where C is the code being used. Let $\mathbf{x}$ be the column vector of received encoded symbols. The decoding problem can then be stated as determining the most likely sent $\mathbf{s}$ given that $\mathbf{x}$ was received and that code C was used:

$$\hat{\mathbf{s}} = \arg\max_{\mathbf{s} \in \mathbb{F}_q^n} p(\mathbf{s}|\mathbf{x}), \tag{4.23}$$

where $\hat{\mathbf{s}}$ denotes the MAP vector of transmitted symbols. Using Bayes' rule, the posterior probability $\arg\max_{s_i \in \mathbb{F}} p(\mathbf{s}|\mathbf{x})$ can be written as:

$$p(\mathbf{s}|\mathbf{x}) = \frac{p(\mathbf{x}|\mathbf{s})p(\mathbf{s})}{p(\mathbf{x})}. \tag{4.24}$$

The term $p(\mathbf{x}|\mathbf{s})p(\mathbf{s})$ in eqn. 4.24 is termed the *likelihood* of the codeword [72, p. 324] and assuming a memoryless channel is separable:

$$p(\mathbf{x}|\mathbf{s})p(\mathbf{s} = \prod_{n=1}^{N} p(x_n|s_n), \tag{4.25}$$

and the term $p(\mathbf{s})$ is known as the *prior probability* of the codeword and is usually assumed to be uniform for all valid codewords in C [72, p. 325]. Using a *truth function* defined as:

$$\mathbf{1}_{\{\text{condition}\}} = \begin{cases} 1 & \text{if condition is true;} \\ 0 & \text{if condition is false,} \end{cases} \tag{4.26}$$

the prior $p(\mathbf{s})$ can be expressed as:

$$p(\mathbf{s}) = K_0 \mathbf{1}_{\mathbf{s}\in C}, \tag{4.27}$$

where K_0 is a constant. Equation 4.27 effectively limits the MAP solution to valid codewords of C. Since the MAP solution is the maximisation of an argument, and seeing that K_0 is a constant for all valid code words in C, the constant K_0 can be omitted in the maximisation.

The denominator in eqn. 4.24 is known as the *normalising constant* of the codeword and is given by:

$$p(\mathbf{s}) = \sum_{\mathbf{s}} p(\mathbf{x}|\mathbf{s})p(\mathbf{s}), \tag{4.28}$$

and is constant for all codewords and can thus be ignored when calculating the MAP solution.

Thus the symbol-by-symbol, or the *bitwise* in $\mathbb{F}_2$, MAP solution of eqn. 4.24 for a discrete memoryless channel, where all codewords are equally probable, is given by:

$$\hat{s}_n = \arg\max_{\mathbf{s}\in\mathbb{F}_q^n} p(x_i|\mathbf{s})\mathbf{1}_{\mathbf{s}\in C} \tag{4.29}$$

$$= \arg\max_{\mathbf{s}\in\mathbb{F}_q^n} \sum_{\tilde{s}_i} p(\mathbf{x}|\mathbf{s})\mathbf{1}_{\mathbf{s}\in C} \tag{4.30}$$

$$= \arg\max_{s_i\in\mathbb{F}_q} \sum_{\tilde{s}_i} \left(\prod_j p(x_j|s_j)\mathbf{1}_{\mathbf{s}\in C} \right). \tag{4.31}$$

Performing the decoding via the marginalisation in eqn. 4.31 is equivalent to performing an exhaustive search over all q^n possible received words or, if the marginalisation is only limited to valid codewords, the q^k codewords in C for *each* symbol in $\mathbb{F}_q$.

4.3.3 Calculating marginals using factorisation

Since performing the marginalisation via eqn. 4.31 is equivalent to an exhaustive search with computation complexity of $O\left(|\mathbb{F}_q|^n\right)$, where $|F_q|$ denotes the cardinality (number of elements in the field) of $\mathbb{F}_q$, it is desired to find a more efficient method of calculating the marginal in eqn. 4.31.

One such method makes use of factorisation and the distributive law [69].

Definition 4.3 (Distributive Law)**.** *Let a, b, and $c \in \mathbb{F}_q$ then:*

$$ab + ac = a(b + c). \tag{4.32}$$

The distributive law can also be applied to the marginalisation of function [69, p. 49]. Assuming a function f with variables $x_i \in F_q$ can be factorised as the product of several functions:

$$f(x_1, x_2, \cdots, x_n) = f(\mathbf{x}) \tag{4.33}$$

$$= \prod_{j, \mathbf{x}_j \subset \mathbf{x}} f_j(\mathbf{x}_j), \tag{4.34}$$

where the variable with respect to which the function is to be marginalised appears in all the factors and the other variables only occur in at most one factor, then the marginalisation of f can be written as:

$$\tilde{f}(x_i) = \sum_{\tilde{x}_i} f(\mathbf{x}) \tag{4.35}$$

$$= \sum_{\tilde{x}_i} \left(\prod_J f_j(\mathbf{x}_j) \right) \tag{4.36}$$

$$= \prod_J \left(\sum_{\tilde{x}_i} f_j(\mathbf{x}_j) \right), \tag{4.37}$$

where the marginalisation of f is computed as the product of the marginals of the factor functions of f. In general, each of the factors f_j may itself be expressed as a product of factors in the form of [69, p. 53]:

$$f_j(\mathbf{x}) = \underbrace{h(\mathbf{x})}_{\text{kernel}} \prod_K \underbrace{h_k(\mathbf{x}_k)}_{\text{factors}}, \tag{4.38}$$

where $\mathbf{x}_l \cap \mathbf{x}_k = \emptyset$, $\forall\ (l \neq k)$, and $x_i \notin \mathbf{x}_k$, $\forall\ k$, where x_i is the current variable with respect to which the function is being marginalised. Thus, the variable being marginalised over, *only* occurs in the *kernel*. The other variables occur only once each in the factors.

The marginalisation can then be expressed as:

$$\sum_{\tilde{x}_i} f_j(\mathbf{x}) = \sum_{\tilde{x}_i} \left[h(\mathbf{x}) \prod_K h_k(\mathbf{x}_k) \right] \tag{4.39}$$

$$= \sum_{\tilde{x}_i} h(\mathbf{x}) \underbrace{\prod_K \left(\sum_{\tilde{x}_k} h_k(\mathbf{x}_k) \right)}_{\text{product of marginals}}. \tag{4.40}$$

The individual marginals of the factor functions in eqn. 4.40 can also be computed using a similar factorisation of each factor. Thus, eqn. 4.40 allows the marginal of a factored function to be calculated *regressively*.

The following example from Richardson and Urbanke [69, p. 50] illustrates the marginalisation of a function by computing the product of the marginals of its factor functions. Consider the following function in six variables:

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5). \tag{4.41}$$

Using the distributive law, eqn. 4.37, the marginalisation of eqn. 4.41 over variable x_1 can be written as:

$$\tilde{f}(x_1) = \left[\sum_{\tilde{x}_1} f_1(x_1, x_2, x_3) \right] \left[\sum_{\tilde{x}_1} f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5) \right]. \tag{4.42}$$

The second product in eqn. 4.42 can be factorised using eqn. 4.38:

$$\underbrace{f_2(x_1, x_4, x_6)}_{\text{kernel}} \underbrace{[f_3(x_4) f_4(x_4, x_5)]}_{x_4} \underbrace{[1]}_{x_6}, \tag{4.43}$$

the marginal of which with respect to x_1,upon application of eqn. 4.40, becomes:

$$\sum_{\tilde{x_1}} f_2(x_1, x_4, x_6)\left(\sum_{\tilde{x_1}} f_3(x_4) f_4(x_4, x_5)\right)\left(\sum_{\tilde{x_1}} 1\right) \tag{4.44}$$

$$= \sum_{x_4, x_6} f_2(x_1, x_4, x_6)\left(f_3(x_4)\sum_{x_5} f_4(x_4, x_5)\right) \tag{4.45}$$

$$= \sum_{x_4} f_3(x_4)\left(\sum_{x_6} f_2(x_1, x_4, x_6)\right)\left(\sum_{x_5} f_4(x_4, x_5)\right). \tag{4.46}$$

Thus, the complete marginal for eqn. 4.41, can be expressed as:

$$\tilde{f}(x_1) = \left[\sum_{x_2, x_3} f_1(x_1, x_2, x_3)\right]\left[\sum_{x_4} f_3(x_4)\left(\sum_{x_6} f_2(x_1, x_4, x_6)\right)\left(\sum_{x_5} f_4(x_4, x_5)\right)\right]. \tag{4.47}$$

Whereas the original marginalisation had a computational complexity on the order of $O(|\mathbb{F}_q|^6)$, the factorised version has a computational complexity on the order of $O(|\mathbb{F}_q|^3)$, a significant reduction.

4.3.4 Marginalisation using factor graphs

The calculation of the marginal, by means of factorisation, of a function can be graphically visualised and computed using a *factor graph*. Figure 4.1 shows the factor graph of the example used in subsection 4.3.3 with x_1 as the base. Factor graphs can be used to calculate the marginal of a function.

The algorithm used to calculate the marginal via a factor graph is also known as the *sum-product* algorithm [72, p. 336]. There are two update rules that define the sum-product algorithm, one rule for connections from variable nodes to factor nodes and the second from factor nodes to variable nodes. These updates are referred to as *messages* and hence the name *message passing decoder*.

The rule for messages passed from variable nodes to factor nodes, u, is given by:

$$u_{n\to m}(x_n) = \prod_{\tilde{m}_n} v_{m'\to n}(x_n), \tag{4.48}$$

that is, the message from variable node x_n to factor node f_m is the product of all messages to the variable node x_n. The rule for messages passed from factor nodes to variable nodes, v, is given by:

$$v_{m\to n}(x_n) = \sum_{\tilde{x}_n} \left(f_m(\mathbf{x}_m) \prod_{\tilde{n}_m} u_{n'\to m}(x_{n'}) \right), \tag{4.49}$$

that is, the message from factor node f_m to variable node x_n is the marginal with respect to variable x_n of the factor node f_m and the product of all the messages factor node f_m has received.

For the case of *leaf nodes*, nodes that have no child nodes themselves, the messages passed on are:

$$u_{n\to m} = 1 \tag{4.50}$$

$$v_{m\to n} = f_m(\mathbf{x}_m). \tag{4.51}$$

As an example of marginalisation via a factor graph, based on the example in Richardson and Urbanke [69, p. 54], consider eqn. 4.41, used previously as an example in subsection 4.3.3, with its associated factor graph of fig. 4.1.

Starting from the variable leaf nodes, x_2, x_3, x_5 and x_6, the following variable to factor node messages are obtained:

$$u_{2\to 1} = 1 \tag{4.52}$$

$$u_{3\to 1} = 1 \tag{4.53}$$

$$u_{5\to 4} = 1 \tag{4.54}$$

$$u_{6\to 2} = 1. \tag{4.55}$$

From the leaf factor nodes, the following is obtained:

$$v_{3\to 4} = f_3(x_4). \tag{4.56}$$

The message from factor node f_4 to variable node x_4 is then calculated as:

$$v_{4\to 4} = \sum_{\tilde{x}_4} f_4(x_4, x_5)(1), \tag{4.57}$$

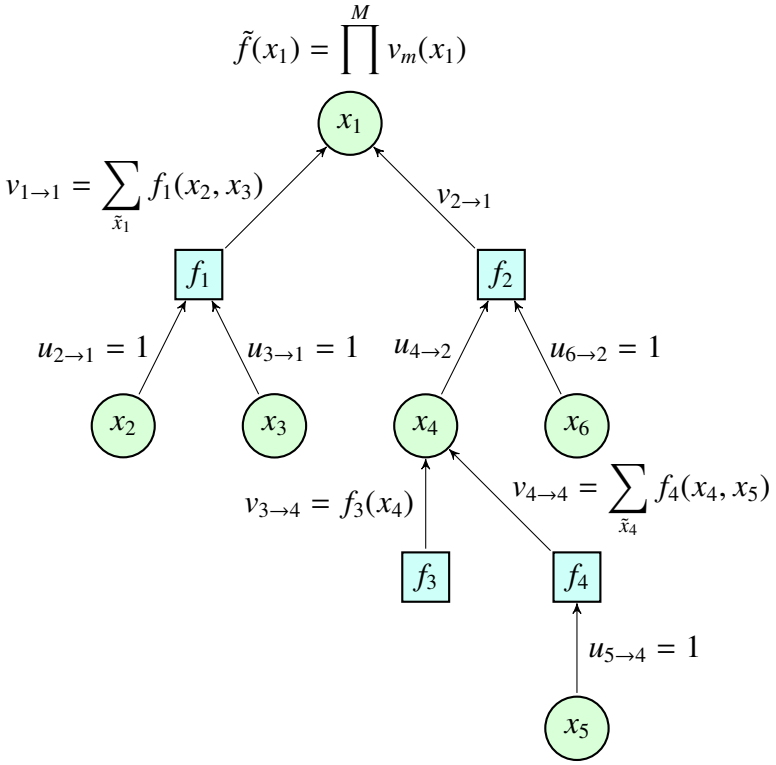

Figure 4.1: Factor Graph of the example in eqn. 4.41. showing marginalisation via message passing.

the message from variable node x_4 to factor node f_2 is:

$$u_{4\to 2} = f_3(x_4) \sum_{\tilde{x}_4} f_4(x_4, x_5). \tag{4.58}$$

From factor node f_1 to variable node x_1 the message is then:

$$v_{1\to 1} = \sum_{\tilde{x}_1} f_1(x_2, x_3) \tag{4.59}$$

and from factor node f_2 to variable node x_1 the message is:

$$v_{2\to 1} = \sum_{\tilde{x}_1} f_2(x_1, x_4, x_6) f_3(x_4) \sum_{\tilde{x}_4} f_4(x_4, x_5). \tag{4.60}$$

The final marginalisation is then the product of the messages to the base variable node x_1:

$$\begin{aligned} \sum_{\tilde{x}_1} f(\mathbf{x}) &= \left[\sum_{\tilde{x}_1} f_1(x_2, x_3)\right]\left[\sum_{\tilde{x}_1} f_2(x_1, x_4, x_6) f_3(x_4) \sum_{\tilde{x}_4} f_4(x_4, x_5)\right] \\ &= \left[\sum_{x_2, x_3} f_1(x_1, x_2, x_3)\right]\left[\sum_{x_4} f_3(x_4)\left(\sum_{x_6} f_2(x_1, x_4, x_6)\right)\left(\sum_{x_5} f_4(x_4, x_5)\right)\right], \end{aligned} \tag{4.61}$$

where the result is equivalent to that obtained in eqn. 4.47.

For the case where the factor graph has a tree structure, the marginalisation is exact [72, 69]. Unfortunately it has been proven that codes that have tree structures do not perform as well as codes that have cycles [69, p. 63]. A cycle occurs when there is a path that links the node to itself. The number of edges this path takes is referred to as the *length of the cycle*. Figure 4.2 shows an example of a factor graph with a cycle length of four. The main difficulty involved with graphs having cycles is that they lack leaf nodes to start the message passing. Thus in a graph with a cycle, before any of the variable nodes can pass on their respective messages, they have to receive the messages from the factor nodes to which they are connected. Unfortunately, before those factor nodes can transmit their messages, they have to receive a message from the variable nodes.

4.3.4.1 Bipartite Graphs

As a factor graph consists out of only two types of nodes, it is also a *bipartite graph*. In a bipartite graph, a factor node is connected to a variable node if the factor node is a function of the variable. Thus the bipartite graph for eqn. 4.41 can be seen in fig. 4.3.

The bipartite graph is useful in that it is trivial to derive the factor graph from it. In essence one simply has to re-arrange the nodes. The bipartite graph can thus also be used to compute the marginals of the variables without first having to construct a factor graph for each of the variables. In this method the marginals of all the variables may be simultaneously calculated on the same graph. The method entails that for each edge on the graph the output message is calculated using all other incoming edges. The current outgoing edge is not used in the calculation of the outgoing message. Only if all of the required incoming messages are available, is the output message computed and sent. The algorithm stops when a message has been sent in both directions on each edge of the graph. The final marginal is then computed as the product of all the received messages at the variable node with respect to which the function is being marginalised.

Alternatively one can also initialise all of the variable node messages, u, to the value of one [72, p. 337]. The message passing algorithm is then iterated over the graph until the marginals converge to the correct results. This method potentially performs wasteful computations, though it does have the advantage of being usable on graphs that are not trees [72]. In subsection 4.4.2 it will be seen that these variable nodes will be initialised with values obtained from the channel.

4.3.4.2 On-the-fly normalisation

It can be advantageous to normalise the marginals. This is especially the case with decoders as the MAP solution is generally desired. In decoders the sum of the calculated probabil-

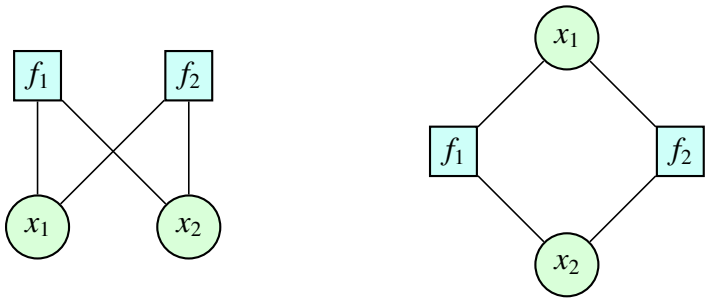

Figure 4.2: Example of a bipartite and a factor graph with a cycle length of four.

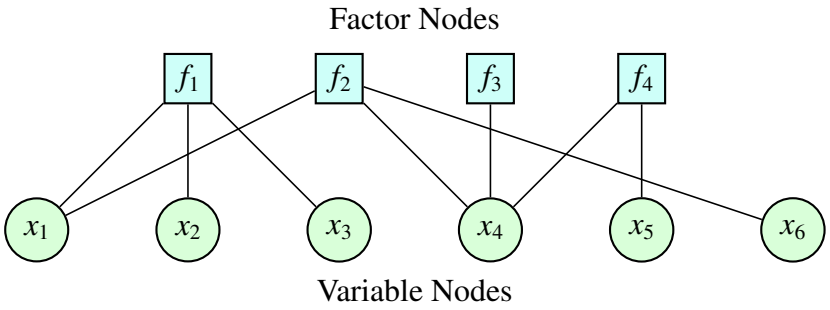

Figure 4.3: Bipartite Graph of eqn. 4.41

ities for the possible symbols must sum up to one. This is usually done by incorporating a normalisation constant into the variable to factor node messages. This can be written as [72]:

$$v_{n\to m}(x_n) = \alpha_{nm} \prod_{\tilde{m} u_{m'\to n}} (x_n), \tag{4.62}$$

where α_{nm} is a scalar such that:

$$\sum_{x_n} u_{n\to m}(x_n) = 1. \tag{4.63}$$

The normalisation can become important when there are many factors, as the product of these factors can become either quite large, or in the case of decoding with probabilities, very small.

4.3.4.3 Log domain computation

By performing marginalisation in the log-domain the variable to factor node messages can be computed using simple addition operations. The disadvantage is that the calculation of the factor to variable node messages requires the evaluation of a number of transcendental functions of the form:

$$l = \ln\left(\sum e^{l_i}\right). \tag{4.64}$$

These evaluations may be performed using lookup tables [72, p. 339]. Also the max-log approximation can be used:

$$\ln\left(e^{x_1} + e^{x_2} + \cdots e^{x_n}\right) \approx \max x_i \tag{4.65}$$

to give an approximate value.

4.4 MESSAGE PASSING DECODER

It was previously shown that decoding can be done as a marginalisation operation. Furthermore it was shown that marginalisation can be efficiently computed using factor graphs. In this section the general decoding algorithm for block codes will be described.

4.4.1 Bipartite Graphs of Block Code

In order to decode a received codeword using message passing, the bipartite graph of the code has to be constructed. The bipartite graph is then used as the factor graph for the codeword. The message passing algorithm is then iterated over this graph. The following example [69, p. 51] will be used to illustrate this process.

Let $\mathbf{H}$ be the parity check matrix of code C:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{4.66}$$

The factorisation for this code can then be formulated using the fact that if $\mathbf{x} \in C$ then $\mathbf{Hx} = \mathbf{0}$. This leads to the following formulation using the truth function:

$$\mathbf{1}_{\mathbf{x}\in C} = \begin{cases} 1 & \text{if } \mathbf{Hx} = \mathbf{0} \\ 0 & \text{otherwise} \end{cases} \tag{4.67}$$

$$= \begin{cases} 1 & \text{if } (\mathbf{h}_1\mathbf{x} = 0) \text{ and } (\mathbf{h}_2\mathbf{x} = 0) \text{ and } (\mathbf{h}_3\mathbf{x} = 0) \\ 0 & \text{otherwise} \end{cases} \tag{4.68}$$

$$= \mathbf{1}_{\{x_1+x_2+x_4=0\}}\mathbf{1}_{\{x_3+x_4+x_6=0\}}\mathbf{1}_{\{x_4+x_5+x_7=0\}}, \tag{4.69}$$

where $\mathbf{h}_i$ denotes the i^{th} row vector of $\mathbf{H}$. It is important to note that the addition operation is performed over the field $\mathbb{F}_q$ of the code. The MAP decoding problem can then be written as:

$$\hat{x}_i(\mathbf{y}) = \arg\max_{x_i\in\mathbb{F}_q} \sum_{\tilde{x}_i} \left(\prod_j p(y_j|x_j)\mathbf{1}_{\{x_1+x_2+x_4=0\}}\mathbf{1}_{\{x_3+x_4+x_6=0\}}\mathbf{1}_{\{x_4+x_5+x_7=0\}} \right). \tag{4.70}$$

This results in the factor graph (also referred to as a *Tanner* graph [69, 75]) in figure 4.4. In

this example the graph has a tree structure and will therefore result in the exact marginalisation.

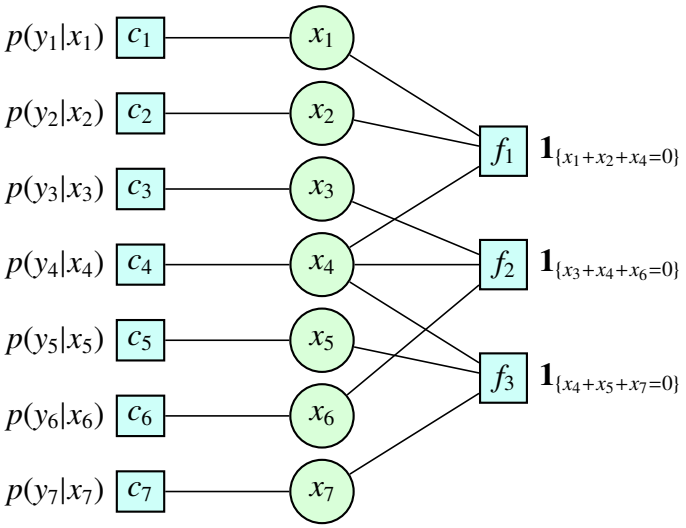

Figure 4.4: Factor graph of the example in eqn. 4.66.

In figure 4.4 the factor nodes on the left represent the received information from the channel. These are used as the initial values for the variable node since they act as leaf nodes in the message passing algorithm.

4.4.2 Decoding Block Codes via Message Passing

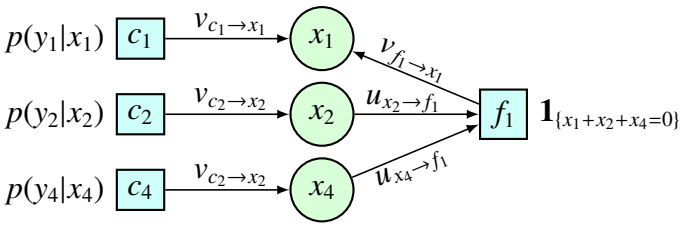

Figure 4.5: Subgraph of the factor graph in the example of eqn. 4.66.

Using the graph in figure 4.5 and the message passing rules in eqn. 4.48 and eqn. 4.49 the marginalisation over x_1 can be performed. At the variable node x_1, the calculation is given

by:

$$p(x_1|\mathbf{y}) = \prod_{\tilde{m}_1} v_{m'\to 1}(x_1) \tag{4.71}$$

$$= v_{f_1\to x_1} v_{c_1\to x_1} \tag{4.72}$$

that is the product of all the messages from the factor nodes connected to itself. The message from factor node c_1 is simply $p(y_1|x_1)$ obtained from the channel. The message from factor node f_1 is given by:

$$v_{f_1\to x_1}(x_1) = \sum_{\tilde{x}_1} \left(\mathbf{1}_{\{x_1+x_2+x_3=0\}} \prod_{\tilde{n}_{f_1}} u_{n'\to f_1}(x_{n'}) \right) \tag{4.73}$$

$$= \sum_{\tilde{x}_1} \left(\mathbf{1}_{\{x_1+x_2+x_3=0\}} u_{x_2\to f_1} u_{x_4\to f_1} \right) \tag{4.74}$$

$$= \sum_{\tilde{x}_1} \left(\mathbf{1}_{\{x_1+x_2+x_3=0\}} p(y_2|x_2) p(y_4|x_4) \right), \tag{4.75}$$

where $\tilde{n}_{f_1}$ is the number of variable nodes connected to factor node f_1 excluding the current variable node being sent a message. Whilst there are $q^{\tilde{n}_{f_1}}$ possible summations in the marginalisation calculation only those that satisfy the condition operator need to be evaluated. Assuming binary ($q = 2$) symbols, the message from the factor node becomes:

$$v_{f_1\to x_1}(x_1) = \begin{cases} p(y_2|x_2=1)p(y_4|x_4=1) + p(y_2|x_2=0)p(y_4|x_4=0), & x_1 = 0 \\ p(y2|x_2=1)p(y_4|x_4=0) + p(y_2|x_2=0)p(y_4|x_4=1), & x_1 = 1 \end{cases} \tag{4.76}$$

This results in the final marginal for x_1 being:

$$p(x_1|\mathbf{y}) = \begin{cases} p(y_1|x_1=0)[p(y_2|x_2=1)p(y_4|x_4=1) + p(y_2|x_2=0)p(y_4|x_4=0)], & x_1 = 0 \\ p(y_1|x_1=1)[p(y2|x_2=1)p(y_4|x_4=0) + p(y_2|x_2=0)p(y_4|x_4=1)], & x_1 = 1 \end{cases}. \tag{4.77}$$

It is still necessary to ensure that the sum of the probabilities associated with x_1 add up to one. This normalisation step can be performed as part of the message passing operation. The message passing operation is then repeated for the other variables in the graph. There are various optimisations that can be done to improve the decoding speed. These include observing that the message from the factor node can be regarded as the circular convolution

of the marginals sent to it [28]. The Fourier transform can then be used to perform the circular convolution [76, 69]. Other optimisation include the use of log domain processing [77] and truncation of some of the message vectors [78].

4.5 SPARSE GRAPH CODES

All linear codes can be represented as a graph [69] though not all linear codes have a *sparse* graph representation. A sparse graph is a graph where the number of edges connecting to vertices in the graph grows *linearly* with the number of vertices [79].

Linear codes that do have a sparse graph representation are called low density parity check (LDPC) codes or Gallagher codes. These codes have few non-zero elements in their parity check matrices, hence their name. Codes where each variable node and each check node have the same degree are termed *Regular*. Codes that do not have a regular degree distribution are called *irregular* codes. Luby et al. [80] show that irregular LDPC codes outperform regular LDPC codes. Furthermore, non-binary LDPC codes also offer an additional improvement over binary LDPC codes [81].

Shannon [68] showed that randomly selecting codewords from the ensemble of possible codewords results in a code that is [69] "...likely to be *good* for many channels." Similarly random LDPC codes tend to be good codes provided that the column density is greater or equal to three [72, p. 557].

Binary codes are used in this dissertation to provide the iterative MIMO decoders with a priori information on subsequent iterations, see chapter 3. The codes used can be found in [30].

4.6 PERFORMANCE OF THE CHOSEN LDPC CODES

In this section the performance, as measured by the achieved BER, of the codes used in this dissertation is investigated. Their basic performance in an AWGN channel is simulated. Additionally the ability of an LDPC code to exploit diversity in a Rayleigh fading channel

Table 4.1: Properties and filenames of the codes used in this dissertation.

Name	Weight	N	K	R	Filename
Code 1	3	20 000	10 000	0.5	20000.10000.3.631
Code 2	4	9972	4886	0.5	4986.93xb.329
Code 3	4	273	82	0.3	273.82.4.3065

with independently faded paths is explored. Lastly the effect of hard limiting the value of the LLR value given to the LDPC decoder is investigated.

4.6.1 Performance in AWGN

The BER performance of a code in AGWN is one of the most elementary performance measures of the code and the performance of the three LDPC codes is given in fig. 4.6. The two large LDPC codes perform similarly, though it can be seen that the length 20 000 LDPC code commences its waterfall region before that of the length 9972 LDPC code. The very short 273 LDPC code performs rather poorly, as expected from such a short LDPC code.

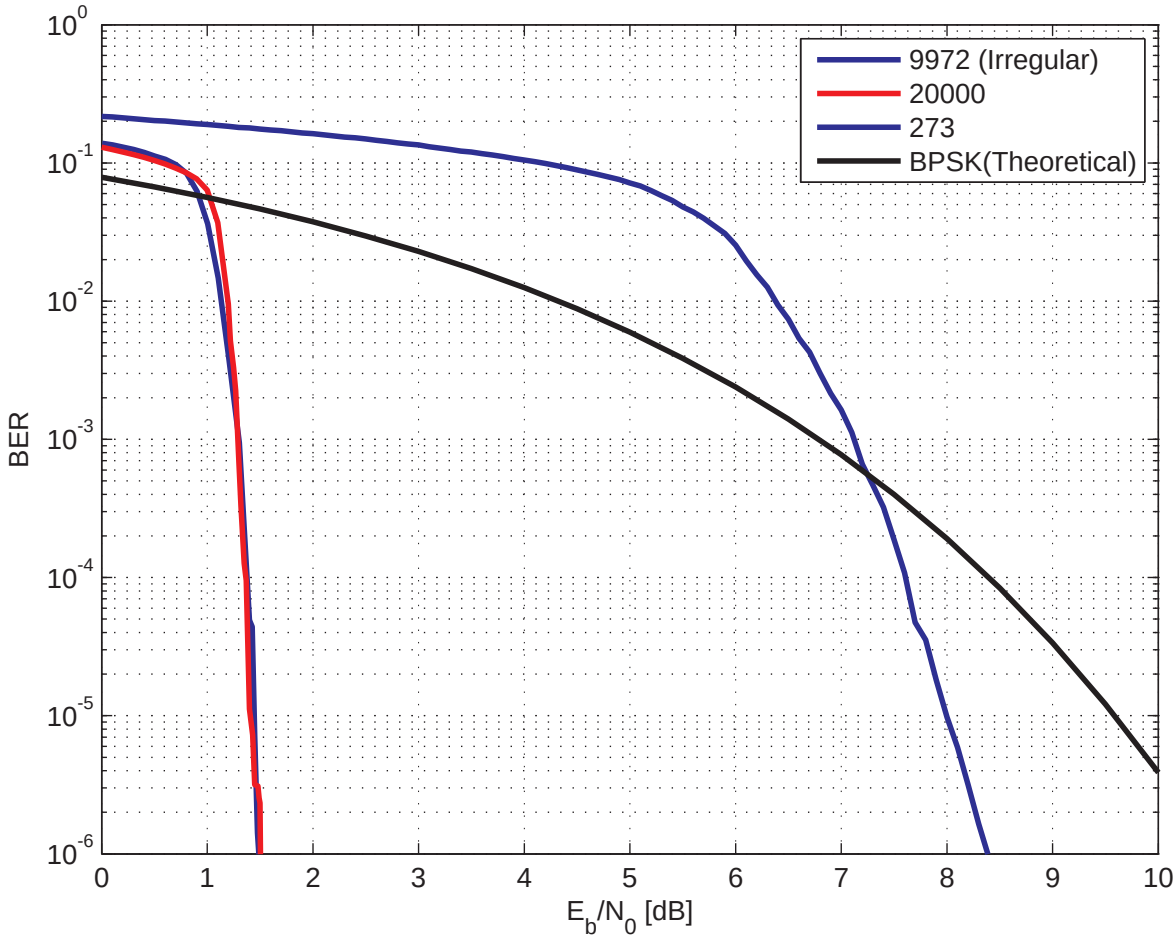

Figure 4.6: Plot of the BER performance of the three LDPC codes in table 4.1 compared to theoretical BPSK in AWGN.

4.6.2 Performance in Rayleigh fading channels

The average E_b/N_0 value required to achieve a BER of 10^{-6} in a Rayleigh fading channel is approximately 54 dB [40, eqns. 14.3-7]. In fig. 4.7 it can be seen that the LDPC code with half of the code faded independently offers virtually no improvement over an uncoded system. The LDPC code requires an E_b/N_0 of approximately 53 dB versus an uncoded system's 54 dB to achieve a BER of 10^{-6}. With a diversity order of four, the required E_b/N_0 drops to approximately 26 dB. Beyond a diversity order of 32 any additional order of diversity yields diminishing return. The horizontal asymptote, where each bit undergoes independent fading is at 3.6 dB, approximately 2 dBs worse than the performance of the code in AWGN.

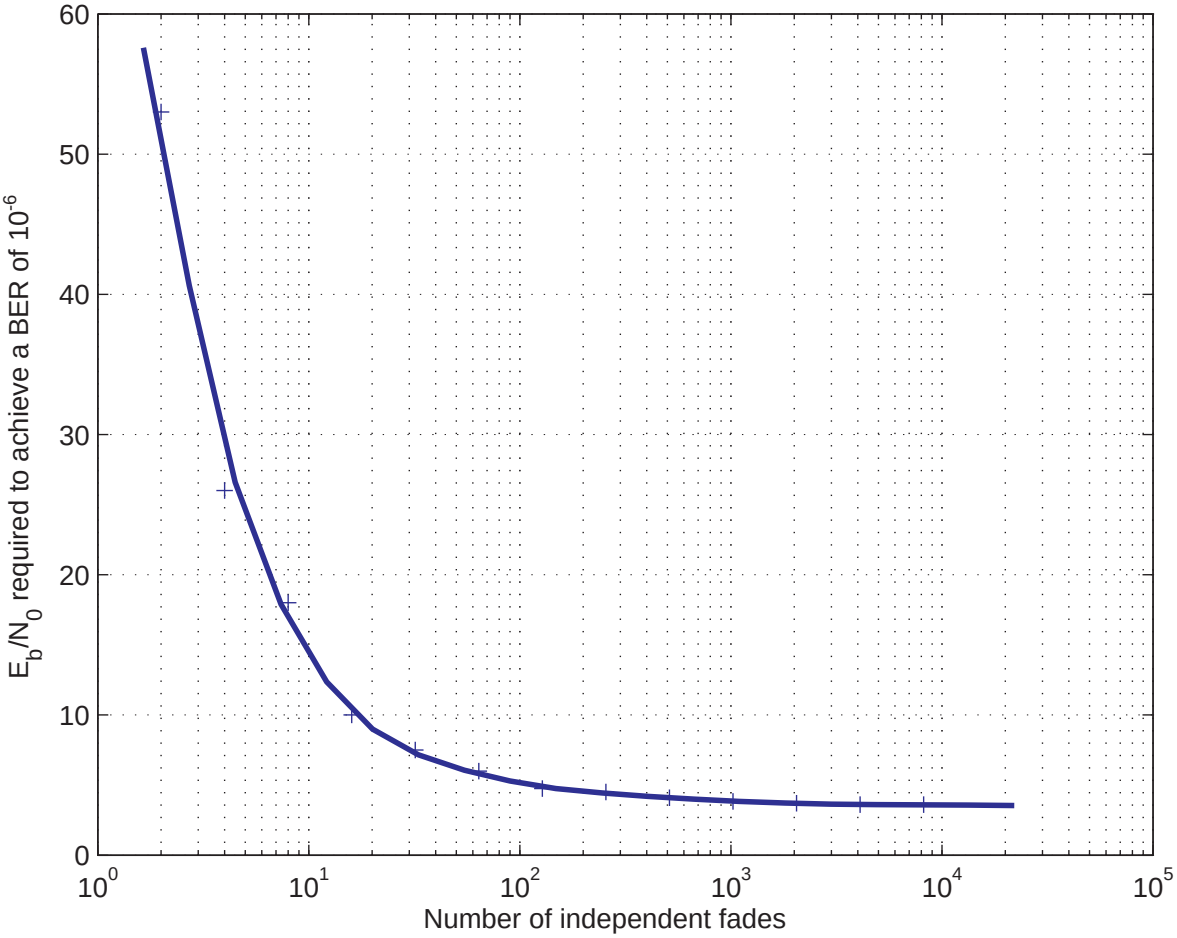

Figure 4.7: Plot of the E_b/N_0 required to achieve a BER of 10^{-6}, for the length 20 000 LDPC code, as a function of the number of independently faded Rayleigh sub blocks.

From this it can be seen that LDPC codes require that parts of the codeword undergo inde-

pendent fading to achieve optimal performance though, even with maximal diversity, there exist a performance penalty.

4.6.3 Effects of LLR limiting on the BER performance on LDPC codes

The max-log MAP hard-to-soft conversion used in the MIMO decoders gives inexact soft outputs. Specifically the output tends to be overoptimistic. To compensate for this, it was found that hard limiting of the LLRs from the MIMO decoders performs better than multiplying with a fixed scaling factor. The sphere decoder tends to produce the least optimistic and most correct LLR values. In fig. 4.8 the effect that the limiting of the LLRs has on the

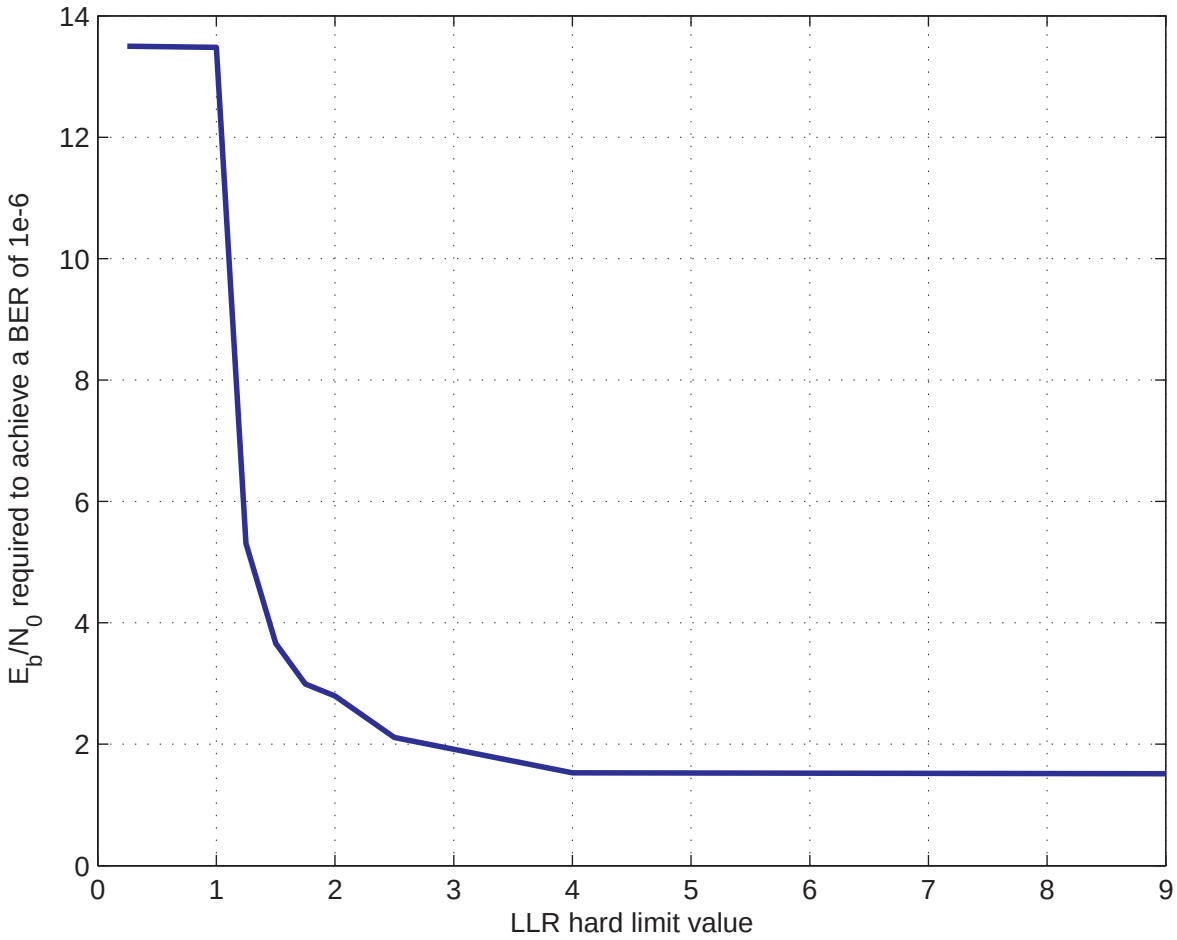

Figure 4.8: Plot of the E_b/N_0 required to achieve a BER of 10^{-6} for the length 20 000 LDPC code as function of the number of LLR hard limit value.

length 20 000 LDPC code may be seen. It can be seen that below a limit of 1, the LDPC

code can not correct errors. Beyond a value of 4, there is little effect on the BER. This upper value most likely is dependent on the BER performance of the code. The lower the waterfall region of the code, the lower the upper threshold and vice versa.

In summary, in this chapter an introduction into FEC was given with an emphasis on linear block codes specifically LDPC codes. The problem of decoding linear block codes was described as a marginalisation problem which can be efficiently computed using message passing. The application of message passing to decoding LDPC codes was then described.

Additionally in this chapter the performance of three chosen LDPC codes were given. Furthermore, the ability of an LDPC to exploit diversity in a Rayleigh fading environment was explored where it was found that without diversity there is almost no gain in using an LDPC code over an uncoded system.

Lastly the effect of hard limiting the LLRs provided to the LDPC decoder on the BER performance in AWGN was investigated, where it was found that there is a threshold beyond which the limiting does not affect the performance of the code.

CHAPTER 5

SPACE TIME FREQUENCY MIMO CODES

In chapter 4 it was shown that FEC codes aide in achieving the capacity of a communications channel. In this chapter STF codes that perform a similar function for MIMO systems are discussed. In chapter 3 various decoders that can be used to decode these MIMO STFCs were described. These STFCs form the inner code in the turbo structure, as is can be seen in fig. 6.1 on page 116. These codes allow MIMO systems to successfully achieve some of the advantages that they have over SISO systems.

Two main advantages that MIMO systems have, are improved capacity and reliability, without increasing the overall transmitted power, as compared to a SISO system. The theoretical capacity of a MIMO system is linearly related to the number of transmit antennas used; doubling the number of transmit antennas doubles the theoretical capacity of the MIMO system [1]. MIMO systems can also benefit from increased reliability in a fading environment by exploiting spatial diversity [1].

In spatial diversity it is assumed that each signal path between the various transmit and receive antennas experience independent fading. The order of the diversity is a measure of the number of independent paths between transmitter and receiver. In a MIMO system the order of the spatial diversity in the system is the product of the number of transmit and receive antennas.

The channel between the transmitter and receiver may also have other forms of diversity. In a multipath environment it is generally assumed that each path is independently faded and

thus the multipath diversity is the number of independently faded multipath components. Multipath diversity is easily exploited by OFDM systems. Additionally in an OFDM system sub carriers that are separated by more than the coherence bandwidth of the channel also experience independent fading. This is called *frequency* diversity.

Lastly fading is a time varying process, therefore signals transmitted at different time intervals may experience independent fading. This is referred to as *time* diversity.

The total diversity D in a system is therefore the product of the number of transmit antennas N_{TX}, the number of receive antennas N_{RX}, the number of individually faded multipath components K and the number of individually faded time blocks B:

$$D = N_{TX}N_{RX}KB. \tag{5.1}$$

FEC codes benefit from the diversity in the channel though at the expense of increased latency and reduced transmission rates since they do not directly exploit the analogue nature of the wireless channel.

In this chapter, coding techniques that exploit the diversity and capacity of a MIMO channel are described. These codes differ in decoding complexity, communication rate and achieved diversity. Codes based on orthogonal design and algebraic number theory are discussed with the emphasis on TAST codes which are used in this dissertation [2, 82]. The BER performance of the TAST codes used in this dissertation is also simulated.

5.1 CHARACTERISTICS OF STF MIMO CODES

There are generally speaking three types of MIMO codes: space-time (ST), space-frequency (SF) and STF codes. These codes are named after the type of diversity they extract. All of these codes are based on the idea that a data vector $\mathbf{s}$ is encoded by a code to produce an encoded data vector $\mathbf{x}$. The elements s_i of the encoded data vector are then mapped onto the MIMO system for transmission. If the elements of $\mathbf{s}$ are mapped onto different antennas, space diversity is obtained. Likewise if the elements of $\mathbf{s}$ are mapped onto different antennas and time periods space and temporal diversity is obtained and the code is referred to as an

ST code. A STF code thus maps the elements of **s** onto different antennas, time periods and frequencies to obtain space, temporal and frequency diversity.

There are two main classes of MIMO codes: linear block code and trellis codes. Trellis codes are a combination of FEC, transmit diversity and modulation [83, 84]. These codes were originally proposed as, at the time, linear block codes did not provide coding gain [83]. These codes are generally decoded using a trellis based decoder such as the Viterbi decoder [57]. In this dissertation, however, the focus is on linear block codes that can be decoded with the linear operation developed in chapter 3.

Tarokh et al. [83] developed two design criteria based on the pairwise error probability (PEP) that one codeword **C** could be mistaken for another. Hence:

$$P(\mathbf{C} \rightarrow \mathbf{C}') \leq \left(\prod_{i=0}^{r-1} \lambda_i \right)^{-N_R} \left(\frac{\rho}{4N_T} \right)^{-rN_R}, \tag{5.2}$$

where N_R is the number of receive antennas, N_T is the number of transmit antennas, ρ is the signal-to-noise ratio at the receiver and r is the rank of the matrix **D** given by:

$$\mathbf{D} = \mathbf{C} - \mathbf{C}', \tag{5.3}$$

where **C** and λ_i are the non-zero eigenvalues of the matrix given by:

$$\mathbf{D}^{\mathcal{H}}\mathbf{D} = (\mathbf{C} - \mathbf{C}')^{\mathcal{H}}(\mathbf{C} - \mathbf{C}'). \tag{5.4}$$

From the PEP an upper bound on the BER performance of the code is obtained as well as the two design criteria [83, 84]:

- **Rank criterion or diversity criterion:** The minimum rank of the matrix **D** for all pairs of distinct codewords must be as large as possible; and
- **Product criterion:** The minimum of the products of the eigenvalues (λ_i) of $\mathbf{D}^{\mathcal{H}}\mathbf{D}$ over all pairs of distinct codewords must be as large as possible.

The rank criterion gives an indication of the diversity that the code can achieve and will thus affect the slope of the BER curve whilst the product criterion gives an indication of the

coding gain that the code can provide and will be evident in a translation of the BER curve. This can be inferred by taking the logarithm of eqn. 5.2:

$$P(\mathbf{C} \to \mathbf{C}') \leq \left(\prod_{i=0}^{r-1} \lambda_i\right)^{-N_R} \left(\frac{\rho}{4N_T}\right)^{-rN_R} \tag{5.5}$$

$$\therefore \quad \ln P(\mathbf{C} \to \mathbf{C}') \leq \underbrace{-N_R \sum_{i=0}^{r} \ln \lambda_i}_{\text{translation}} - \underbrace{rN_R}_{\text{slope}} \ln \frac{\rho}{4N_T}. \tag{5.6}$$

Other important criteria to keep in mind is the rate achieved by the code as well as decoding complexity. For linear block codes two main construction methods have been developed. The first is based on orthogonal design, such as the Alamouti ST code [8]. The other is based on algebraic number theory such as TAST codes [2, 82].

5.2 ORTHOGONAL CODES

Orthogonal codes are codes whose codewords are orthogonal to each other. That is the cross-correlation matrix is a scaled identity matrix, as can be represented by:

$$\mathbf{C}^{\mathcal{T}}\mathbf{C} = \sum_{i=0}^{N-1} E\left[|s_i|^2\right] \mathbf{I}_N. \tag{5.7}$$

Since the codewords are orthogonal to each other, the symbols they encode can be decoded individually using maximum ratio combining (MRC) [1, p. 253]. Thus the decoding complexity growth is linear with respect to the growth in the number of encoded symbols.

One of the first space time block codes is the Alamouti code which is an orthogonal code [8]. This code was originally only designed for a two transmit antenna system:

$$\mathbf{C} = \begin{bmatrix} x_0 & -x_1^* \\ x_1 & x_0^* \end{bmatrix}, \tag{5.8}$$

that is during symbol period t_0 symbol x_0 is transmitted on antenna 0 and symbol x_1 on antenna 1. During the following symbol period $-x_1^*$ is transmitted on antenna 0 and x_0^* is transmitted on antenna 1. It is assumed that the channel stays stationary over the two symbol periods.

The Alamouti code manages to exploit a diversity order of two at a rate of one. Orthogonal codes that achieve rates of $3/4$ for $N_{TX} \in \{3, 4\}$ have been developed [9, 10]. In the event that $N_{TX} > 4$, these codes are limited to a rate of a half.

It is therefore evident that whilst orthogonal codes exploit the available diversity in the system, they cannot make use of the full capacity of the wireless MIMO channel. There are fortunately codes based on linear transformations that can make use of full diversity and capacity of the MIMO channel [2].

5.3 LINEAR PRECODING

Linear precoding involves a linear transformation that is applied to the channel symbol vector $\mathbf{s}$ before transmission over the channel. The precoding can be used with CSI to compensate for the channel effect at the transmitter as in maximal ratio transmission (MRT) [1]. Optimal linear precoders that incorporate perfect CSI at the transmitter and receiver have been derived by Palomar and Jiang [85].

Another class of codes allow the MIMO communication system to exploit available diversity and capacity without relying on CSI such as codes based on a linear transformation such as a rotation of the signal space.

5.3.1 Diversity through rotation

Diversity can be obtained using a linear transform such as rotational transform to the transmit signal vector $\mathbf{s}$. To illustrate how a rotational transform can achieve diversity, consider as example the two dimensional signal space in fig. 5.1.

If various signal constellation points share the same coordinates on the same basis vector, then those signal constellation points may not be uniquely decoded if the other basis vectors undergo fading. In the example points p_0 and p_3 have the same coordinate value on basis vector $\vec{v_1}$. It is only through their coordinates on basis vector $\vec{v_0}$ that the decoder may differentiate between them. Should basis vector $\vec{v_0}$ undergo fading, then the decoder will not

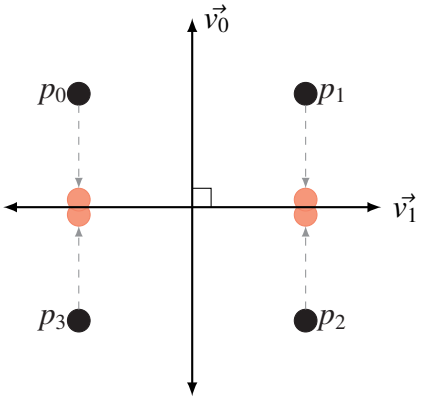

Figure 5.1: Figure showing the effect of a faded basis vector on the signal constellation. In the figure the $\vec{v_0}$ basis undergoes fading resulting in an ambiguous signal constellation.

be able to correctly distinguish between points p_0 and p_3. The same hold for points p_1 and p_2.

By applying an appropriate rotational transform to the signal constellation in fig. 5.1 the signal constellation points can still be distinguished if a basis vector is faded.

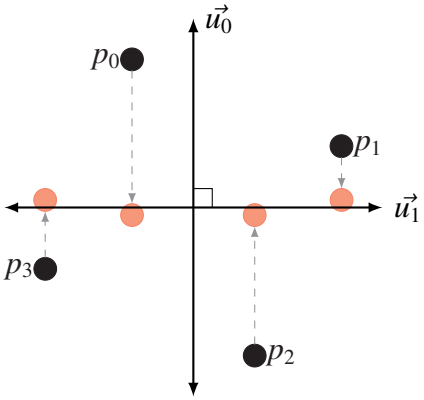

Figure 5.2: Figure showing the effect of a faded basis vector on a rotated signal constellation. In the figure the $\vec{u_0}$ basis undergoes fading resulting in an ambiguous signal constellation.

In fig. 5.2 it can be seen that the rotated signal constellation points do not share any co-ordinates on any of the basis vectors. The decoder will therefore be able to correctly decode the signal, provided that one basis vector is not faded. The euclidean distance between the constellation points is reduced when a basis undergoes fading thus potentially resulting in a

lower SNR which may cause increased decoding errors.

Based on these observations of a rotation in two dimensions it is evident that the rotation should be chosen such that each constellation point has a unique coordinate on all of the basis vectors and that the minimum euclidean distance between the coordinates of two adjacent signal constellation points on any basis vector be as large as possible.

The first criterion, the diversity criterion, for a rotation matrix $\mathbf{\Theta}_r$ can be given as:

$$\Theta_k(\mathbf{s} - \mathbf{s}') \neq 0, \quad \forall \ \mathbf{s} \neq \mathbf{s}', \quad \forall \ k, \tag{5.9}$$

where Θ_k denotes the k^{th} row of matrix $\mathbf{\Theta}$. The second criterion is generally stated in terms of the *product* of the euclidean distances of the coordinates on the basis vectors between two adjacent signal points [13]: find the rotation matrix that maximises $d_{\mathbf{\Theta}_r}$ given by:

$$d_{\mathbf{\Theta}_r} = \min_{\forall \ \mathbf{s}\neq\mathbf{s}'} \prod_{k=0}^{r-1} |\Theta_k(\mathbf{s} - \mathbf{s}'|^2. \tag{5.10}$$

An additional criterion is that the rank of the rotation matrix $\mathbf{\Theta}_r$ should be r. Furthermore, if $\mathbf{\Theta}_r$ is a unitary matrix ($\mathbf{U}^{\mathcal{H}}\mathbf{U} = \mathbf{U}\mathbf{U}^{\mathcal{H}} = \mathbf{I}$) [86] then the distance properties of the signal constellation is retained. Thus, in an AWGN system the performance of the rotated constellation is identical to the original constellation.

It should be noted that each dimension uniquely represents the entire signal constellation. Thus, all of the data that is to be transmitted is represented on all of the basis vectors instead of the n^{th} root of the data, where n is the number of basis vector. This may have an effect on systems with limited ADCs [87]. Rate distortion theory implies that signals cannot be represented with less information than what they contain without experiencing distortion [40].

A two dimensional system was used to illustrate how a rotational transform allows a two dimensional system to gain diversity. This can be extended to an arbitrary number of dimensions.

5.3.2 Algebraically constructed linear precodes

Damen et al. [2] and Xin et al. [12] give a construction method for unitary linear precodes as:

$$\mathbf{\Theta} = \mathbf{F}_r \text{diag}(\theta^0, \theta^1, \ldots, \theta^{r-1}), \tag{5.11}$$

where $\mathbf{F}_r$ is the $r \times r$ discrete Fourier transform matrix and $\theta = \exp(\jmath 2\pi/4P)$ and $P = N_{TX}l$ with $l \in \mathbb{N}$. The variable P is to ensure that the matrix is unitary for non powers of two matrix sizes. For instances where the desired rotation matrix size is a power of two, $P = r$ and the matrix is unitary.

The construction in eqn. 5.11 is used in the construction of diagonal algebraic space time (DAST) codes by multiplying the resultant rotation matrix with a Hadamard transform [88, p. 134] to improve the peak-to-average power ratio [89]. This also transforms the signal constellation into a lattice that can be efficiently decoded by decoders that can take advantage of the lattice structure such as the sphere decoder or decoder based on lattice reduction [90, p. 245].

The resultant codeword elements are then transmitted in such a way that they do not interfere with each other as in a diagonal layering in space and time:

$$\mathbf{C}_{N_{TX}} = \begin{pmatrix} x_1 & 0 & \cdots & 0 \\ 0 & x_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & x_{N_{TX}} \end{pmatrix}, \tag{5.12}$$

the rows denote antenna, the columns denote different time periods and x_i denotes the i^{th} element of the codeword given by $\mathbf{x} = \mathbf{\Theta s}$.

DAST codes, though having full diversity, did not exploit the available capacity of the MIMO channel though they are used as a component code in a linear precode construction that does achieve full capacity.

5.3.2.1 Threaded algebraic space time codes

To increase the capacity DAST codes were used in a TAST code [11]. In TAST codes the component codes are *layered* together in *threads*. This allows the TAST code to achieve diversity and throughput. It is necessary but not sufficient that the component code for the TAST code have full diversity for the resultant TAST code to also have full diversity [11]. The use of Diophantine numbers [91] to scale the individual threads allows the TAST code to achieve full diversity. The Diophantine numbers given by:

$$\phi_l = \exp\left[\jmath(l-1)/N\right], \tag{5.13}$$

where $l \in \{1, \ldots, L\}$ denotes the number of the layer and N is the size of the component rotation matrix.

Thus, using a circulant layering approach the code matrix in eqn. 5.12 for the TAST code then has the following structure:

$$\mathbf{C} = \begin{pmatrix} \phi_1 x_1^1 & \phi_L x_2^L & \phi_{L-1} x_3^{L-1} & \cdots & \phi_2 x_M^2 \\ \phi_2 x_1^2 & \phi_1 x_2^1 & \phi_L x_3^L & \cdots & \phi_3 x_M^3 \\ \phi_3 x_1^3 & \phi_2 x_2^2 & \phi_1 x_3^1 & \cdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi_L x_1^L & \phi_{L-1} x_2^{L-1} & \cdots & \cdots & \phi_1 x_M^1 \end{pmatrix}, \tag{5.14}$$

where L is the number of layers and M is the component code length.

A general form for TAST codes is derived by Damen et al. [82] where it is shown that TAST codes based on algebraic rotations do not necessarily result in increased peak-to-average power ratio. The Hadamard transform of the DAST codes may thus be omitted. In this dissertation none of the MIMO decoders make use of lattice structures and therefore the loss of the lattice structure is of no consequence.

5.4 PERFORMANCE OF THE STF MIMO CODES USED IN THIS DISSERTATION

The performance of the various STF TAST codes used in this dissertation is given in fig. 5.3. All of the STF codes are based on TAST based on DAST codes without the Hadamard transform.

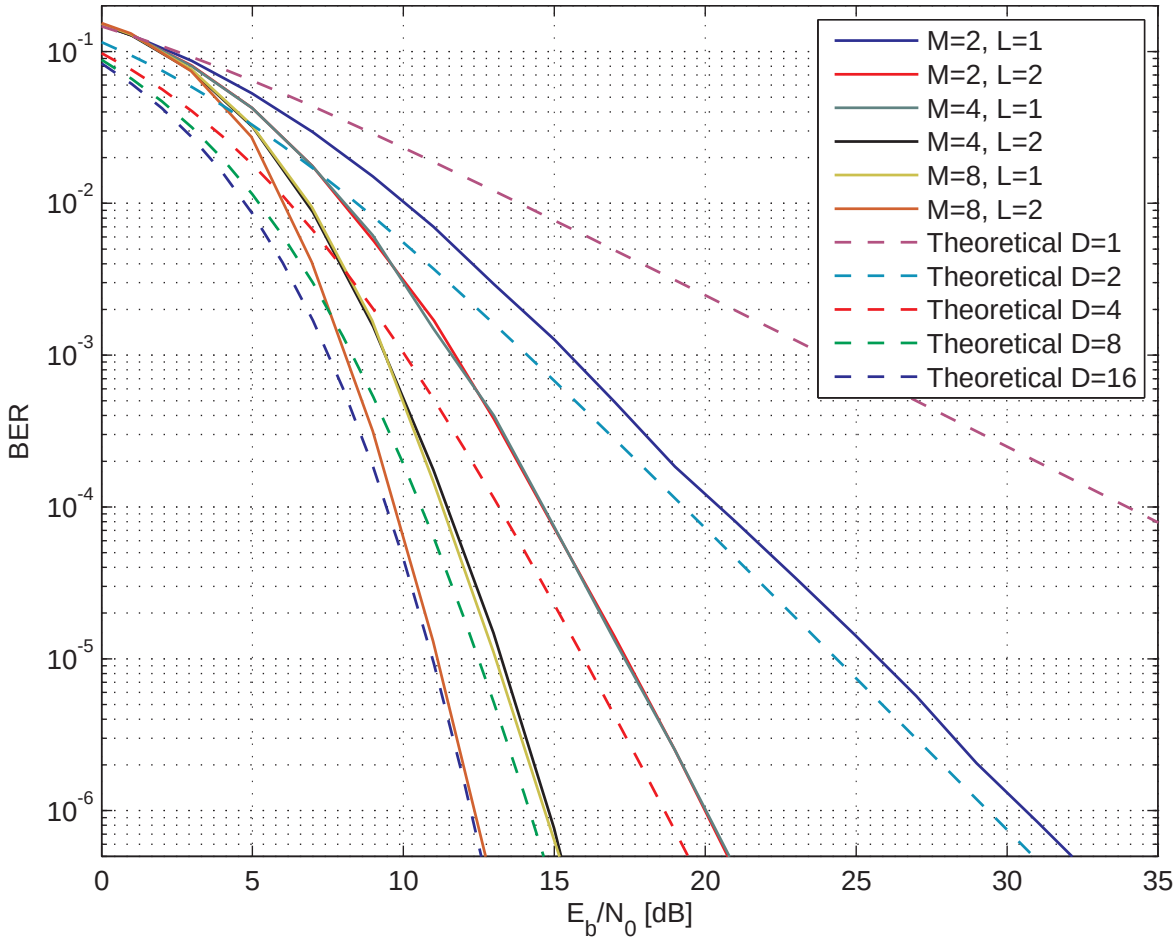

Figure 5.3: BER plot of TAST code for different number of layers L and component code lengths M for QPSK signal constellation using sphere decoder decoding in flat Rayleigh fading for a $N_{TX} = N_{RX} = 2$ MIMO system. The theoretical BER curves for Rayleigh fading with diversity D is also shown for comparison.

In fig. 5.3 it can be seen that the diversity of the code is proportional to the product ML. Therefore the $M = 2, L = 2$ and $M = 4, L = 1$ codes perform exactly the same, as well as the

$M = 4, L = 2$ and $M = 8, L = 1$ codes. Comparing the simulated performance of the codes with the theoretical curves shows that the slopes of the simulated performance of the codes are slightly less than the slopes expected. Recall that the theoretical diversity slope is given by eqn. 5.4 as $N_{RX}r$ where $r = M$. It seems therefore that the TAST codes, as simulated, only extract half the available diversity in the system.

The difference between the various TAST codes is encouraging as the diversity obtained does scale as expected with increases in the layers used and code sizes.

In summary, in this chapter the basic characteristics of the most common STF codes were given. The two main types of STF codes, that of orthogonal and linear precodes, were presented with the focus on STF based on layering of algebraic rotations. The performance of the STF TAST codes used in this dissertation was also given.

CHAPTER 6

TURBO STRUCTURE AND SYSTEM DESCRIPTION

In this chapter the overall system diagram combining the MIMO STFC decoder, FEC coding and STFCs, as described in chapters 3, 4 and 5 respectively, is given. The overall system combines the MIMO TAST and LDPC codes together, which are iteratively decoded in a turbo decoder structure. The transmitter and receiver will be discussed as well as the limiting of the output from the STFC decoder.

6.1 TRANSMITTER

Figure 6.1 shows a block diagram of the transmitter structure. The source provided equiprobable binary data **d** that is encoded by the FEC LDPC encoder to yield the encoded binary symbols **b**. The encoded binary symbols are mapped to the appropriate channel symbols **s** by the constellation mapper. The channel symbols are then linearly precoded, using the STF encoder block, in to the final channel symbols that are to be transmitted **x**. The MIMO multiplexer then maps the encoded channel symbols **x** onto the appropriate antenna, subcarrier and time slot for transmission over the antennas.

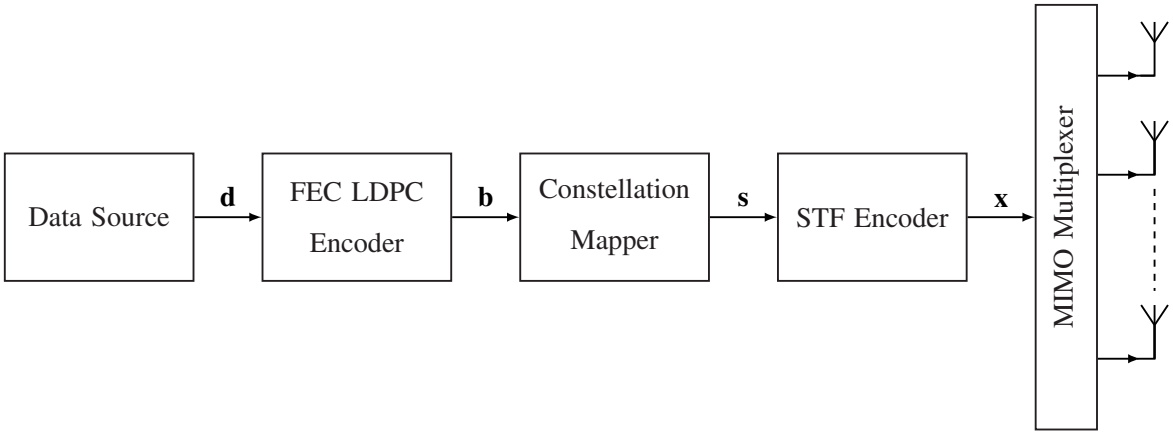

Figure 6.1: Block diagram of the structure of the transmitter.

6.2 RECEIVER

The block diagram for the receiver structure is given in fig. 6.2. The received signal **y** is demultiplexed from the various receiver antennas and is passed, together with the channel information, **H**, to the STFC decoder.

The STFC decoder uses the channel information **H**, the received signal **y** and any a priori information λ^a_{STF} to soft decode the STF encoded data symbols to produce the MAP decoded values, using the max-log approximation, λ^p_{STF}. On the initial decoding iteration there is no a priori information provided to the STF decoder. The STFC decoder also perform the conversion from channel symbols to binary data, i.e. it *demaps* the signal constellation.

The soft decoded data from the STFC decoder is then used to calculate the extrinsic information that is sent to the FEC decoder by subtracting the a priori information the STFC decoder used. Before the extrinsic data is passed to the FEC decoder, the LLR values are first hardlimited in magnitude. The data passed to the FEC decoder is thus given by:

$$\lambda^a_{\text{FEC}} = \text{HL}\left(\lambda^p_{\text{STF}} - \lambda^a_{\text{STF}}\right), \tag{6.1}$$

where HL() denotes the hard limiting function. The hard limiting corrects for the over optimistic results obtained from the max-log approximation and from the sub-optimal decoders such as the ZF decoder.

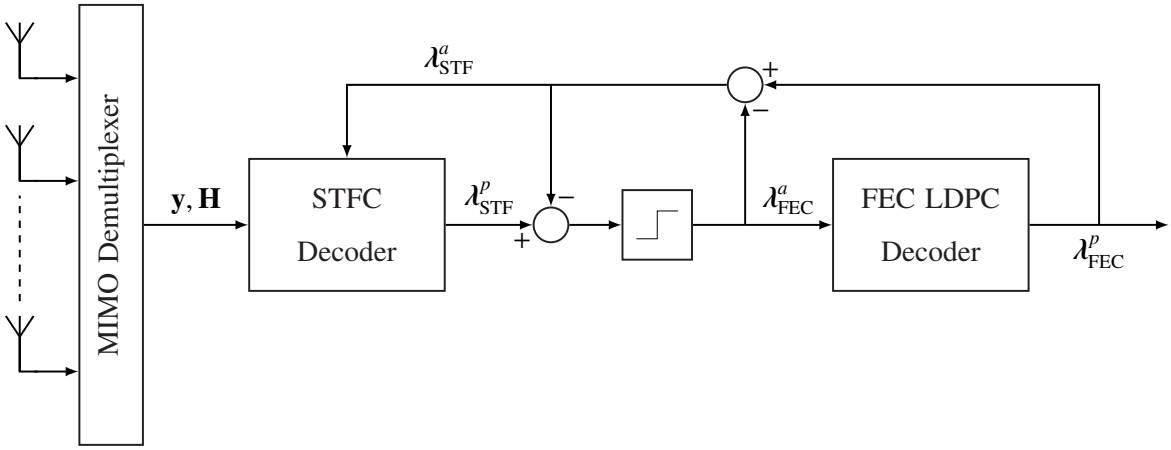

Figure 6.2: Block diagram of the structure of the receiver.

The FEC decoder then attempts to correct some of the errors in the data decoded by the STFC decoder. Once the FEC decoder has finished decoding the data, the output can be used as the final output of the system, or used for subsequent iterations. For subsequent iterations, the data decoded by the FEC decoder is used to calculate the extrinsic information, used as a priori information by the STFC decoder, by subtracting from it the input to the FEC decoder:

$$\lambda^a_{STF} = \lambda^p_{FEC} - \lambda^a_{FEC}. \tag{6.2}$$

6.2.1 Mathematical model of receiver

The mathematical model for the receiver structure, excluding the FEC decoder, is given as:

$$\mathbf{y} = \mathbf{HP\Theta s} + \mathbf{n}, \tag{6.3}$$

where $\mathbf{H}$ is the MIMO channel matrix, $\mathbf{\Theta}$ is the linear precode matrix, $\mathbf{s}$ is the uncoded signal constellation symbols and $\mathbf{n}$ is AWGN. The vector $\mathbf{s}$ has the form:

$$\mathbf{s} = \left[\mathbf{s}_1^{\mathcal{T}}, \cdots, \mathbf{s}_L^{\mathcal{T}}\right]^{\mathcal{T}}, \tag{6.4}$$

where $\mathbf{s}_i$ is the vector of symbols associated with each thread in the TAST code and L is the number of threads or layers in the TAST code. The encoder matrix $\mathbf{\Theta}$ is composed of the linear precode matrix $\mathbf{\Theta}'$ and the Diophantine numbers $\phi^{\mathbf{i}}$ used to separate the various layers

in the TAST code and can be obtained as:

$$\mathbf{\Theta} = \text{diag}\left(\phi^0, \phi^1, \ldots, \phi^{L-1}\right) \otimes \mathbf{\Theta}'. \tag{6.5}$$

The effective channel matrix $\mathbf{H}$ is comprised of MIMO channel blocks $\mathbf{H}'$ on the diagonal. Each individual MIMO channel block represents the MIMO channel for that time and sub-carrier instance. These MIMO channel blocks are orthogonal to each other hence their placement on the diagonal:

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}'_1 & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{H}'_2 & \mathbf{0} & \mathbf{0} \\ \vdots & \mathbf{0} & \ddots & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{H}'_{N_{H'}} \end{bmatrix}, \tag{6.6}$$

where $N_{H'}$ is the number of MIMO channel sub blocks. The number of MIMO blocks on the diagonal is given by the number of different sub carriers and time blocks that were used to transmit the TAST code and is given by:

$$N_{H'} = \left\lceil \frac{ML}{N_{TX}} \right\rceil, \tag{6.7}$$

where M is the size of the linear precode, L is the number of threads or layers in the TAST code and N_{TX} is the number of transmit antennas. In this dissertation the individual MIMO subblocks are modelled as each having a i.i.d. complex Gaussian distribution of unit variance.

6.2.2 Effect of hard limiting the a priori information to the FEC decoder

It was found that one of the important factors affecting the FEC decoder used was that of the magnitude of the LLR values, specifically if those LLR values were in error. Simulations were run in an attempt to find an optimum value for the hard limit value. The simulations were performed using the sphere decoder, a size four TAST code with two layers using the small $N = 273$ LDPC code and was performed for both QPSK and 16QAM. As shown in figs. 6.3 and 6.4.

In figure 6.3 the effects of hard limiting the LLR values from the STFC decoder can be seen. It appears that a value of around six is optimal. The incremental gain from iteratively

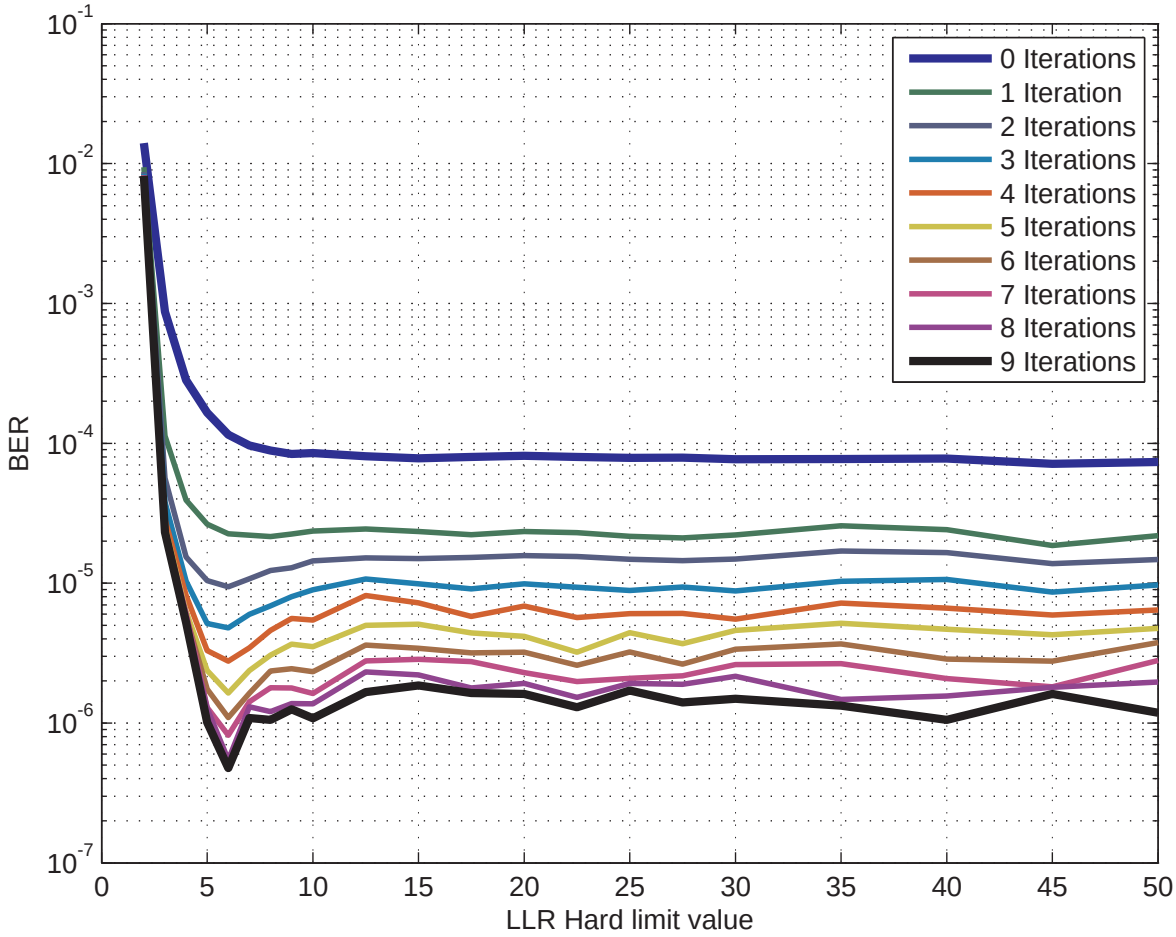

Figure 6.3: Plot of the BER as a function of the hard limit value for the LLRs from the STC decoder showing the BER of each iteration. Code used was the $N = 273$ LDPC code with a $M = 4$, $L = 2$ TAST code and QPSK signal constellation. $N_{TX} = N_{RX} = 2$. An $E_b/N_0 = 11$dB was used.

decoding can also be seen in that around two orders of magnitude in BER performance was gained. The diminishing returns for each successive iteration can also be seen. Note that only two hundred errors were searched, thus the accuracy of the graph in terms of performance might not be exact. For the rest of the simulation in this dissertation a hard limit value of five was used.

Figure 6.4 shows a similar graph and it can be seen that an LLR value of around five is best. The gains from iterative decoding is about a factor ten reduction in the BER. The diminishing returns from successive iterations is more pronounced. Finally, the increase in BER can be

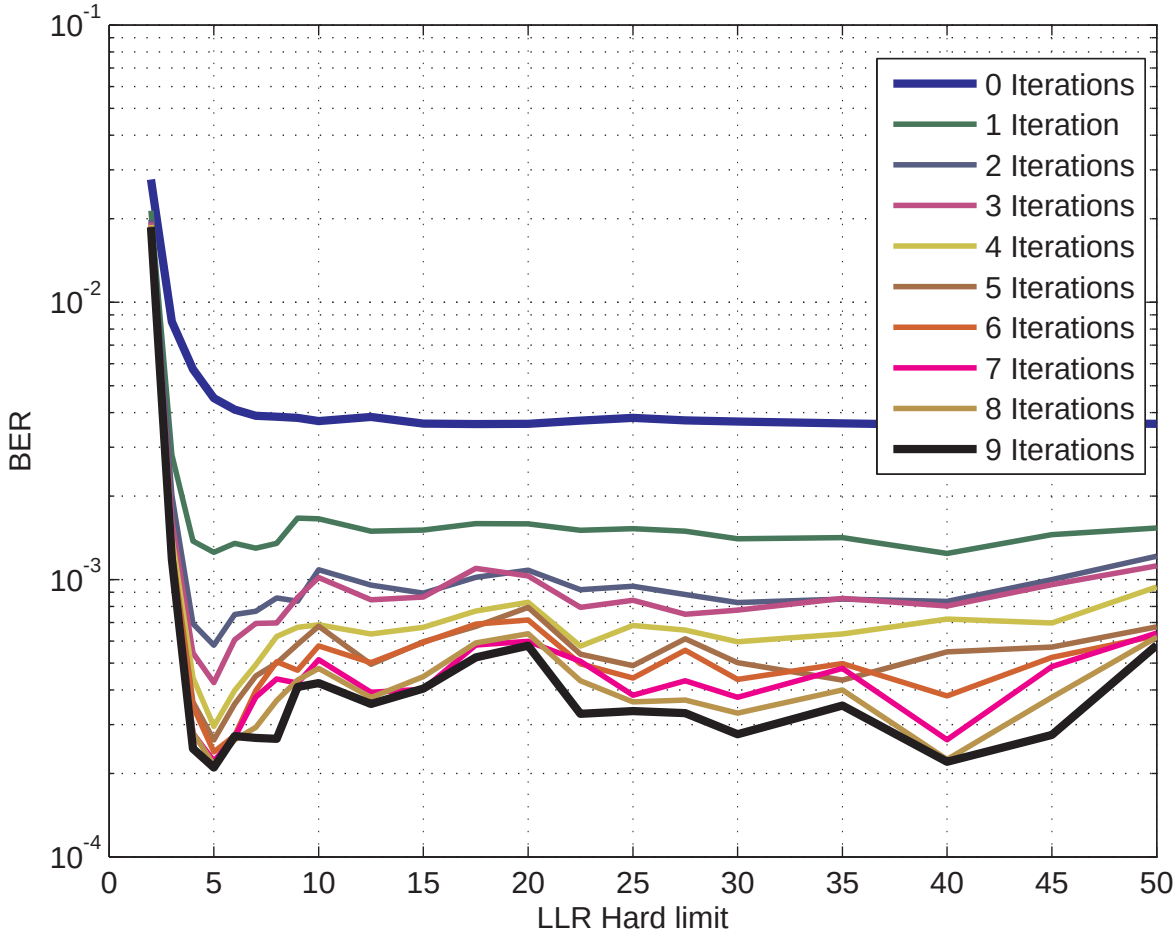

Figure 6.4: Plot of the BER as a function of the hard limit value for the LLRs from the STC decoder showing the BER of each iteration. Code used was the $N = 273$ LDPC code with a $M = 4$, $L = 2$ TAST code and 16QAM signal constellation. $N_{TX} = N_{RX} = 2$. An $E_b/N_0 = 15$dB was used.

seen for higher values of the LLR hard limit value, whereas in fig. 6.3 the performance stayed more or less constant over the interval simulated.

In summary, this chapter initially described overall transmitter and receiver structure incorporating the turbo structure. The overall mathematical model describing the received signal was also detailed. The effects of hard limiting the LLR values from the STF decoder was shown. It was shown that for the 16QAM constellation simulation that a high limit can have a detrimental effect on the performance of the decoder. It is expected that the larger the number of bits being decoded, the sooner a high LLR limit will have a negative influence on the

performance of the system.

It is speculated that this is due to higher inaccuracies in the soft values calculated by the STF decoder. The same should be noticeable with less accurate decoders such as the ZF decoder.

Lastly, the incremental gains offered by iterative decoding was demonstrated with up a hundredfold decrease in the number of errors made. It was also shown that each additional iteration yields diminishing returns. After five iterations it appears as though no significant additional improvements are obtained with each successive iteration.

CHAPTER 7

RESULTS

In this chapter the effects of the various decoding algorithms presented in chapter 3 are presented. The difference between the method of adding a priori information to the ZF, ZF-DF, MMSE and MMSE-DF decoders developed in this dissertation is compared to the RVT method for adding a priori information to said decoders. The effect of including DF is also shown for both the ZF and MMSE decoders. The difference between the ZF and MMSE decoders will also be apparent.

The effect of symbol ordering on the performance of the ZF and ZF-DF decoders are also shown. Lastly, the performance of the SD and the effect of symbol ordering on the computational complexity of the SD is shown.

The BER performance of the various decoders are compared in two ways. Both methods aim to compare the codes asymptotically. The first method is to compare the horizontal difference between two points, each at the same BER. This is generally measured in dB and the lowest BER level common to both, i.e. as close as possible to the asymptotic BER, is used. The second uses the highest E_b/N_0 point that is common to both and compares the BER ratio between the two.

7.1 PERFORMANCE OF THE BACK SUBSTITUTION DECODERS

This section will show the difference between the ZF, ZF-DF, MMSE and MMSE-DF decoders. The main comparison is comparing the method of including a priori information in the decoder. It is shown that iteratively decoded MMSE-DF can with small matrixes reach the performance of the standard sphere decoder.

Finally, in this section the effect of that sorting of the decoding order has on the BER performance of the ZF and ZF-DF decoders are shown. In this section, the decoders employing the method of adding a priori information developed in this dissertation are simply referred to as the ZF, ZF-DF, MMSE and MMSE-DF decoders. Those decoders that use the RVT method of adding a priori information are referred to as the ZF RVT, ZF-DF RVT, MMSE RVT and MMSE-DF RVT decoders.

7.1.1 Comparison between the ZF and ZF RVT decoders

Figure 7.1 shows the performance of the ZF decoders, unsorted, in an iterative decoding environment. The results show that the ZF RVT results in a two orders of magnitude BER reduction as measured at 4 dB. This translates into a gain of approximately 0.25 dB as measured at 10^{-4}. The small gain is due to the steep slope of the BER curve. The ZF decoder does not perform as well, showing only half the gain of the ZF RVT decoder. The sphere decoder is plotted as a reference, and shows a 1.25 dB gain at 10^{-4} to the ZF RVT decoder.

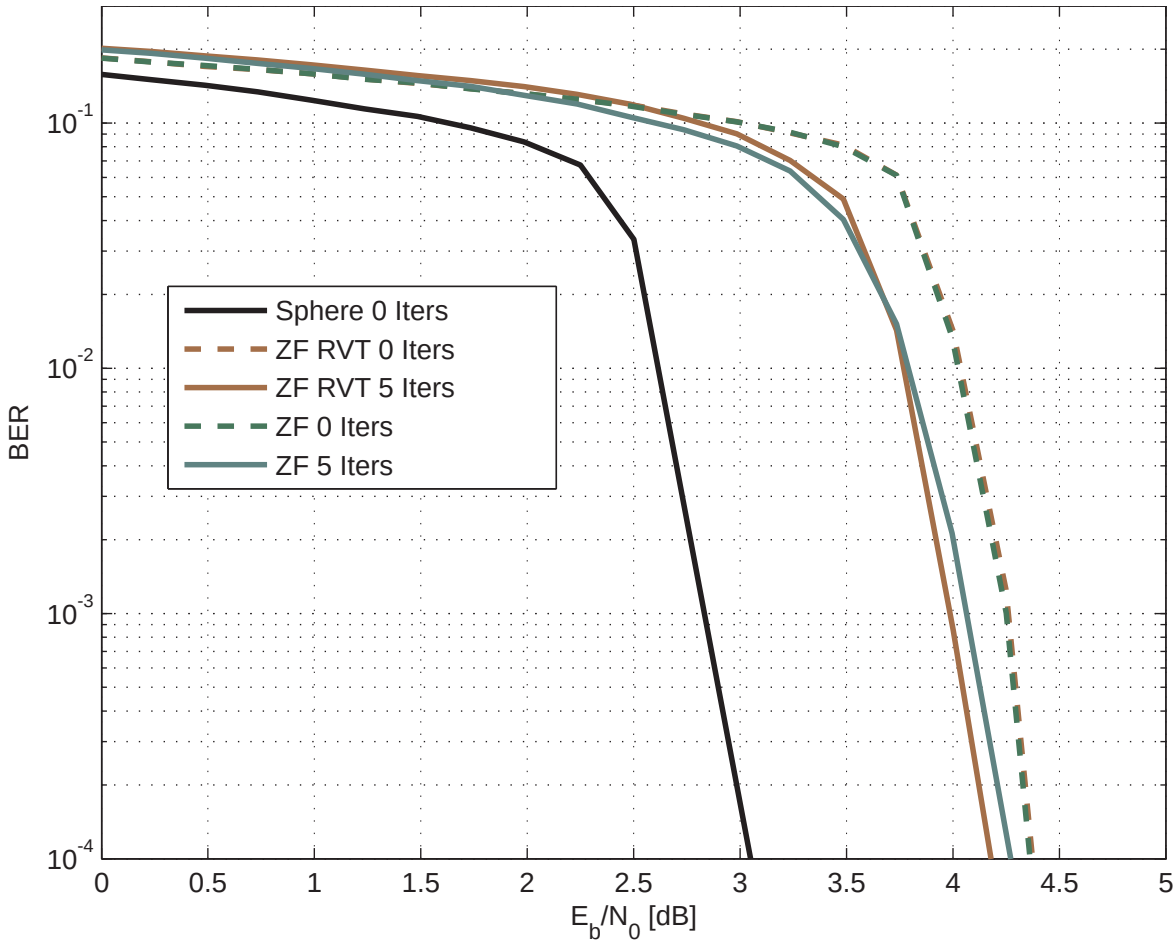

Figure 7.1: BER plot of the ZF decoder comparing the performance of the RVT method of adding a priori information to the method developed. $N_{TX} = N_{RX} = M = 2$. $L = 1$. LDPC $N = 20\,000$.

Doubling of the decoding matrix size by doubling the linear precode matrix size M should result in a gain due to the extra diversity and extra coding gain. In figure 7.2 the sphere decoder does show a gain of approximately 0.6 dB at 10^{-4} BER. The ZF and ZF RVT decoders, on the other hand, show a loss of around 1.5 dB. Once more it is seen that the ZF RVT is marginally better than the ZF decoder.

There does appear to be a discrepancy between the ZF and ZF RVT decoders in that the initial iterations should show the same performance. This may be due to the complex-to-real transform used in the ZF RVT method impacting the calculation of the LLRs.

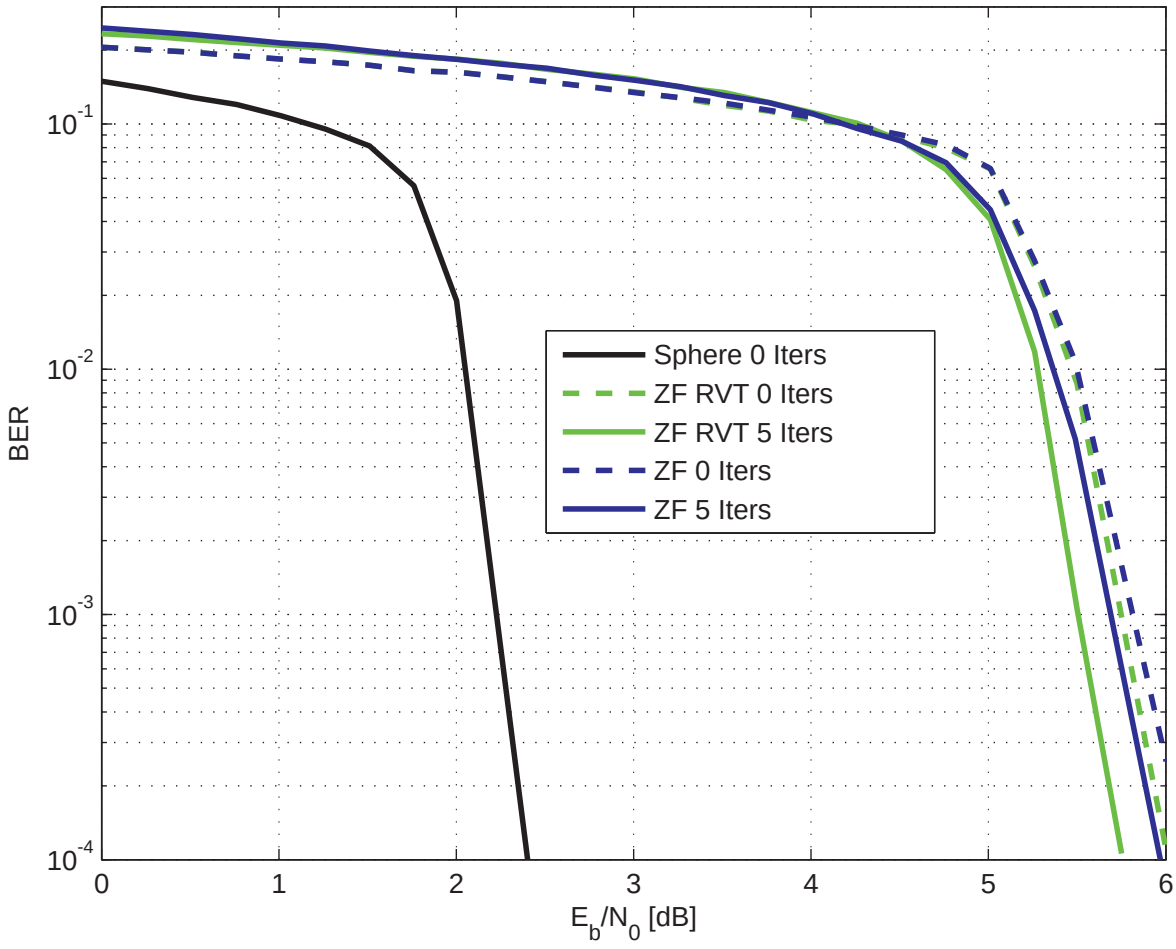

Figure 7.2: BER plot of the ZF decoder comparing the performance of the RVT method of adding a priori information to the method developed. $N_{TX} = N_{RX} = 2$, $M = 4$. $L = 1$. LDPC $N = 20\,000$.

7.1.2 Comparison between the ZF-DF and ZF-DF-RVT decoders

In figure 7.3 the effect of adding decision feedback to the ZF decoder can be seen. Comparing figure 7.3 with figure 7.1 it can be seen that decision feedback adds a gain of approximately 0.4 dB to the initial iteration as measured at a BER of 10^{-4}. After five iterations the gain is 0.7 dB.

It is also interesting to note that the ZF-DF and ZF-DF RVT decoders perform similarly after five iterations, showing a gain of approximately 0.5 dB or more than two orders of magnitude reduction in the BER.

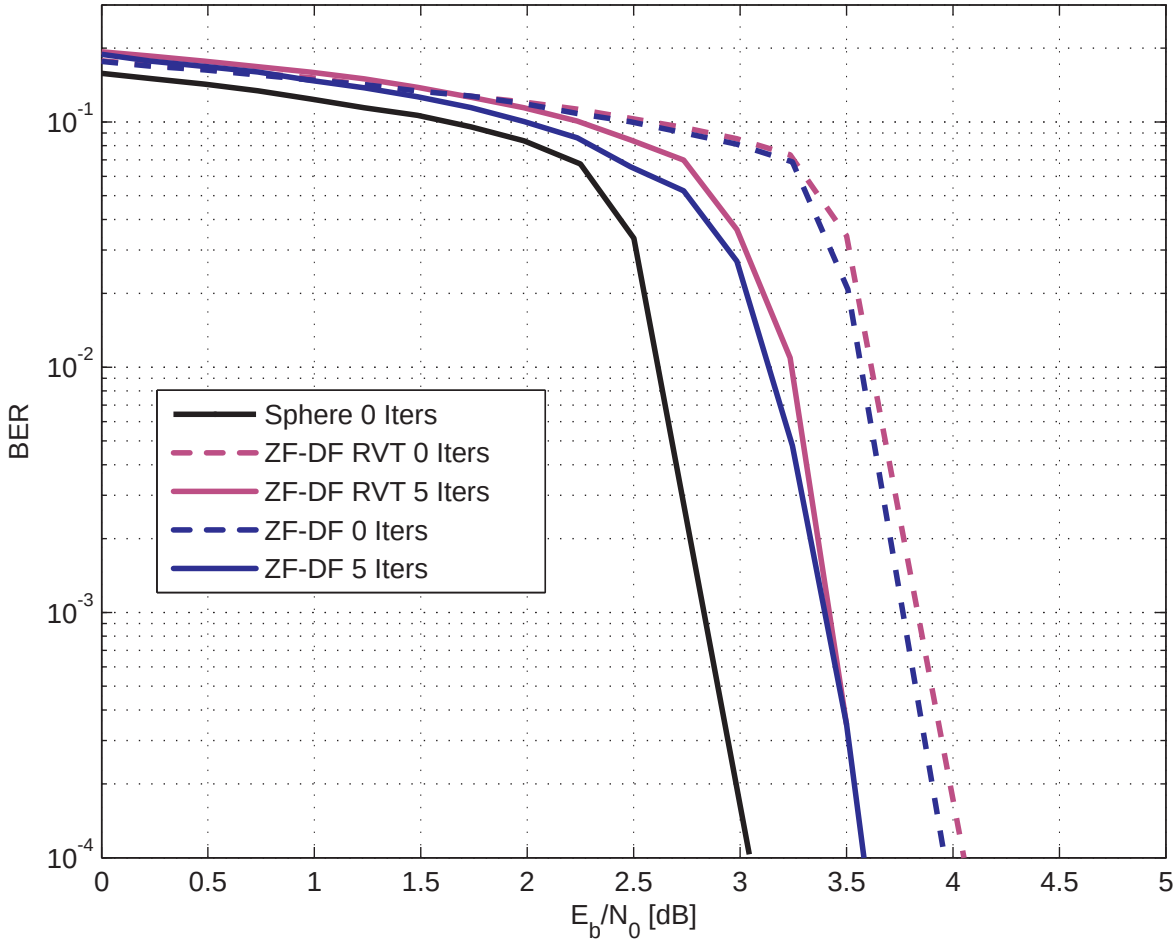

Figure 7.3: BER plot of the ZF-DF decoder comparing the performance of the RVT method of adding a priori information to the method developed. $N_{TX} = N_{RX} = 2$, $M = 2$. $L = 1$. LDPC $N = 20\,000$.

Figure 7.4 shows that doubling the linear precode size has a detrimental effect on the ZF-DF and ZF-DF RVT decoders. The decoders lose approximately 0.6 dB in gain with the initial iteration at a BER of 10^{-4}. The gain after five iterations is still approximately 0.4 dB. The initial loss is hypothesised to be due to the increased size of the channel matrix that needs to be inverted. The bigger the matrix, the larger the effect of an incorrectly decoded symbol as the error propagates through the back substitution.

The effect of adding decision feedback results in both decoders performing the same after five iterations.

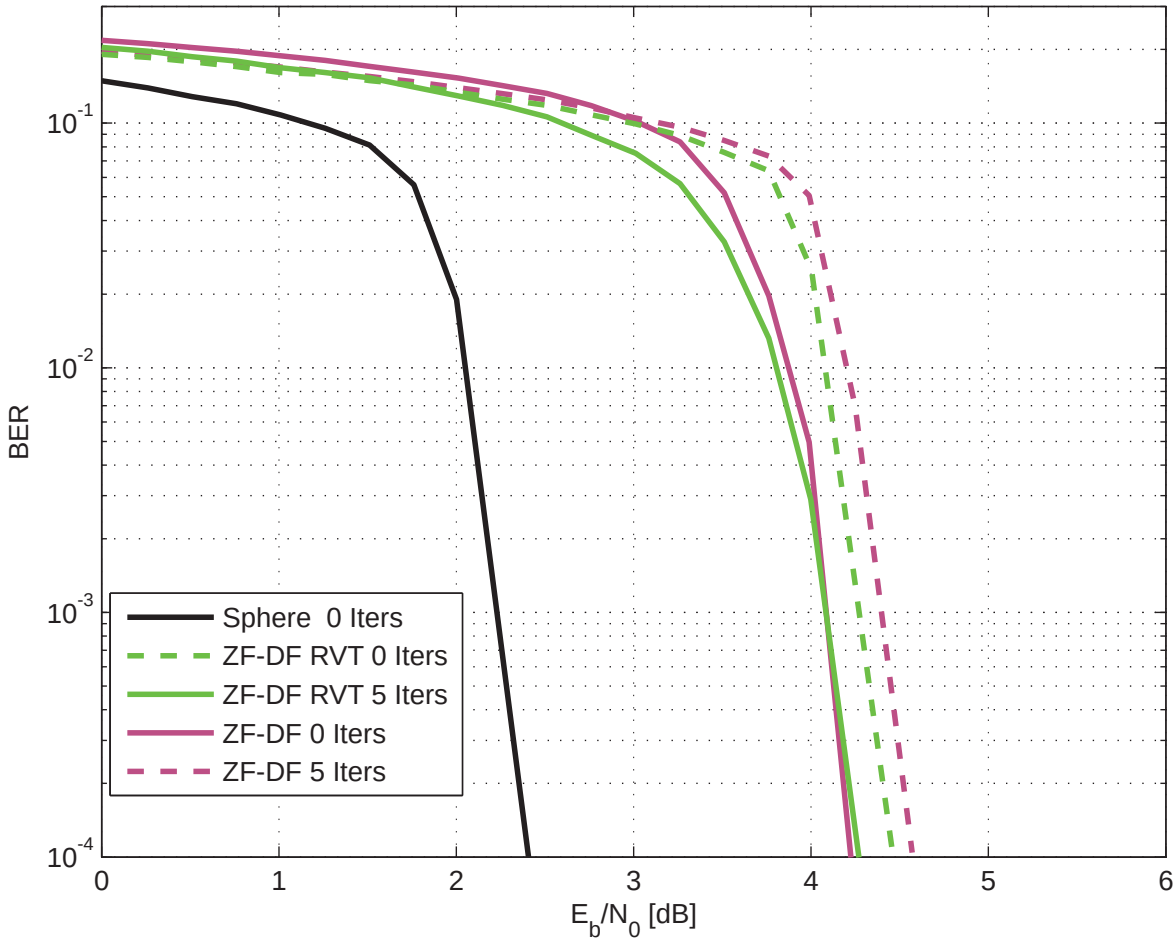

Figure 7.4: BER plot of the ZF-DF decoder comparing the performance of the RVT method of adding a priori information to the method developed. $N_{TX} = N_{RX} = 2$, $M = 4$. $L = 1$. LDPC $N = 20\,000$.

7.1.3 Comparison between the MMSE and MMSE-RVT decoders

The effect of using the SNR in the decoder by using MMSE decoding is shown in figure 7.5. Comparing with figure 7.1 shows a gain in the region of 0.6 dB as measued at a BER of 10^{-4}. The MSME RVT decoder outperforms the MMSE decoder after five iterations by approximately 0.4 dB at a BER of 10^{-5}. The MMSE shows approximately half the gain after five iterations.

After five iterations the MMSE RVT decoder is approximately 0.4 dB from the sphere decoder as measured at a BER of 10^{-5}.

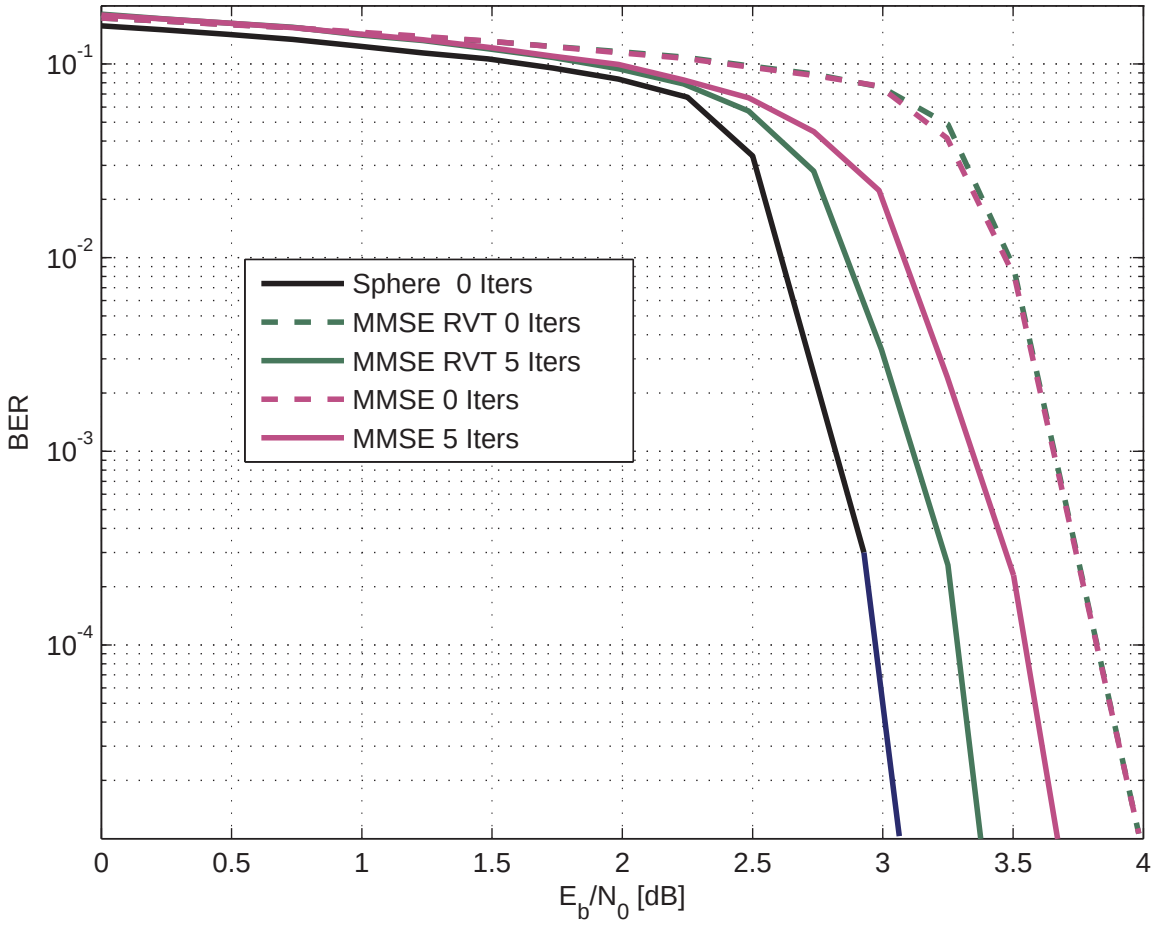

Figure 7.5: BER plot of the MMSE decoder comparing the performance of the RVT method of adding a priori information to the method developed. $N_{TX} = N_{RX} = 2$, $M = 2$. $L = 1$. LDPC $N = 20\,000$.

The effect that doubling of the linear precode size has on the decoders is shown in figure 7.6. In contrast to the ZF and ZF RVT decoders the MMSE and MMSE RVT decoders show no significant performance loss at a BER of 10^{-3} due to a doubling of the linear precode size.

The MMSE decoder does, however, show decreased performance after five iterations showing a gain of only 0.2 dB at a BER of 10^{-3} versus a gain of 0.4 dB at the same BER in figure 7.5.

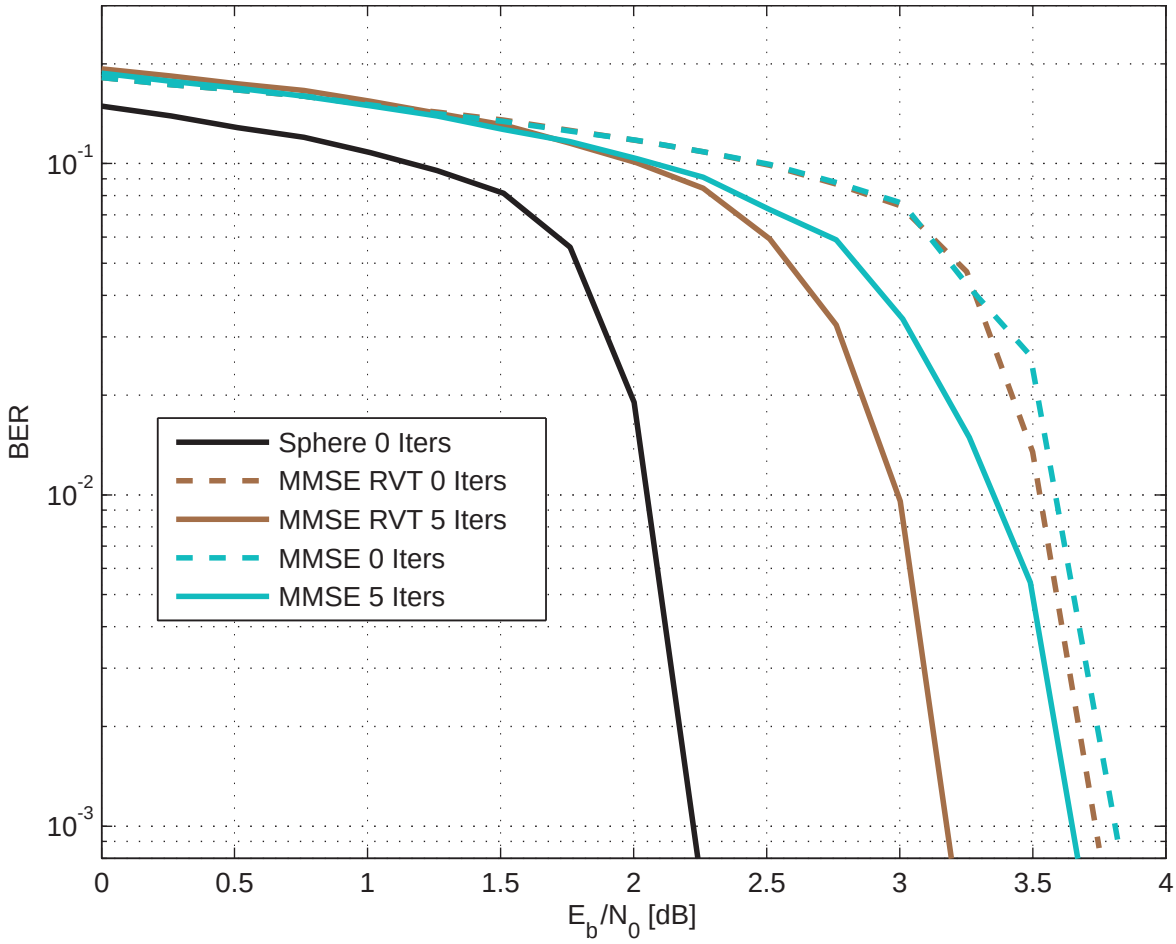

Figure 7.6: BER plot of the MMSE decoder comparing the performance of the RVT method of adding a priori information to the method developed. $N_{TX} = N_{RX} = 2$, $M = 4$. $L = 1$. LDPC $N = 20\,000$.

7.1.4 Comparison between the MMSE-DF and MMSE-DF-RVT decoders

Figure 7.7 shows the effect that the addition of decision feedback has on the MMSE and MMSE RVT decoders. The MMSE-DF RVT shows a gain of approximately 0.4 dB at 10^{-3} over the MMSE RVT decoder and is close to the performance of the SD after five iterations. The MMSE-DF shows approximately two thirds the gain of the MMSE-DF RVT decoder.

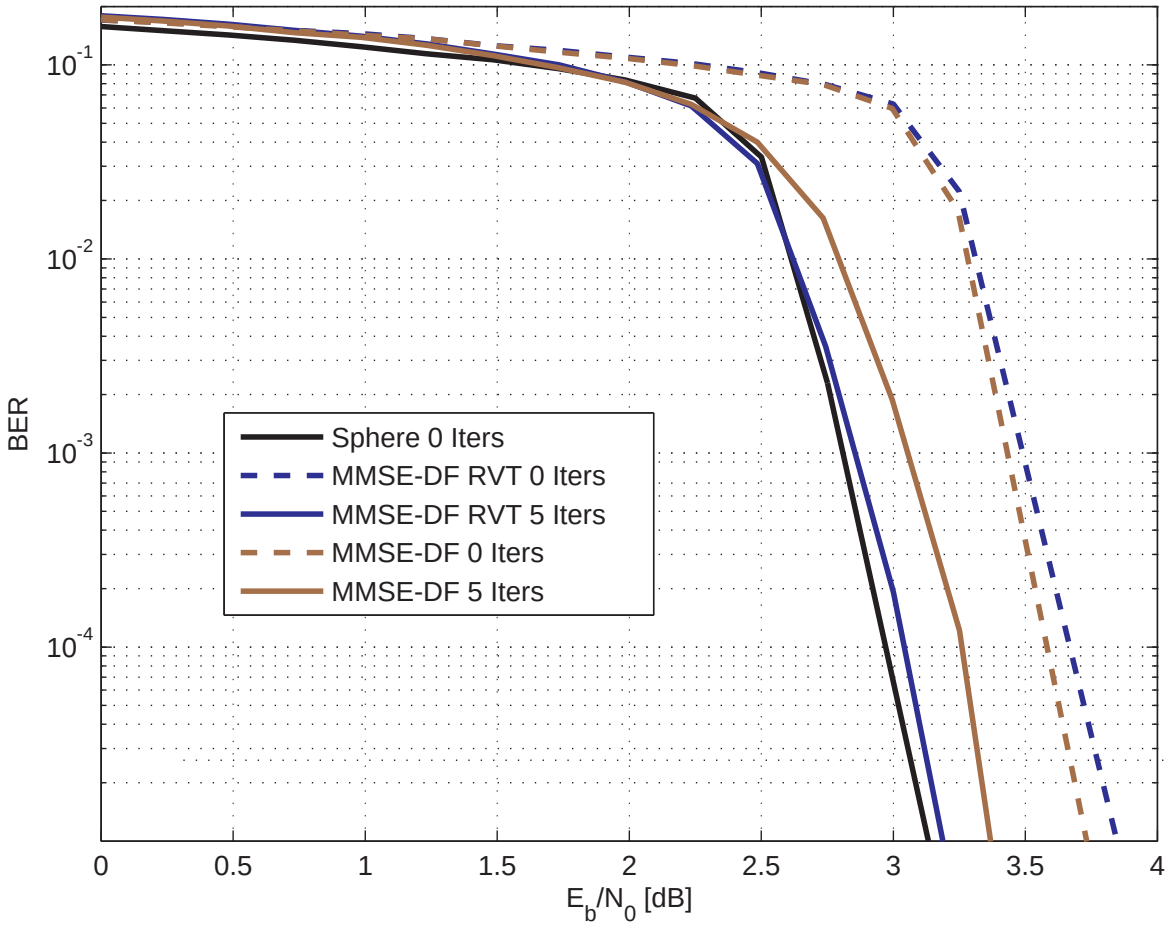

Figure 7.7: BER plot of the MMSE-DF decoder comparing the performance of the RVT method of adding a priori information to the method developed. $N_{TX} = N_{RX} = 2$, $M = 2$. $L = 1$. LDPC $N = 20\,000$.

Figure 7.8 shows that the MMSE type decoders do not show a similar reduction in BER performance due to a doubling of the linear precode size. The MMSE-DF RVT method shows a modest gain of approximately 0.1 dB at a BER of 10^{-3} after five iterations. The MMSE-DF shows a minor loss.

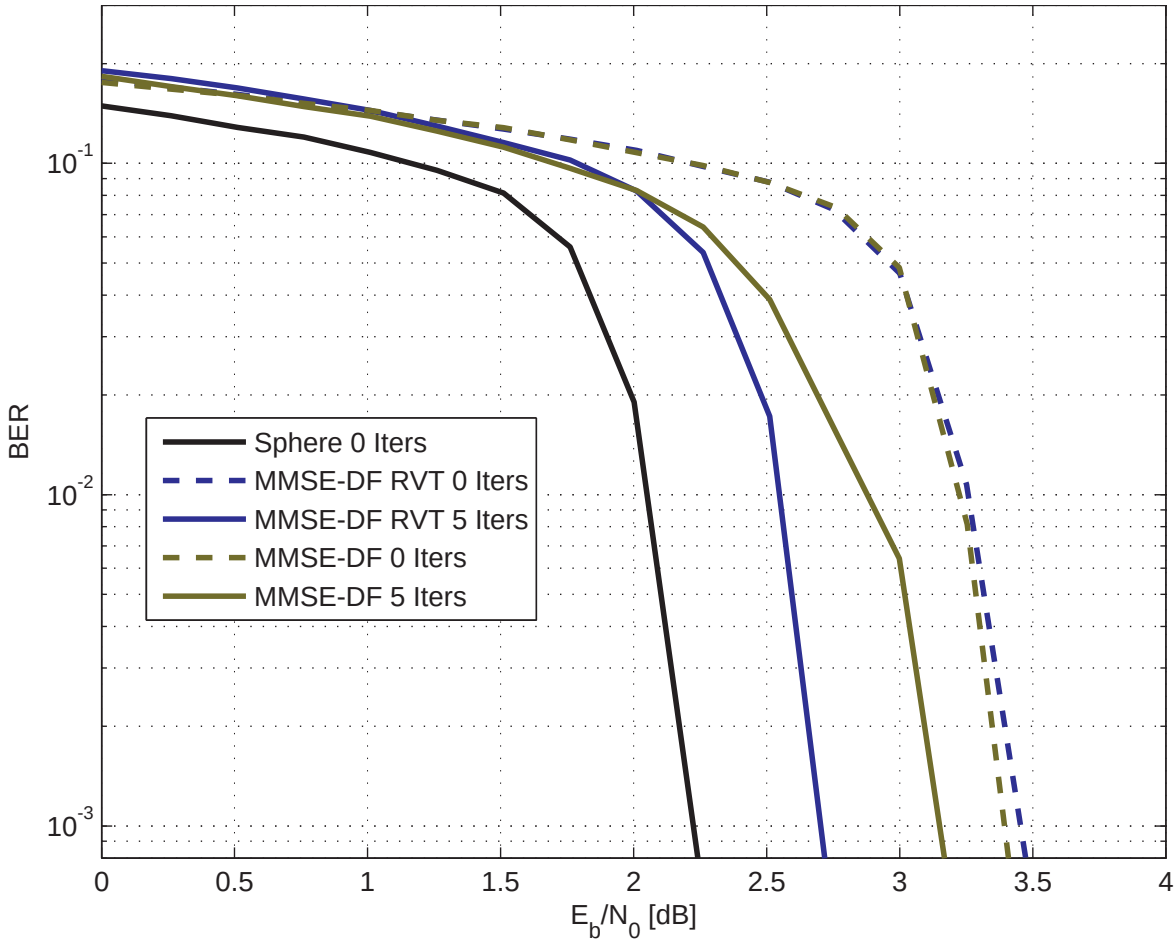

Figure 7.8: BER plot of the MMSE-DF decoder comparing the performance of the RVT method of adding a priori information to the method developed. $N_{TX} = N_{RX} = 2$, $M = 4$. $L = 1$. LDPC $N = 20\,000$.

7.1.5 Effect of symbol sorting on the ZF and ZF-DF decoders

The BER performance of the ZF and ZF-DF decoders for a smaller LDPC code and a larger TAST code for QPSK system for various symbol ordering strategies is shown in figure 7.9. The slope of the curve is significantly less than those of figures 7.1 to 7.8. This can be explained by the difference in LDPC code length. It was shown in chapter 4 that an LDPC code is able to extract diversity. Thus, the smaller LDPC code performs significantly worse in terms of diversity than the longer code. As a side effect, although the BER reduction is less, the gain in dB is far greater.

Figure 7.9 shows that sorting does not have a significant effect in the BER performance of the ZF decoder. The SQRD algorithm is an exception in that it actually seems to perform worse.

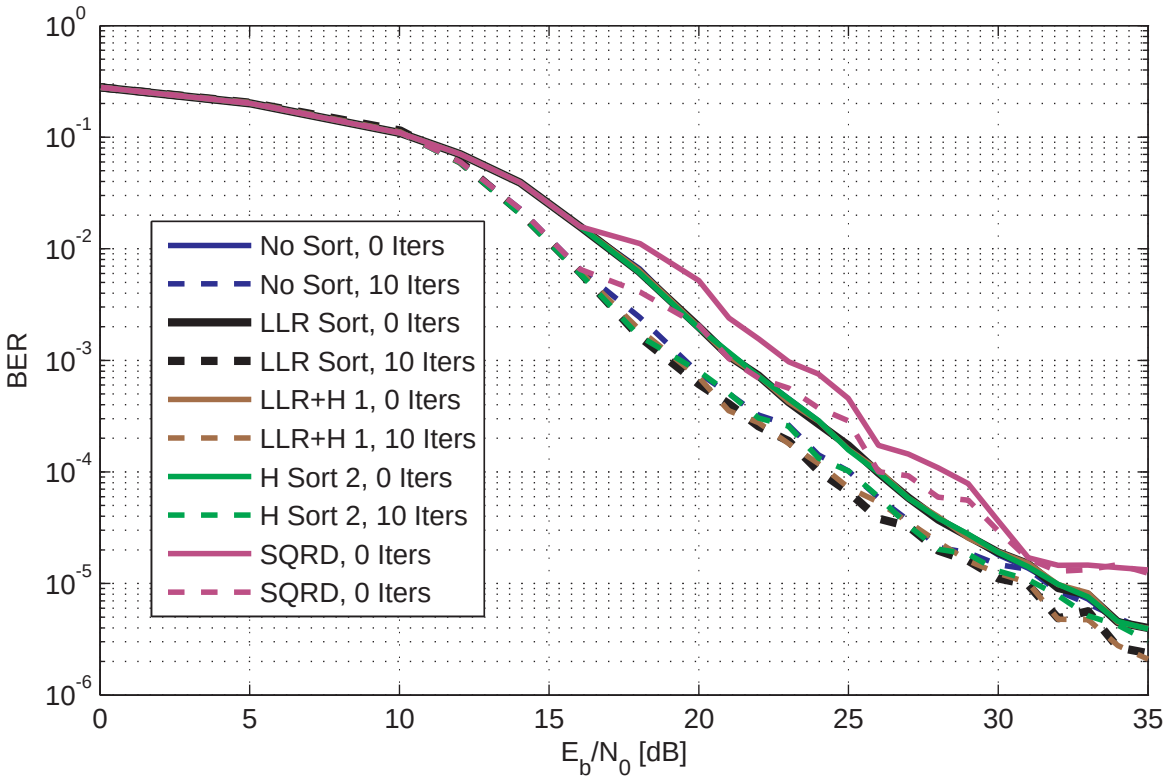

Figure 7.9: BER plot of the effect that various sorting strategies has on the ZF decoder for a QPSK system. $N_{TX} = N_{RX} = L = 2$. $M = 4$, LDPC $N = 273$.

As can be seen in figure 7.10 the addition of decision feedback has a significant effect on the BER performance of the decoder. The addition of decision feedback also makes a material difference to the various symbol ordering strategies. The method based solely on the a priori information, yields the best results with a gain of 12.5 dB at a BER of 10^{-5}. The method based on LLR sorting is followed by the combined LLR+H method. The SQRD algorithm shows now significant improvement over the unsorted case.

It is expected that due to the limiting of the LLRs, the gain of the methods incorporating a priori information will decrease as compared to the unsorted case. This is due to the noise scaling term in the MAP equation 3.58 decreasing the contribution of the LLRs. Scaling of the LLR limit as a function of the SNR might compensate for this effect.

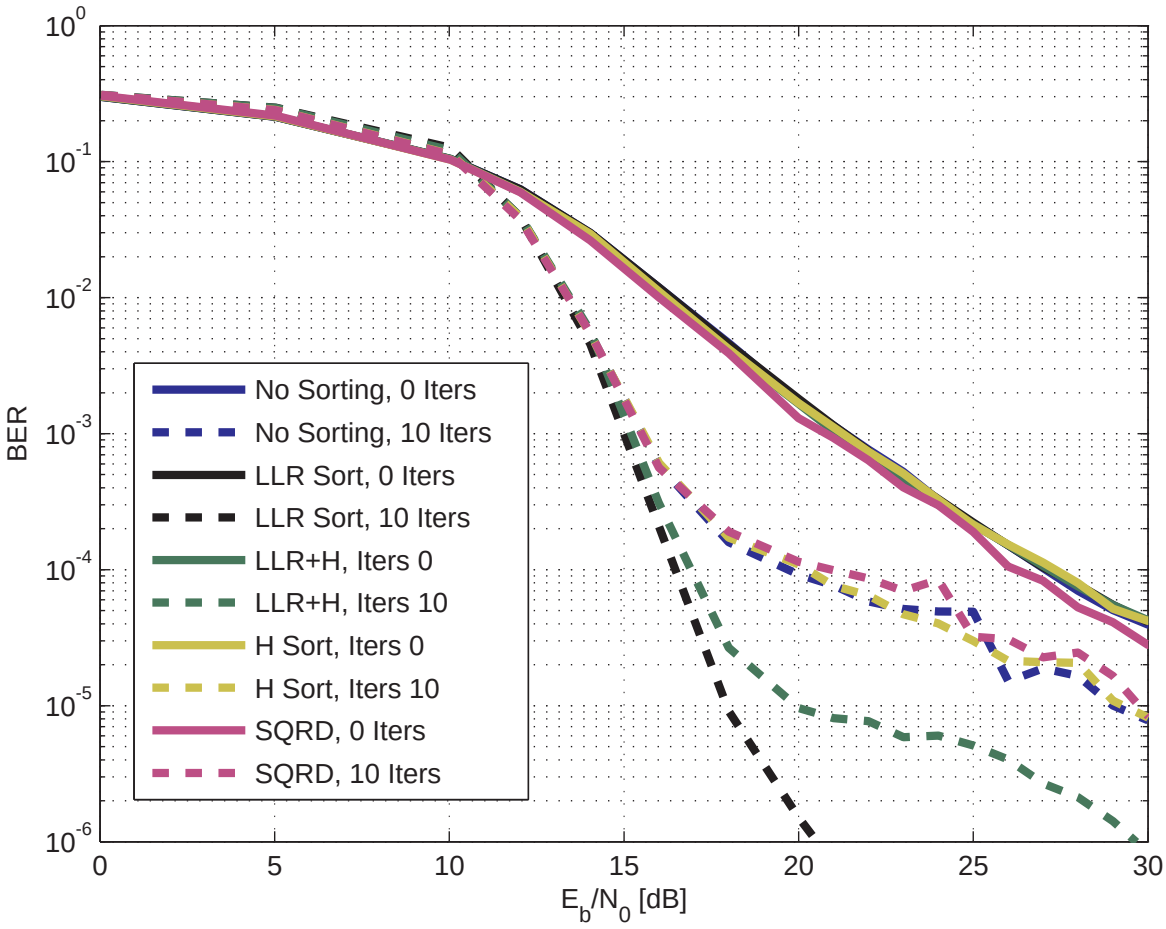

Figure 7.10: BER plot of the effect that various sorting strategies has on the ZF-DF decoder for a QPSK system. $N_{TX} = N_{RX} = L = 2$. $M = 4$, LDPC $N = 273$.

Figure 7.11 shows the effect of sorting on the ZF-DF decoder with a 16QAM signal constellation. It can be seen that the methods using a priori information outperform the other methods. It can be seen that after ten iterations the gain is approximately 5 dB at a BER of 10^{-3} over the non-iteratively decoded performance.

The sorting strategy based solely on a priori information shows a gain of 8 dB at a BER of 10^{-4} over the unsorted and SQRD based decoder. This gap between the two decoders is expected to remain constant and perhaps even decrease slightly at much higher SNRs due to the hard limiting of the LLRs from the STFC decoder. Using an adaptive hard limit based on SNR may yield improvements at higher SNRs.

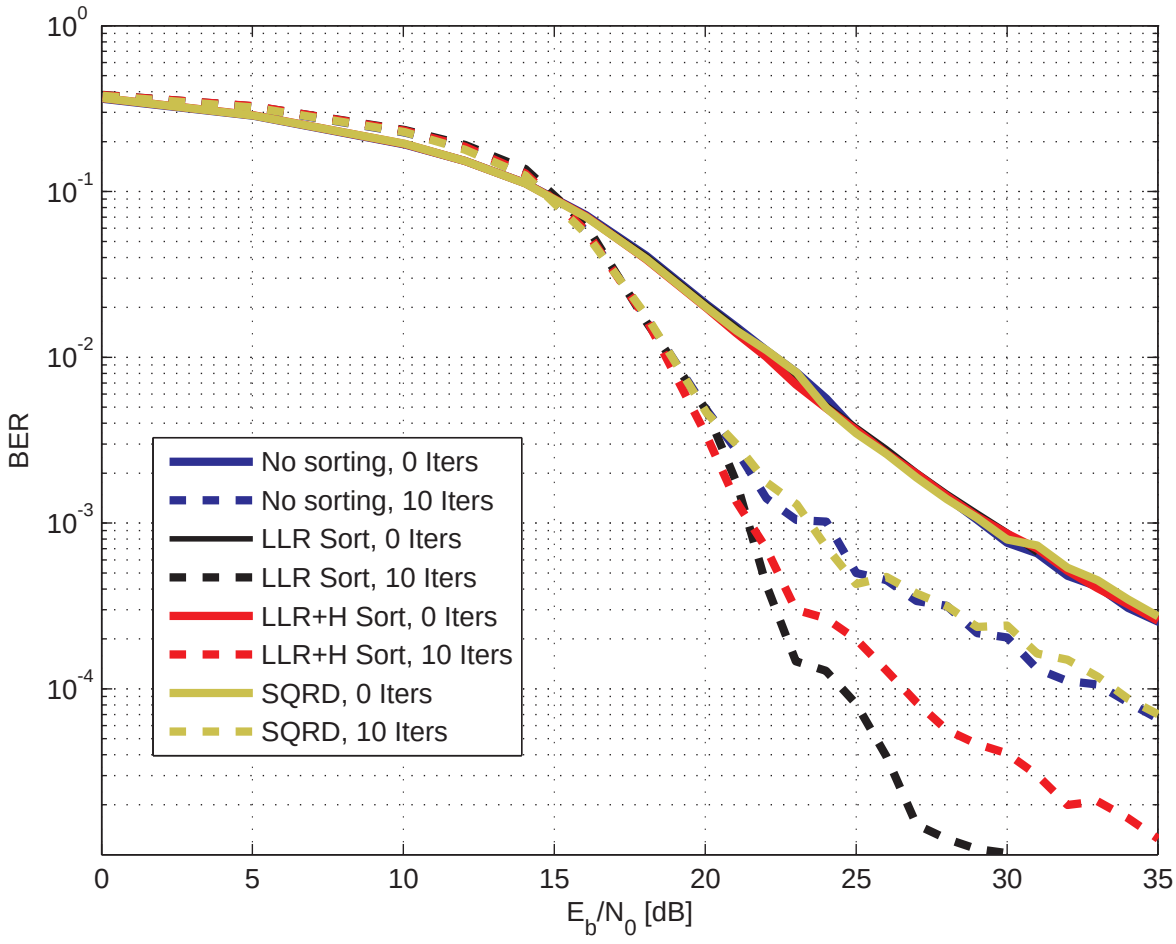

Figure 7.11: BER plot of the effect that various sorting strategies has on the ZF-DF decoder for a 16QAM system. $N_{TX} = N_{RX} = L = 2$. $M = 4$, LDPC $N = 273$.

7.2 SPHERE DECODER PERFORMANCE

In this section the performance of the a priori sphere decoder developed in this dissertation is shown. The BER performance of the iterative sphere decoder is shown as well as the additional complexity that the inclusion of a priori information has on the decoder. Lastly, the effect of sorting the symbol order has on the computational complexity of the sphere decoder is shown.

The BER performance of the sphere decoder can be seen in figures 7.12. The figure shows a hundredfold reduction in the BER for both QPSK and 16QAM signal constellations. Due to the steep slope of the BER curve, this translates into a 1 dB gain as measured at a BER of 10^{-4}.

Notice that the slope of the BER curve in figure 7.12 is much steeper than the BER curves in figure 7.9, 7.10 and 7.11. This shows that the ZF based decoders are unable to fully extract the available diversity in the system. The ZF-DF decoder with LLR sorting best approximates the slope of the sphere decoder.

The simulation were halted when at least a thousand errors has been made by the last iteration of the decoder at a BER of 10^{-6} and 10^{-5} for the QPSK and 16QAM systems respectively. This results in the abrupt ending of the BER curves of the initial iteration.

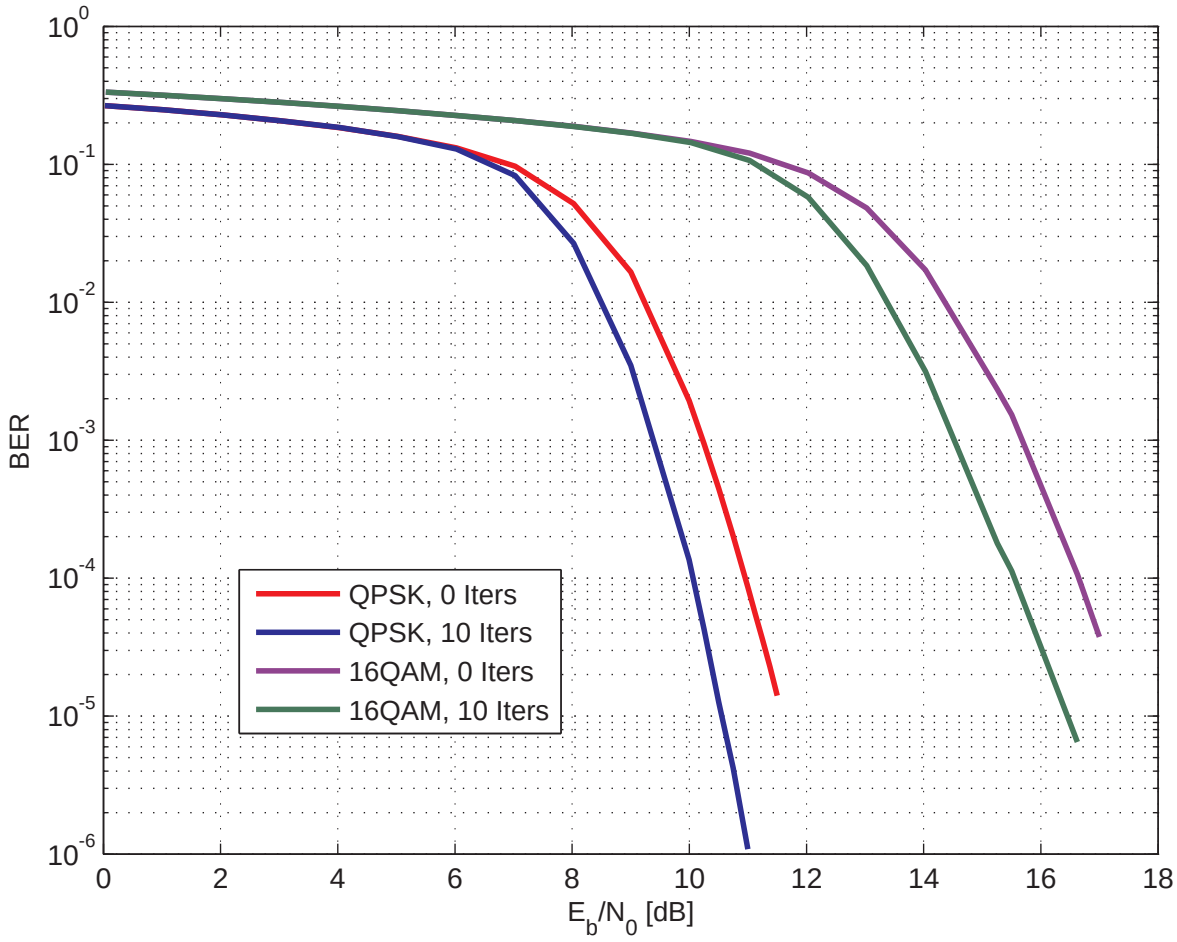

Figure 7.12: BER plot of the performance of the iterative sphere decoder for QPSK and 16-QAM. $N_{TX} = N_{RX} = L = 2$. $M = 4$, LDPC $N = 273$.

Figure 7.13 shows the number of `SYMLIST()` function calls made over ten iteration for the iterative sphere decoder when decoding QPSK and 16QAM signal constellations. The number of calls to this function is used as a performance metric, as described in section 3.7.3.2. It can be seen that the sphere decoder is most computationally expensive at low SNRs. In a communication system employing adaptive signal constellations once the E_b/N_0 is less than say 14 dB, the system would employ a lower order signal constellation. Thus keeping the actual decoding complexity low and resulting in a saw-tooth pattern.

At very high SNR, the complexity of the sphere decoder approaches that of the ZF decoder. By limiting the number of `SYMLIST()` function calls, a compromise between performance and complexity may be achieved. The performance would at worst be that of the ZF-DF

decoder.

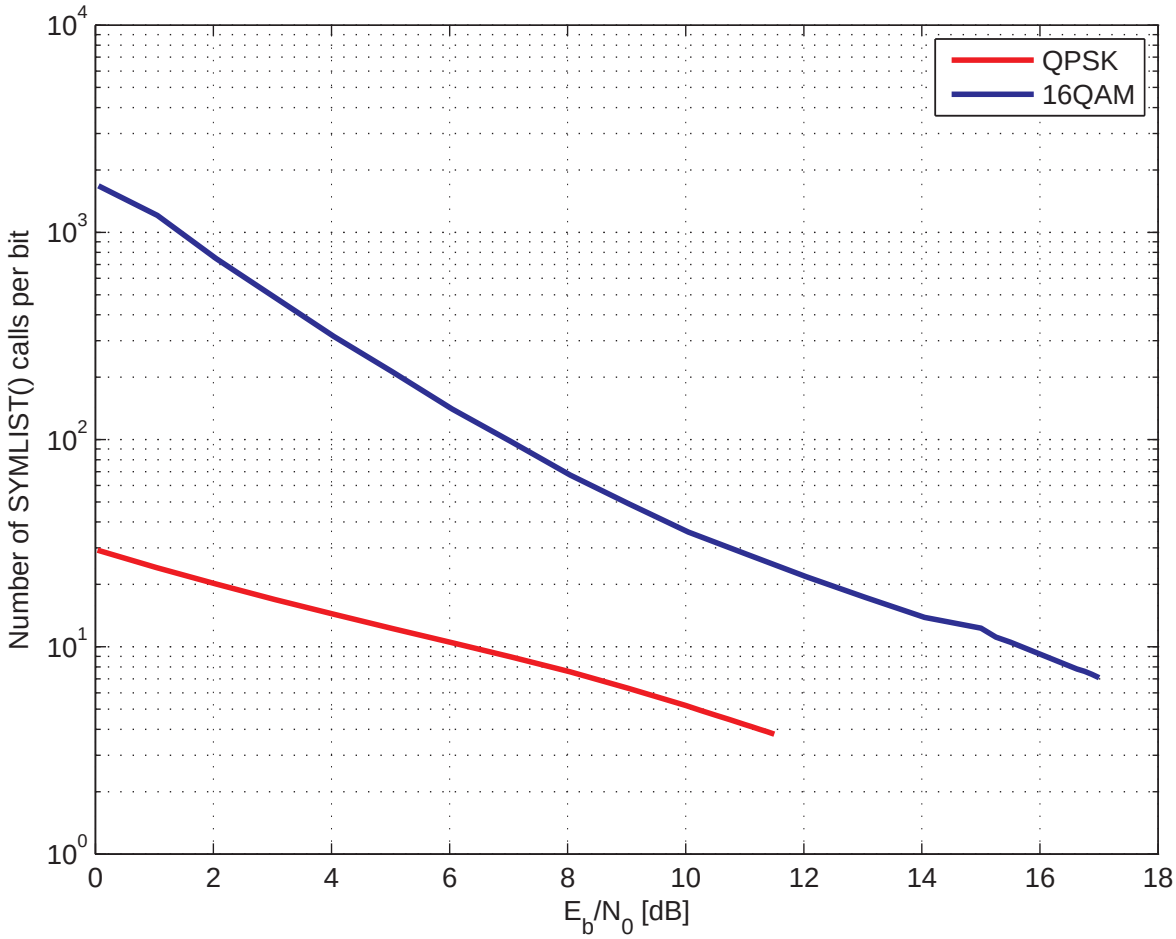

Figure 7.13: Plot of the number of `SYMLIST()` function calls for the unsorted a priori sphere decoder for QPSK and 16QAM. $N_{TX} = N_{RX} = L = 2$. $M = 4$, LDPC $N = 273$.

The added complexity that the inclusion of a priori information has on the sphere decoder, without any sorting, is shown in figure 7.14. The figure shows that the second iteration, the first to include a priori information from the FEC decoder, shows an up to sixty percent increase in computations. This is reduced on subsequent iterations with the fifth iteration showing an increase of approximately forty-five percent.

Notice that at low SNR hardly any increase in complexity is seen, whilst at high SNRs the addition of a priori information reduces the computations required by the sphere decoder. At high SNRs the standard sphere already shows a significant decrease in computations as

shown in figure 7.13.

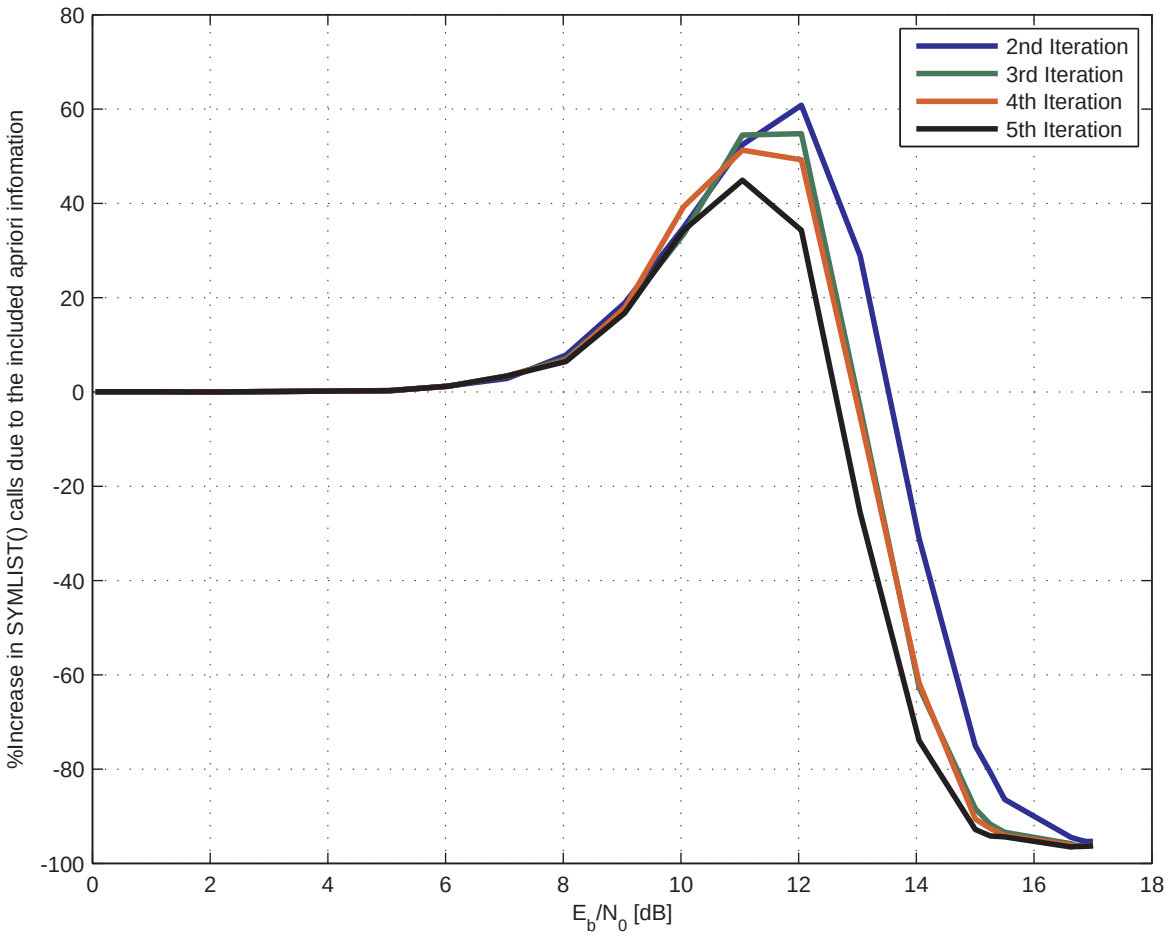

Figure 7.14: Plot of the percentage increase of `SYMLIST()` function calls for the unsorted a priori sphere decoder as compared to a standard sphere decoder. Plot shows the additional complexity due to the inclusion of a priori information for a 16QAM system. $N_{TX} = N_{RX} = L = 2$. $M = 4$, LDPC $N = 273$.

Figure 7.15 shows the effect that the various sorting strategies have on the complexity of the a priori sphere decoder as compared to the standard decoder for a single iteration. The figure shows that sorting the symbols solely according to a priori information add the least additional complexity. This confirms the hypothesis made in section 3.7 that sorting based on the a priori information should reduce the impact of the addition of a priori information on the sphere decoder.

It is also apparent that the methods that include a priori information outperform those that do not.

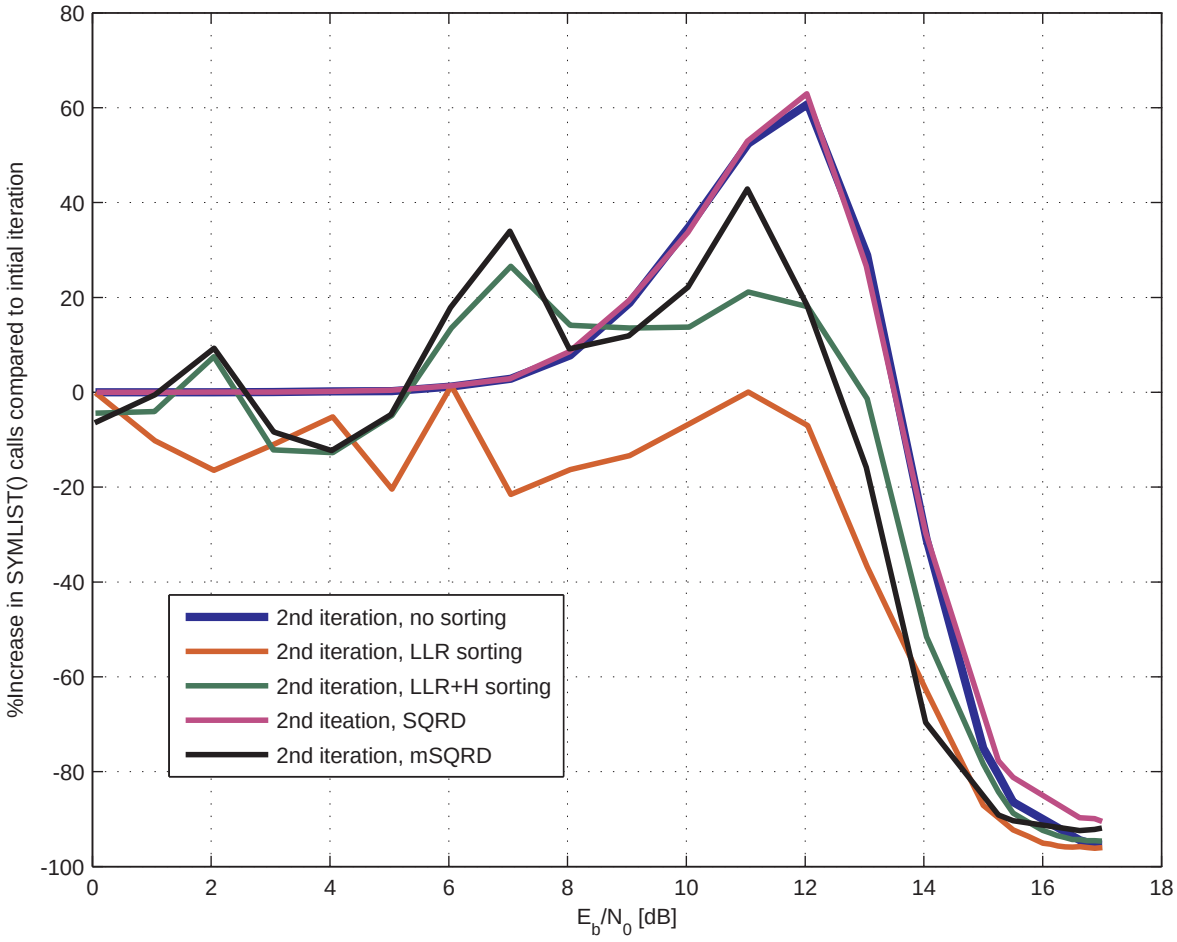

Figure 7.15: Plot of the percentage increase in `SYMLIST()` function calls for the various sorting metrics for the a priori sphere decoder as compared to the standard sphere decoder for a 16QAM system. $N_{TX} = N_{RX} = L = 2$. $M = 4$, LDPC $N = 273$.

For a QPSK signal constellation, the overall decrease in computations over ten decoding iterations for the various sorting strategies is shown in fig. 7.16. The figure shows that the greatest reduction, around 42 percent, is for the modified SQRD algorithm followed by the suboptimal algorithm based on the column norms of **H** and a priori information. Both SQRD algorithms have an initial gain of fifteen percent.

The LLR sorting algorithm does not fulfil its promise of having the lowest complexity as presented in fig. 7.15. It appears that the cumulative gains of the other algorithms exceed that of the LLR sorting strategy.

It is expected that at very high SNRs, as the complexity of the sphere decoder approaches that of the ZF decoder, the percentage difference of the various algorithms will approach zero.

The effect of the various symbol sorting strategies on the complexity of the iterative sphere decoder with a 16QAM signal constellation is shown in fig. 7.17. It is shown that the SQRD based algorithms show a significant reduction in complexity. The initial reduction is approximately 35 percent with the modified SQRD algorithm achieving a peak reduction of up to seventy percent.

It is informative to compare figs. 7.13 and 7.17. Both figures are for a 16QAM signal constellation. Figure 7.13 shows the additional computational complexity from adding a priori information to the classical sphere decoder without any symbol reordering. The additional complexity is up to sixty percent. Figure 7.17 shows that the mSQRD algorithm shows a reduction of seventy percent. This translates into an overall decrease in complexity of 25 percent as compared to ten iterations of a standard sphere decoder.

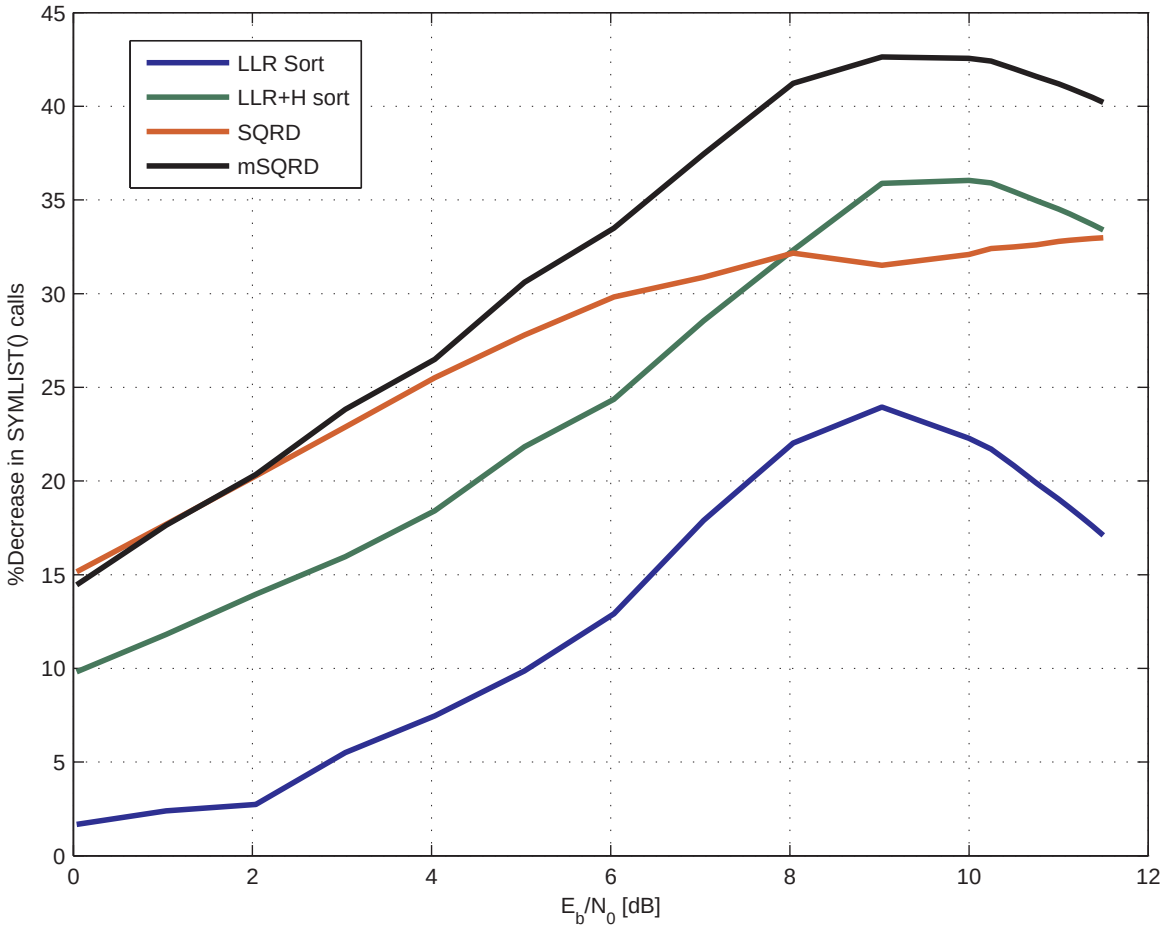

Figure 7.16: Plot of the percentage decrease, over the unsorted sphere decoder, of the number of `SYMLIST()` calls made in the sphere decoder for various sorting strategies for a QPSK system. The plot is for the total increase of calls over five iterations. $N_{TX} = N_{RX} = L = 2$. $M = 4$, LDPC $N = 273$.

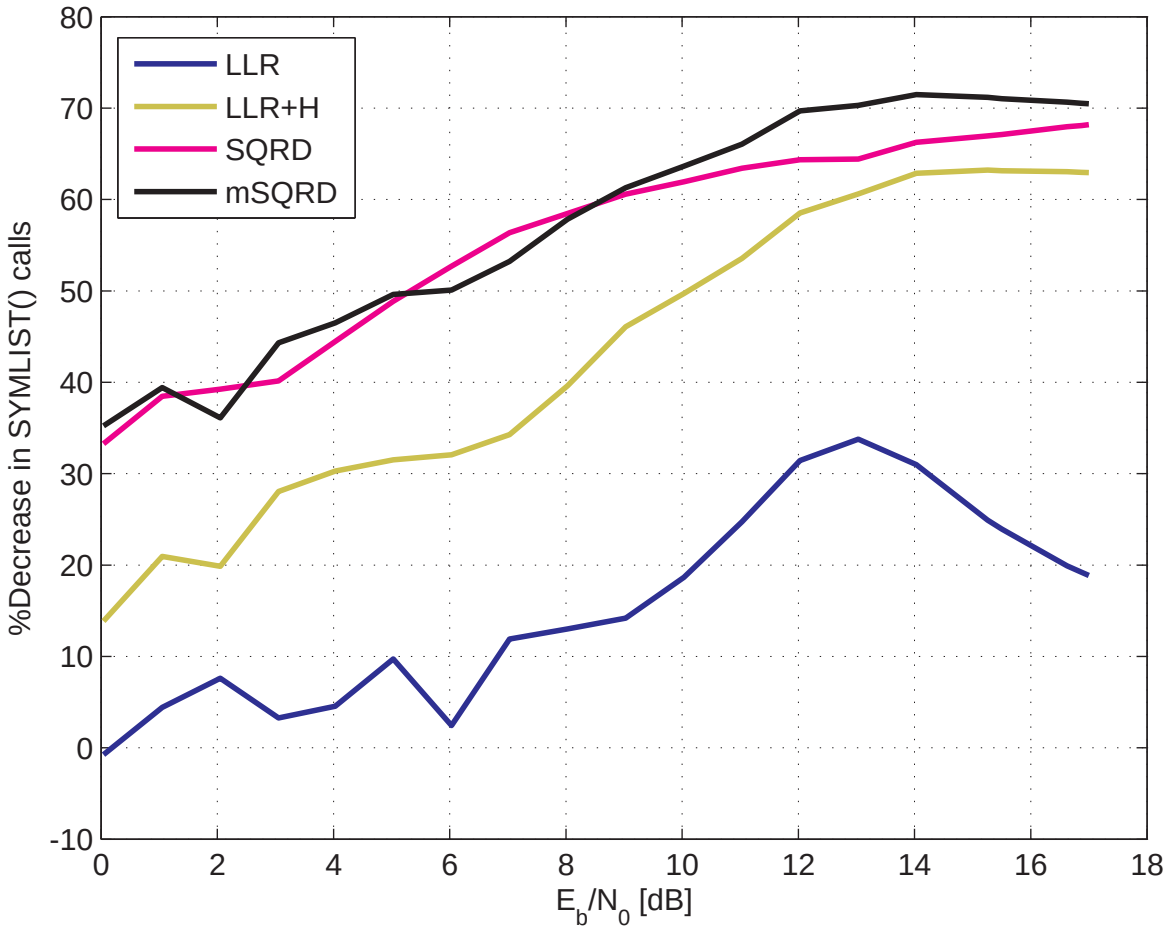

Figure 7.17: Plot of the percentage decrease, over the unsorted sphere decoder, of the number of `SYMLIST()` calls made in the sphere decoder for various sorting strategies for a 16QAM system. The plot is for the total increase of calls over five iterations. $N_{TX} = N_{RX} = L = 2$. $M = 4$, LDPC $N = 273$.

7.2.1 Performance of the SD-HNN decoder

Figure 7.18 shows the performance of the SD-HNN decoder. The figure shows that after six iterations the SD-HNN is able to match the performance of the original sphere decoder. The exact gains in computational complexity is dependant on the FEC decoder used. The case studied by Louw et al. [63, 64] showed that the sphere decoder used five times more BP iterations than each SD-HNN iteration. It is expected that simply using ZF-DF for the alternate

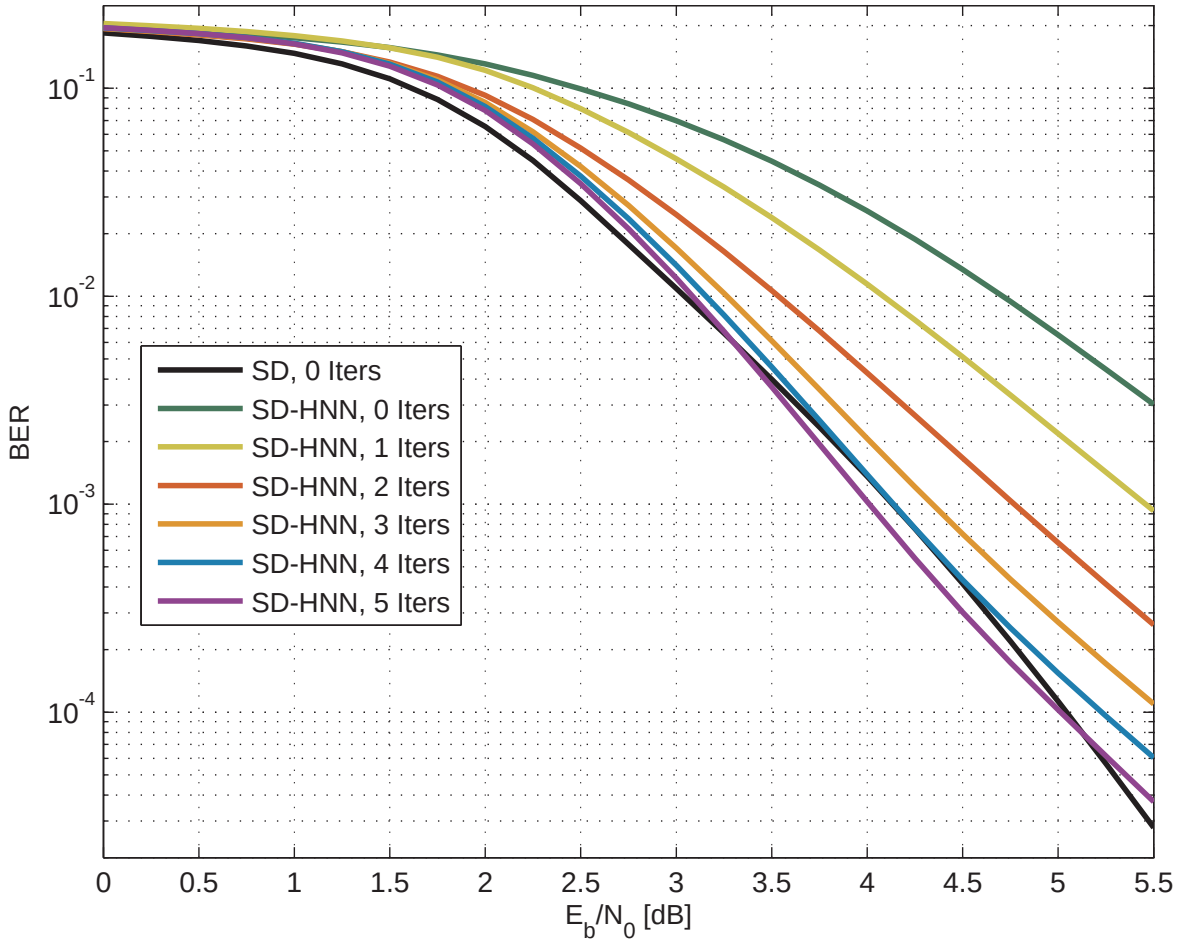

Figure 7.18: BER plot of the performance of the SD-HNN decoder versus the standard sphere decoder showing the performance per iteration. $N_{TX} = N_{RX} = L = 2$. $M = 4$.

hypothesis in the LLR computation will yield improved results at a lower complexity. This is because during the optimisation of the annealing parameters (see section 3.9.3) of the HNN the ZF decoder was used as a reference and was found to outperform the HNN. In addition,

making use of the tree structure the overall complexity for the ZF-DF can be reduced when calculating the LLRs. This is achieved by calculating the LLRs in reverse order. Due to the tree structure the upper symbols will stay the same. This is not necessarily the case when the sphere decoder is used to calculate the alternate hypothesis.

7.3 DISCUSSION OF THE RESULTS

In this chapter the performance of the various decoders described in chapter 3 was given. The results show that the RVT method of adding a priori information generally outperform the method developed in this dissertation. This is especially the case for decoders not employing decision feedback. This is perhaps due to the fact that the a priori information in one symbol affects all symbols being decoded in the RVT method through the CSI matrix in eqn. 3.127.

Once decision feedback is incorporated, the difference between the two methods is reduced though still remains significant for the MMSE-DF decoder. It was also shown that the iterated MMSE-DF RVT decoder is able to approach the sphere decoder in BER performance, at least when using a large LDPC code. For small LDPC codes it appears that the ZF and ZF-DF decoders are not able to make full use of the available diversity provided by the TAST code.

Furthermore it was shown that symbol sorting has a significant effect on the performance of the ZF-DF decoder with gains of up to 11 dB seen. This was the case for both QPSK and 16QAM. The results also suggest that the hard limit threshold for the LLRs should be made adaptive according to the SNR.

The performance of the iterative a priori sphere decoder was also given. It was shown that the sphere decoder was superior at extracting the diversity offered in the MIMO system and TAST code than the ZF and ZF-DF decoders. Due to the extra diversity, the resultant BER curve is very steep thus reducing the gain in dB from iterative decoding to approximately 1 dB. The additional complexity due to the inclusion of a priori information in the sphere decoder was shown to be up to sixty percent. The ability of various symbol sorting strategies

in mitigating the increase in computational complexity was shown. The algorithm based on a modified sorted QR decomposition was shown to perform the best. Overall a complexity reduction of twenty-five percent was achieved.

Lastly, the performance of a SD-HNN reduced complexity sphere decoder was given. It was shown that after six iterations the decoder is able to match the performance of the SD. The shortcomings of the SD-HNN was also discussed and it was explained that a lower complexity SD-ZF-DF decoder would outperform the SD-HNN decoder.

CHAPTER 8

CONCLUSIONS AND FUTURE RESEARCH

In this chapter the conclusion from the previous chapters are drawn and potential future research is proposed.

8.1 CONCLUSIONS

In this dissertation several iterative MIMO decoders, that can be used in a turbo structure at the decoder to exploit the concatenation of STFCs with FEC codes, were developed. Their performance in a MIMO system employing space time coding was presented.

1. In chapter two, mobile wireless channels were discussed. Various physical aspects such as fading, Doppler spreading and delay spread were discussed. Furthermore an alternative implementation offering computational savings and additional simulation abilities was presented. It was mathematically shown that the triply selective MIMO fading channel model by Xiao et al. [31] is a special instance of the model developed.

2. MAP decoding was discussed in chapter three. The basic MAP metric was derived. It was shown that there is a slight difference between the metric for complex Gaussian noise and real Gaussian noise when a priori information is included. Various algorithms that strive to perform MAP decoding were described.

   A method to incorporate a priori information in the ZF, ZF-DF, MMSE and MMSE-DF decoders was developed and constrasted with the RVT method proposed by Wang

and Giannakis [20]. It was also shown that the RVT method is limited to signal constellations that have a linear mapping to their binary information, such as is the case for QPSK and BPSK signal constellations.

The sphere decoder was also described as a special instance of the A* tree search algorithm. This permitted the extension of the classical sphere decoder to a sphere decoder incorporating a priori information.

It was predicted that the inclusion of a priori information into the sphere decoder would increase its complexity. In an attempt to reduce the additional complexity various symbol sorting strategies were proposed. A modification to the SQRD algorithm of Wubben et al. [15], incorporating the use of a priori information, was also presented.

The use of a the energy function of a HNN to perform MAP decoding was also presented. It was shown that MAP decoding of the MIMO system corresponds to a constrained optimization. The HNN was then used in collaboration with Mr. D.J. Louw in a reduced complexity sphere decoder. Some of the disadvantages of the HNN when used in this capacity to the ZF-DF decoder was also mentioned.

3. In chapter four FEC was discussed. The emphasis was placed on linear block codes and specifically sparse graph codes such as LDPC codes. The decoding of linear block codes using message passing was given. This follows from the formulation of the decoding problem as that of performing a marginalisation using factor graphs.

   The performance in AWGN and Rayleigh of the LDPC codes fading used in this dissertation was also presented. It was shown that LDPC codes are able to extract diversity in a Rayleigh fading environment.

4. Space time frequency codes used in MIMO channel to improve performance was discussed in chapter five. Some of the theoretical characteristics, such as coding gain and diversity, were presented. Two main types of MIMO codes were discussed: orthogonal codes and linear precodes. It was shown how a constellation rotation can achieve diversity.

A general approach to calculating the rotation matrix based on the Fourier transform was given based on algebraically constructed linear precode matrixes. It was shown that the throughput of the linear precode could be increased by using threading or layering of a component code resulting in a TAST code. The various threads are then algebraically separated using Diophantine numbers.

Finally the performance of the various TAST codes used in this dissertation was given. It was shown that the TAST codes having the same number of layers as antennas were able to extract all of the theoretical diversity.

5. Chapter six provided a description of the simulated system. The block diagram of the transmitter was given and described. The receiver structure was also discussed and a block diagram detailing the iterative decoding structure also presented.

   A mathematical model of the received signal was also given. This model detailed the structure of the TAST codes used as well as the format of the CSI matrix and decoded signal vector.

   Lastly, the effect of hard limiting the LLR values from the STFC decoder was also investigated. This investigation showed a benefit of limiting the LLRs specifically at high SNRs and when less optimal decoders are used. The results also showed the iterative reduction in BER for each decoding iteration. A gradual reduction in the BER improvement in each successive iteration was also shown.

6. The simulation results of the various iterative decoders were presented in chapter seven. It was shown that iterative decoding can provide up to two orders of magnitude reduction in the BER. The gain in dB this reduction provides is dependent on the slope of the BER curve, with steep curves showing a small improvement. It was also shown that the ZF based decoders perform worse with increasing decoding matrix sizes. This was not the case for the MMSE decoders.

   It was also shown that the RVT method generally outperforms the method developed in this dissertation, except for the ZF-DF decoder where the performance is the same.

This was attributed to the fact that the a priori information in the RVT method is distributed over all of the symbols as evident in eqn. 3.127.

It was also shown that the ZF and ZF-DF decoders do not fully exploit all of the diversity offered by the MIMO channel and STF codes. In addition it was shown that LDPC codes are able to extract diversity.

The effect that the order in which the symbols are decoded for the ZF and ZF-DF decoders was also presented. The results showed a significant gain of approximately 11 dB for the ZF-DF decoder when decoding QPSK symbols and around 8.5 dB when decoding 16QAM symbols respectively. This was achieved by sorting the symbols according to their a priori information. It was shown that the SQRD method did not provide any substantial gains.

The BER performance of the iterative sphere decoder was also presented showing a hundredfold reduction in the BER. Due to the ability of the sphere decoder to extract all of the available diversity, this translated into a 1 dB gain. It was shown that the addition of a priori information results in a sixty percent increase in computations. It was also shown that the sorting strategies proposed result in a mitigation of the increase in computations required. The modified SQRD algorithm proving to provide the largest reduction in complexity. This results in an overall decrease in complexity of 25 percent as compared with a similar number of iterations of a classical sphere decoder.

8.2 FUTURE RESEARCH

The research presented in this dissertation can be used as a basis for future research. Not all of the possible aspects were considered in this dissertation. Some of these include channel state estimation (CSE), differential MIMO encoding and implementation issues. The following are of particular importance.

8.2.1 Channel state estimation

In this dissertation it was assumed that perfect CSI was available to the decoders. It would be worthwhile to investigate the performance of the decoders when imperfect CSI is given. The CSE algorithms could be combined in the iterative decoder to reduce the degradation due to imperfect CSI.

8.2.2 Differential STF coding

Differential encoded STF codes do not require the knowledge of CSI. It will be instructive to compare the performance of differentially encoded STF codes to that of STF codes with imperfect CSI when decoded using the decoders in this dissertation.

8.2.3 Sphere decoder complexity in an adaptive system

It was shown that the sphere decoder has maximum complexity at low SNRs. It would perhaps be worthwhile to investigate the complexity characteristics of a sphere decoder in an adaptive system. In the adaptive system the transmitter would switch to a lower complexity constellation in order to reduce the required computations in low SNR environments. Some of the computational reduction methods found by Studer et al. [21] could also be implemented and analysed.

The exact criteria for switching may depend not only on the BER but also on the maximum computational complexity supported by the decoder at the receiver. The tradeoff between the rate and the potential energy savings could be investigated.

8.2.4 Adaptive LLR hard limiting

It was shown that hard limiting of the LLRs has a significant impact on the performance of all the decoders. Scaling the limit with SNR may provide performance benefits at high SNRs.

8.2.5 Application of post sorting algorithm to the MMSE decoder

In this dissertation the SQRD algorithm was modified to include a priori information. This can potentially be duplicated for the post sorting algorithm (PSA) developed by Wubben et al. [15] and applied to the MMSE and MMSE-DF decoders. The sphere decoder can then also be modified to perform optimal MMSE-DF decoding using the mSQRD and modified PSA.

In conclusion, iterative MIMO STFC decoders can potentially offer a hundred fold decrease in BER performance as compared with the equivalent non-iterative decoder, though at the expense of increased computations. This can be done without requiring any changes at the transmitter, as it uses the inherent structure in the typical use of FEC concatenated STFCs in MIMO communication systems.

REFERENCES

[1] A. F. Molisch, *Wireless Communications*. John Wiley & Sons, LTD, May 2007.

[2] M. Damen, H. El Gamal, and N. Beaulieu, "Systematic construction of full diversity algebraic constellations," *Information Theory, IEEE Transactions on*, vol. 49, no. 12, pp. 3344 –3349, December 2003.

[3] G. D. Forney, "Concatenated codes," Ph.D. dissertation, M.I.T. Dept. of Electrical Engineering, 1965. [Online]. Available: http://hdl.handle.net/1721.1/13449

[4] S. Benedetto and G. Montorsi, "Iterative decoding of serially concatenated convolutional codes," *Electronics Letters*, vol. 32, no. 13, pp. 1186–1188, 1996.

[5] V. Szeto and S. Pasupathy, "Iterative decoding of serially concatenated convolutional codes and MSK," *Communications Letters, IEEE*, vol. 3, no. 9, pp. 272–274, 1999.

[6] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes. 1," in *Communications, 1993. ICC '93 Geneva. Technical Program, Conference Record, IEEE International Conference on*, vol. 2, 1993, pp. 1064–1070 vol.2.

[7] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Correction to "space-time block codes from orthogonal designs"," *Information Theory, IEEE Transactions on*, vol. 46, no. 1, pp. 314–314, 2000.

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

[9] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," *Information Theory, IEEE Transactions on*, vol. 45, no. 5, pp. 1456–1467, 1999.

[10] T. Lo, "Maximum ratio transmission," *Communications, IEEE Transactions on*, vol. 47, no. 10, pp. 1458 –1461, October 1999.

[11] H. El Gamal and M. Damen, "Universal space-time coding," *Information Theory, IEEE Transactions on*, vol. 49, no. 5, pp. 1097 – 1119, May 2003.

[12] Y. Xin, Z. Wang, and G. B. Giannakis, "Space-time diversity systems based on linear constellation precoding," *Wireless Communications, IEEE Transactions on*, vol. 2, no. 2, pp. 294–309, 2003.

[13] Z. Liu, Y. Xin, and G. B. Giannakis, "Linear constellation precoding for OFDM with maximum multipath diversity and coding gains," *Communications, IEEE Transactions on*, vol. 51, no. 3, pp. 416–427, 2003.

[14] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. John Hopkins University Press, 1996.

[15] D. Wubben, R. Bohnke, J. Rinas, V. Kuhn, and K. Kammeyer, "Efficient algorithm for decoding layered space-time codes," *Electronics Letters*, vol. 37, no. 22, pp. 1348 –1350, October 2001.

[16] O. Damen, A. Chkeif, and J.-C. Belfiore, "Lattice code decoder for space-time codes," *Communications Letters, IEEE*, vol. 4, no. 5, pp. 161 –163, May 2000.

[17] Z. Safar, W. Su, and K. J. R. Liu, "A fast sphere decoding algorithm for space-frequency block codes," *Applied Signal Processing, EURASIP Journal on*, vol. 2006, pp. 148–148, uary.

[18] E. Larsson, *Space-time block coding for wireless communications*. Cambridge, UK New York: Cambridge University Press, 2003.

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

[20] R. Wang and G. Giannakis, "Approaching MIMO channel capacity with soft detection based on hard sphere decoding," *Communications, IEEE Transactions on*, vol. 54, no. 4, pp. 587–590, April 2006.

[21] C. Studer, A. Burg, and H. Bolcskei, "Soft-output sphere decoding: algorithms and VLSI implementation," *Selected Areas in Communications, IEEE Journal on*, vol. 26, no. 2, pp. 290 –300, February 2008.

[22] R. G. Gallager, "Low-density parity-check codes," *Information Theory, Transactions of the IRE Professional Group on*, vol. IT-8, pp. 21–28, 1962. [Online]. Available: http://web.mit.edu/gallager/www/pages/ldpc.pdf

[23] D. MacKay and R. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, p. 1645, August 1996.

[24] D. J. MacKay and M. C. Davey, "Evaluation of Gallager codes for short block length and high rate applications," in *In Codes, Systems and Graphical Models*. Springer-Verlag, 1999, pp. 113–130.

[25] J. Pearl, *Reverend Bayes on Inference Engines: A Distributed Hierarchical Approach*, ser. UCLA-ENG. Cognitive Systems Laboratory, School of Engineering and Applied Science, University of California, Los Angeles, 1982. [Online]. Available: https://www.aaai.org/Papers/AAAI/1982/AAAI82-032.pdf

[26] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," *Information Theory, IEEE Transactions on*, vol. 47, no. 2, pp. 599–618, Feb 2001.

[27] H. Song and J. Cruz, "Reduced-complexity decoding of Q-ary LDPC codes for magnetic recording," *Magnetics, IEEE Transactions on*, vol. 39, no. 2, pp. 1081 – 1087, March 2003.

[28] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *Information Theory, IEEE Transactions on*, vol. 52, no. 2, pp. 549 – 583, February 2006.

[29] L. Lan, L. Zeng, Y. Tai, L. Chen, S. Lin, and K. Abdel-Ghaffar, "Construction of quasi-cyclic LDPC codes for AWGN and binary erasure channels: A finite field approach," *Information Theory, IEEE Transactions on*, vol. 53, no. 7, pp. 2429 –2458, July 2007.

[30] D. J. Mackay, "Encyclopedia of sparse graph codes," website, http://www.inference.phy.cam.ac.uk/mackay/codes/data.html, (visited 13 December 2013). [Online]. Available: http://www.inference.phy.cam.ac.uk/mackay/codes/data.html

[31] C. Xiao, J. Wu, S.-Y. Leong, Y. R. Zheng, and K. Letaief, "A discrete-time model for triply selective MIMO Rayleigh fading channels," *Wireless Communications, IEEE Transactions on*, vol. 3, no. 5, pp. 1678–1688, September 2004.

[32] D. J. Louw, "Non-binary LDPC coded STF-MIMO-OFDM with an iterative joint receiver structure," Master's thesis, University of Pretoria, 2010. [Online]. Available: http://books.google.co.za/books?id=sxwlpwAACAAJ

[33] J. Hopfield and D. Tank, "Computing with neural circuits: a model," *Science*, vol. 233, no. 4764, pp. 625–633, 1986. [Online]. Available: http://www.sciencemag.org/cgi/content/abstract/233/4764/625

[34] H. C. Myburgh, "Low complexity iterative MLSE equalization in extremely long Rayleigh fading channels," Master's thesis, University of Pretoria, 2010. [Online]. Available: http://innopac.up.ac.za/record=b1794423~S1

[35] D. J. Louw, "Non-binary LDPC coded STF-MIMO-OFDM with an iterative joint receiver structure," Ph.D. dissertation, Departement Electrical, Electronics and Computer Engineering. University of Pretoria, 2010. [Online]. Available: http://upetd.up.ac.za/thesis/available/etd-09202010-185002/

[36] A. Scholten, L. Tiemeijer, R. van Langevelde, R. Havens, A. Zegers-van Duijnhoven, and V. Venezia, "Noise modeling for RF CMOS circuit simulation," *Electron Devices, IEEE Transactions on*, vol. 50, no. 3, pp. 618 – 632, March 2003.

[37] L. Staphorst, "Viterbi decoded linear block codes for narrowband and wideband wireless communication over mobile fading channels," Master's thesis, University of Pretoria, July 2005.

[38] B. Sklar, "Rayleigh fading channels in mobile digital communication systems .i. characterization," *Communications Magazine, IEEE*, vol. 35, no. 7, pp. 90 –100, July 1997.

[39] T. Rappaport, *Wireless Communications: Principles and Practice*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1996.

[40] J. G. Proakis, *Digital Communications*, 4th ed. New York: McGraw-Hill, 2001.

[41] A. Molisch, *Wireless communications*. Chichester, West Sussex, U.K: Wiley IEEE, 2011.

[42] P. Robertson and S. Kaiser, "The effects of Doppler spreads in OFDM(A) mobile radio systems," in *Vehicular Technology Conference, 1999. VTC 1999 - Fall. IEEE VTS 50th*, vol. 1, 1999, pp. 329–333 vol.1.

[43] P. Bello and B. D. Nelin, "Optimization of subchannel data rate in FDM-SSB transmission over selectively fading media," *Communications Systems, IEEE Transactions on*, vol. 12, no. 1, pp. 46–53, 1964.

[44] E. Biglieri, *Coding for wireless channels*. New York: Springer, 2005.

[45] D. Halliday, R. Resnick, and J. Walker, *Fundamentals of Physics*, 6th ed. New York: John Wiley & Sons, LTD, 2001.

[46] M. Pätzold, *Mobile Fading Channels*, 1st ed. New York, NY: John Wiley and Sons LTD, 2002.

[47] M. Failli, "Digital land mobile radio communications COST 207," European Commission, Tech. Rep., 1989.

[48] J. Anderson, *Digital transmission engineering*. Hoboken, N.J. Chichester: IEEE Press Wiley-Interscience John Wiley distributor, 2005.

[49] P. Hoeher, "A statistical discrete-time model for the WSSUS multipath channel," *Vehicular Technology, IEEE Transactions on*, vol. 41, no. 4, pp. 461–468, November 1992.

[50] M. Herdin, H. Ozcelik, H. Hofstetter, and E. Bonek, "Variation of measured indoor MIMO capacity with receive direction and position at 5.2 GHz," *Electronics Letters*, vol. 38, no. 21, pp. 1283 – 1285, October 2002.

[51] J. Kermoal, L. Schumacher, K. Pedersen, P. Mogensen, and F. Frederiksen, "A stochastic MIMO radio channel model with experimental validation," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 6, pp. 1211–1226, August 2002.

[52] J. Mietzner and P. Hoeher, "A rigorous analysis of the statistical properties of the discrete-time triply-selective MIMO rayleigh fading channel model," *Wireless Communications, IEEE Transactions on*, vol. 6, no. 12, pp. 4199–4203, December 2007.

[53] J. Mietzner, C. Xiao, P. Hoeher, and K. B. Letaief, "A note on discrete-time triply-selective MIMO rayleigh fading channel models," *Wireless Communications, IEEE Transactions on*, vol. 7, no. 3, pp. 837–837, March 2008.

[54] E. Telatar, "Capacity of multi-antenna gaussian channels," *Telecommunications, European Transactions on*, vol. 10, no. 6, pp. 585–595, 1999. [Online]. Available: http://dx.doi.org/10.1002/ett.4460100604

[55] P. P. Vaidyanathan, *Signal processing and optimization for transceiver systems*. Cambridge, UK New York: Cambridge University Press, 2010.

[56] R. G. Gallager, "Circularly-symmetric gaussian random vectors," 2008. [Online]. Available: www.rle.mit.edu/rgallager/documents/CircSymGauss.pdf

[57] L. L. Hanzo, T. H. Liew, B. L. Yeap, R. Y. S. Tee, and S. X. Ng, *Turbo Coding, Turbo Equalisation and Space-Time Coding: EXIT-Chart-Aided Near-Capacity Designs for Wireless Channels*, 2nd ed. Chichester Honoken, N.J: Wiley-IEEE Press, March 2011. [Online]. Available: http://eu.wiley.com/WileyCDA/WileyTitle/productCd-0470972904.html

[58] E. Zimmermann, *Complexity Aspects in Near-Capacity MIMO Detection-Decoding*, ser. Beiträge aus der Informationstechnik. Vogt, 2007. [Online]. Available: http://books.google.co.za/books?id=PbcFZ7bhAkwC

[59] D. Wubben, R. Bohnke, V. Kuhn, and K. D. Kammeyer, "MMSE extension of V-BLAST based on sorted QR decomposition," in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 1, 2003, pp. 508–512 Vol.1.

[60] P. Hart, N. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *Systems Science and Cybernetics, IEEE Transactions on*, vol. 4, no. 2, pp. 100 –107, July 1968.

[61] R. Dechter and J. Pearl, "Generalized best-first search strategies and the optimality of a*," *Association for Computing Machinery, Journal of the*, vol. 32, no. 3, pp. 505–536, July 1985. [Online]. Available: http://doi.acm.org/10.1145/3828.3830

[62] S. V. Balakrishnan-Ayer, "Solving combinatorial optimization problems using neural networks with applications in speech recognition," Ph.D. dissertation, Trinity College, Cambridge, 1991.

[63] D. Louw, P. Botha, and B. Maharaj, "A low complexity soft-input soft-output MIMO detector which combines a sphere decoder with a Hopfield network," in *Mediterranean Electrotechnical Conference, MELECON 2010 - 2010 15th IEEE*, April 2010, pp. 521 –526.

[64] D. J. Louw, P. R. Botha, and B. T. J. Maharaj, "A reduced complexity soft-input soft-output MIMO detector combining a sphere decoder with a Hopfield network," in *Wireless Communications and Networking Conference (WCNC), 2010 IEEE*, April 2010, pp. 1 –6.

[65] P. Botha and B. Maharaj, "Turbo STFC decoding with the Zero Forcing decoder," in *AFRICON, 2011*, September 2011, pp. 1 –5.

[66] B. Datta, *Numerical Linear Algebra and Applications*. Society for Industrial and Applied Mathematics, 2010. [Online]. Available: http://books.google.co.za/books?id=-tW8-FUoxWwC

[67] D. S. Watkins, *Fundamentals of matrix computations*. New York: Wiley-Interscience, 2002.

[68] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, no. 4, pp. 623–656, October 1948. [Online]. Available: http://bstj.bell-labs.com/BSTJ/images/Vol27/bstj27-4-623.pdf;http://www.alcatel-lucent.com/bstj/vol27-1948/articles/bstj27-4-623.pdf

[69] T. Richardson and R. Urbanke, *Modern Coding Theory*. New York, NY, USA: Cambridge University Press, 2008.

[70] W. W. Peterson and E. Weldon, *Error-Correcting Codes*, 2nd ed. England: MIT Press, 1988.

[71] S. Lin, *An Introduction to Error-Correcting Codes*, F. F. Kuo, Ed. Englewood Cliffs, New Jersey: Prentice Hall Inc., 1970.

[72] D. MacKay, *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003, http://www.inference.phy.cam.ac.uk/mackay/itprnn/book.html.

[73] R. Roth, *Introduction to coding theory*. Cambridge: Cambridge University Press, 2006.

[74] A. Jimenez Felström and K. Zigangirov, "Time-varying periodic convolutional codes with low-density parity-check matrix," *Information Theory, IEEE Transactions on*, vol. 45, no. 6, pp. 2181 –2191, September 1999.

[75] R. Tanner, "A recursive approach to low complexity codes," *Information Theory, IEEE Transactions on*, vol. 27, no. 5, pp. 533 – 547, September 1981.

[76] D. Declercq and M. Fossorier, "Decoding algorithms for nonbinary LDPC codes over GF," *Communications, IEEE Transactions on*, vol. 55, no. 4, pp. 633–643, April 2007.

[77] H. Wymeersch, H. Steendam, and M. Moeneclaey, "Log-domain decoding of LDPC codes over GF(q)," in *Communications, 2004 IEEE International Conference on*, vol. 2, June 2004, pp. 772 – 776 Vol.2.

[78] A. Voicila, D. Declercq, F. Verdier, M. Fossorier, and P. Urard, "Low-complexity decoding for non-binary LDPC codes in high order fields," *Communications, IEEE Transactions on*, vol. 58, no. 5, pp. 1365 –1375, May 2010.

[79] I. Streinu and L. Theran, "Sparse hypergraphs and pebble game algorithms," *European Journal of Combinatorics*, vol. 30, no. 8, pp. 1944 – 1964, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0195669808002795

[80] M. Luby, M. Mitzenmacher, A. Shokrollah, and D. Spielman, "Analysis of low density codes and improved designs using irregular graphs," in *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, ser. STOC '98. New York, NY, USA: ACM, 1998, pp. 249–258. [Online]. Available: http://doi.acm.org/10.1145/276698.276756

[81] M. Davey and D. MacKay, "Low-density parity check codes over GF(q)," *Communications Letters, IEEE*, vol. 2, no. 6, pp. 165 –167, June 1998.

[82] M. Damen, H. Gamal, and N. Beaulieu, "A new representation of TAST codes," *Information Theory, IEEE Transactions on*, vol. 52, no. 9, pp. 4248 –4251, September 2006.

[83] V. Tarokh, N. Seshadri, and A. Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," *Information Theory, IEEE Transactions on*, vol. 44, no. 2, pp. 744 –765, March 1998.

[84] Z. Safar and K. Liu, "Systematic space-time trellis code construction for correlated Rayleigh-fading channels," *Information Theory, IEEE Transactions on*, vol. 50, no. 11, pp. 2855 – 2865, November 2004.

[85] D. P. Palomar and Y. Jiang, "MIMO transceiver design via majorization theory," *Found. Trends Commun. Inf. Theory*, vol. 3, no. 4, pp. 331–551, November 2006. [Online]. Available: http://0-dx.doi.org.innopac.up.ac.za/10.1561/0100000018

[86] S. Roman, *Advanced linear algebra*. New York: Springer, 2005.

[87] P. R. Botha, D. J. Louw, and B. T. Maharaj, "Achievable diversity limits in a quantized MIMO-OFDM linear pre-coded system," in *Wireless Pervasive Computing (ISWPC), 2010 5th IEEE International Symposium on*, May 2010, pp. 455 –459.

[88] S. Georgiou, C. Koukouvinos, and J. Seberry, *Designs 2002 : further computational and constructive design theory*. Boston: Kluwer Academic Publishers, 2003.

[89] M. Damen, K. Abed-Meraim, and J.-C. Belfiore, "Diagonal algebraic space-time block codes," *Information Theory, IEEE Transactions on*, vol. 48, no. 3, pp. 628 –636, March 2002.

[90] J. Choi, *Optimal combining and detection : statistical signal processing for communications*. Cambridge, UK New York: Cambridge University Press, 2010.

[91] H. Cohen, *Number theory*. New York: Springer, 2007.