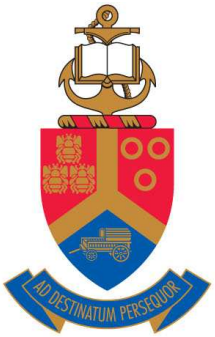

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

Non-binary LDPC coded STF-MIMO-OFDM with an iterative joint receiver structure

D.J. Louw

2010

UNIVERSITEIT VAN PRETORIA
UNIVERSITY OF PRETORIA
YUNIBESITHI YA PRETORIA

NON-BINARY LDPC CODED STF-MIMO-OFDM WITH AN ITERATIVE JOINT RECEIVER STRUCTURE

By

Daniel Johannes Louw

Study leader: Dr. B.T. Maharaj

Submitted in partial fulfilment of the requirements for the degree

Master of Engineering (Electronic)

in the

Department of Electrical, Electronic & Computer Engineering

of the

Faculty of Engineering, Built Environment & Information Technology

UNIVERSITY OF PRETORIA

January 2010

SUMMARY

NON-BINARY LDPC CODED STF-MIMO-OFDM WITH AN ITERATIVE JOINT RECEIVER STRUCTURE

by

Daniel Johannes Louw

Study leader: Dr. B.T. Maharaj

Department of Electrical, Electronic & Computer Engineering

Master of Engineering (Electronic)

The aim of the dissertation was to design a realistic, low-complexity non-binary (NB) low density parity check (LDPC) coded space-time-frequency (STF) coded multiple-input multiple-output (MIMO) orthogonal frequency division multiplexing (OFDM) system with an iterative joint decoder and detector structure at the receiver. The goal of the first part of the dissertation was to compare the performance of different design procedures for NB-LDPC codes on an additive white Gaussian noise (AWGN) channel, taking into account the constraint on the code length. The effect of quantisation on the performance of the code was also analysed. Different methods for choosing the NB elements in the parity check matrix were compared. For the STF coding, a class of universal STF codes was used. These codes use linear pre-coding and a layering approach based on Diophantine numbers to achieve full diversity and a transmission rate (in symbols per channel use per frequency) equal to the number of transmitter antennas. The study of the system considers a comparative performance analysis of different ST, SF and STF codes. The simulations of the system were performed on a triply selective block fading channel. Thus, there was selectivity in the fading over time, space and frequency. The effect of quantisation at the receiver on the achievable diversity of linearly pre-coded systems (such as the STF codes used) was mathematically derived and verified with simulations. A sphere decoder (SD) was used as a MIMO detector. The standard method used to create a soft-input soft output (SISO) SD uses a hard-to-soft process and the max-log-map approximation. A new approach was developed which combines a Hopfield network with the SD. This SD-Hopfield detector was connected with the fast Fourier transform belief propagation (FFT-BP) algorithm in an iterative

structure. This iterative system was able to achieve the same bit error rate (BER) performance as the original SISO-SD at a reduced complexity. The use of the iterative Hopfield-SD and FFT-BP decoder system also allows performance to be traded off for complexity by varying the number of decoding iterations. The complete system employs a NB-LDPC code concatenated with an STF code at the transmitter with a SISO-SD and FFT-BP decoder connected in an iterative structure at the receiver. The system was analysed in varying channel conditions taking into account the effect of correlation and quantisation. The performance of different SF and STF codes were compared and analysed in the system. An analysis comparing different numbers of FFT-BP and outer iterations was also done.

Keywords:

MIMO-OFDM, Non-binary LDPC, Hopfield Network, Sphere Decoder, Quantisation, Space-Time Codes, Space-Frequency Codes, Space-Time Frequency Codes.

OPSOMMING

NON-BINARY LDPC CODED STF-MIMO-OFDM WITH AN ITERATIVE JOINT RECEIVER STRUCTURE

deur

Daniel Johannes Louw

Studie leiers: Dr. B.T. Maharaj

Departement Elektriese, Elektroniese & Rekenaar Ingenieurswese

Meester in Ingenieurswese (Elektronies)

Die doel van die verhandeling was om ’n realistiese, lae-kompleksiteit nie-binêre (NB) LDPC gekodeerde ruimte-tyd-frekwensie-gekodeerde MIMO-OFDM-sisteem met iteratiewe gesamentlike dekodeerder- en detektorstrukture by die ontvanger te ontwerp. Die eerste deel van die verhandeling was om die werkverrigting van verskillende ontwerpprosedures vir NB-LDPC kodes op ’n gesommeerde wit Gausruiskanaal te vergelyk met inagneming van die beperking op die lengte van die kode. Verskillende metodes om die nie-binêre elemente in die pariteitstoetsmatriks te kies, is gebruik. Vir die ruimte-tyd-frekwensiekodering is ’n klas universele ruimte-tyd-frekwensiekodes gebruik. Hierdie kodes gebruik lineêre pre-kodering en ’n laagbenadering gebaseer op Diofantiese syfers om volle diversiteit te bereik en ’n oordragtempo (in simbole per kanaalgebruik per frekwensie) gelyk aan die aantal senderantennes. Die studie van die sisteem oorweeg ’n vergelykende werkverrigtinganalisie van verskillende ruimte-tyd-, ruimte-freksensie- en ruimte-tyd-frekwensiekodes. Die simulasies van die sisteem is gedoen op ’n drievoudig selektiewe blokwegsterwingskanaal. Daar was dus selektiwiteit in die wegsterwing oor tyd, ruimte en frekwensie. Die effek van kwantisering by die ontvanger op die bereikbare diversiteit van lineêr pre-gekodeerde sisteme (soos die ruimte-tyd-frekwensiekodes wat gebruik is) is matematies afgelei en bevestig deur simulasies. ’n Sfeerdekodeerder (SD) is gebruik as ’n MIMO-detektor. Die standaardmetode wat gebuik is om ’n sagte-inset-sagte-uitset (SISO) SD te skep, gebruik ’n harde-na-sagte proses en die maksimum logaritmiese afbeelding-benadering. ’n Nuwe benadering wat ’n Hopfield-netwerk met die SD kombineer, is ontwikkel. Hierdie SD-Hopfield-detektor

is verbind met die FFT-BP-algoritme in iteratiewe strukture. Hierdie iteratiewe sisteem was in staat om dieselfde bisfouttempo te bereik as die oorspronklike SISO-SD, met laer kompleksiteit. Die gebruik van die iteratiewe Hopfield-SD en FFT-BP-dekodeerdersisteem maak ook daarvoor voorsiening dat werkverrigting opgeweeg kan word teen kompleksiteit deur die aantal dekodering-iterasies te varieer. Die volledige sisteem maak gebruik van ’n QC-NB-LDPC-kode wat met ’n ruimte-tyd-frekwensiekode by die sender aaneengeskakel is met ’n SISO-SD en FFT-BP-dekodeerder wat in ’n iteratiewe struktuur by die ontvanger gekoppel is. Die sisteem is onder ’n verskeidenheid kanaalkondisies ge-analiseer met inagneming van die effek van korrelasie en kwantisering. Die werkverrigting van verskillende ruimte-frekwensie- en ruimte-tyd-frekwensiekodes is vergelyk en in die sisteem ge-analiseer. ’n Analise om ’n wisselende aantal FFT-BP en buite-iterasies te vergelyk, is ook gedoen.

Sleutelwoorde:

MIMO-OFDM, Nie-binêr LDPC, Hopfield Netwerk, Sfeerdekodeerder, Kwantisering, Ruimte-tydkodes, Ruimte-frekwensiekodes, Ruimte-tyd-frekwensiekodes.

I dedicate this work to Ayn Rand, whose works inspired me to become an engineer.

ACKNOWLEDGEMENTS

I would like to thank the following people and institutions.

- First and foremost, my appreciation and thanks go to my parents for their enduring love, support and encouragement.
- I am especially grateful to my supervisor, Prof. B.T. Maharaj for his support and guidance.
- I would like to thank my engineering friends, Philip Botha, Daniel Basilio, K.P. Mare, and Christo du Plessis, for the insightful discussions during the course of this study.
- Credit goes to the University of Pretoria (UP) and the Sentech Chair in Broadband Wireless Communication (BWMC) for the financial sponsorship of my Masters degree.

Contents

List of Figures

List of Tables

List of Abbreviations

ADC	Analogue to Digital Converter
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
DAC	Digital to Analogue Converter
DAST	Diagonal Algebraic Space-Time
DMC	Discrete Memoryless Channel
DMS	Discrete Memoryless Source
EXIT	Extrinsic Information Transfer
FEC	Forward Error Correction
FFT	Fast Fourier Transform
FIR	Finite Impulse Response
ISI	Inter-symbol Interference
LDPC	Low Density Parity Check
LLR	Log Likelyhood Ratio
LOS	Line of Sight
LP	Linear Precoding
MAP	Maximum A-Posteriori Probability
MIMO	Multiple Input Multiple Output
ML	Maximum Likelihood
MRC	Maximum Ratio Combining
NB	Non-Binary
OFDM	Orthogonal Frequency Division Multiplexing

PEP	Pairwise Error Probability
PCM	Pulse Coded Modulation
PDF	Probability Density Function
PDP	Power Delay Profile
QAM	Quadrature Amplitude Modulation
QO	Quasi-Orthogonal
QPSK	Quadrature Phase Shift Keying
RV	Random Variable
SD	Sphere Decoder or Sphere Decoding
SI	Soft Input
SISO	Single Input Single Output
SNR	Signal-to-noise Ratio
SO	Soft Output
ST	Space-Time
SF	Space-Frequency
STF	Space-Time-Frequency
TAST	Threaded Algebraic Space-Time
ZF	Zero Forcing

NOTATION

$\mathbf{A}$	The matrix $\mathbf{A}$
$\mathbf{a}$	The column vector $\mathbf{a}$
$E[.]$	The expected value
$(.)^{\mathcal{T}}$	The transpose operation
$(.)^{\mathcal{H}}$	The Hermitian transpose operation
$(.)^*$	The complex conjugate
$\jmath$	The imaginary unit ($\jmath = \sqrt{-1}$)
$\mathbf{I}_N$	The $N \times N$ identity matrix
$\mathbf{0}_{P\times N}$	The $P \times N$ all zero matrix
$\otimes$	The Kronecker product
tr(.)	The trace of a matrix
$\lvert.\rvert$	The determinant of a matrix, or cardinality of a set, or the magnitude of an element
$\mathbb{N}$	The positive integer set
$\mathbb{Z}$	The integer ring
$\mathbb{Q}$	The rational number field
$\mathbb{R}$	The real number field
$\mathbb{C}$	The complex number field
$diag(d_1, \ldots, d_P)$	A diagonal matrix with diagonal entries $d_1, \ldots, d_P$
$\mathbb{Z}(j)$	The ring of Gaussian integers, whose elements are in the form of $p + jq$, where $p, q \in \mathbb{Z}$
$\mathbb{Q}(j)$	The smallest sub-field of $\mathbb{C}$ including both $\mathbb{Q}$ and j
$\mathbb{Q}(j)(\alpha)$	The smallest sub-field of $\mathbb{C}$ including both $\mathbb{Q}(j)$ and α, where α is algebraic over $\mathbb{Q}(j)$
$\mathcal{F}_N$	The $N \times N$ DFT matrix, with its $(p,q)_{th}$ element equal to $(1/\sqrt{N})\exp(-j2\pi(p-1)(q-1)/N)$
$\lceil x \rceil$	The smallest integer larger than x

$\lfloor x \rfloor$	The largest integer smaller than x
$real(x)$	The real part of x
$imag(x)$	The imaginary part of x
$(x)mod(y)$	The remainder of division of x by y
$gcd(x, y)$	The largest positive integer that divides both x and y without remainder
$a \not\equiv b \ (mod \ n)$	$(a - b)$ is not divisible by n (a and b are not congruent modulo n)
$\mathbf{X}^{1/2}$	The square root of the matrix $\mathbf{X}$

COMMONLY USED SYMBOLS

$\mathcal{A}$	The signal constellation, such as PSK or QAM
$\mathbf{N}_T$	The number of transmit antennas
$\mathbf{N}_R$	The number of receive antennas
K	The number of multipaths
N_B	The number of time periods (blocks)
T_s	The sampling interval
σ_n^2	The noise variance
σ_s^2	The transmitted symbol variance
W_c	The coherence bandwidth of the channel
T_c	The coherence time of the channel
D_s	The Doppler spread of the channel

CHAPTER ONE

INTRODUCTION

Multiple-input multiple-output (MIMO) schemes have greatly increased system capacity and reliability (due to spatial diversity) over conventional single-input single-output (SISO) systems. However, the design of systems capable of achieving this capacity is still an open research topic.

Along with spatial diversity due to multiple antennas, frequency diversity and time diversity may also be available to the system, depending on the channel conditions. Schemes that utilise this diversity are called space-time (ST) codes, space-frequency (SF) codes and space-time-frequency (STF) codes.

In SISO schemes, forward error correction (FEC) coding is used to improve reliability. Codes have been designed that are able to operate at a fraction of a decibel from the Shannon bound. The best class of codes, to date, is sparse graph codes, of which turbo codes and low-density parity check (LDPC) codes are the best known examples. It has been shown that by defining the code over non-binary (NB) alphabets a performance increase may be achieved.

In this dissertation, a system combining NB-LDPC codes with a MIMO system using ST, SF and STF codes is proposed. At the receiver, an iterative joint detector and decoder structure is employed.

1.1 BACKGROUND

In this section the background of the different aspects of the dissertation will be discussed.

1.1.1 LDPC codes

LDPC codes were first discovered by Gallager in 1963 [2]. The codes were then re-discovered by Mackay and Neal [3] in 1996. Davey and MacKay [4] then extended the codes to arbitrary Galois fields. The decoding of LDPC codes is done by the use of Tanner graphs and Pearl's belief propagation (BP). Richardson and Urbanke [5] developed a BP algorithm using the fast Fourier transform (FFT) for LDPC codes which reduces the decoding complexity. When BP is extended directly to NB-LDPC codes, the decoding complexity grows exponentially with the size of the Galois field. However, by using the FFT-based BP decoding algorithm (FFT-BP), the growth in complexity becomes linear.

Several methods for designing binary LDPC codes have been developed. Specifically, the use of density evolution [5] and extrinsic information transfer (EXIT) charts [6] has been shown to yield codes capable of performing very close to the Shannon bound. The use of these methods for the direct design of NB-LDPC codes is problematic owing to the complexity and the astronomical memory requirements. Simplified versions of density evolution have been developed in conjunction with coset-LDPC codes to yield a class of NB codes [7]. Another method of creating NB-LDPC codes is to design a binary LDPC code and simply replace all the 1s with elements from the Galois field. Quasi-cyclic (QC) LDPC codes are a class of code that do not use density evolution and may be encoded efficiently using shift registers. Several methods for constructing binary QC-LDPC and NB-QC-LDPC codes based on a structured finite field approach have been developed [8, 9]. These QC codes have an extremely low error floor. A class of NB-QC-LDPC codes that are able to guarantee a girth of 12 was developed by Ge and Xia [10].

1.1.2 MIMO, OFDM and STF codes

There are two main advantages of MIMO systems over SISO systems. Firstly, MIMO systems have a higher capacity than SISO systems. Capacity may be interpreted as the maximum allowable transmission rate for communication at an arbitrarily low error rate. In MIMO

systems the capacity increases linearly with the number of transmit antennas or the number of receive antennas, whichever is the least. Secondly, MIMO systems experience spatial diversity [11].

Orthogonal frequency division multiplexing (OFDM) is a bandwidth efficient transmission scheme which transmits parallel streams of data on orthogonal sub-carriers. It can be implemented efficiently using the FFT operation. OFDM has very useful properties when operating in a frequency selective channel. Specifically, by adding a cyclic prefix at the transmitter and removing it at the receiver, inter-symbol interference (ISI) is avoided. Also, by limiting the bandwidth of each sub-carrier to the coherence bandwidth of the channel, OFDM is able to ensure that flat fading occurs on each sub-carrier. This significantly simplifies channel equalisation and decoding. These flat fading sub-channels provide a convenient method for coding over frequencies and OFDM thus provides a basis for SF and STF codes.

Through the use of MIMO and OFDM, henceforth called MIMO-OFDM, the system designer is able to place data symbols in space, time and frequency. This allows for codes to be designed that are able to extract diversity in time, space and frequency. These codes can be compared by considering the diversity as well as the transmission rate that they are able to achieve. Many different designs have been proposed in literature. These include orthogonal ST codes [12, 13], algebraic ST codes [14, 15], algebraic SF codes [16, 17] and high rate STF codes [17]. The codes considered in this dissertation are the rate-1 Alamouti ST code [12], the rate-1 SF code [16], the rate-1 STF code [18] and the rate-2 STF code [17]. The SF and STF codes considered use the concept of linear precoding (LP) and layering using Diophantine approximations. These codes require a joint detector at the receiver. The detector used for this is the sphere decoder (SD).

The SD was first developed as a universal lattice decoder [19]. Because of the high initial complexity of the SD, $O(n^6)$, many efforts were made to reduce the complexity of the process. Current implementations are able to reduce the complexity to be on the order of n^3. The SD is however still one of the most computationally intensive parts of the system and reducing the complexity further is still an open research topic.

1.1.3 LDPC coded STF codes

Concatenating LDPC codes with multi-antenna codes is a promising approach and many systems have been proposed [20–25]. Since turbo codes were introduced in 1993 [26], the concept of iterative processing at the receiver has found many applications. One application is to use the turbo principle to create a joint detector and decoder at the receiver. In terms of MIMO communication systems, several systems combining FEC techniques with ST and SF codes have been proposed. Some of these systems also employ a turbo receiver structure. These studies do not, however, consider the performance of NB-LDPC codes on SF and STF codes. In addition, the analyses in these papers generally do not consider realistic channel conditions where antenna correlation is taken into account. A notable exception is [27], which uses binary LDPC codes but no turbo receiver.

1.2 Motivation and Objective of Dissertation

The objective of this dissertation is to design a system combining NB-LDPC codes with MIMO-OFDM using ST, SF and STF codes and an iterative joint detector and decoder structure. Figure 1.1 shows a block diagram of the transmitter of the proposed system while Figure 1.2 shows a block diagram of the receiver of the proposed system.

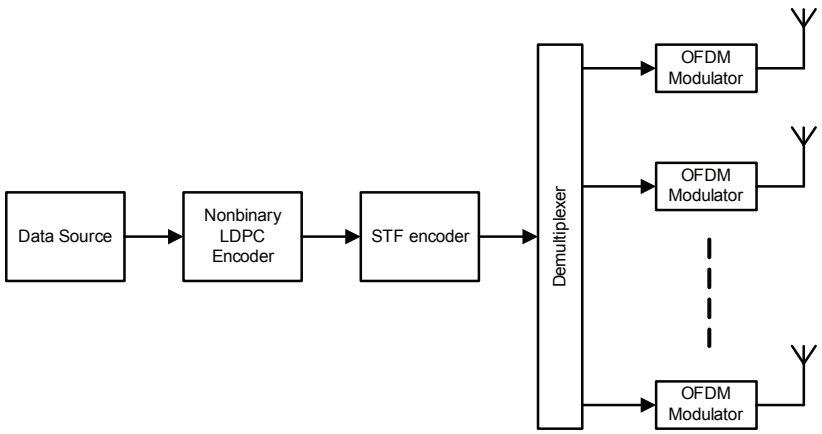

FIGURE 1.1: A block diagram of the transmitter of the proposed Turbo NB-QC-LDPC coded STF-MIMO-OFDM system.

The design will specifically focus on keeping the computational complexity low to allow for a realistic implementation. The system will be analysed in realistic triply selective (space, time

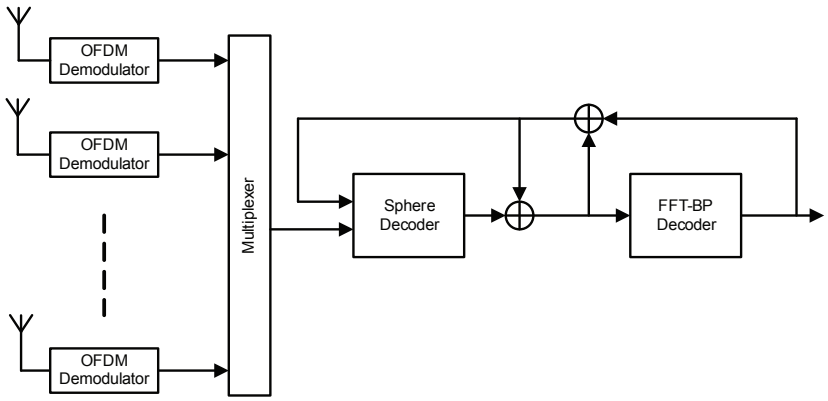

FIGURE 1.2: A block diagram of the receiver of the proposed Turbo NB-QC-LDPC coded STF-MIMO-OFDM system.

and frequency) fading channel conditions. The effect of quantisation at the receiver will also be analysed.

1.3 AUTHOR'S CONTRIBUTION

The purpose of this dissertation is the design and analysis of a low complexity NB-LDPC coded STF coded MIMO-OFDM system with an iterative joint receiver structure. In the process of designing this system, several contributions were made.

1.3.1 LDPC code design

In chapter 3 the design of NB-LDPC codes is considered. In order to keep encoding complexity low, a QC code design is chosen. For the sake of realistic complexity and latency, the code length, as well as the number of FFT-BP decoding iterations, are limited. Several design procedures (subject to the restrictions) for NB-QC-LDPC codes are implemented and compared. It was found that girth-12 codes performed best. The method used to select the non-binary elements in the parity check matrix was analysed and it was found that by replacing the method used in [8] with a random selection procedure a performance improvement could be obtained. The codes were also analysed with quantisation at the receiver and the effect on performance measured.

1.3.2 Quantisation Limits of STF codes

In chapter 5 the ST, SF and STF codes using LP are discussed. It was found that the process of LP places extra requirements on the number of quantisation bits at the receiver. The limit of the achievable diversity in a quantised LP coded system was mathematically derived. These limits were then verified by simulating SF and STF codes with quantisation at the receiver. The contributions from this chapter are the result of joint research with P.R. Botha.

1.3.3 Reduced complexity SD decoder

In chapter 6, the soft output SD is developed and discussed. A reduced complexity SO-SD utilising a Hopfield network and an iterative receiver structure is developed. The performance of the detector is compared with that of the normal detector. The computational complexity of the SD-Hopfield detector is explicitly calculated and compared with the complexity of the normal SO-SD. It is shown that the Hopfield-SD provides a convenient method of trading complexity for performance.

1.3.4 Turbo NB-QC-LDPC coded STF-MIMO-OFDM system

In Chapter 7 the full system combining the code developed in Chapter 3 with the MIMO-OFDM system and the STF codes is given. The turbo structure is explained. The system is analysed in triply selective block fading conditions. The effect of correlation at the antennas as well as quantisation at the receiver is analysed with simulations.

1.4 PUBLICATIONS

The author wrote or co-wrote several papers based on the research work performed in the course of this dissertation.

1. A paper entitled, “A reduced complexity Soft-Input Soft-Output MIMO detector combining a Sphere Decoder with a Hopfield Network” authored by D.J. Louw, P.R. Botha and B.T. Maharaj has been accepted for WCNC 2010, which will be held in Sydney, Australia in April 2010.

2. A paper entitled, “Achievable Diversity Limits in a Quantized MIMO-OFDM Linear Pre-coded System” authored by P.R. Botha, D.J. Louw and B.T. Maharaj was submitted

for review to ISWPC 2010 which will be held in Modena, Italy in May 2010.

3. A paper entitled "A low complexity Soft-Input Soft-Output MIMO detector which combines a Sphere Decoder with a Hopfield Network" authored by D.J. Louw and B.T. Maharaj was submitted for review to MELECON 2010, which will be held in Malta in April 2010.

1.5 OUTLINE OF DISSERTATION

The dissertation consists of eight chapters. Chapter 2 provides a basic introduction to FEC codes and introduces terminology and notation used throughout the dissertation. Chapter 3 introduces sparse graph codes and provides an explanation and a derivation for FFT-BP over arbitrary fields. From the explanation, the design process for NB-LDPC codes is discussed. The analysis of different QC-NB-LDPC codes is then given. Chapter 4 discusses mobile wireless channels and introduces the triply selective fading channel model used in the simulations. Chapter 5 gives an overview of multi-antenna coding and presents the ST, SF and STF codes used in the rest of the dissertation. The achievable diversity of quantised LP coded systems is derived in this chapter. Chapter 6 discusses MIMO detection for LP coded systems. The SD is discussed and the reduced complexity Hopfield-SD detector is developed. Chapter 7 presents the full Turbo NB-QC-LDPC coded STF-MIMO-OFDM system. This chapter also contains the analysis and simulation results of the system. In Chapter 8 all the conclusions and results drawn from the dissertation are presented and discussed. Future research possibilities are also proposed.

Chapter Two

Introduction to Forward Error Correction Codes

Communication is the process whereby information is transferred between two or more entities. One can create a simple model of communication with three components: a transmitter (the entity which has the information), a channel (the path through which the information must travel) and a receiver (the entity which wants the information). If the channel were perfect and did not in any way change the information, then communication would be easy and one could transmit infinite amounts of information without any errors. However, channels used by current communication systems are far from perfect. In a digital communication system these imperfections result in errors in the received signal. Forward error correction (FEC) codes encode the information at the transmitter in such a way that the errors can be detected and corrected at the receiver. Simply put, the encoder adds redundancy to the information which is used at the decoder to correct the errors. The amount of redundancy added can be measured as a ratio of the number of symbols after encoding to the number of symbols before encoding. If k represents the number of symbols before encoding and n represents the number of encoded symbols, then the rate of the code is defined as $r = k/n$. This chapter starts by introducing some information theory fundamentals and then describes block codes and specifically linear block codes.

2.1 INFORMATION THEORY FUNDAMENTALS

In order to determine any mathematical bounds on what is achievable in a communication system (and with FEC codes) one first needs a mathematical definition of information.

Consider two discrete memoryless information sources (DMS) X and Y with possible outcomes $x_1, x_2, ..., x_n$ and $y_1, y_2, ..., y_m$ respectively. Let the probability that $X = x_i$ be denoted by $P(x_i)$ and the conditional probability that $X = x_i$ given that $Y = y_i$ be denoted as $P(x_i|y_i)$. The information content provided by the occurrence of the event y_i about x_i is defined as [28]:

$$I(x_i; y_i) = \log \frac{P(x_i|y_i)}{P(x_i)}. \tag{2.1}$$

$I(x_i; y_i)$ is called the mutual information between x_i and y_i. The units of information depend on the base of the logarithm. When the base is 2, the units are bits and when the base is e, the units are nats. When x_i and y_i are perfectly correlated, the event y_i would always indicate that the event x_i has occurred, thus

$$P(x_i|y_i) = 1 \tag{2.2}$$

$$\therefore I(x_i; y_i) = \log \frac{1}{P(x_i)} \tag{2.3}$$

$$= -\log P(x_i) \tag{2.4}$$

which simply contains the information provided by the event x_i. Thus the self-information of the event $X = x_i$ may be given as:

$$I(x_i) = -\log P(x_i). \tag{2.5}$$

From the mutual and self-information between events, one can derive the average mutual information between sources, as well as the average self-information of a source. The average mutual information is obtained by weighting $I(x_i; y_j)$ for each possible event pair (i, j) with the probability of this event occurring and then summing over all these possibilities :

$$I(X; Y) = \sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i, y_j) I(x_i, y_j) \tag{2.6}$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{m} P(x_i, y_j) \log \frac{P(x_i|y_i)}{P(x_i)}. \tag{2.7}$$

Similarly the average self-information is defined as:

$$H(X) = \sum_{i=1}^{n} P(x_i)I(x_i) \tag{2.8}$$

$$= -\sum_{i=1}^{n} P(x_i)\log P(x_i). \tag{2.9}$$

When X represents the alphabet of possible output symbols from a source then $H(X)$ represents the average self-information per source symbol and is called the entropy of a source. The average conditional self-information is called the conditional entropy and is defined as:

$$H(X|Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_i)\log\frac{1}{P(x_i|y_i)}. \tag{2.10}$$

$H(X|Y)$ represents the the information in X after Y has been observed. The different average information terms may be combined to yield the following:

$$I(X;Y) = H(X) - H(X|Y) \tag{2.11}$$

The concept of average mutual information may be carried over from discrete random variables to continuous random variables. If X and Y are defined as continuous random variables with a joint probability density function (PDF) $p(x, y)$ and marginal PDFs $p(x)$ and $p(y)$, the average mutual information between X and Y is defined as:

$$I(X;Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x)p(y|x)\log\frac{p(y|x)p(x)}{p(x)p(y)}dxdy. \tag{2.12}$$

The concept of average self-information does not carry over to continuous variables. This is because an infinite number of bits are required to represent the variable exactly. The substitute used is called differential entropy and is defined as:

$$H(X) = -\int_{-\infty}^{\infty} p(x)\log p(x)dx. \tag{2.13}$$

Similarly, the average conditional entropy of X given Y may be defined as

$$H(X|Y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} p(x, y)\log p(x|y)dxdy \tag{2.14}$$

and thus

$$I(X;Y) = H(X) - H(X|Y). \tag{2.15}$$

2.2 CHANNEL CAPACITY AND THE NOISY CHANNEL CODING THEOREM

If one considers X to be the random variable that represents all possible symbols that may be transmitted from the source over a channel and Y to be the random variable that represents all possible symbols that may be received, then the channel capacity is defined as:

$$C = \max_{P(x_i)} I(X;Y) \tag{2.16}$$

for a discrete memoryless channel (DMC). The capacity of a channel is thus the mutual information between the input and output variables maximised over the set of input probabilities. For a bandwidth limited waveform channel the capacity of a channel per unit time is defined as

$$C = \lim_{T\to\infty} \max_{p(x)} \frac{1}{T} I(X;Y). \tag{2.17}$$

Most real world communication systems are modelled in part as additive white Gaussian noise (AWGN) channels. The capacity of an AWGN channel was derived by Shannon in 1948 [29] and can be given as:

$$C = W \log\left(1 + \frac{P_{av}}{WN_0}\right) \text{bits/s} \tag{2.18}$$

where W represents the bandwidth, P_{av} represents the average signal power and N_0 is the power spectral density of additive noise. However, this capacity is reliant on an input PDF, which is a statistically independent zero mean Gaussian random variable and will decrease if another input random variable is used. One may also normalise channel capacity with respect to bandwidth used:

$$\frac{C}{W} = \log\left(1 + \frac{E_b}{N_0}\frac{C}{W}\right) bits/s/Hz \tag{2.19}$$

where E_b is the energy per bit. From this equation one can derive the channel capacity in bits for any given signal-to-noise ratio (SNR) per bit. Alternatively, given a certain transmission rate, one can find the required SNR per bit. The meaning of channel capacity becomes clear when one considers the noisy channel coding theorem [29]:

NOISY CHANNEL CODING THEOREM : *There exist channel codes that make it possible to achieve reliable communication, with as small an error probability as desired, if the transmission rate $R < C$, where C is the channel capacity. If $R > C$, it is not possible to make*

the probability of error tend toward zero with any code.

Thus, once the capacity for any channel over which one would like to communicate has been derived, there are clear bounds on what is achievable and a clear goal for the design of the code. The problem is that while the theorem defines what is possible, it gives no indication as to how this may be achieved.

2.3 BLOCK CODES

There are two main code types : Convolutional codes and Block codes. Block codes encode a finite number (a block) of symbols at a time, while convolutional codes encode streams of symbols continuously. However, in real world applications these streams are generally of finite length, effectively resulting in a block code. Thus, while the encoding and decoding algorithms differ for convolutional and block codes, most of the results given in this section hold for both types. In this dissertation, only block codes are considered, thus no more mention will be made of convolutional codes.

The notation and terminology used for block codes will now be introduced. Let $\mathcal{A} = \{\alpha_0, ..., \alpha_{q-1}\}$ be an alphabet containing q elements. If $q = p^n$, where p is a prime and n an integer, then $\mathcal{A}$ can be considered a finite field $GF(q)$. For digital communication systems p is generally chosen as 2. When $n = 1$, the alphabet is the general binary alphabet $\mathcal{A} = \{0, 1\}$. Binary codes are defined for input and output alphabets of $GF(2)$. Non-binary codes on the other hand are defined over larger alphabets, $GF(p^n)$. For the rest of this dissertation, $GF(q)$ will be denoted by $\mathbb{F}_q$ and $\mathbb{F}_q$ will be the assumed alphabet over which the codes are defined.

Before specific aspects of a code can be analysed, it must be formally defined. An encoder takes, as an input, k symbols from alphabet $\mathbb{F}_q$ and produces n symbols from alphabet $\mathbb{F}_q$. An encoder may thus be viewed as a mapping from a k-dimensional vector to an n-dimensional vector. The code is the set of codewords produced by this mapping. Thus, if $\phi : \mathbb{F}_q^k \rightarrow \mathbb{F}_q^n$ is the encoder, then the code $C := \phi[\mathbb{F}_q^k]$ is the range of ϕ.

The ability of a code to correct errors comes from the dissimilarity between codewords

in the code. This dissimilarity allows one to distinguish between codewords even when the codewords have been corrupted. The dissimilarity between codewords is thus an important parameter, the measure of which will have an impact on the power of the code. The metric used is called the Hamming distance and is defined as the number of positions in which two codewords differ:

$$d_H(\mathbf{x}, \mathbf{y}) \quad := \quad |\{i : x_i \neq y_i\}|, \tag{2.20}$$

where $\mathbf{x}$ and $\mathbf{y}$ are codewords. From this definition one may define the minimum distance of a code as the smallest Hamming distance between any two codewords in the code:

$$d(C) = min\{d_H(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in C, \mathbf{x} \neq \mathbf{y}\}. \tag{2.21}$$

From the minimum distance of a code, one can determine its guaranteed error-correcting capability. The minimum distance indicates the minimum number of symbols which need to change in order to make one codeword look like another codeword. Thus, if fewer symbols are changed (received in error), then the receiver will know that errors have occurred. Thus a code, C is guaranteed to detect up to s errors if

$$d(C) \quad > \quad s. \tag{2.22}$$

In order to correct the errors, the code determines which codeword is closer to the received word in terms of the hamming distance. Thus the hamming distance divided by two would provide the decision point between the correct codeword and the closest incorrect codeword. The code can thus correct up to t errors where

$$t \quad = \quad \left\lfloor \frac{d(C) - 1}{2} \right\rfloor. \tag{2.23}$$

In order to guarantee a certain error correction capability, t, for all codewords in a code as well as all possible error patterns, the minimum distance is the most important parameter. It should, however, be remembered that there may be many codeword pairs with a Hamming distance much larger than the minimum distance. Thus for some error patterns, the code might be able to decode many more errors than t. Decoders which cannot exploit this property are called bounded distance decoders [6]. It can be shown [6] that communication at capacity is not possible with the use of bounded distance decoding. If a decoder is used which can exploit this property, then the minimum distance is no longer the most important parameter, rather the distribution of distances becomes the important factor.

2.4 LINEAR BLOCK CODES

A class of block codes which has found much use is linear block codes. A linear code has two defining properties:

1. Any two codewords can be added to produce a new codeword.
2. Multiplying any codeword with an element in $\mathbb{F}_q$ produces another codeword.

These two conditions imply that the all-zero codeword is always a codeword in a linear code. It also implies that since $\mathbb{F}_q^n$ is a vector space, a linear code is a linear subspace of $\mathbb{F}_q^n$. For linear codes the minimum distance can be found by the use of the concept of Hamming weight. The Hamming weight of a codeword, $w_H(\mathbf{x})$, is defined as the Hamming distance between the codeword and the all zero codeword, thus

$$w(\mathbf{x}) \quad := \quad d_H(\mathbf{x}, \mathbf{0}). \tag{2.24}$$

The minimum distance of a linear code is equal to the smallest Hamming weight of the nonzero codewords in the code:

$$d(C) \quad = \quad min\{w_H(\mathbf{x}) : \mathbf{x} \in C, \mathbf{x} \neq \mathbf{0}\}. \tag{2.25}$$

Another major advantage of linear codes is that since the code is a k-dimensional linear subspace, all one needs to describe the code is k basis vectors (codewords). All other codewords may be derived as linear combinations of these basis codewords. It should be noted that any k distinct nonzero codewords may be used to form the basis. A useful method for expressing the basis is in matrix form. Since the basis can be used to generate any codeword in the code, it is called a generator matrix. Formally, a generator matrix $\mathbf{G}$ for a $[n, k]$ linear code is a $k \times n$ full rank matrix, where the rows are the k basis vectors:

$$\mathbf{G} = \begin{pmatrix} \mathbf{g}_0 \\ \mathbf{g}_1 \\ \vdots \\ \mathbf{g}_{k-1} \end{pmatrix} = \begin{pmatrix} g_{0,0} & \cdots & g_{0,n-1} \\ g_{1,0} & \cdots & g_{1,n-1} \\ \vdots & & \vdots \\ g_{k-1,0} & \cdots & g_{k-1,n-1} \end{pmatrix} \tag{2.26}$$

The generator matrix provides a simple method for encoding data words. If $\mathbf{x}$ is an input data column vector of length k, then $\mathbf{y}$ is its corresponding codeword where

$$\mathbf{y} \quad = \quad \mathbf{G}^{\mathcal{T}}\mathbf{x}. \tag{2.27}$$

Note that this mapping from dataword to codeword is not unique, in the sense that a different set of basis vectors for the same code might have produced a different mapping. The performance of a code does not depend on the specific mapping used during encoding. A useful form of the generator matrix is called the systematic form. The result of encoding with a generator matrix in systematic form is that the data word is mapped directly into the first k symbols in the codeword. Thus, after errors have been corrected at the decoder, the data can be effortlessly extracted. A generator matrix is said to be in systematic form when

$$\mathbf{G} = [\mathbf{I}_k|\mathbf{P}], \tag{2.28}$$

where $\mathbf{I}_k$ is a $k \times k$ identity matrix and $\mathbf{P}$ is a $k \times (n-k)$ parity matrix. For each code, C, one may define a dual code $C^{\perp}$ as a code consisting of all codewords in $\mathbb{F}_q^n$ that are othogonal to all the codewords in C. Since all codewords in C can be generated by its generator matrix, it suffices to check that all codewords in $C^{\perp}$ are orthogonal to all the rows of $\mathbf{G}$. Let $\mathbf{H}$ be the $(n-k) \times n$ size generator matrix of $C^{\perp}$. Since all codewords in $C^{\perp}$ are created by $\mathbf{H}$ and orthogonal to $\mathbf{G}$,

$$\mathbf{G}\mathbf{H}^{\mathcal{T}} = 0. \tag{2.29}$$

$\mathbf{H}$ is generally referred to as the parity check matrix of C. Since all codewords in C are orthogonal to all codewords in $C^{\perp}$, any codeword in C multiplied by $\mathbf{H}$ will also be zero:

$$\mathbf{H}\mathbf{y} = \mathbf{0}, \tag{2.30}$$

where $\mathbf{y}$ is a codeword in C. Thus $\mathbf{H}$ may be used to check if any received word is a correct codeword (check the parity of the symbols). This result forms an integral part of the decoding of LDPC codes. It should also be noted that for a specific code, $\mathbf{H}$ is not unique and any set of basis vectors for $C^{\perp}$ can be used. However, because of the specific technique used for decoding LDPC codes, not all versions of $\mathbf{H}$, for the same code, will perform equally well [6].

Chapter Three

Sparse Graph Codes

The motivation for sparse graph codes can be found in the 1948 paper by Shannon [29]. In this paper the Shannon random ensemble was defined as:

SHANNON RANDOM ENSEMBLE : *Let the field $\mathbb{F}_q$ be fixed. Consider the ensemble of codes C (n, M) of length n and cardinality M. There are nM degrees of freedom in choosing a code, one for each element in each codeword. The ensemble consists of all q^{nm} possible codes. All codes are given equal probability. To sample from the ensemble simply pick a random code.*

In the paper it was shown that almost any code (a large percentage of codes in the ensemble) can be used for transmission at vanishing probabilities of error as long as the transmission rate is less than the capacity of the channel and n is large. Large random codes can thus provide a simple solution to the problem of finding capacity approaching code designs. The problem with purely random codes is that the complexity and the memory requirements of encoding and decoding can become prohibitive as the code size increases. A solution which combines the concept of large random codes with reasonable complexity is sparse graph codes that rely on belief propagation.

3.1 Graphs and Belief Propagation

From the description of the Shannon random ensemble it is evident that there are very many good codes if the code length becomes large enough. The problem that remains is to make the process of encoding and decoding efficient. Traditional codes introduce a lot of structure in

the encoding process which can then be used at the receiver to speed up decoding. However, random codes have no structure of which to take advantage. For large codes (required to achieve capacity), an optimal solution where all possible solutions are considered exhaustively is clearly unfeasible. One approach is to consider the decoding problem as the calculation of a marginal of a multivariate function. This is the approach of BP decoding.

3.1.1 Marginals of functions and message passing

The marginal of a function f with respect to a variable x_1 is defined as the summation of the evaluation of f over all possible permutations of the remaining variables $(x_2, \cdots, x_n)$. Thus, the marginal $\tilde{f}(x_1)$ is given by:

$$\tilde{f}(x_1) = \sum_{x_2, \cdots, x_n} f(x_1, x_2, \cdots, x_n) \tag{3.1}$$

which will be alternatively represented as:

$$\tilde{f}(x_1) = \sum_{\tilde{x}_1} f(x_1, x_2, \cdots, x_n) \tag{3.2}$$

where $\sum_{\tilde{x}_1}$ indicates the summation over all variables except x_1. The method used to compute a marginal of a function efficiently, makes use of the distributive law. Let $a, b, c \in \mathbb{F}_q$, the distributive law states:

$$ab + ac = a(b + c). \tag{3.3}$$

This property is applied to functions. As an example [6] one can consider a function in six variables $(x_1, \cdots, x_6)$ which may factorised as follows:

$$f(x_1, x_2, x_3, x_4, x_5, x_6) = f_1(x_1, x_2, x_3) f_2(x_1, x_4, x_6) f_3(x_4) f_4(x_4, x_5). \tag{3.4}$$

For this function the marginal with respect to x_1 can be calculated as follows:

$$\tilde{f}(x_1) = \sum_{\tilde{x}_1} f(x_1, x_2, x_3, x_4, x_5, x_6) \tag{3.5}$$

$$= \Big[\sum_{x_2, x_3} f_1(x_1, x_2, x_3) \Big] \Big[\sum_{x_4} f_3(x_4) \Big(\sum_{x_6} f_2(x_1, x_4, x_6) \Big) \Big(\sum_{x_5} f_4(x_4, x_5) \Big) \Big]. \tag{3.6}$$

The calculation of the marginal has now become the calculation of many smaller marginals. For this specific case the number of operations required has decreased from being on the order of $O(|\mathbb{F}_q|^6)$ to $O(|\mathbb{F}_q|^3)$, where $|F_q| = q$ is the cardinality of the alphabet from which $x_1, \cdots, x_n$ are

taken. A factor graph is a graphical representation of a factored function, and when combined with message passing it provides a convenient systematic way of calculating the marginal.

A factor graph is a bi-partite graph because it consists of two types of nodes. One kind of node (drawn as a circle) represents a variable and the other kind of node (drawn as a square) represents a factor of the function (and is thus a function itself). A variable node is connected to a factor node if the variable appears in the factor. Variable nodes are not connected to other variable nodes and factor nodes are not connected to other factor nodes. A factor graph is called a factor tree if there are no cycles in the graph. A cycle occurs when there is more than one path between any given variable node and factor node pair. A node connected to only one other node is called a leaf node. The connection between two nodes is called an edge. The bi-partite graph for the function in Equation 3.4 can be seen in figure 3.1.

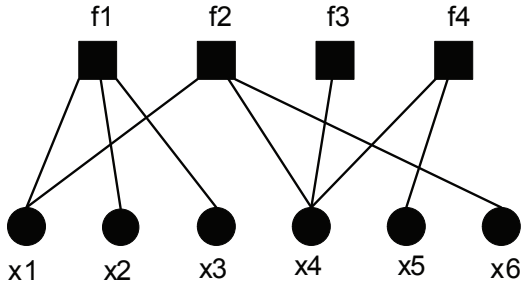

FIGURE 3.1: Bi-partite graphs of the function in Equation 3.4.

Calculating a marginal with a bi-partite graph is done through a process known as message passing. Message passing is used in many applications of graph theory [30]. Assuming that the graph is a tree, message passing starts with the leaf nodes. These nodes generate a message and pass it to the node connected to it. The transmitting node is known as a child node and the receiving node is known as a parent node. In terms of the tree structure, the parent node will be closer to the base of the tree. When a node has received messages from all its children nodes these messages are used to calculate a message, which is then passed to the parent. This process is repeated until the message has reached the base node (this node represents the variable over which the marginal is being calculated).

In the case where message passing is being used to calculate a marginal, the message being passed from one node to another is a function on $\mathbb{F}_q$ and is represented by a vector of

length q. The message from a factor node (f_j) to a variable node (x_i) represents the marginal of the factor f_j with respect to x_i. If the k^{th} position in the message corresponds to the element $\alpha_k \in \mathbb{F}$, then the value at that position corresponds to the value of f_j when $x_i = \alpha_k$. The message from a variable (x_i) node to a factor node represents the product of the marginals over x_i of the other factor nodes connected to x_i. The k^{th} position in this marginal vector can thus be interpreted as the value that the product of marginals over x_i would take if $x_i = \alpha_k$.

For a multivariate function, the marginal can be calculated with respect to each variable in the function. This can be done by the message passing process described above by simply viewing the tree as starting from the variable in question. Figure 3.2 shows the graph of the function in Equation 3.4 using first x_1 and then x_2 as the base nodes. To calculate the marginal of the function with respect to x_1, one would use Tree x_1 and to calculate the marginal of the function with respect to x_2, one would use Tree x_2. However, since the graph structure remains

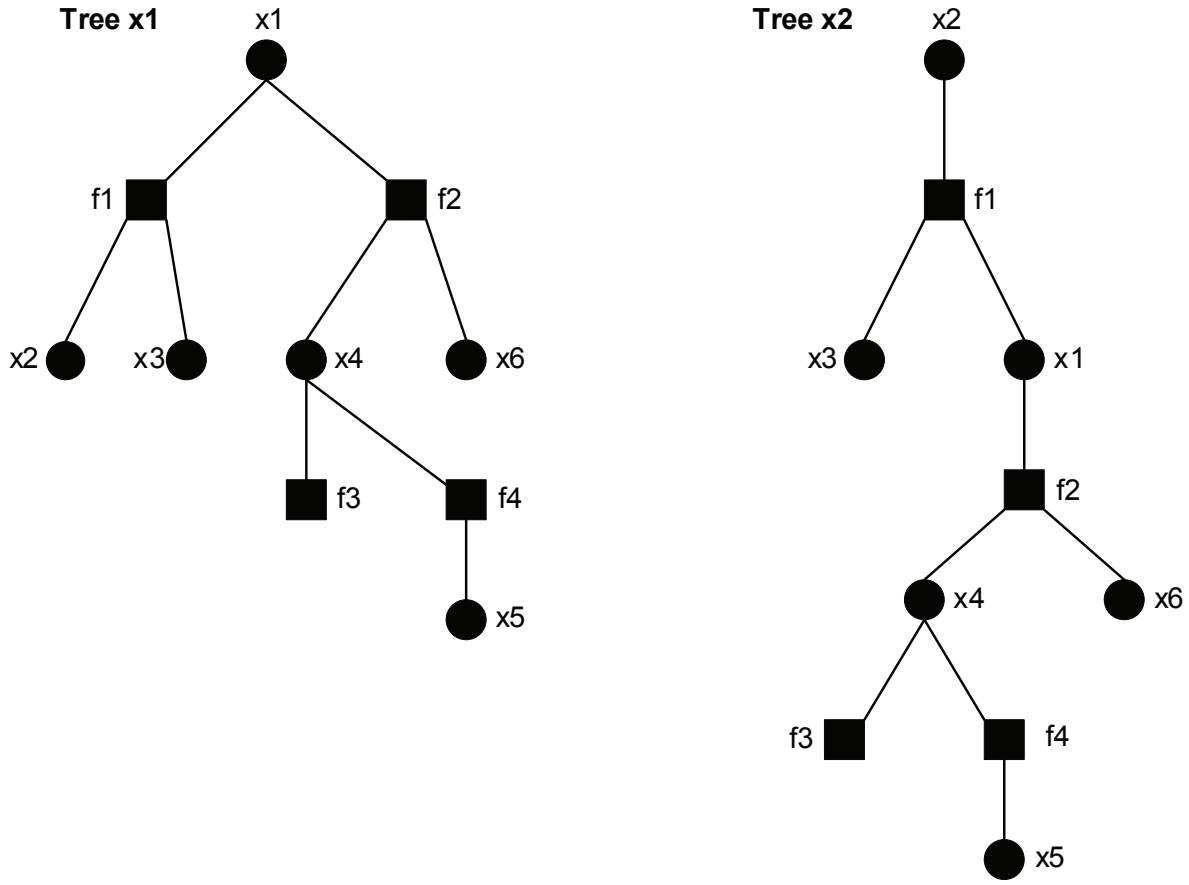

FIGURE 3.2: Tree graphs of the function in Equation 3.4 with x_1 and x_2 as the base.

the same, the marginals for all the variables may be calculated at the same time on the same graph. In this case, a node considers each connection, one at a time. For each connection an output message is calculated using all the input messages except the one from the connection for which it is currently calculating an output message. This approach can be seen in Figure 3.4. The message passing algorithm will continue until a message has been sent along each edge in both directions, at which time the marginals with respect to all the variables will have been calculated.

A marginal can only be calculated exactly using message passing on a factor graph when the graph is a tree. Figure 3.3 shows a simple graph with a cycle. The length of the cycle is equal to the number of edges that must be traversed to move from a node back to itself. The cycle in Figure 3.3 is thus of length 4. The problem with cycles is immediately apparent from the

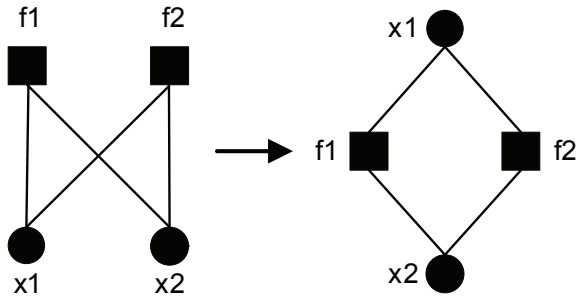

FIGURE 3.3: Graph of a cycle of length 4.

figure. Since there is no leaf node, the message passing process has no way of starting. Before any of the variable nodes can send a message, it must first receive a message from one of the factor nodes. However, before any of the factor nodes can send a message, it must first receive a message from one of the variable nodes. Exact marginalisation can thus not be performed if the graph is not a tree. The message passing algorithm may be adapted to allow for graphs with cycles (see section 3.1.4); however, the performance will always be strictly sub-optimal.

3.1.2 Decoding as marginalisation

In order to use marginalisation over graphs as a decoding procedure for block codes, the decoding process must be written as a marginal. Let $\mathbf{x}$ be the vector of transmitted symbols where $x_i \in \mathbb{F}_q \ \forall \ i$ and $\mathbf{c} \in C$ where C is the code in question. Let $\mathbf{y}$ be the vector of received

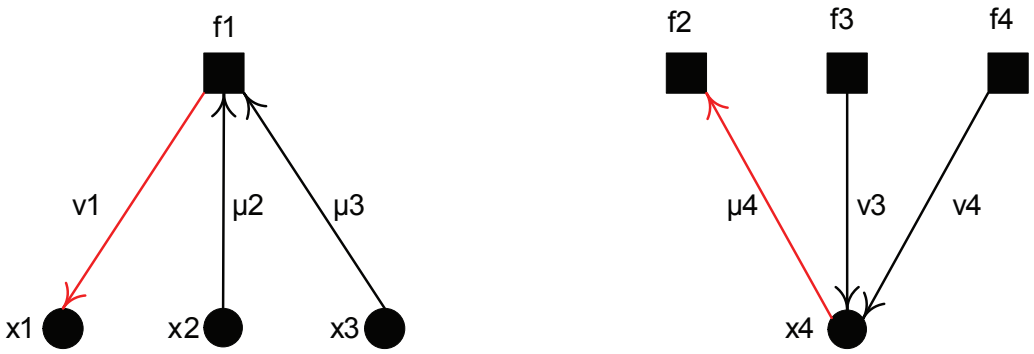

FIGURE 3.4: Graph depicting an isolated factor node and an isolated variable node.

symbols. The decoding problem can be stated as finding the most likely $\mathbf{x}$ given that $\mathbf{y}$ was received. In bitwise maximum a posteriori probability (MAP) decoding, this rule can be stated as:

$$\begin{aligned}
\hat{x}_i^{MAP}(\mathbf{y}) &= \text{argmax}_{x_i \in \mathbb{F}} p(x_i|\mathbf{y}) && (3.7)\\
&= \text{argmax}_{x_i \in \mathbb{F}} \sum_{\tilde{x}_i} p(\mathbf{x}|\mathbf{y}) && (3.8)\\
&= \text{argmax}_{x_i \in \mathbb{F}} \sum_{\tilde{x}_i} p(\mathbf{y}|\mathbf{x}) p_X(\mathbf{x}) && (3.9)\\
&= \text{argmax}_{x_i \in \mathbb{F}} \sum_{\tilde{x}_i} \left(\prod_j p(y_j|x_j) \mathbf{1}_{\{\mathbf{x} \in C\}} \right), && (3.10)
\end{aligned}$$

where the derivation used the assumption that the channel is memoryless, which gives:

$$p(\mathbf{y}|\mathbf{x}) = \prod_j p(y_j|x_j). \quad (3.11)$$

Equation 3.10 uses the assumption of uniform priors over all codewords and introduces the indicator function:

$$\mathbf{1}_{\{\text{condition}\}} = \begin{cases} 1 & \text{if condition true} \\ 0 & \text{otherwise.} \end{cases} \quad (3.12)$$

In this case the indicator function is used to determine whether a vector $\mathbf{x}$ is a member of the code. If $\mathbf{x}$ is not a member of the code then $p(\mathbf{x}) = 0$ and the term $\prod_j p(y_j|x_j)$ is removed from the summation of the marginal. The marginal $\sum_{\tilde{x}_i} (\prod_j p(y_j|x_j) \mathbf{1}_{\{\mathbf{x} \in C\}}$ in Equation 3.10 will be a vector of values corresponding to the elements in $\mathbb{F}$. The value in the marginal vector at position j, corresponding to element $\alpha_j \in \mathbb{F}_q$, can be interpreted as the probability that the symbol $x_i = \alpha_j$. By finding the element in the marginal with the largest value (argmax) the MAP decoding

is completed. It is, however, useful that soft outputs over all possibilities are directly available in the marginal after normalisation. In its current form, Equation 3.10 is no different from an exhaustive search through all possible codewords. The reduction in complexity comes from when the marginal is factorised. Specifically, the code membership function can be factorised by considering each parity check equation in the code (row of the parity check matrix) as a factor. To illustrate the process of creating a factor graph from a parity check matrix, an example will be used. Let the parity check matrix $\mathbf{H}$ of code C be given as:

$$\mathbf{H} = \begin{pmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}. \tag{3.13}$$

By using the fact that a vector $\mathbf{x}$ is a member of a code if $\mathbf{Hx} = \mathbf{0}$, the indicator function may be factorised as follows:

$$\mathbf{1}_{\{\mathbf{x}\in C\}} = \begin{cases} 1 & \text{if } \mathbf{Hx} = \mathbf{0} \\ 0 & \text{otherwise} \end{cases} \tag{3.14}$$

$$= \begin{cases} 1 & \text{if } (\mathbf{h}_1\mathbf{x} = 0) \text{ and } (\mathbf{h}_2\mathbf{x} = 0) \text{ and } (\mathbf{h}_3\mathbf{x} = 0) \\ 0 & \text{otherwise} \end{cases} \tag{3.15}$$

$$= \mathbf{1}_{\{x_1+x_2+x_4=0\}}\mathbf{1}_{\{x_3+x_4+x_6=0\}}\mathbf{1}_{\{x_4+x_5+x_7=0\}} \tag{3.16}$$

where $\mathbf{h}_i$ corresponds to the i^{th} row of $\mathbf{H}$ and the addition is done over the code field $\mathbb{F}_q$. The MAP decoding rule for this code may now be written as:

$$\hat{x}_i^{MAP}(\mathbf{y}) = \text{argmax}_{x_i\in\mathbb{F}} \sum_{\tilde{x}_i} \left(\prod_j p_{Y_j|X_j}(y_j|x_j)\mathbf{1}_{\{x_1+x_2+x_4=0\}}\mathbf{1}_{\{x_3+x_4+x_6=0\}}\mathbf{1}_{\{x_4+x_5+x_7=0\}} \right). \tag{3.17}$$

When using factor graphs for decoding, the graph is referred to as a Tanner graph [31]. The variable nodes are called symbol nodes and the factor nodes representing parity checks in the code are called check nodes. The Tanner graph for the marginal in this equation can be seen in Figure 3.5. Decoding (for all the symbols in $\mathbf{x}$) can now be done by running the message passing algorithm on this graph. Up until now, the message passing approach to calculating a marginal has been given in general terms. The specific calculations for the message passing algorithm when the marginal represents the decoding of a block code will now be explained.

3.1.3 Message passing rules for MAP decoding of block codes

Let the transmitted sequence be $\mathbf{x}$ where $x_i \in \mathbb{F}_q$. It is assumed that $\mathbb{F}_q$ is a field, and may thus be represented as $\mathbb{F}_q = \{0, \alpha^0, \alpha^1, \ldots, \alpha^{q-2}\}$. Let $\mu_{f_j}(x_i)$ represent the message from factor node f_j

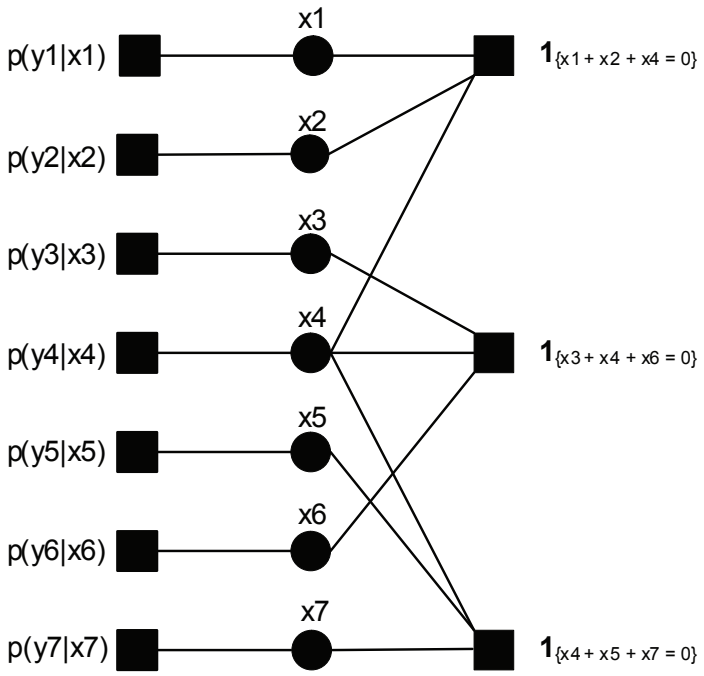

FIGURE 3.5: Tanner graph of the marginal function matched to the parity check matrix given in Equation 3.13.

to variable node x_i. Let $\mathbf{v}_{x_i}(f_j)$ represent the message being sent from symbol node x_i to check node f_j. The factor nodes which represents the channel are all leaf nodes and will simply send their initialised vectors $p(y_i|x_i)$ to the respective symbol nodes. Since these nodes are leaf nodes, they can now be removed from the factor graph and will no longer play any role in decoding. Since the output of a variable node is the product of the marginals sent to it, its update rule is a point wise multiplication of the received marginals. Thus, if a variable node is connected to d_v function nodes, the output along the j^{th} edge towards the j^{th} factor node connected to it, can be given by:

$$\mathbf{v}_{x_i}(f_j) = \prod_{\substack{n=1 \\ n\neq j}}^{d_v} \mu_{f_n}(x_i). \tag{3.18}$$

At the function node, the update rule is more complex. The derivation of the update rule will be shown by using an example. Consider the simple graph in Figure 3.6. Let the marginal be calculated with respect to x_1:

$$\tilde{f}(x_1) = \sum_{\tilde{x}_1} p(y_1|x_1)p(y_2|x_2)p(y_3|x_3)\mathbf{1}_{\{x_1+x_2+x_3=0\}} \tag{3.19}$$

Marginalisation requires the calculation of the function over all possible values of the variables. Let the field be $\mathbb{F}_2 = \{0, 1\}$. There are thus eight possible combinations for the variables in

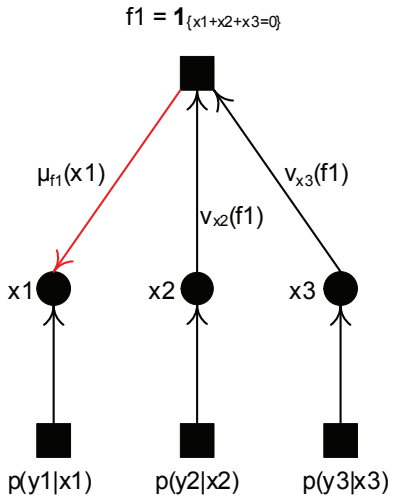

FIGURE 3.6: Simple graph for check node update example.

question. If the channel probability vectors are given by:

$$p(y_1|x_1) = (0.2, 0.8) \tag{3.20}$$
$$p(y_2|x_2) = (0.7, 0.3) \tag{3.21}$$
$$p(y_3|x_3) = (0.4, 0.6) \tag{3.22}$$

then the values of the function for the eight different sequences can be calculated as follows:

$$p(y_1|x_1=1)p(y_2|x_2=1)p(y_3|x_3=1)\mathbf{1}_{\{x_1+x_2+x_3=0\}} = 0.144 \cdot 0 = 0 \tag{3.23}$$
$$p(y_1|x_1=1)p(y_2|x_2=1)p(y_3|x_3=0)\mathbf{1}_{\{x_1+x_2+x_3=0\}} = 0.096 \cdot 1 = 0.096 \tag{3.24}$$
$$p(y_1|x_1=1)p(y_2|x_2=0)p(y_3|x_3=1)\mathbf{1}_{\{x_1+x_2+x_3=0\}} = 0.336 \cdot 1 = 0.336 \tag{3.25}$$
$$p(y_1|x_1=1)p(y_2|x_2=0)p(y_3|x_3=0)\mathbf{1}_{\{x_1+x_2+x_3=0\}} = 0.224 \cdot 0 = 0 \tag{3.26}$$
$$p(y_1|x_1=0)p(y_2|x_2=1)p(y_3|x_3=1)\mathbf{1}_{\{x_1+x_2+x_3=0\}} = 0.036 \cdot 1 = 0.036 \tag{3.27}$$
$$p(y_1|x_1=0)p(y_2|x_2=1)p(y_3|x_3=0)\mathbf{1}_{\{x_1+x_2+x_3=0\}} = 0.024 \cdot 0 = 0 \tag{3.28}$$
$$p(y_1|x_1=0)p(y_2|x_2=0)p(y_3|x_3=1)\mathbf{1}_{\{x_1+x_2+x_3=0\}} = 0.084 \cdot 0 = 0 \tag{3.29}$$
$$p(y_1|x_1=0)p(y_2|x_2=0)p(y_3|x_3=0)\mathbf{1}_{\{x_1+x_2+x_3=0\}} = 0.056 \cdot 1 = 0.056. \tag{3.30}$$

The marginal can now be calculated by summing over these probabilities:

$$f(x_1) = [0.036 + 0 + 0 + 0.056, 0 + 0.096 + 0.336 + 0] \tag{3.31}$$
$$= [0.092, 0.423]. \tag{3.32}$$

From this it can be seen that the MAP solution for x_1 is $x_1 = 1$. In order to get the marginal into a probability it must simply be normalised to add to 1. Thus:

$$p(x_1|\mathbf{y}) = \left(\frac{0.092}{0.092 + 0.423}, \frac{0.423}{0.423 + 0.092}\right) \tag{3.33}$$

$$= [0.178, 0.822] \tag{3.34}$$

The marginal will now be calculated again, this time by using the graph in Figure 3.6. The messages from the factor nodes representing the output of the channel to the variable nodes are simply the probability vectors $p(x_i|y_i)$. Nodes x_2 and x_3 then pass these messages to the factor node. The factor node then computes the marginal:

$$\mu_{f_1}(x_1) = p(y_2|x_2)p(y_3|x_3)\mathbf{1}_{\{x_1+x_2+x_3=0\}} \tag{3.35}$$

$$= [p(y_2|x_2 = 1)p(y_3|x_3 = 1) + p(y_2|x_2 = 0)p(y_3|x_3 = 0), \tag{3.36}$$

$$p(y_2|x_2 = 1)p(y_3|x_3 = 0) + p(y_2|x_2 = 0)p(y_3|x_3 = 1)].$$

This message is then passed to variable node x_1. Variable node x_1 has now received messages from all the factor nodes connected to it and simply performs a point wise multiplication to yield the final result:

$$f(x_1) = p(y_1|x_1) \cdot \mu_{f_1}(x_1) \tag{3.37}$$

$$= \Big[p(y_1|x_1 = 0)[p(y_2|x_2 = 1)p(y_3|x_3 = 1) + p(y_2|x_2 = 0)p(y_3|x_3 = 0)], \tag{3.38}$$

$$p(y_1|x_1 = 1)[p(y_2|x_2 = 1)p(y_3|x_3 = 0) + p(y_2|x_2 = 0)p(y_3|x_3 = 1)]\Big]$$

$$= [0.092, 0.423], \tag{3.39}$$

which is the same as above. The calculation of $\mu_{f_1}(x_1)$ as shown above requires the calculation of the value of the function for all combinations of all the variables. One can simplify this by noting that the combinations for which the function $\mathbf{1}_{\{x_1+x_2+x_3\}}$ is equal to 1 are the only ones worth considering. Specifically for the $x_1 = 0$ position the combinations of x_2 and x_3 are 00 and 11, whereas for the $x_1 = 1$ position the combinations are 10 and 01. This can be seen viewed as a circular convolution of the $p(y_2|x_2)$ and $p(y_3|x_3)$ vectors. Let the k^{th} component of the vector resulting from the $\mathbb{F}_q$ convolution operator $\odot_{\mathbb{F}}$ between two vectors $\mathbf{v}^1$ and $\mathbf{v}^2$ be given by [7]:

$$\left[\mathbf{v^1} \odot_{\mathbb{F}} \mathbf{v^2}\right]_k = \sum_{a \in \mathbb{F}} v_a^1 \cdot v_{k-a}^2, \quad k \in \mathbb{F}_q, \tag{3.40}$$

where $k-a$ is done over the field $\mathbb{F}_q$. The calculation of the marginal can now be written in terms of a convolution:

$$\mu_{f_1}(x_1) = p(y_2|x_2)p(y_3|x_3)\mathbf{1}_{\{x_1+x_2+x_3=0\}} \quad (3.41)$$

$$= p(y_2|x_2) \odot_{\mathbb{F}_2} p(y_3|x_3). \quad (3.42)$$

This update rule can be generalised to yield the following rule for creating the output message from a factor node. Consider the j^{th} factor node connected to d_c variable nodes. The output message to the i^{th} variable node can be given by:

$$\mu_{f_j}(x_i) = \mathbf{v}_{x_1}(f_1) \odot_{\mathbb{F}} \mathbf{v}_{x_2}(f_1) \odot_{\mathbb{F}} \cdots \odot_{\mathbb{F}} \mathbf{v}_{x_{i-1}}(f_1) \odot_{\mathbb{F}} \mathbf{v}_{x_{i+1}}(f_1) \odot_{\mathbb{F}} \cdots \odot_{\mathbb{F}} \mathbf{v}_{x_K}(f_1) \quad (3.43)$$

$$= \bigodot_{k=1,k\neq i}^{K} \mathbf{v}_k(f_1). \quad (3.44)$$

This equation is valid for all codes defined over finite fields. Up until now, only codes where the parity check matrix has 1s and 0s have been considered. In the general case, the elements in the parity check matrix may be any of the elements in the field over which the code is defined. Thus, for codes defined over higher order fields than $\mathbb{F}_2$, the non-zero elements in the parity check matrix will have values other than 1. In this case, the parity check equation corresponding to the j^{th} row of the parity check matrix will have the following general form:

$$\sum_{i=1}^{K} h_i x_i = 0, \quad (3.45)$$

where h_i is the element in the parity check. The addition is done over $\mathbb{F}_q$. This changes the function at the function node to:

$$f_j = \mathbf{1}_{\{\sum_{i=1}^{K} h_i x_i=0\}}. \quad (3.46)$$

This affects the rule for the output of a function node since it can no longer be simply represented by a convolution. The convolution is, however, a very convenient representation. A solution to this problem is to introduce a function node on each edge of the graph. This node will be referred to as the permutation node. When the message is moving from the variable node towards the factor node the effect that multiplication by h_i has on the marginal of the i^{th} edge is a multiplication by h_i of the indices of the marginal. The new function node will thus perform a permutation on the marginal corresponding to the value of h_i. If $h_i = \alpha^j$ then all the values in the vector except the value corresponding to the zero element will rotate to the left by j positions. In other words, the value which used to be at the position which corresponds to α^k will now

be at the position which corresponds to $\alpha^k \times \alpha^j$. When the message is moving from the factor node towards the variable node, the inverse happens and the permutation is reversed. From the introduction of the new node, the function at the factor node will once again be:

$$f_j = \mathbf{1}_{\{\sum_{i=1}^{K} x_i = 0\}}. \tag{3.47}$$

This allows for the convolution-based update rule. The calculation of the convolution is not fast and the complexity grows exponentially with both the field size and the number of nodes connected to a check node. This complexity can be reduced by noting that a convolution becomes a multiplication when the Fourier transform has been applied. Thus [5][32]:

$$\mu_{f_j}(x_i) = \bigodot_{k=1,k\neq i}^{K} \mathbf{v}_k(f_1) \tag{3.48}$$

$$= \mathcal{F}\left[\prod_{k=1,k\neq i}^{K} \mathcal{F}(\mathbf{v}_k(f_1))\right]. \tag{3.49}$$

Over finite fields, the Fourier transform reduces to a Hadamard transform and can be efficiently implemented using the butterfly approach of the FFT. The algorithm using the FFT is called FFT-BP. Adding the permutation node and the Fourier transform, a generic representation of the Tanner graph can be seen in figure 3.7.

3.1.4 Graphs with cycles

The assumption that the graph of the code is a tree has allowed for the use of message passing to perform MAP decoding. The problem is that codes without cycles have been shown to have poor performance [6]. The message passing algorithm must thus be modified to allow for graphs with cycles. This can be done by changing the update procedure. The algorithm for FFT-BP on codes with cycles over an arbitrary field is given in Algorithm 1.

The leaf nodes in the graph (the channel nodes) pass their messages to the variable nodes. In a graph without cycles, each node can wait for messages from all the edges connected to it (except one) before starting to create outputs. In graphs with cycles this is not possible. A solution to this problem is for each variable node to forward the message that came from the channel node to all the check nodes. In this way a message has been sent on every edge. Each check node can then calculate outputs for each edge in the standard manner. From the outputs of the check nodes, each variable node can then create outputs using the standard update rules.

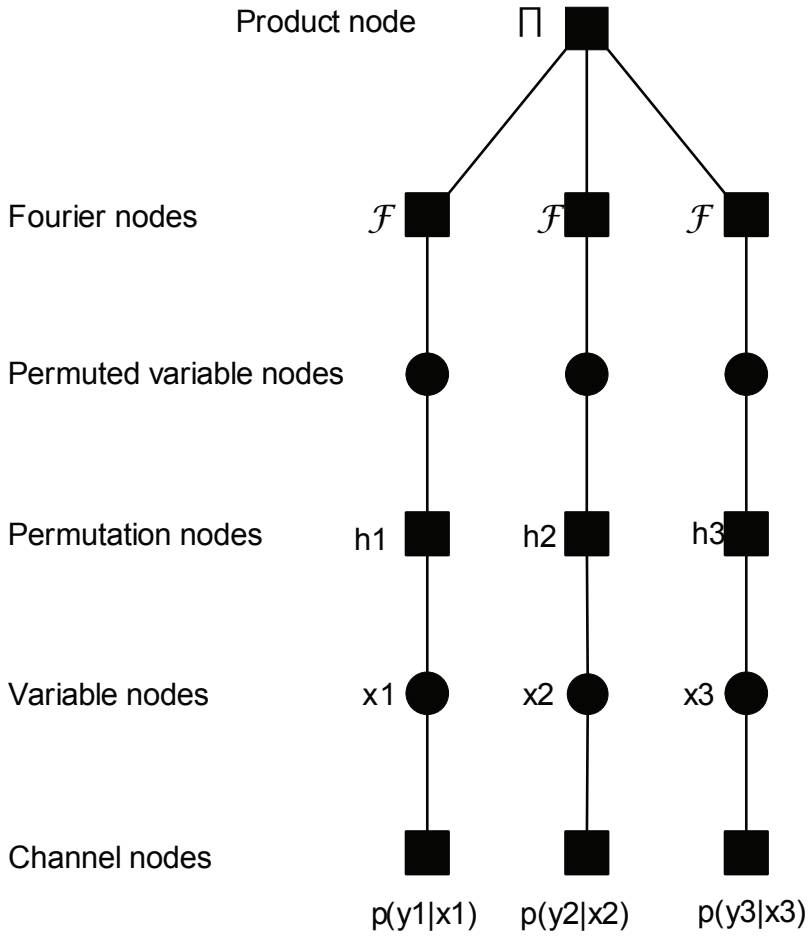

FIGURE 3.7: General graph for FFT-BP decoding at the factor node.

Since the initial messages are not exact and the graph is not a tree, a single iteration of the network updates will not give the exact solution to the marginal. Instead, many iterations must be run where the result of each iteration converges towards the correct result. This leads to another interpretation of the effect that cycles have on BP. Since there is more than one path connecting a variable node to a check node, the information sent from the variable node to the check node will pass back to the variable node. This results in a positive feedback effect, which degrades performance. The effect of this positive feedback can be reduced by designing the code such that the cycles are as long as possible. The length of the shortest cycle in the code is referred to as the girth of the code.

Algorithm 1 FFT-Belief Propagation over $\mathbb{F}_q$

procedure INITIALISATION
  $\mathbf{v}_k(f_i) = p(y_k|x_k)\ \forall\ i$ ▷ Outputs of Variable Nodes are Set to Channel Node Outputs
end procedure
procedure DECODING ITERATIONS
  for $Iter\_Count = 0 \rightarrow Num\_Iter$ **do**
    Normalise $\mathbf{v}_k(f_i)\ \forall\ k, i$ ▷ All vectors must sum to 1
    Permute $\mathbf{v}_k(f_i)\ \forall\ k, i$ ▷ Permutation Node Update
    $\mathbf{v}_k^{\mathcal{F}}(f_i) = \mathcal{F}(\mathbf{v}_k(f_i))\ \forall\ k, i$ ▷ Fourier Node Update
    $\mu_{f_j}^{\mathcal{F}}(x_i) = \prod_{k=1,k\neq i}^{K} \mathbf{v}_k^{f}(f_i)\ \forall\ i, j$ ▷ Product (Check) Node Update
    $\mu_{f_j}(x_i) = \mathcal{F}(\mu_{f_j}^{\mathcal{F}}(x_i))\ \forall\ k, i$ ▷ Fourier Node Update
    Normalise $\mu_{f_j}(x_i)\ \forall\ i, j$ ▷ All vectors must sum to 1
    Permute $\mu_{f_j}(x_i)\ \forall\ i, j$ ▷ Permutation Node Update
    $\mathbf{v}_k(f_i) = \prod_{\substack{i=1 \\ i\neq j}}^{k} \mu_{f_i}(x_i)\ \forall\ k, i$ ▷ Variable Node Update
  end for
end procedure
procedure CALCULATE FINAL OUTPUT
  $\mathbf{v}_k = \prod_{i=1}^{k} \mu_{f_i}(x_k)\ \forall k$
  Normalise $\mathbf{v}_k\ \forall k$ ▷ All vectors must add to 1
end procedure

3.2 LDPC CODES

LDPC codes which were first developed by Gallager in [2] and then rediscovered by MacKay and Neal [3], are defined as being codes which have at least one sparse Tanner graph. A sparse Tanner graph is a graph with few edges. This corresponds to a parity check matrix with few non-zero elements. The reason for the sparsity (low density) is that it reduces the number and increases the length of the cycles in the graph. The number of nodes connected to a node is the degree of the node. LDPC codes for which each check node has the same degree and each variable node has the same degree are known as regular LDPC codes, while codes for which the degrees of the nodes vary are known as irregular LDPC codes. Regular LDPC codes are denoted as (l, r) codes where l is the degree of the check nodes and r is the degree of the variable nodes. It has been found [6] that irregular LDPC codes in general perform better than regular LDPC codes, especially when the code is binary. Irregular LDPC codes are described by defining a

check node distribution. This is done by using a polynomial:

$$\Lambda(x) = \sum_{i=1}^{l_{max}} \Lambda_i x^i \tag{3.50}$$

where l_{max} denotes the maximum degree of a check node and Λ_i indicates the number of check nodes with degree i. A similar polynomial may be defined for the distribution of the degree of the variable nodes:

$$P(x) = \sum_{i=1}^{r_{max}} P_i x^i \tag{3.51}$$

where r_{max} denotes the maximum degree of a variable node and P_i indicates the number of variable nodes with degree i. Irregular codes may alternatively be described from the perspective of the edges. Thus:

$$\lambda(x) = \frac{\Lambda'(x)}{\Lambda'(1)} \tag{3.52}$$

$$= \sum_{i=1}^{l_{max}} \lambda_i x^{i-1} \tag{3.53}$$

and

$$\rho(x) = \frac{P'(x)}{P(1)} \tag{3.54}$$

$$= \sum_{i=1}^{r_{max}} \rho_i x^{i-1}. \tag{3.55}$$

In this case $\lambda_i(\rho_i)$ is interpreted as the fraction of edges connected to a check node (variable node) of degree i. All codes following a certain degree distribution are said to belong to an ensemble and it can be shown that any code randomly drawn from an ensemble will perform approximately the same [6].

3.2.1 Design procedures

The design of an LDPC code comes down to a design of the parity check matrix and its equivalent Tanner graph. For the binary case a procedure known as Density Evolution (DE) was developed to optimise the degree distributions of the code by considering the PDF of the received symbols. By using this procedure, codes that can operate within a fraction of a decibel from capacity in AWGN and fading channels have been designed. DE assumes that the code is of infinite length. This creates the problem that codes following the degree distribution

must be very large. The code length directly effects the delay (latency) introduced by the code into the communication system, thus making large codes unsuitable for delay-limited systems such as voice communication. Thus, the design of small to medium frame length codes is an important problem for reliable communication in delay-limited systems. For smaller codes (that cannot effectively be approximated as having infinite length) some heuristic methods have been developed to improve performance when using DE to design the code. In the case of binary LDPC codes with a short frame length, the problem is that there is not enough space to place enough parity checks to meet the optimised degree distributions while maintaining a high girth. A solution to this problem is to design the code over a higher order field. These so-called NB-LDPC codes have the advantage of a sparser Tanner graph while maintaining a higher binary equivalent density. This is because the non-zero elements in the parity check matrix are drawn from a higher order field and have a binary representation with multiple 1's while still only representing a single edge in the tanner graph. Several studies have shown that the NB-LDPC codes outperform binary LDPC codes when the frame length is constrained [4] [33] [34]. The general trend is that the performance improvement increases with field size and is often more pronounced with higher code rates. The trade off incurred when using larger field sizes is that the decoding process becomes more complex. However, with the use of the FFT-BP decoding algorithm, the increase in complexity is not prohibitive.

Another problem that LDPC codes face, is that the encoding process is computationally and memory-storage intensive. Since there is no structure in the code, the encoder is required to store the relatively large generator matrix and perform a matrix multiplication. Several attempts have been made to mitigate the complexity and storage requirements of the encoder [33] [35]. Many of these attempts operate by introducing some structure into the parity check matrix. The method chosen for this dissertation is to design the code to be QC. QC codes can be efficiently encoded using shift registers.

3.2.2 QC-LDPC Codes

A QC code is defined as a code where every codeword can be cyclicly shifted by l positions and still produce a codeword. The degree of the QC code is l. A QC-LDPC code has a parity check matrix that is made up of rotated identity matrices of size $l \times l$. The advantage of QC-LDPC codes over random LDPC codes is that the encoder can be implemented with shift registers

[36]. The encoding process is thus very fast and computationally inexpensive compared to random LDPC codes, which are implemented with a matrix multiply.

The design of QC binary codes has been well studied and several possible design procedures have been proposed [10] [8] [37] [38]. The design problem is in finding rotations for the identity matrices that result in large girths. Also, the size of the rotated matrices relative to the total size of the code affects the resultant degree distribution of the code. Owing to the definition of QC-LDPC codes, they have regular degree distributions. In an effort to create irregular degree distributions, one may replace some of the identity matrices with zero matrices.

Designing NB-QC-LDPC codes can be as simple as designing a binary QC-LDPC code and replacing the 1s with elements from the field over which the NB-QC-LDPC code should be defined. The choice of elements can have a large effect on the performance of the code and will be discussed later. The difficulty with this approach is that degree distributions that work well for binary codes do not necessarily work well for NB codes and vice verca. Put more generally, the size of the field over which the NB-LDPC code is being defined will affect the optimal degree distribution. Some results have shown [39] that with large field sizes the matrix should be ultra-sparse. In the next section the performance of several different construction techniques will be compared.

3.3 QC-LDPC CODES BASED ON FINITE FIELDS

3.3.1 Construction of QC-LDPC codes by matrix dispersion

A convenient method for representing and constructing QC-LDPC codes was developed in [8]. A QC-LDPC code's parity check matrix consists of cyclicly shifted identity matrices. In this method the cyclic shift is indiced by a finite field element. The finite field used, $\mathbb{F}_q$, will be called the construction field. Let α be the primitive element of the construction field then the powers of α $(\alpha^{-\infty}, \alpha^0, \ldots, \alpha^{q-2})$ will give all the elements of the field. From this, a location vector $\mathbf{z}$ may be defined as follows:

$$\mathbf{z}(\alpha^{\mathbf{i}}) = \{z_0, z_1, \ldots, z_{q-2}\}, \tag{3.56}$$

where $z_i = 1$ and $z_k = 0\ \forall k \neq i$. For $\alpha^{-\infty}$, $\mathbf{z}$ is defined as the all-zero vector. Let σ be an element of $GF(q)$. $\mathbf{z}(\sigma\alpha)$ will correspond to the location vector of σ cyclicly shifted to the right by one

position. A $(q-1)\times(q-1)$ sized matrix $\mathbf{A}(\sigma)$ can then be created by using $\mathbf{z}(\sigma)$, $\mathbf{z}(\sigma\alpha)$, $\mathbf{z}(\sigma\alpha^2)$, $\ldots$, $\mathbf{z}(\sigma\alpha^{\mathbf{q-2}})$ as the rows of the matrix. As an example consider, $GF(4)$, $\sigma = \alpha^2$, $\mathbf{A}(\sigma)$ will then be:

$$\mathbf{A}(\sigma) = \begin{pmatrix} 0 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}. \tag{3.57}$$

This matrix is thus the $(q-1)\times(q-1)$-fold matrix expansion of σ. The expansion of the zero element ($\alpha^{-\infty}$) is a $(q-1)\times(q-1)$ all zero matrix. Using this approach, the design of a QC-LDPC code requires one to create a matrix consisting of elements in the construction field. This matrix is then expanded by using the $(q-1)\times(q-1)$-fold matrix expansion of each element. The design problem is to choose the correct elements.

3.3.2 Requirements for code with a minimum girth equal to 6

In order for a code to have no cycles of length 4 and thus a girth of at least 6, it must meet the row-column (RC) constraint. The RC constraint states that no two rows may have non-zero elements at the same positions in more than one column. Consider a $m \times n$ matrix $\mathbf{W}$:

$$\mathbf{W} = \begin{pmatrix} \mathbf{w_0} \\ \mathbf{w_1} \\ \vdots \\ \mathbf{w_{m-1}} \end{pmatrix} = \begin{pmatrix} w_{0,0} & w_{0,1} & \ldots & w_{0,n-1} \\ w_{1,0} & w_{1,1} & \ldots & w_{1,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ w_{m-1,0} & w_{m-1,1} & \ldots & w_{m-1,n-1} \end{pmatrix}, \tag{3.58}$$

where $w_{ij} \in \mathbb{F}_q$. Two constraints may be imposed on this matrix:

1. $\alpha^k\mathbf{w_i}$ and $\alpha^l\mathbf{w_i}$ must differ in at least $n-1$ places for: $0 < i < m$, $0 < k, l < q-12$ and $k \neq l$.

2. $\alpha^k\mathbf{w_i}$ and $\alpha^l\mathbf{w_j}$ must differ in at least $n-1$ places for: $0 < i, j < m$, $i \neq j$ and $0 < k, l < q-2$.

These constraints are known as the α-multiplied row constraints and imply that each row of $\mathbf{W}$ may contain at most one zero element. Let $\mathbf{H}$ be the matrix created when each element in $\mathbf{W}$ is expanded using the $(q-1)\times(q-1)$-fold matrix expansion. Thus:

$$\mathbf{H}(\mathbf{W}) = \begin{pmatrix} \mathbf{A}(w_{0,0}) & \mathbf{A}(w_{0,1}) & \ldots & \mathbf{A}(w_{0,n-1}) \\ \mathbf{A}(w_{1,0}) & \mathbf{A}(w_{1,1}) & \ldots & \mathbf{A}(w_{1,n-1}) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}(w_{m-1,0}) & \mathbf{A}(w_{m-1,1}) & \ldots & \mathbf{A}(w_{m-1,n-1}) \end{pmatrix}. \tag{3.59}$$

From the structure of the location vectors and the α-multiplied row constraints, it can be shown that $\mathbf{H}$ will meet the RC constraint [8]. For any pair (γ, ρ) of integers with $1 < \gamma < m$ and $1 < \rho < n$, let $\mathbf{H}(\gamma, \rho)$ be a $\gamma \times \rho$ subarray of $\mathbf{H}$. $\mathbf{H}(\gamma, \rho)$ is then a $\gamma(q-1) \times \rho(q-1)$ matrix and also satisfies the RC-constraint. Then the null space over of $\mathbf{H}(\gamma, \rho)$ gives a QC-LDPC code of length $\gamma(q-1)$ with rate at least $(\gamma - \rho)/\gamma$, whose Tanner graph has a girth of at least 6.

In order to change the binary QC-LDPC code described above into an NB-QC-LDPC code, the 1's must be replaced by elements from a finite field, $\mathbb{F}_{q2}$. This field will be called the code field. The code field and the construction field may be the same field, but it is not required. There are several methods for replacing the 1s with elements from the code field. The method used in [9] does this by redefining the location vector as:

$$\mathbf{z}(\alpha^{\mathbf{i}}) = \{z_0, z_1, \ldots, z_{q-2}\} \tag{3.60}$$

where $z_i = \alpha^i$ and $z_k = 0 \; \forall k \neq i$. Another method is to pick the elements at random from the code field. Work was also done in [34] to find optimum choices based on the binary image of the code. However, it was found that for code fields where $q_2 \geq 256$ the optimised method will not outperform random selection. Several methods to create codes that meet the α-multiplied row constraints will be described in the next sections.

3.3.3 Girth 6 codes based on dispersion of the multiplicative group

Consider a construction field $\mathbb{F}_q$. Let m be the largest prime factor of $q-1$, thus $cm = q-1$. Let α be a primitive element of $\mathbb{F}_q$ and $\beta = \alpha^c$. Then β is an element of $\mathbb{F}_q$ of order m (m is the smallest integer such that $\beta^m = 1$). The set $G_m = \{1, \beta, \beta^2, \ldots, \beta^{m-1}\}$ forms a cyclic subgroup of the multiplicative group $G_{q-1} = \{1, \alpha, \ldots, \alpha^{q-1}\}$ of $\mathbb{F}_q$. Let $\mathbf{W}$ be a $m \times m$ matrix over $\mathbb{F}_q$:

$$\mathbf{W} = \begin{pmatrix} \mathbf{w_0} \\ \mathbf{w_1} \\ \vdots \\ \mathbf{w_{m-1}} \end{pmatrix} = \begin{pmatrix} 1 & 1 & \ldots & 1 \\ \beta & \beta^2 & \ldots & \beta^{m-1} \\ \vdots & \vdots & \ddots & \vdots \\ \beta^{m-1} & (\beta^2)^{m-1} & \ldots & (\beta^{m-1})^{m-1} \end{pmatrix}. \tag{3.61}$$

It was shown in [8] that this matrix meets the requirements for a girth-6 code as given above. Therefore, after the $(q-1) \times (q-1)$-fold matrix expansion, as defined above, the resultant code will have a girth of at least 6.

3.3.4 Structured NB-QC-LDPC codes with large girth

This construction method was developed in [10]. A parity check matrix is said to be ultra-sparse if it has a uniform column weight of 2. Ultra-sparse binary LDPC codes have been shown to have poor performance [6] as they will have many codewords with a low hamming weight and thus have a small minimum distance. It has, however, been shown that the performance of ultra-sparse NB-LDPC codes is quite good and improves with the field size. Let $\mathbf{P}$ be an $m \times m$ permutation matrix defined by:

$$P_{ij} = \begin{cases} 1 & \text{if } i+1 = j \bmod m \\ 0 & \text{otherwise.} \end{cases} \tag{3.62}$$

The parity check matrix of a (J,L)-regular QC-LDPC code is made up of $J \times L$ permutation matrices :

$$\mathbf{H} = \begin{pmatrix} \mathbf{P}^{a_{11}} & \dots & \mathbf{P}^{a_{1L}} \\ \vdots & \vdots & \vdots \\ \mathbf{P}^{a_{J1}} & \dots & \mathbf{P}^{a_{JL}} \end{pmatrix}. \tag{3.63}$$

Choosing values for $\mathbf{a}$ is part of the design problem. For ultra-sparse codes $J = 2$ and the rate is controlled with L. It has been shown that the maximum girth for a code with $L \geq 3$ is 12. In [10] this bound is met by taking the top row as identity matrices and using the Hoey sequence for $\mathbf{a}$. The Hoey sequence is given below for the first 18 numbers:

$$0, 1, 3, 7, 12, 20, 30, 44, 65, 80, 96, 122, 147, 181, 203, 251, 289, 360. \tag{3.64}$$

To obtain an NB code from this binary image, a substitution is done where each 1 is replaced with an element from the desired field. In [10] an approach which uses a unit weight circulant was used. This approach is similar to the approach defined in equation 3.60. It works by defining the unit weight circulant matrix of size m over $GF(q)$:

$$\tilde{\mathbf{I}}_m = \begin{pmatrix} \alpha^0 & 0 & \dots & 0 \\ 0 & \alpha^1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \alpha^{m-1} \end{pmatrix} \tag{3.65}$$

where $m < q$. Using this matrix the code structure is defined as:

$$\mathbf{H} = \begin{pmatrix} \tilde{\mathbf{I}}_m & \dots & \tilde{\mathbf{I}}_m \\ \mathbf{P}^{a_{21}}\tilde{\mathbf{I}}_m & \dots & \mathbf{P}^{a_{2L}}\tilde{\mathbf{I}}_m \end{pmatrix}. \tag{3.66}$$

However, if the field size of the code is much larger than the size of the permutation matrix, then the weight of the expanded matrix is relatively low, which results in poor performance. In [39] it was found that for $\mathbb{F}_{256}$ and larger, the optimal choice of NB elements can be closely achieved by randomly selecting the elements from the field. This is the approach that was taken in this dissertation.

3.4 CODES USED IN THIS DISSERTATION

The codes used in this dissertation were limited to the following parameters: The number of

TABLE 3.1: NB-QC-LDPC code parameters

Code parameters	
Code rate	$\frac{1}{2}$
Field size (q)	256
Code length (n) in bits	2304
Data length (k) in bits	1152
Code length (n) in symbols	288
Data length (k) in symbols	144

encoded bits was chosen as 2304 in order to be an integer multiple of the number of bits in the ST, SF and STF codes used later in the dissertation. $\mathbb{F}_{256}$ was chosen as this represents a byte of data which is a standard unit. The code will thus not need to break message bytes into sections. $\mathbb{F}_{256}$ is also a large enough field to demonstrate the performance of NB-LDPC codes. Four codes matching the parameters above were created using the method described in section 3.3.3:

1. The first code used a construction field of $\mathbb{F}_{32}$ and selected NB elements using the method given in Equation 3.60. This code will be called $C1$.

2. The second code used a construction field of $\mathbb{F}_{32}$ and selected NB elements using the random method. This code will be called $C2$.

3. The third code used a construction field of $\mathbb{F}_{64}$ and selected NB elements using the method given in Equation 3.60. This code will be called $C3$.

4. The fourth code used a construction field of $\mathbb{F}_{64}$ and selected NB elements using the random method. This code will be called $C4$.

Two codes matching the parameters above were created using the method described in section 3.3.4.

1. The first code used $J = 2$, $L = 4$, $m = 77$ and the first four elements of the Hoey sequence. The NB elements were selected using the method given in Equation 3.66. This code will be called $C5$.

2. The second code used $J = 2$,$L = 4$,$m = 77$ and the first four elements of the Hoey sequence. The NB elements were selected using the random method. This code will be called $C6$.

3.4.1 Performance of codes in an AWGN channel

In this study the performance of the 6 different codes will be compared. The simulations were performed on an AWGN channel using BPSK modulation. Table 3.2 shows a summary of the E_B/N_0 required by all the codes in order to achieve a BER of 10^{-4}. The complete performance curves can be seen in Figures 3.8, 3.9, 3.10, 3.11, 3.12 and 3.13. None of the codes showed any signs of approaching an error floor in the simulated region.

TABLE 3.2: NB-LDPC code comparison, AWGN Channel

E_b/N_0 required to achieve a BER of 10^{-3}[dB]						
Code	$C1$	$C2$	$C3$	$C4$	$C5$	$C6$
E_b/N_0	3.505	3.635	2.83	2.22	3.25	1.3

When comparing $C3$ with $C1$, one can see that $C3$ outperforms $C1$ by 0.675 dB. This is because the construction field for $C3$ is twice the size of the construction field for $C1$. This means that there are fewer rotated matrices in $C1$ than in $C3$, and $C3$ thus has a sparser parity check matrix. This result thus agrees with the general concept that for NB-LDPC codes, matrices should be sparse. Comparing $C3$ with $C5$ one can see that the girth 6 code ($C3$) outperforms the girth 12 code ($C5$) code by 0.42 dB. This indicates that girth is not the only factor when determining the performance of a code. When comparing the performance of $C4$ with $C3$, one can see an improvement of 0.61 dB. This improvement is due to using random nonzero elements over using the unit weight circulant approach. Comparing the performance of $C6$ with $C5$, one can

see an improvement of 1.92 dB. This performance is also due to the use of random non-zero elements. With this improvement the girth 12 code outperforms the girth 6 code. Since $C6$ has the best performance, it was used for all the rest of the simulations in this dissertation. Figures 3.14 and 3.15 show the performance of $C6$ in quantised channel conditions. From the figure one can see that the code can operate without any loss in performance with only 6 quantisation bits per BPSK modulation symbol.

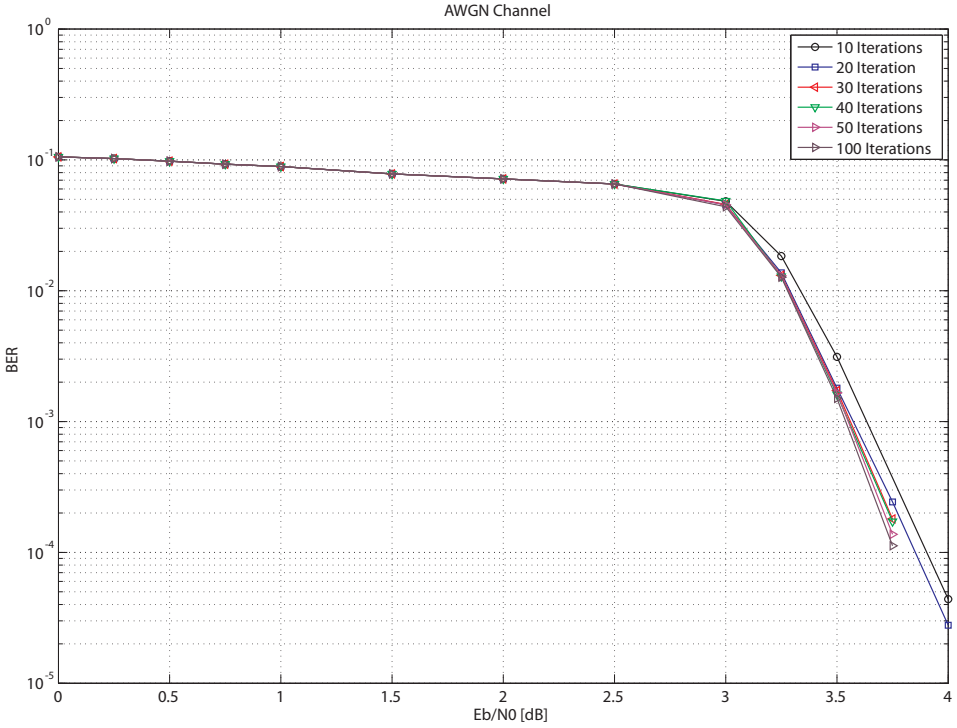

FIGURE 3.8: Performance of the girth-6 NB-QC-LDPC code using the $\mathbb{F}_{32}$ construction field with the unit weight circulant NB elements in an AWGN channel ($C1$).

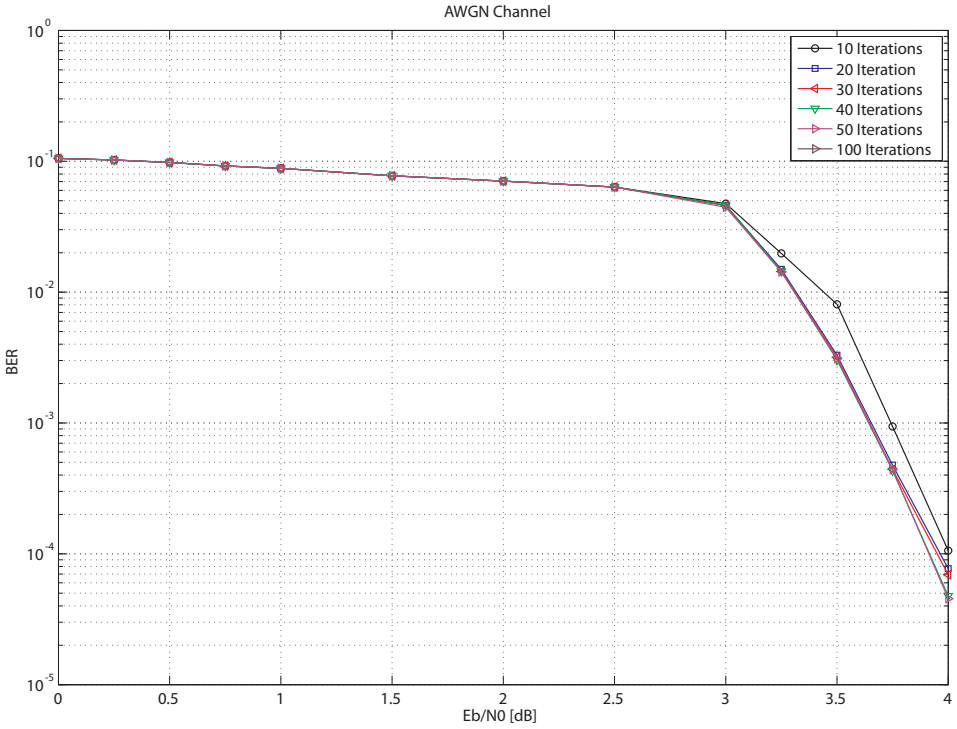

FIGURE 3.9: Performance of the girth-6 NB-QC-LDPC code using the $\mathbb{F}_{32}$ construction field with with random NB elements in an AWGN channel ($C2$).

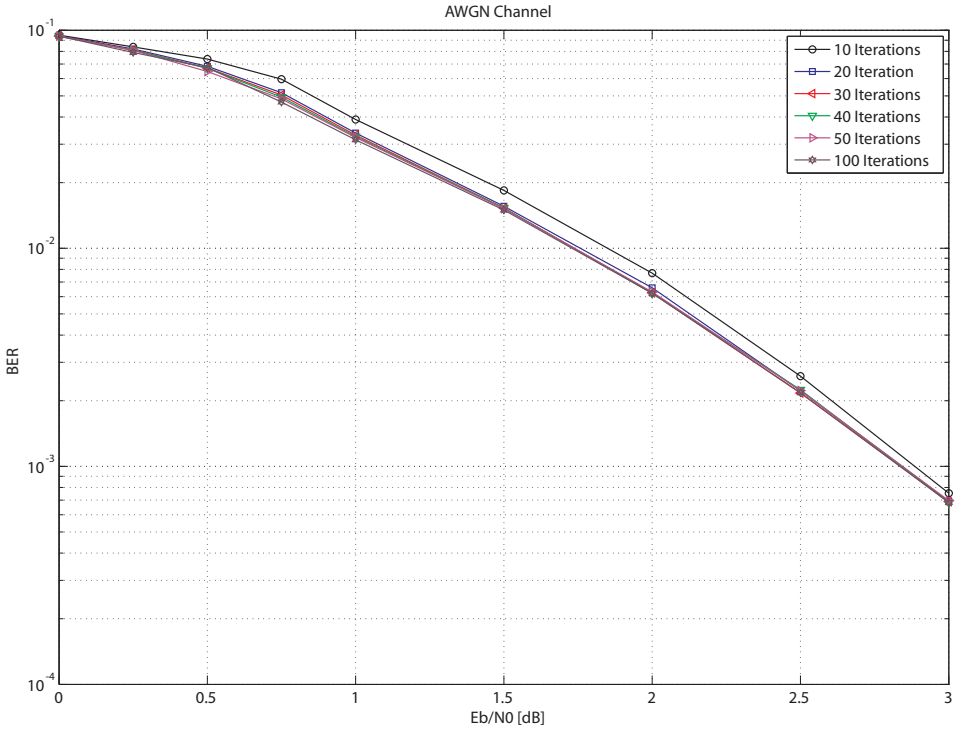

FIGURE 3.10: Performance of the girth-6 NB-QC-LDPC code using the $\mathbb{F}_{64}$ construction field with the unit weight circulant NB elements in an AWGN channel ($C3$).

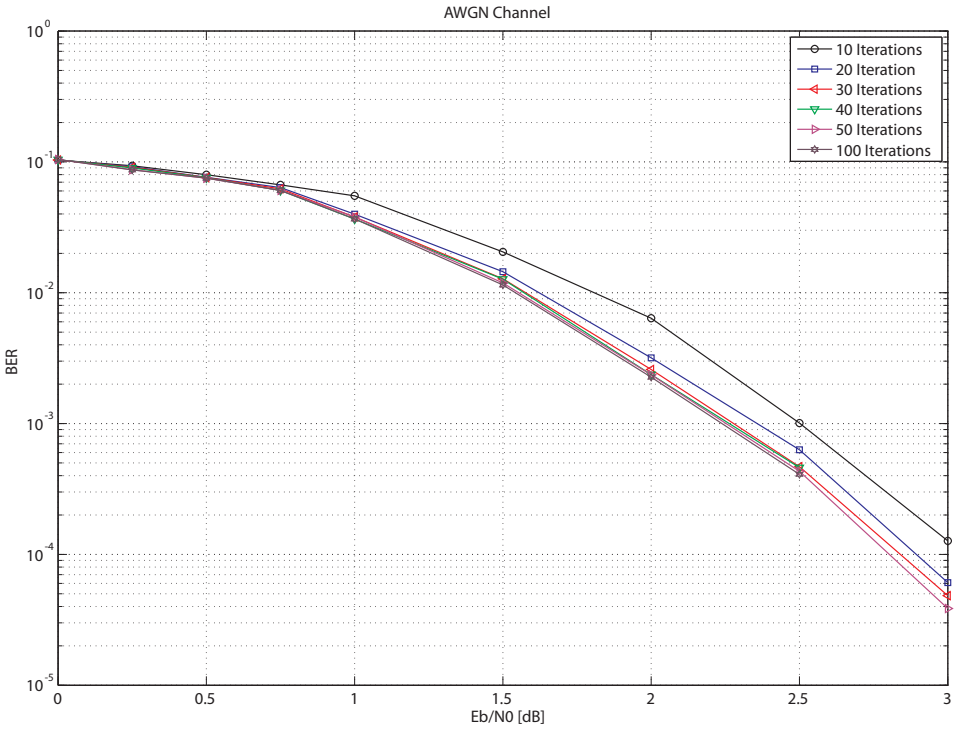

FIGURE 3.11: Performance of the girth-6 NB-QC-LDPC code using the $\mathbb{F}_{64}$ construction field with with random NB elements in an AWGN channel ($C4$).

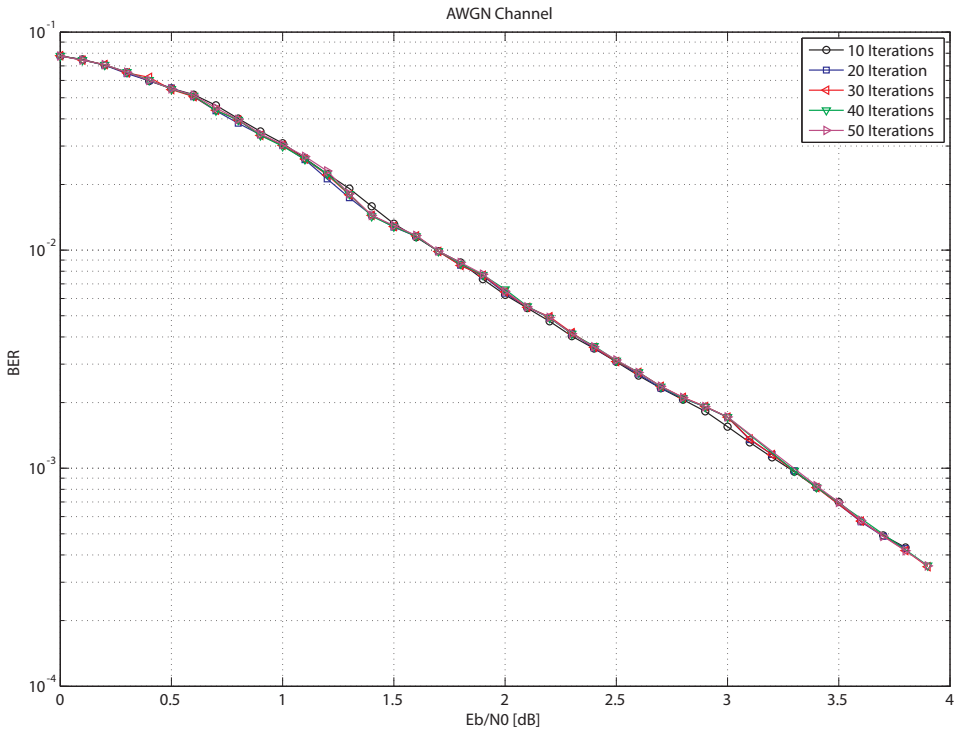

FIGURE 3.12: Performance of the girth-12 NB-QC-LDPC code using the unit weight circulant NB elements in an AWGN channel ($C5$).

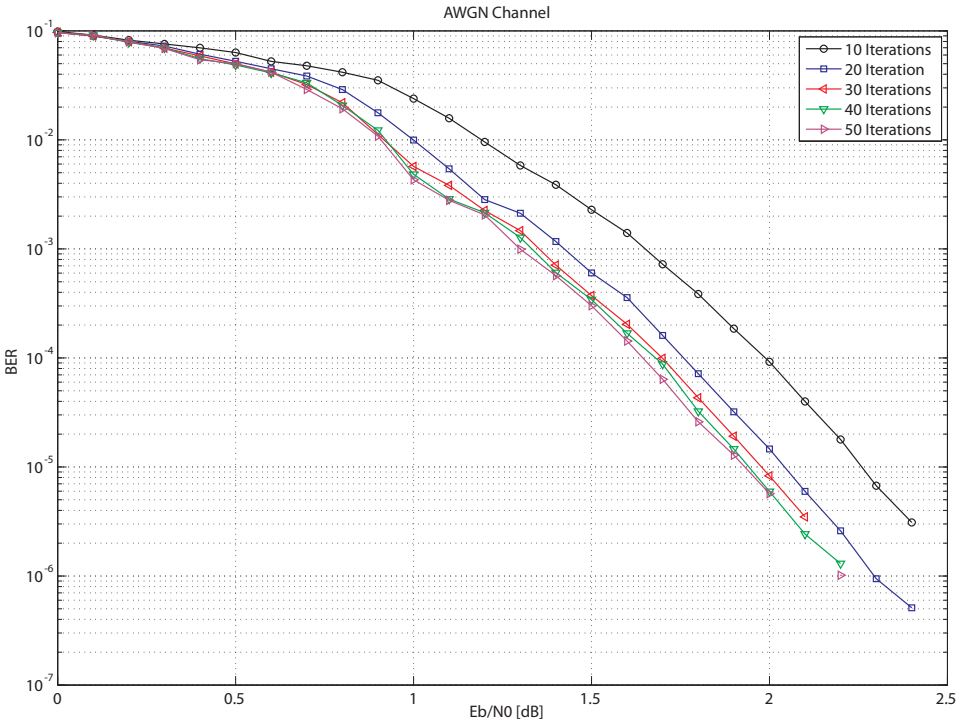

FIGURE 3.13: Performance of the girth-12 NB-QC-LDPC code with random NB elements in an AWGN channel ($C6$).

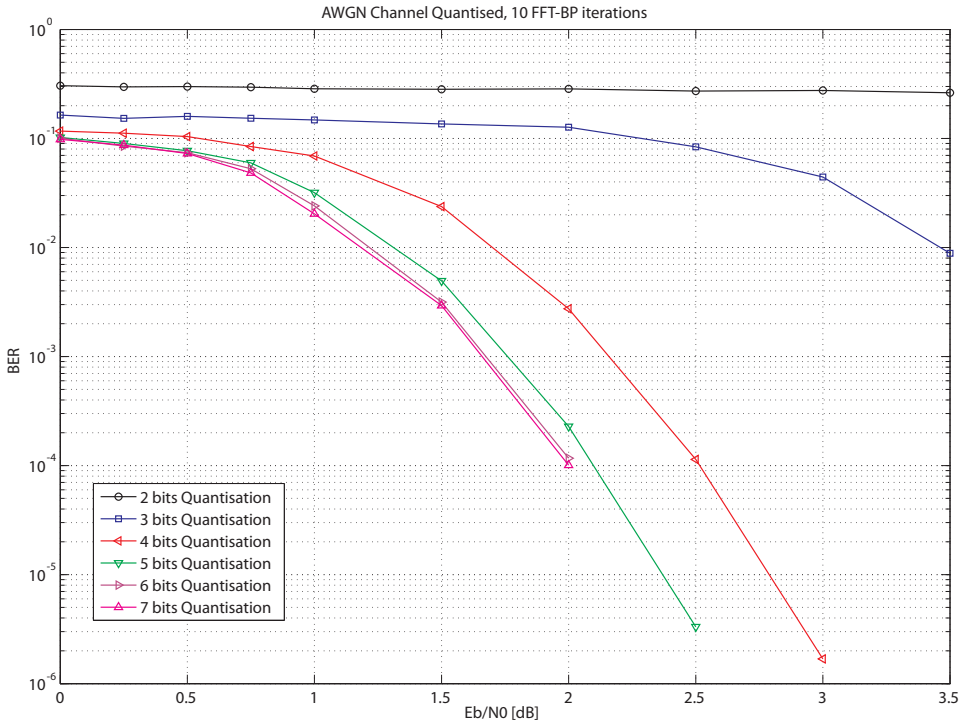

FIGURE 3.14: Performance of the girth-12 NB-QC-LDPC code with random NB elements in a quantised AWGN channel running 10 FFT-BP iterations ($C6$).

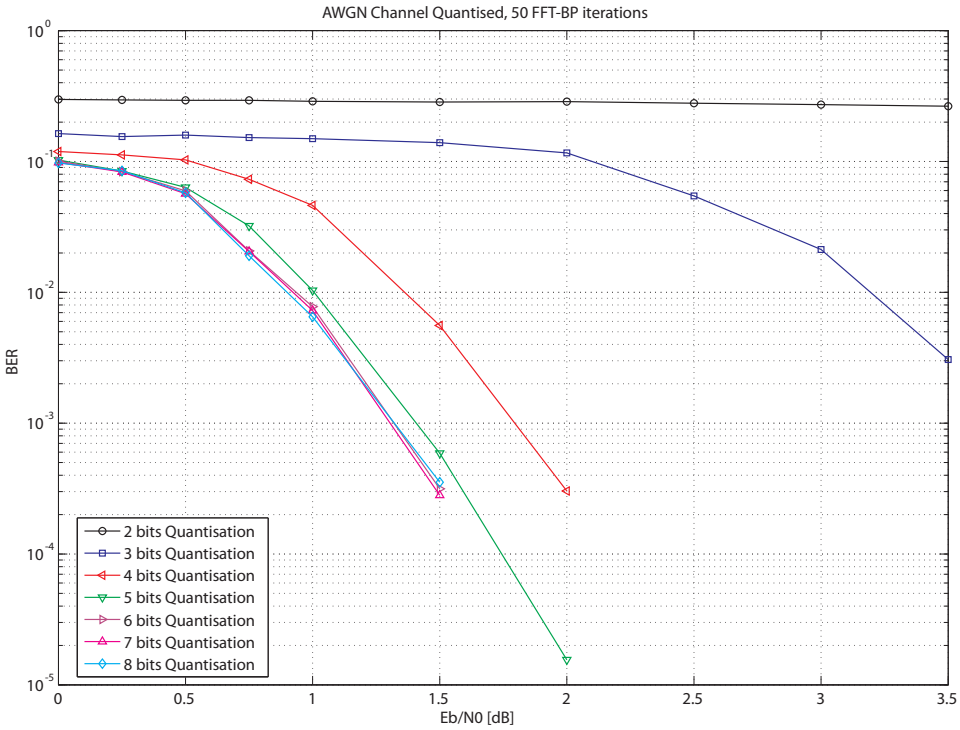

FIGURE 3.15: Performance of the girth-12 NB-QC-LDPC code with random NB elements in a quantised AWGN channel running 50 FFT-BP iterations ($C6$).

Chapter Four

A Triply Selective MIMO Channel Model

A MIMO communication system is a system which has multiple transmit and multiple receive antennas. Figure 4.1 shows a block diagram of a MIMO system. The incoming data signal is processed, placed on N_T antennas and transmitted. The transmitted signal passes through the MIMO channel and is received by the N_R receive antennas. The received signal is then processed to retrieve an estimate of the original data. In order to provide useful performance results

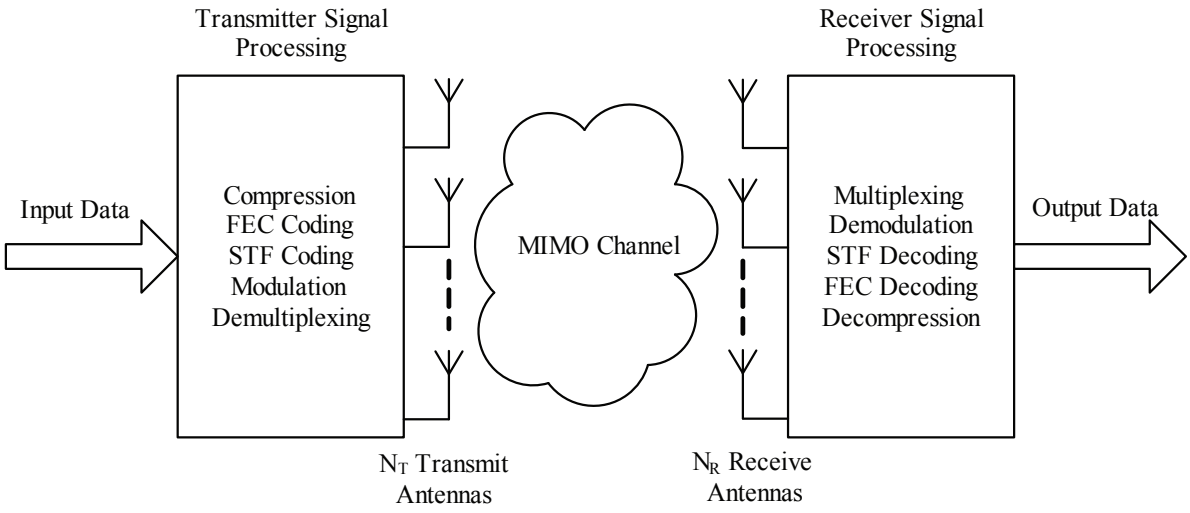

Figure 4.1: A block diagram of a MIMO mobile wireless system.

and comparisons of the signal processing techniques (FEC coding, Multi-Antenna coding etc.) discussed in this dissertation, a realistic and accurate mathematical model of a MIMO channel is required. This chapter starts by discussing the different physical effects in a mobile wireless

channel. A SISO mobile wireless channel model is then presented and extended to a MIMO mobile wireless channel capable of simulating fading conditions with selectivity over space, time and frequency.

4.1 SISO MOBILE WIRELESS CHANNEL

Several effects occur in mobile wireless communication channels. These effects [11] will be explained below.

4.1.1 Physical effects

4.1.1.1 Noise

One kind of signal corruption that exists in all forms of communication is noise. Noise exists when there are signals present in the system (specifically at the receiver) that were not sent by the transmitter. The sources of noise are diverse. Some common sources include radiation from outer space, amplifier noise, interference from other communication systems and thermal noise. The most commonly used statistical model for noise is AWGN. The noise is additive because it is added to the signal at the receiver (after the rest of the channel). The noise is white, because it is assumed that the noise exists at the same power at all frequencies and the noise is Gaussian because the amplitude of the noise signal is zero mean Gaussian distributed. The PDF of the noise is therefore given as:

$$p(\eta(t)) = \frac{1}{\sqrt{2\pi\sigma_n^2}} \exp\left(-\frac{\eta^2(t)}{2\sigma_n^2}\right) \tag{4.1}$$

where $\eta(t)$ is the noise signal and σ_n^2 is the noise power. In a system where complex modulation schemes are used, the noise is added to both the complex and the real axis. In this case the noise is usually modelled as being a circular symmetric complex Gaussian variable. The noise can thus be given as:

$$n(t) = n_R(t) + \jmath n_I(t) \tag{4.2}$$

where $n_R(t)$ and $n_I(t)$ are independent Gaussian processes, with the same variance, following the distribution given in Equation 4.1.

4.1.1.2 Multiple paths

Multipath refers to the condition where there are several paths between the transmitter and the receiver. These paths could include a LOS path as well as paths created when the signal bounces off scatterers in the environment. The result of multiple paths is that multiple copies of the signal arrive at the receiver. When the multiple paths are combined at the receiver, several effects occur. The first effect is ISI and occurs when the path delays are on the order of a symbol period. The second effect is fading. Fading occurs when the amplitude of the signal decreases. Fading can generally be divided into two types [11]:

- Large-scale fading - This type of fading is due to path loss as a function of distance and shadowing by large objects such as buildings and hills. This occurs as the mobile moves through distances on the order of the cell size and is typically frequency independent.
- Small-scale fading - This type of fading is due to constructive and destructive interference of multiple signal paths between the transmitter and receiver. This occurs as the mobile moves through distances on the order of the carrier wavelength and is frequency dependent.

Large-scale fading is applicable to issues such as cell-site planning and is not considered in this thesis. Small-scale fading is relevant to the design of the communication system and is considered in this dissertation. From this point onwards, small scale fading will be referred to as fading. As an example consider figure 4.2. In the figure there are two main paths. These paths will create ISI. Each path is made up of many paths with very similar path delays, which will result in independent fading on each main path.

4.1.1.3 Relative motion

When the transmitter and the receiver are in relative motion, the signal experiences a shift in frequency (Doppler shift). If the relative motion does not have a constant velocity, the Doppler shift will not be constant. The Doppler shift can be positive or negative, depending on the relative direction of movement between the transmitter and receiver. For example, consider a transmitter moving at a velocity of $v(t)$ $[m/s]$ relative to the receiver, transmitting on a carrier with a wavelength of λ $[m]$. The time-variant Doppler frequency shift, denoted by $f_d(t)$ $[Hz]$, is given by:

$$f_d(t) = \frac{v(t)}{\lambda}\cos(\theta(t)) \tag{4.3}$$

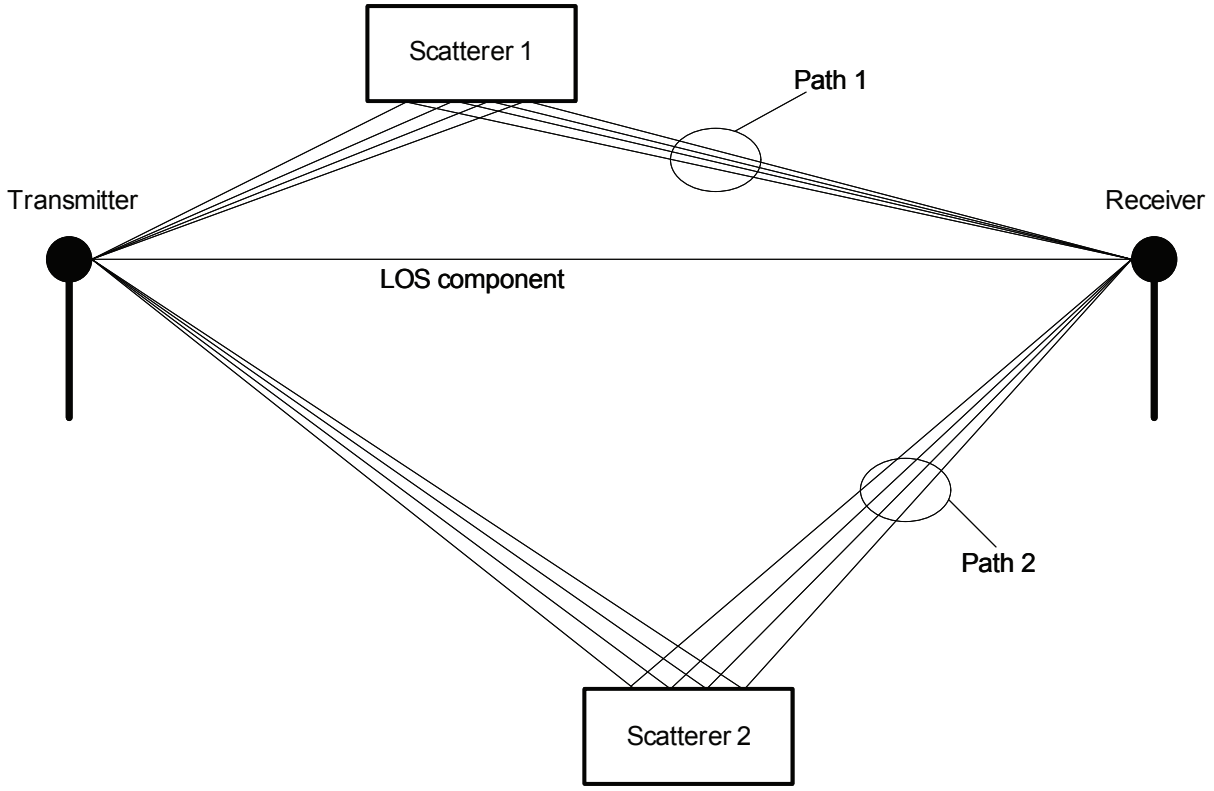

FIGURE 4.2: A simplistic representation of a wireless channel.

where $\theta(t)$ is the angle of arrival of the signal as a function of time. When the Doppler shift varies with time, the signal is 'smeared' in the frequency domain. This effect is known as the Doppler Spread. The Doppler spread is defined in Hertz and is equal to the width of the Doppler spectrum. In this dissertation, Doppler spread will be denoted by D_s.

4.1.2 Signal effects

The physical effects of the channel on the signal are mathematically modelled. Some definitions that are used will be given below [11].

- **Coherence Time**: The coherence time (T_c) of a channel is defined as the time over which the amplitude of the signal does not change significantly and is proportional to the inverse of the Doppler spread:

$$T_c \propto \frac{1}{D_s} \tag{4.4}$$

- **Delay Spread**: The delay spread (T_d) of the channel is defined as the difference in propagation time between the shortest and the longest significant paths. It is thus a measure of how much the signal is 'smeared' in the time domain.

- **Coherence Bandwidth**: The coherence bandwidth (W_c) of the channel is defined as the range of frequencies over which the amplitude of the frequency response of the channel does not change significantly. The coherence bandwidth is proportional to the inverse of the delay spread:

$$W_c \propto \frac{1}{T_d}. \tag{4.5}$$

Using the definitions provided above, the fading experienced by the channel may be classified into several categories. In terms of the frequency domain, the fading experienced by a signal will be either flat or frequency selective:

- **Flat Fading**: A signal is defined as experiencing flat fading when all the frequency components of the signal experiences the same fading. This occurs when the bandwidth occupied by the signal is less than the coherence bandwidth of the channel, or equivalently when the signal period is larger than the delay spread.

- **Frequency Selective Fading**: A signal experiences frequency selective fading when the different frequency components experience different fading. This occurs when the bandwidth occupied by the signal is larger than the coherence bandwidth or equivalently when the signal period is less than the delay spread of the channel. Signals in a frequency selective channel experience ISI and require an equaliser at the receiver (except in the case of OFDM)

In terms of the time domain, fading is categorised as slow fading or fast fading.

- **Fast Fading**: The signal is defined as experiencing fast fading when the channel parameters vary faster than the signal. This corresponds to a high Doppler spread and equivalently occurs when the symbol period is longer than the coherence time.

- **Slow Fading**: The signal is defined as experiencing slow fading when the channel parameters vary more slowly than the signal. This corresponds to a low Doppler spread and equivalently occurs when the symbol period is shorter than the coherence time.

4.1.3 Modelling of fading

Several models for the fading amplitude have been suggested in literature. These include: Raleigh fading, Rician fading, Weibull fading and Nakagami-n fading. Each model has its own advantages and disadvantages and is able to model different scenarios effectively. In this dissertation, the Rayleigh fading model is used. The Rayleigh fading model assumes that there is no LOS component path and is thus a worst case scenario. In the Rayleigh fading model, the random variable (RV) used to represent the fading is constructed by considering a complex Gaussian RV in phasor form. The amplitude of the fading RV is thus the amplitude of $X_1 + \jmath X_2$ where X_1 and X_2 are Gaussian random variables. The PDF of the amplitude of the fading RV is thus Rayleigh distributed:

$$p_A(r) = \begin{cases} \frac{r}{\sigma^2} \exp\left[\frac{-r^2}{2\sigma_r^2}\right] & for \ r \geq 0 \\ 0 & otherwise, \end{cases} \tag{4.6}$$

where σ_r^2 is the variance of both X_1 and X_2 and is set to $1/2$ to ensure that no power is added to the signal. The mean of the amplitude of the fading RV is:

$$E[r] = \sqrt{\frac{\pi}{2}}\sigma_r^2. \tag{4.7}$$

The variance of the amplitude of the fading RV can be given as:

$$\sigma_R^2 = \left(2 - \frac{\pi}{2}\right)\sigma_r^2. \tag{4.8}$$

The phase of the fading RV will be uniformly distributed and its PDF is thus:

$$p_\theta(p) = \frac{1}{2\pi} \quad for \quad -\pi \leq p \leq \pi. \tag{4.9}$$

4.1.4 Modelling of frequency selective multipath

A frequency selective multipath channel can be effectively modelled as a finite impulse response (FIR) filter where each tap in the filter corresponds to a signal path. In the case where there is relative motion between the transmitter and the receiver, the amplitude and phase of the taps will vary with time (fading). If the time varying channel impulse response (CIR) of the channel is given by $h(t, \tau)$ and the transmitted signal is given by $s(t)$, the received signal will be:

$$r(t) = s(t) \circledast h(t, \tau) \tag{4.10}$$

where $r(t)$ is the received signal and $\circledast$ represents the time domain convolution operator. In the case where the time delays of the different multipath components remain constant (which will be assumed for the remainder of this dissertation), the impulse response for a channel consisting of K paths may be given as:

$$h(t,\tau) = \sum_{i=0}^{K-1} \beta_i(t)\delta(\tau-\tau_i)\left[\exp(\jmath\phi_i(t)\right] \tag{4.11}$$

where $\beta_i(t)$, $\phi_i(t)$ and τ_i are the amplitude, phase and time delay of the i^{th} multipath respectively. A time invariant channel impulse response may be obtained by averaging $h(t,\tau)$ over time:

$$h(\tau) = \overline{h(t,\tau)} \tag{4.12}$$

$$= \sum_{i=0}^{K-1} h_i\delta(\tau-\tau_i)\left[\exp(\jmath\overline{\phi_i(t)})\right] \tag{4.13}$$

where h_i is the time average of $\beta_i(t)$ and $\overline{x(t)}$ is used to denote the time average of the variable $x(t)$. From the channel impulse response a channel power delay profile [PDP] ($P(t,\tau)$) may be defined:

$$P(t,\tau) = |h(t,\tau)|^2 \tag{4.14}$$

By averaging the PDP over time, a time invariant PDP can be given as:

$$P(\tau) = \sum_{i=1}^{K} P_i\delta(\tau-\tau_i) \tag{4.15}$$

where P_i is the power in the i^{th} tap and may be given by:

$$P_i = |h_i|^2. \tag{4.16}$$

In order to model realistic channels, measurements have been taken to produce different time invariant PDPs. In this dissertation, the 20-tap suburban alternative PDP [1] as well as a standard 2-tap channel was used for simulation purposes (see Appendix A for full PDP specifications). Using the time-invariant PDPs as a starting point, one can now add fading to the channel model. Let $\alpha_i(t)$ represent the complex fading process of the i^{th} multipath. The time varying amplitude and time varying phase of the i^{th} multipath may then be given as:

$$\beta_i(t) = |\alpha_i(t)|\overline{\beta_i(t)} \tag{4.17}$$

$$\phi_i(t) = \angle\alpha_i(t). \tag{4.18}$$

From this equation one may then produce any fading characteristic by defining the characteristic of $\alpha_i(t)$.

4.1.5 SISO channel model

Combining the fading, multipath and noise yields a system model, which may be expressed as:

$$r(t) \quad = \quad s(t) \circledast h(\tau) + n(t). \tag{4.19}$$

A diagram representing the model for a SISO mobile wireless channel as derived above can be seen in Figure 4.3. A delay D_i as given in the figure corresponds to the incremental time delay from one tap to the next, thus $D_i = \tau_i - \tau_{i-1}$. In this dissertation, only digital communication

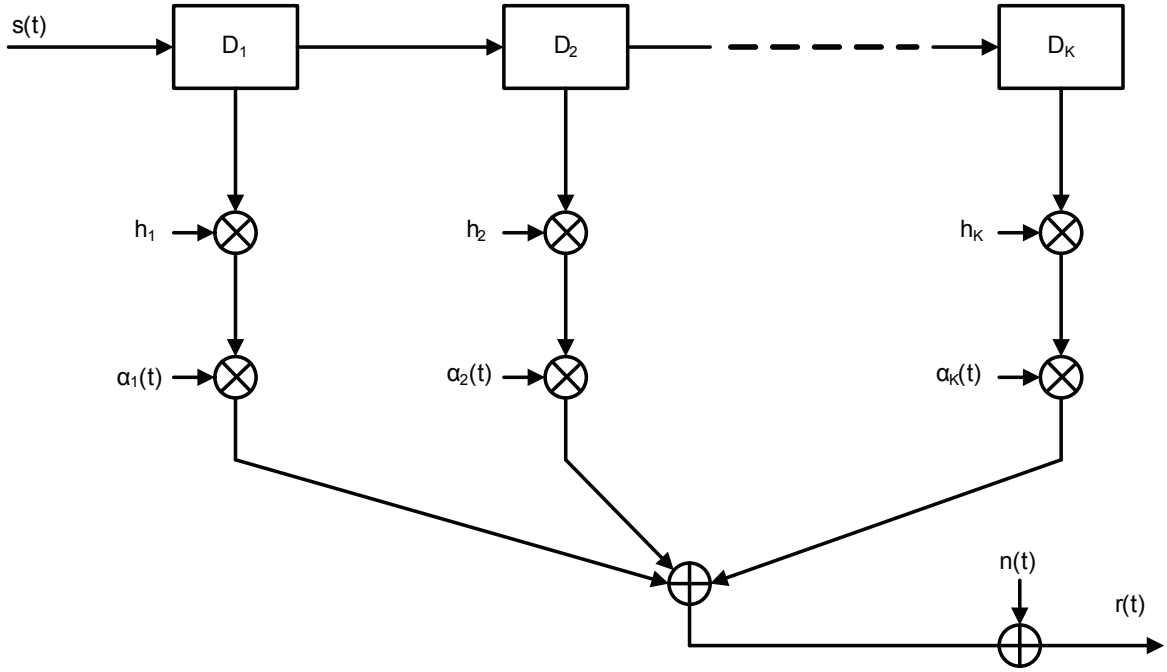

FIGURE 4.3: The SISO mobile wireless channel model

systems will be considered. It is thus useful to move the channel model to the discrete time domain. Let T_s be the sampling time. For the moment, also assume that all the time delays of the taps of the CIR $(\tau_1, \tau_2, \ldots, \tau_K)$ are integer multiples of the sampling time. The system model can now be expressed as:

$$y[t] \quad = \quad \sum_{i=0}^{L-1} h[i]x[t-i] + n[t] \tag{4.20}$$

where $L = T_d/T_s$ and represents the number of samples required to represent the CIR.

4.2 MIMO MOBILE WIRELESS CHANNEL

The SISO mobile channel model described above can now be extended to a MIMO channel model. Essentially, a SISO channel as described above, exists between each transmitter and receiver antenna pair. The received signal at an antenna will thus be the sum of the signals from each of the transmit antennas.

There are, however, some difficulties which arise in modelling realistic MIMO channels. Since the antenna pairs exist in the same physical space the channels experienced will be similar to some degree. The degree of similarity is called correlation. The correlation will depend on both the spacing of the antennas and on the richness of the scattering environment. An effective method to introduce correlation into the model is thus required. It was also shown in [40] that the transmit and receive filters can have an effect on the correlation and thus need to be taken into account.

This section will start by setting up the mathematical notation and framework for the MIMO channel model and then describe the method used to introduce correlation.

4.2.1 Model description and notation

Consider a MIMO system with N_T transmit antennas and N_R receive antennas. Let $p_T(t)$ and $p_R(t)$ be the normalised time-invariant impulse responses of the transmit and receive filters respectively. If $g_{ij}(t,\tau)$ represents the CIR of the sub-channel between the i^{th} transmit antenna and the j^{th} receive antenna, the combined CIR of the sub-channel can be given as:

$$h_{i,j}(t,\tau) \quad = \quad p_R(t) \circledast g_{i,j}(t,\tau) \circledast p_T(t). \tag{4.21}$$

Let $h_{i,j}[k,\tau]$ be the discrete time version of $h_{i,j}(t,\tau)$ sampled at T_s where k is the sampling indice. It should be noted that the impulse response of a transmit or receive filter is generally designed to be infinite to limit the bandwidth usage of the signal. As a result, $h_{ij}[k,\tau]$ will also have an infinite number of taps. However, the filters are usually designed such that the power in the time domain taps of the impulse response falls off rapidly. Thus, for the purpose of modelling, when the power in a tap falls below a defined threshold (such as 0.01%), the effect of the taps are negligible and can be ignored. Let the range of indices, required to yield the predefined amount of power, be from $-L_1$ to L_2 (where $L_1, L_2 \in \mathbb{Z}; L_1, L_2 \geq 0$). The total

number of values in $h_{i,j}[k,\tau]$ is thus equal to L, where $L = L_1 + L_2 + 1$. The combined CIR is thus a non-causal FIR filter. This does not imply that the physical channel is non-causal. The physical channel $g_{ij}(t,\tau)$ is always causal; it is the introduction of the transmit and receive filters that causes $h_{i,j}[k,\tau]$ to be non-causal. For this dissertation it is assumed that the sub-channels between all pairs of transmit and receive antennas have the same PDP but with independent fading. The time averaged, combined CIR ($\overline{h_{i,j}[k,\tau]}$), is thus the same for all i and j.

In a realistic channel it is seldom the case that the taps are spaced at intervals that are integer multiples of the symbol period (T_{sym}). Thus in order to represent the CIR accurately, the sampling period T_s is generally higher than the symbol period and given by $T_s = T_{sym}/\gamma$, where γ is an integer. Since the CIR has been oversampled, the transmit signal must also be oversampled. Since the transmit filter has been included in the CIR, this can simply be done by inserting $\gamma - 1$ zeros between each symbol. Thus, if $s_i[t]$ is the sequence of symbols to be transmitted from the i^{th} transmit antenna and k is the sampling indice, the oversampled transmit sequence $x_i[k]$ can be given by:

$$x_i[k] = \begin{cases} s_i\left[\frac{k}{\gamma}\right] & if\ \frac{k}{\gamma} \in \mathbb{Z} \\ 0, & otherwise. \end{cases} \tag{4.22}$$

The signal received at antenna j can now be given as:

$$y_j[k] = \sum_{i=1}^{N_T} \sum_{l=-L_1}^{L_2} h_{i,j}[k,l]x_i[k-l] + z_j[k] \tag{4.23}$$

where $z_j[k] = n_j[k] \otimes p_R[k]$ is the noise at the j^{th} receive antenna, that has been passed through the receive filter. This process can now efficiently moved to a matrix representation. Let $\mathbf{y}[k]$ and $\mathbf{x}[k]$ and $\mathbf{z}[k]$ respectively be the vector of signals received, the vector of symbols transmitted and the vector of noise samples at sampling instance k. The channel model can now be expressed as:

$$\mathbf{y}[k] = \sum_{i=-L_1}^{L_2} \mathbf{H}_i[k]\mathbf{x}[k-i] + \mathbf{z}[k], \tag{4.24}$$

where the channel matrix $\mathbf{H}_i[k]$ is structured as follows:

$$\mathbf{H}_i[k] = \begin{pmatrix} h_{1,1}[k,i] & \dots & h_{1,N_T}[k,i] \\ \vdots & \ddots & \vdots \\ h_{N_R,1}[k,i] & \dots & h_{N_R,N_T}[k,i] \end{pmatrix}. \tag{4.25}$$

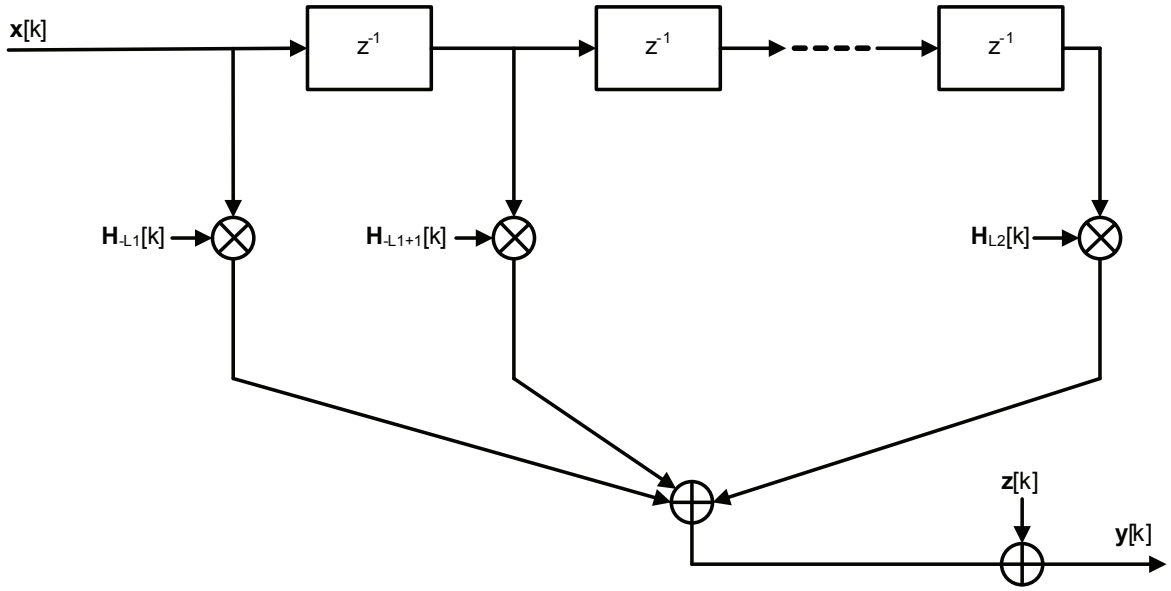

FIGURE 4.4: The discrete time MIMO mobile wireless channel model

A diagram of the model can be seen in Figure 4.4. If one considers $\mathbf{H}_i[k]$ for all i it is evident that the channel requires $N_T N_R L$ channel co-efficients at each sampling instance k. These co-efficients may be represented as a single vector by placing the channel co-efficients of the different subchannels in the same vector, $\mathbf{h_v}[k]$:

$$\mathbf{h_v}[k] = [\mathbf{h}^{\mathcal{T}}_{\mathbf{1,1}}[k], \ldots, \mathbf{h}^{\mathcal{T}}_{\mathbf{1,N_R}}[k], \ldots, \mathbf{h}^{\mathcal{T}}_{\mathbf{N_T,1}}[k], \cdots, \mathbf{h}^{\mathcal{T}}_{\mathbf{N_T,N_R}}[k]]^{\mathcal{T}} \tag{4.26}$$

The modelling problem has now been reduced to generating $\mathbf{h_v}[k]$ and the noise vector $\mathbf{z}[k]$ in such a way as to produce a system with the correct stochastic properties.

4.2.1.1 Introducing Antenna Correlation

Antenna correlation can be modelled in different ways. This dissertation uses the Kronecker model. The Kronecker model works on the assumption that the antenna correlation between the transmitted and received signals can be split into the correlation at the transmit antennas and the correlation at the receive antennas separately. Let $\mathbf{\Psi}$ be the total correlation, $\mathbf{\Psi_T}$ be the correlation matrix for the transmit antennas and $\mathbf{\Psi_R}$ be the correlation matrix for the receive antennas. Thus,

$$\mathbf{\Psi} = \mathbf{\Psi}_R \mathbf{\Psi}_T \tag{4.27}$$

where $\mathbf{\Psi}_R$ and $\mathbf{\Psi}_T$ can be given as:

$$\mathbf{\Psi_T} = \begin{pmatrix} \rho_T^{1,1} & \cdots & \rho_T^{1,N_T} \\ \vdots & \ddots & \vdots \\ \rho_T^{N_T,1} & \cdots & \rho_T^{N_T,N_T} \end{pmatrix} \tag{4.28}$$

$$\mathbf{\Psi_R} = \begin{pmatrix} \rho_R^{1,1} & \cdots & \rho_R^{1,N_R} \\ \vdots & \ddots & \vdots \\ \rho_R^{N_R,1} & \cdots & \rho_R^{N_R,N_R} \end{pmatrix} \tag{4.29}$$

where $\rho_T^{i,j}$ represents the correlation between the i^{th} and the j^{th} transmit antennas while $\rho_R^{i,j}$ represents the correlation between the i^{th} and the j^{th} receive antennas. Different models may now be used to determine values for $\rho_R^{i,j}$ and $\rho_T^{i,j}$. In this dissertation, the exponential process is used [41]:

$$\rho_R^{m,p} = r^{|m-p|} \tag{4.30}$$

$$\rho_T^{n,q} = r^{|n-q|}, \tag{4.31}$$

where $|r| \leq 1$. Scaling r will thus add more or less correlation.

4.2.2 Statistical properties of the channel co-efficients

4.2.2.1 Noise

The original noise vector $\mathbf{n}[k]$ is white zero-mean Gaussian distributed. The filtered noise vector $\mathbf{z}[k]$ is also zero mean Gaussian distributed with an auto-covariance matrix that can be given as:

$$\mathbf{R_{zz}}(k_1 - k_2) = E\left[\mathbf{z}(k_1) \cdot \mathbf{z}^{\mathcal{H}}(k_2)\right] \tag{4.32}$$

$$= N_0 \cdot R_{p_R p_R}\left[(k_1 - k_2)T_s\right] \mathbf{I}_{N_R} \tag{4.33}$$

where $R_{p_R p_R}(\cdot)$ is the autocorrelation of the receive filter. If $R_{p_R p_R}(\cdot)$ satisfies the following condition:

$$R_{p_R p_T}(kT_s) = 0, \quad k \neq 0 \tag{4.34}$$

then $z_i[k]$ is white from sample to sample and from antenna to antenna with a variance of N_0.

4.2.2.2 Channel fading parameters

In order to generate a channel fading co-efficient vector with the correct stochastic properties, the stochastic properties must first be defined. Since $g_i(t,\tau)$ is zero mean complex Gaussian distributed, $h_i(t,\tau)$ will also be zero mean complex Gaussian distributed. As as result $\mathbf{h}_v[k]$ will be zero mean Gaussian distributed. To find the correlation properties of $\mathbf{h}_v[k]$ one studies its covariance matrix. The covariance of $\mathbf{h}_v[k]$ can be given by [40]:

$$\mathbf{C}_h[k_1 - k_2] = E[\mathbf{h}_v[k_1].\mathbf{h}_v^{\mathcal{H}}[k_2]] \tag{4.35}$$

$$= (\mathbf{\Psi}_R \otimes \mathbf{\Psi}_T \otimes \mathbf{C}_{ISI}) J_0(2\pi f_d(k_1 - k_2)T_s) \tag{4.36}$$

where $\otimes$ denotes the Kronecker product, $\mathbf{\Psi}_T$ and $\mathbf{\Psi}_R$ are the antenna correlation matrices as described above and $\mathbf{C}_{ISI}$ is the filter tap ISI covariance matrix. f_d is the Doppler frequency and $J(\cdot)$ is the Bessel function of the first kind of order zero. $\mathbf{C}_{ISI}$ is defined as [40]:

$$\mathbf{C}_{ISI} = \begin{pmatrix} c[-L_1,-L_1] & \dots & c[-L_1,L_2] \\ \vdots & \ddots & \vdots \\ c[L_2,-L_1] & \dots & c[L_2,L_2] \end{pmatrix} \tag{4.37}$$

where

$$c[l_1,l_2] = \sum_{i=0}^{K} P_i \overline{R}_{P_T,P_R}[l_1 T s - \tau_1] \overline{R}^*_{P_T,P_R}[l2T_s - \tau_i] \tag{4.38}$$

and $\overline{R}_{P_T,P_R}[\cdot]$ is the convolution function of the transmit and receive filters. P_i is the power in the i^{th} tap of the PDP (see Equation 4.15) and K is the number of original taps in the PDP. $\mathbf{C}_h[k_1 - k_2]$ thus takes into account the correlation due to antennas as well as the the correlation due to the transmit and receive filters and the channel taps. If one wishes a vector of values to have correlation properties given by a matrix $\mathbf{X}$, then the vector must be multiplied by a matrix $\mathbf{X}^{1/2}$. To create the correlation matrix for any sampling instance k, one sets $k_1 = k_2$. Thus $\mathbf{h}_v[k]$ can be created by multiplying a zero mean complex Gaussian vector with the square root of the correlation matrix ($\mathbf{C}_h[0]$) as follows:

$$\mathbf{h}_v[k] = \mathbf{C}_h^{1/2}(0) \cdot \phi[k] \tag{4.39}$$

$$= \left(\mathbf{\Psi}_R^{1/2} \otimes \mathbf{\Psi}_T^{1/2} \otimes \mathbf{C}_{ISI}^{1/2}\right) \cdot \phi[k], \tag{4.40}$$

where $\phi[k]$ is an $N_T N_R L \times 1$ vector whose elements are uncorrelated Raleigh flat fading and

$$E\left[\phi[k1] \cdot \phi^{\mathcal{H}}[k2]\right] = J_0[2\pi f_d(k_1 - k_2)T_s] \cdot \mathbf{I}_{N_T N_R L}. \tag{4.41}$$

A useful property of this model is that it does not require the channel tap delays in the PDP to be integer multiples of the sampling frequency. Through Equation 4.38 the C_{ISI} matrix will take into account the effects of fractionally spaced taps without the need to increase the sampling frequency.

4.2.3 Block fading channel conditions and OFDM

Most of the work in this dissertation uses OFDM. For simulation purposes and to analyse the multi-antenna coding techniques properly, assumptions are made on the temporal characteristics of the channel. Some codes, such as the Alamouti ST code, assume quasi-static fading conditions in which the channel fading parameters ($\phi[k]$ in equation 4.39) remain constant for the entire ST codeword (2 OFDM symbols in the case of Alamouti). Most of the SF and STF codes assume block fading channel conditions. Block fading assumes that the channel fading parameters remain constant for the duration of one OFDM symbol (T_{sym}).

Since OFDM allows the placement of information symbols on individual subcarriers, it is useful to rewrite equation 4.24 in the frequency domain. Consider the combined CIR between the i^{th} transmit antenna and the j^{th} receive antenna for the k^{th} transmitted OFDM symbol, $\mathbf{h}_{ij}[k]$. If one defines N_f to be the number of discrete frequencies, the frequency domain representation of this channel ($\mathbf{h}^f_{ij}[k]$) can be given by:

$$\mathbf{h}^f_{ij}[k] = \mathbf{F}\mathbf{h}_{ij}[k] \tag{4.42}$$

where $\mathbf{F} = [\mathbf{f}^{\tau_0}, \mathbf{f}^{\tau_1}, \ldots, \mathbf{f}^{\tau_{L-1}}]$, $\mathbf{f} = [1, \zeta, \ldots, \zeta^{N_f-1}]^{\mathcal{T}}$ and $\zeta = \exp\left(-\jmath\frac{2\pi}{T_{sym}}\right)$. $h^f_{ijp}[k]$ will thus contain the channel response from the i^{th} transmit antenna to the j^{th} receive antenna along the p^{th} subcarrier. In this dissertation it is assumed that the OFDM modulation system has perfect timing and synchronisation. Thus, the symbol received at the j^{th} receive antenna on the p^{th} subcarrier for the k^{th} OFDM symbol can be given as:

$$y_{j,p}[k] = \sum_{i=1}^{N_T} h^f_{ijp}[k]c_{i,p}[k] + w_{j,p}[k] \tag{4.43}$$

where $c_{i,p}[k]$ is the codeword transmitted from the i^{th} transmit antenna on the p^{th} subcarrier and $w_{j,p}[k]$ is the noise at the j^{th} receive antenna for the p^{th} subcarrier. In matrix form, over all subcarriers, this can be written as:

$$\mathbf{y}_j[k] = \sum_{i=1}^{N_T} \mathbf{h}^f_{ij}[k]diag(\mathbf{c}_i[k]) + \mathbf{w}_j[k]. \tag{4.44}$$

More generic descriptions will be given in the section on multi-antenna codes when they are required.

4.2.4 Generating random variables

4.2.4.1 Generating AWGN random variables

The channel requires the generation of a vector of independent zero mean Gaussian RVs with a specific variance. The process followed to produce these is to create a zero mean Gaussian RV with variance equal to 1 and then to scale the RV with the correct scaling factor. Many algorithms exist for the creation of zero mean Gaussian RVs. These algorithms normally start by creating a uniformly distributed RV and then transforming the variable into a Gaussian RV using a transformation algorithm. Examples of algorithms that create uniformly distributed RVs are the Mersenne Twister and the Wichman-Hill algorithms. Examples of transformation algorithms are the Bray-Marsaglia and the Box-Muller sine-cosine algorithm. This dissertation used the Wichman-Hill algorithm in conjunction with the Bray-Marsaglia algorithm [42].

The output of these algorithms may be directly scaled by the required noise variance if SNR versus BER simulations are required. However, most coded systems require E_b/N_0 versus BER simulations. To obtain noise which results in the correct E_b/N_0 in a digital simulation, the scaling factor (n_s) is calculated as follows [43]:

$$n_s = \sqrt{\frac{\sigma_s^2 f_s}{10^{\frac{E_b/N_0}{10}} 2 f_b}} \tag{4.45}$$

where σ_s^2 is the variance of the transmitted signal, $f_s = 1/T_s$ is the sampling frequency of the system and f_b is the input frequency of the information bits.

4.2.4.2 Generating Rayleigh flat fading co-efficients

There are many algorithms available to create Raleigh fading variables with the required time domain characteristics (Doppler spread). For example, Clarke's model starts by creating independent complex Gaussian variables using methods as described in the previous section and then filters these values with a filter shaped like the required Doppler spectrum. In this dissertation however, the modified Jakes model is used [44]. This model creates fades by summing over many sinusoids. The fading process ($\alpha(t)$) is given by:

$$\alpha(t) = \alpha_r(t) + \jmath\alpha_i(t) \tag{4.46}$$

where

$$\alpha_r(t) = \sqrt{\frac{2}{N_S}} \sum_{n=1}^{N_S} \cos(2\pi f_d \cos\beta_n + \zeta_n) \tag{4.47}$$

$$\alpha_i(t) = \sqrt{\frac{2}{N_S}} \sum_{n=1}^{N_S} \cos(2\pi f_d \sin\beta_n + \eta_n) \tag{4.48}$$

and

$$\beta_n = \frac{2\pi n - \pi + \theta}{4N_S}. \tag{4.49}$$

N_S represents the number of sinusoids added and should be larger than eight. η_n, θ and ζ_n are independent uniformly distributed random variables in the interval $[-\pi, \pi]$. For the purpose of the simulator, L of these processes are independently created. These processes are then matched to $\phi[k]$ (from equation 4.39). Thus $\phi_j[k] = \alpha_j(kT_s)$ where $\alpha_j(kT_s)$ is the sampled version of the j^{th} fading process.

Chapter Five

Multi-Antenna Coding Techniques

MIMO systems theoretically offer several performance improvements over SISO systems. Specifically, the capacity of a MIMO system grows linearly with the minimum of the number of transmit and receive antennas. This increased capacity allows for an increase in the throughput of a MIMO system over a SISO system. A second improvement is increased reliability in a fading environment. This is due to the diversity obtained from the independent paths between the transmitter and receiver antennas. In order to extract these gains, multi-antenna codes must be developed. These codes can be characterised by the diversity order and the transmission rate achieved. Codes that are able to extract all the available diversity and communicate at a high rate will inevitably have a high decoding complexity. As a result, the complexity of a code is another important parameter to consider. This chapter presents the design of ST, SF and STF codes with varying rates, achieved diversity and decoding complexity. The effect of quantisation at the receiver on the achievable diversity of full rate codes using linear precoding is also derived mathematically and verified using simulations.

5.1 ST codes

Consider a MIMO system with N_T transmit and N_R receive antennas. Let $\mathbf{x} = [x_1, x_2, \ldots, x_p]$ denote a block of information symbols, of size p, taken from a complex modulation alphabet such as bipolar phase shift keying (BPSK) or quadrature amplitude modulation (QAM). This

block of information symbols is encoded to produce a code $\mathbf{C}$

$$\mathbf{C} = \begin{pmatrix} c_{1,1} & \dots & c_{1,N_B} \\ c_{2,1} & \dots & c_{2,N_B} \\ \vdots & \ddots & \vdots \\ c_{N_T,1} & \dots & c_{N_T,N_B} \end{pmatrix}, \tag{5.1}$$

where N_B represents the number of symbol periods over which the code is defined. The code entry $c_{i,j}$ represents the complex symbol to be transmitted from antenna i at time interval j. The rate of the code is defined as the number of information symbols per channel use and may be calculated as:

$$r = \frac{p}{N_B}. \tag{5.2}$$

The maximum achievable rate is equal to the number of transmit antennas. Deriving exact performance curves for an ST code is non-trivial. A useful measure which is used instead is a pairwise error probability (PEP), which is the probability that one codeword ($\mathbf{C}$) will be mistaken for another ($\mathbf{C}'$). The PEP can be upper bounded by [45]

$$P(\mathbf{C} \to \mathbf{C}') \leq \left(\prod_{i=1}^{r} \lambda_i \right)^{-N_R} \left(\frac{\rho}{4N_T} \right)^{rN_r}, \tag{5.3}$$

where ρ denotes the SNR at each receive antenna, r denotes the rank of the matrix $\mathbf{D}$ given by $\mathbf{D} = \mathbf{C} - \mathbf{C}'$ and $\lambda_1, \lambda_2, \dots, \lambda_r$ are the non-zero eigenvalues of the matrix given by $(\mathbf{C} - \mathbf{C}')^H(\mathbf{C} - \mathbf{C}')$. Thus, this gives an upper bound on the symbol error rate. From this PEP, two design criteria for ST codes were developed:

- **Rank Criterion or Diversity Criterion:** The minimum rank of the matrix $\mathbf{D}$ over all pairs of distinct codewords must be as large as possible.
- **Product Criterion:** The minimum of the products of the eigenvalues $(\lambda_1, \lambda_2, \dots, \lambda_r)$ over all pairs of distinct codewords must be as large as possible.

The rank criterion will affect the achievable diversity of the code and thus the slope of the BER curve, while the product criterion will affect the coding gain achieved by the code and thus the horizontal shift of the BER curve. Important parameters to consider, other than these criteria, include the code rate and the decoding complexity. The first ST block codes to be developed were orthogonal codes.

5.1.1 Orthogonal codes

Orthogonal codes are defined as codes where the code matrix meets the following requirement:

$$\mathbf{C}^T\mathbf{C} = \sum_{i=1}^{n} |s_i|^2\mathbf{I}, \tag{5.4}$$

where s_i is a complex modulation symbol and n is the number of symbols in a codeword. When this criteria is met, the channel decouples into n separate channels. The SNR of the i^{th} subchannel is:

$$SNR = \frac{\rho^2}{N_T} \cdot \frac{\|\mathbf{H}\|^2}{\sigma^2} \cdot E\left[|s_i|^2\right]. \tag{5.5}$$

This decoupling allows the symbols in the code to be maximum likelihood (ML) decoded separately using maximum ratio combining (MRC), thus making the complexity of the decoding process grow linearly with code size. The original orthogonal code is known as the Alamouti code [12] and was designed for $N_T = 2$:

$$\mathbf{C} = \begin{pmatrix} s_1 & -s_2^* \\ s_2 & s_1^* \end{pmatrix}. \tag{5.6}$$

The Alamouti code is able to achieve a diversity of 2 at a rate of 1. After the seminal Alamouti code, some work was done in [13][46] to develop generalised orthogonal codes for any number of antennas. For the cases where $N_T = 3, 4$, codes were found that could achieve a rate of $3/4$. However, for arbitrary complex modulation constellations and $N_T > 4$, the codes developed in [13] were limited to a rate of $1/2$.

5.1.2 Quasi-orthogonal codes

As a result of the limited rate of orthogonal ST codes, the requirement of orthogonality was relaxed to create quasi-orthogonal ST block codes (QO-ST) [47]. QO-ST block codes were developed that could achieve a rate of 1; however, these codes could not achieve full diversity. Using the concept of constellation rotation described in the next section, QO-ST block codes were developed that could achieve full diversity [48]. However, these codes were not able to achieve a rate of 1 for more than 4 antennas. The trade-off for the higher rate compared to the orthogonal codes is that the symbols could no longer be separately decoded. They are ML decoded in pairs (or even larger groups). This leads to an increase in decoding complexity.

5.1.3 Constellation rotation and linear precoding

In order to achieve full diversity and full rate, simple orthogonal type codes are not sufficient. Advanced codes make use of the concepts of constellation rotation (CR), LP and layering [49–53].

5.1.3.1 Constellation rotation

In CR, the complex modulation constellation is 'rotated' to yield a new constellation. The purpose of the rotation is to make a transmitted symbol robust against fading. In order for CR to work, the symbol has to fade independently along the real and imaginary axes. Figure 5.1 shows a symbolic representation of a standard quadrature phase shift keying (QPSK) constellation. If the symbol experiences fading along either the real or the imaginary axis,

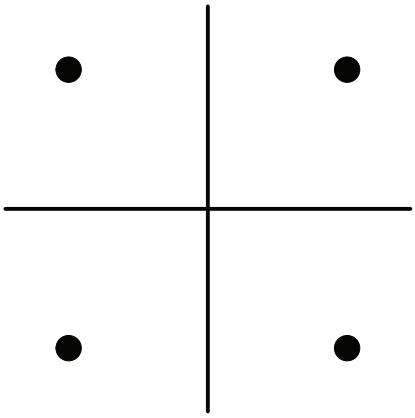

FIGURE 5.1: Symbolic QPSK constellation.

the symbols fade toward one another and one bit of information is lost. This can be seen in Figure 5.2. Consider now the rotated constellation in figure 5.4. If fading is experienced along a dimension, the symbols do not move towards each other, as can be seen in Figure 5.4. As a result, all the information is maintained. In order for the information to be lost, the signal has to experience fading along both dimensions. CR is thus able to achieve diversity of order 2, as long as the fading along the two dimensions is independent. In practice, the fading can be made to be independent by interleaving the complex and real components of different symbols.

This diversity can be explained in terms of information theory. In the case of normal constellations each dimension carries n_b bits of information (for QPSK, $n_b = 1$). After CR

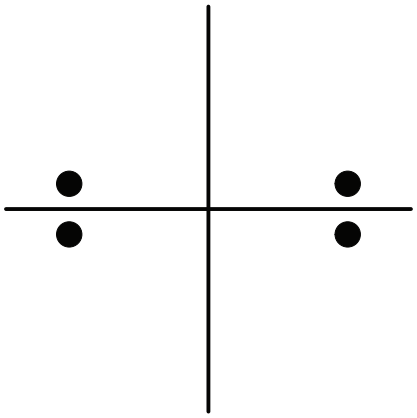

FIGURE 5.2: Symbolic Faded QPSK constellation.

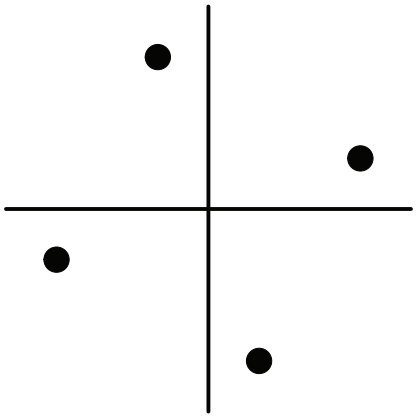

FIGURE 5.3: Symbolic Rotated QPSK constellation.

each dimension carries $2n_b$ bits of information. Since only $2n_b$ of information exists in each symbol, the two dimensions are carrying the same information. L^{th}-Order diversity is defined as occurring when the information is transmitted across L independent fading paths. CR thus achieves a diversity order of 2. The concept of entangling the information in several dimensions together, to yield diversity can be generalised to yield LP.

5.1.3.2 Linear precoding

In CR the information bits from the 2 phase dimensions are entangled. In LP the information from any arbitrary set of dimensions may be entangled. Consider the vector $\mathbf{x}$, which consists of n complex data symbols drawn from some modulation alphabet $\mathcal{A}$. $\mathbf{x}$ can be viewed as a

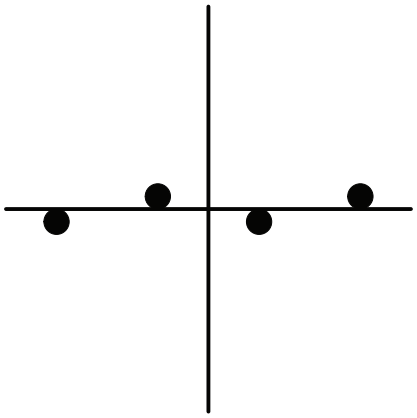

FIGURE 5.4: Symbolic Faded rotated QPSK constellation.

point in an n-dimensional vector space where the n complex mathematical dimensions could correspond to any n orthogonal complex physical dimensions such as different points in time, frequency or space. The rotation matrix, $\mathbf{\Theta_n}$, is defined as an $n \times n$ complex matrix. The design criteria may be given as follows:

1. The rank of Θ_n (R_Θ) must be equal to n.

2. The LP process must not add power, thus:

$$tr(\mathbf{\Theta}\mathbf{\Theta}^{\mathcal{H}}) \quad = \quad n \tag{5.7}$$

   where the notation $tr(\mathbf{M})$ refers to the trace of the matrix $\mathbf{M}$.

3. Maximum diversity gain criterion:

$$|\theta_k^{\mathcal{T}}(\mathbf{x} - \mathbf{x}')| \quad \neq \quad 0, \quad \forall\, \mathbf{x} \neq \mathbf{x}' \in \mathcal{A} \tag{5.8}$$

   where $\theta_k^{\mathcal{T}}$ denotes the k^{th} row of $\mathbf{\Theta}_n$.

4. Maximum coding gain criterion: Find the matrix $\mathbf{\Theta}_n$ which maximises the minimum Euclidean distance, $d_{\mathbf{\Theta}_n}$, between the closest pair of encoded vectors.

$$d_{\Theta_n} \quad = \quad \min_{\forall \mathbf{x} \neq \mathbf{x}'} \prod_{k=1}^{n} |\theta_k^{\mathcal{T}}(\mathbf{x} - \mathbf{x}')|^2 \tag{5.9}$$

If criterion 2 is met, then the code will be able to achieve full diversity. Criterion 3 allows for maximum separation of the different coded vectors and thus maximises the coding gain. In

general, if a code has maximised its coding gain, then the code will also meet criterion 2. If n is a power of two then the precoding matrix is unitary. Unitary matrices have the advantage that the distance properties of the original modulation constellation are maintained. This means that the performance in an AWGN channel is invariant from an uncoded system. In this dissertation only unitary matrices are used. A construction method for a unitary matrix which meets the above criteria can be given by [51] :

$$\mathbf{\Theta}_n = \mathcal{F}_n diag(\theta^0, \theta^1, \ldots, \theta^{n-1}), \tag{5.10}$$

where $\mathcal{F}_n$ is the $n \times n$ DFT matrix and θ is given by $\theta = e^{j2\pi/(4n)}$. This construction method is used for all the rotation matrices used in this dissertation. For a discussion on non-unitary matrices see [52]. The process of LP can now be defined as:

$$\mathbf{s} = \mathbf{\Theta_n x}. \tag{5.11}$$

If $\mathbf{\Theta_n}$ meets the requirements given above, each symbol in $\mathbf{s}$ will contain all the information in $\mathbf{x}$. If each symbol is then transmitted along an independently fading dimension (such as two time periods experiencing independent fading), the system will achieve a diversity order equal to n.

5.1.3.3 Layering

Layers may be created in a code matrix to increase the throughput of the system. Each layer is independently encoded with the linear precoder. Diophantine approximation theory is then used to keep these layers separable at the decoder [54]. This is done by multiplying each layer with a Diophantine number. The Diophantine number may be given by:

$$\phi = \theta^{1/n} \tag{5.12}$$

where θ is the same number as used for the LP matrix. Each layer is then assigned a power of ϕ. The i^{th} layer is thus multiplied with ϕ^{i-1}. The exact placement of layers will be explained in the following sections.

5.1.4 Diagonal algebraic space time codes

Diagonal algebraic space time (DAST) codes were developed in [14] and are based on algebraic rotations. Basically DAST codes entangle a vector of symbols and place them on the diagonal

of a code matrix. A Hadamard transform is then performed on the code matrix to improve the peak to average power ratio. The structure can thus be given by:

$$\mathbf{C}_{N_T} = \mathcal{H}_{M_T} diag(x_1, x_2, \ldots, x_{N_T}) \tag{5.13}$$

where $\mathbf{x} = \mathbf{\Theta}_{N_T}\mathbf{s}$ and $\mathcal{H}_{N_T}$ is the size N_T Hadamard matrix. Since Hadamard matrices only exist in sizes that are a power of two, these DAST codes only exist for $N_T = 2^k, k \in \mathbb{Z}; k \geq 0$. Universal DAST codes that exist for any N_T are of the form:

$$\mathbf{C}_{N_T} = \begin{pmatrix} x_1 & 0 & \ldots & 0 \\ 0 & x_2 & \ldots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \ldots & x_{N_T} \end{pmatrix}, \tag{5.14}$$

where $\mathbf{x} = \mathbf{\Theta}_{N_T}\mathbf{s}$ and $\mathbf{\Theta}_{N_T}$ is the $N_T \times N_T$ rotation matrix defined by equation 5.10. Since the information in each symbol s_i will be transmitted from each transmit antenna and at each time instance, DAST codes are able to provide full diversity while achieving a rate of 1. DAST Codes (and all codes that use LP) can be efficiently decoded using a SD.

5.1.5 Threaded algebraic space time codes

Threaded algebraic space time codes (TAST) codes are also known as universal space time codes and can achieve full diversity at a transmission rate equal to N_T (the highest possible rate) [54]. TAST codes are essentially an extension of universal DAST codes. DAST codes have one thread placed on the main diagonal of the code matrix. TAST codes have up to N_T threads. These threads are placed on the cyclicly shifted diagonals of the code matrix. To improve performance, Diophantine approximation theory may be used to create Diophantine numbers which make the different threads 'transparent' to each other at the receiver. The code will thus have the following structure:

$$\mathbf{C}_{N_T} = \begin{pmatrix} \phi^0 x_{1,1} & \phi^1 x_{2,1} & \phi^2 x_{3,1} & \ldots & \phi^{N_T-1} x_{N_T,1} \\ \phi^{N_T-1} x_{N_T,2} & \phi^0 x_{1,2} & \phi^1 x_{2,2} & \ldots & \phi^{N_T-2} x_{N_T-1,2} \\ \phi^{N_T-2} x_{N_T-1,3} & \phi^{N_T-1} x_{N_T,3} & \phi^0 x_{1,3} & \ldots & \phi^{N_T-3} x_{N_T-2,3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \phi^1 x_{2,N_T} & \phi^2 x_{3,N_T} & \phi^3 x_{4,N_T} & \ldots & \phi^0 x_{1,N_T} \end{pmatrix} \tag{5.15}$$

where

$$\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,N_T})^{\mathcal{T}} \quad (5.16)$$

$$= \mathbf{\Theta}_{N_T}\mathbf{s}_i \quad (5.17)$$

and

$$\mathbf{s}_i = (s_{1+N_T(i-1)}, s_{2+N_T(i-1)}, \ldots, s_{N_T+N_T(i-1)}) \quad (5.18)$$

The sequence of information symbols ($\mathbf{s}$) is thus demultiplexed up into N_T vectors ($\mathbf{s}_i$), individually precoded using Equation 5.17, multiplied by a Diophantine number and then placed on the diagonals of the code matrix. The algebraic rotation matrix used is the same one as was used for the DAST codes (equation 5.10) and the Diophantine number $\phi = \theta^{1/N_T}$ where θ is the same as that used for the DAST code. TAST codes are able to extract all the diversity available in space and time ($N_T N_R N_B$).

5.2 SF CODES

In frequency selective channels, different frequencies separated by more than the coherence bandwidth may experience independent fading. If one were to place the same information on N_f independently faded frequencies, a diversity order of N_f can be achieved. This is the concept of SF codes. Since OFDM allows for the easy placement of information on carriers, SF codes are a natural coding technique to use for OFDM systems. However, multi-carrier systems other than OFDM can also make use of SF codes. Consider an OFDM system with N_f-subcarriers. A codeword $\mathbf{C}[k]$ can be structured as follows:

$$\mathbf{C}[k] = (\mathbf{c}_1[k] \;\; \mathbf{c}_2[k] \;\; \ldots \;\; \mathbf{c}_{N_T}[k]) \quad (5.19)$$

$$= \begin{pmatrix} c_{1,1}[k] & c_{1,2}[k] & \ldots & c_{1,N_T}[k] \\ c_{2,1}[k] & c_{2,2}[k] & \ldots & c_{2,N_T}[k] \\ \vdots & \vdots & \ddots & \vdots \\ c_{N_f,1}[k] & c_{N_f,2}[k] & \ldots & c_{N_f,N_T}[k] \end{pmatrix} \quad (5.20)$$

where $c_{p,i}[k]$ contains the symbol transmitted at the i^{th} transmit antenna on the p^{th} subcarrier during the k^{th} OFDM symbol. The signal received at the j^{th} receive antenna was given in equation 4.44 and is reproduced here:

$$\mathbf{y}_j[k] = \sum_{i=1}^{N_T} \mathbf{h}_{ij}^f[k] diag(\mathbf{c}_i[k]) + \mathbf{w}_j[k]. \quad (5.21)$$

A channel matrix over all transmit-receive antenna pairs and subcarriers may be constructed as follows:

$$\mathbf{H}^f[k] = \begin{pmatrix} \mathbf{D}^h_{1,1}[k] & \mathbf{D}^h_{1,2}[k] & \dots & \mathbf{D}^h_{1,N_T}[k] \\ \mathbf{D}^h_{2,1}[k] & \mathbf{D}^h_{2,2}[k] & \dots & \mathbf{D}^h_{2,N_T}[k] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{D}^h_{N_R,1}[k] & \mathbf{D}^h_{N_R,2}[k] & \dots & \mathbf{D}^h_{N_R,N_T}[k] \end{pmatrix} \tag{5.22}$$

where

$$\mathbf{D}^h_{ij}[k] = diag(\mathbf{h}^f_{ij}[k]) \tag{5.23}$$

$$= \begin{pmatrix} h^f_{ij1}[k] & 0 & \dots & 0 \\ 0 & h^f_{ij2}[k] & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & h^f_{ijN_f}[k] \end{pmatrix}. \tag{5.24}$$

Defining $\mathbf{c}^v[k]$ as a $N_T N_f \times 1$ vector version of $\mathbf{C}[k]$:

$$\mathbf{c}^v[k] = [\mathbf{c}^{\mathcal{T}}_1[k]\, \mathbf{c}^{\mathcal{T}}_2[k]\, \dots\, \mathbf{c}^{\mathcal{T}}_{N_T}[k]]^{\mathcal{T}}, \tag{5.25}$$

one can can rewrite Equation 5.21 as follows:

$$\mathbf{y}[k] = \mathbf{H}^f[k]\mathbf{c}^v[k]. \tag{5.26}$$

Since N_f is generally large (128 or more) the code $\mathbf{C}[k]$ is often broken up into J sub-codes that span $N_{fc} = N_f/J$ subcarriers. The m^{th} $N_{fc} \times N_T$ sub-code matrix $\mathbf{C}_m[k]$ can be given as:

$$\mathbf{C}_m[k] = [\mathbf{c}_{m,1}[k]\, \mathbf{c}_{m,2}[k]\, \dots\, \mathbf{c}_{m,N_T}[k]] \tag{5.27}$$

$$= \begin{pmatrix} c_{m,1,1}[k] & c_{m,1,2}[k] & \dots & c_{m,1,N_T}[k] \\ c_{m,2,1}[k] & c_{m,2,2}[k] & \dots & c_{m,2,N_T}[k] \\ \vdots & \vdots & \ddots & \vdots \\ c_{m,N_{fc},1}[k] & c_{m,N_{fc},2}[k] & \dots & c_{m,N_{fc},N_T}[k] \end{pmatrix} \tag{5.28}$$

where $\mathbf{c}_{m,i}$ is the vector of symbols from the m^{th} code transmitted on the i^{th} transmit antenna. Thus $\mathbf{C}[k]$ may be given by

$$\mathbf{C}[k] = [\mathbf{C}^{\mathcal{T}}_1[k]\ \mathbf{C}^{\mathcal{T}}_2[k]\ \dots \mathbf{C}^{\mathcal{T}}_J[k]]^{\mathcal{T}} \tag{5.29}$$

The design procedure for the code will now be considered.

5.2.1 SF code design

The methods of linear precoding and layering used in the ST codes can easily be extended to create SF codes. If a vector of symbols that has been linearly precoded is placed over independent fading dimensions then the diversity in these dimensions will be extracted irrespective of whether these dimensions occur in time or frequency. Using the structure of the generic DAST code will thus produce a rate-1 full diversity code while using the structure of the TAST code will produce a rate-N_T full diversity code.

To guarantee the extraction of all the diversity in a frequency selective channel with K independent fading paths as well as the diversity from the N_T transmit antennas, the number of subcarriers over which the code should be defined is $N_{fc} = N_T K$ [55]. Thus the total number of sub-codes is $J = N_f/N_{fc}$ where it will be assumed that N_f is an integer multiple of N_{fc}.

5.2.1.1 Rate-1 SF code

Since the code structure will be the same for each OFDM symbol, for the purpose of notational brevity the OFDM symbol indice will be dropped. Let $\mathbf{s}$ be the size $N_f \times 1$ vector of symbols to be transmitted. $\mathbf{s}$ is broken up into J sub-vectors $\mathbf{s}_m$ of length N_{fc} and linearly precoded to yield J encoded sub-vectors $\mathbf{x}_m$

$$\mathbf{s} = [\mathbf{s}_1^{\mathcal{T}}\ \mathbf{s}_2^{\mathcal{T}}\ \dots \mathbf{s}_J^{\mathcal{T}}]^{\mathcal{T}} \tag{5.30}$$

$$\mathbf{x}_m = \mathbf{\Theta}_{N_{fc}}\mathbf{s}_m \quad for\ \ 1 < m < J \tag{5.31}$$

These encoded subvectors can then be further broken up into vectors of length K:

$$\mathbf{x}_m = [\mathbf{x}_{m1}^{\mathcal{T}}\ \mathbf{x}_{m2}^{\mathcal{T}}\ \dots \mathbf{x}_{mN_T}^{\mathcal{T}}]^{\mathcal{T}} \tag{5.32}$$

The m^{th} full diversity rate-1 SF code sub-matrix can be given as [16]:

$$\mathbf{C}_m = \sqrt{N_T}\begin{pmatrix} \mathbf{x}_{m,1} & \mathbf{0}_K & \dots & \mathbf{0}_K \\ \mathbf{0}_K & \mathbf{x}_{m,2} & \dots & \mathbf{0}_K \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_K & \mathbf{0}_K & \dots & \mathbf{x}_{m,N_T} \end{pmatrix} \tag{5.33}$$

where $\mathbf{0}_K$ is a column vector of 0s with length K. This code will be able to extract $N_T N_R K$ orders of diversity. As an example, consider the case where $K = 2, N_T = 2, N_{fc} = 4$. The

sub-code matrix C_m will be:

$$\mathbf{C}_i = \sqrt{2}\begin{pmatrix} x_m[1] & 0 \\ x_m[2] & 0 \\ 0 & x_m[3] \\ 0 & x_m[4] \end{pmatrix} \tag{5.34}$$

where $\mathbf{x}_m = \Theta_4 \mathbf{s}_m$.

5.2.1.2 Rate-N_T SF code

Let $\mathbf{s}$ be the size $N_T N_f \times 1$ vector of symbols to be transmitted. $\mathbf{s}$ is broken up into J sub-vectors $\mathbf{s}_m$ of length $N_T N_{fc}$ to be placed in the m^{th} subcode. This vector is then further broken down into N_T subvectors, $\mathbf{s}_{mj}$, of length N_{fc} and linearly precoded to yield $N_T J$ encoded sub-vectors $\mathbf{x}_{mj}$.

$$\mathbf{s} = [\mathbf{s}_1^{\mathcal{T}} \ \mathbf{s}_2^{\mathcal{T}} \ \ldots \mathbf{s}_J^{\mathcal{T}}]^{\mathcal{T}} \tag{5.35}$$

$$\mathbf{s}_m = [\mathbf{s}_{m1}^{\mathcal{T}} \ \mathbf{s}_{2,m2}^{\mathcal{T}} \ \ldots \mathbf{s}_{mN_T}^{\mathcal{T}}]^{\mathcal{T}} \tag{5.36}$$

$$\mathbf{x}_{mj} = \mathbf{\Theta}_{N_{fc}} \mathbf{s}_{mj} \ for \ 1 < m < N_T J. \tag{5.37}$$

These encoded subvectors can then be split up into N_T vectors of length K:

$$\mathbf{x}_{mj} = [\mathbf{x}_{mj1}^{\mathcal{T}} \ \mathbf{x}_{mj2}^{\mathcal{T}} \ \ldots \ \mathbf{x}_{mjN_T}^{\mathcal{T}}]^{\mathcal{T}}. \tag{5.38}$$

The m^{th} sub-code of a full diversity rate-N_T SF code can then generically be given by [18]:

$$\mathbf{C}_m = \begin{pmatrix} \phi^0 \mathbf{x}_{m11} & \phi^1 \mathbf{x}_{m21} & \ldots & \phi^{N_T-1} \mathbf{x}_{mN_T1} \\ \phi^{N_T-1} \mathbf{x}_{mN_T2} & \phi^0 \mathbf{x}_{m12} & \ldots & \phi^{N_T-2} \mathbf{x}_{m(N_T-1)2} \\ \vdots & \vdots & \ddots & \vdots \\ \phi^2 \mathbf{x}_{m2N_T} & \phi^3 \mathbf{x}_{m3N_T} & \ldots & \phi^0 \mathbf{x}_{m1N_T} \end{pmatrix} \tag{5.39}$$

This code will also be able to extract $N_T N_R K$ orders of diversity. As an example consider the case where $N_T = 2, K = 2, N_{Fc} = 4$:

$$\mathbf{C}_m = \begin{pmatrix} \mathbf{x}_{m1}[1] & \phi \mathbf{x}_{m2}[1] \\ \mathbf{x}_{m1}[2] & \phi \mathbf{x}_{m2}[2] \\ \phi \mathbf{x}_{m2}[3] & \mathbf{x}_{m1}[3] \\ \phi \mathbf{x}_{m2}[4] & \mathbf{x}_{m1}[4] \end{pmatrix} \tag{5.40}$$

where $\mathbf{x}_{mj} = \mathbf{\Theta}_4 \mathbf{s}_{mj}$. The performance of the code improves when the channels experienced by the symbols are as independent as possible. Performance may thus be improved in SF codes if the sub-carriers are interleaved to spread the subcarriers of a single sub-code over the entire frequency range that is used [16, 18].

5.3 STF CODES

If the channel experiences frequency selective fading conditions that change independently from one OFDM block to the next, the channel is said to experience triply selective block fading conditions. In this case, diversity may be extracted over space, time and frequency. A code which spans all three of these dimensions is an STF code. Consider a MIMO-OFDM system with N_T transmit antennas, K taps in the channel PDP and N_f subcarriers. Let N_B denote the number of OFDM symbols over which the code is to be defined. The structure of the code will be similar to that of the SF code but different in the sense that it must now be defined over N_B blocks [17]. Thus, generically an STF code will be structured as follows:

$$\mathbf{C} = \left[\mathbf{C}^{\mathcal{T}}[k]\ \mathbf{C}^{\mathcal{T}}[k+1]\ \ldots\ \mathbf{C}^{\mathcal{T}}[k+N_B-1]\right]^{\mathcal{T}} \tag{5.41}$$

$$= \begin{pmatrix} \mathbf{c}_1[k] & \mathbf{c}_2[k] & \ldots & \mathbf{c}_{N_T}[k] \\ \mathbf{c}_1[k+1] & \mathbf{c}_2[k+1] & \ldots & \mathbf{c}_{N_T}[k+1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_1[k+N_B-1] & \mathbf{c}_2[k+N_B-1] & \ldots & \mathbf{c}_{N_T}[k+N_B-1] \end{pmatrix} \tag{5.42}$$

where $\mathbf{c}_i[t]$ denotes the vector of symbols to be placed on the i^{th} antenna during the t^{th} OFDM symbol. Using the frequency domain representation of the channel as given in Equation 5.22, one can create a combined channel matrix $\mathbf{H}^f$:

$$\mathbf{H}^f = \begin{pmatrix} \mathbf{H}^f[k] & & & \\ & \mathbf{H}^f[k+1] & & \\ & & \ddots & \\ & & & \mathbf{H}^f[k+N_B-1] \end{pmatrix} \tag{5.43}$$

Defining a vectorised version of the code $\mathbf{C}$ as:

$$\begin{aligned} \mathbf{c}^v = [&\mathbf{c}_1^{\mathcal{H}}[k]\ \mathbf{c}_2^{\mathcal{H}}[k]\ \ldots\ \mathbf{c}_{N_T}^{\mathcal{H}}[k], \mathbf{c}_1^{\mathcal{H}}[k+1]\ \mathbf{c}_2^{\mathcal{H}}[k+1]\ \ldots\ \mathbf{c}_{N_T}^{\mathcal{H}}[k+1],\ \ldots, \\ &\mathbf{c}_1^{\mathcal{H}}[k+N_B-1]\ \mathbf{c}_2^{\mathcal{H}}[k+N_B-1]\ \ldots\ \mathbf{c}_{N_T}^{\mathcal{H}}[k+N_B-1]], \end{aligned} \tag{5.44}$$

allows one to write the received vector as:

$$\mathbf{y} = \begin{pmatrix} \mathbf{y}[k] \\ \mathbf{y}[k+1] \\ \vdots \\ \mathbf{y}[k+N_B-1] \end{pmatrix} = \mathbf{H}^f \mathbf{c}^v. \tag{5.45}$$

Because of the size of N_f, $\mathbf{C}$ is generally also broken up into subcodes $\mathbf{C}_m$. Let $\mathbf{c}_{im}[k]$ be the vector of symbols transmitted from the i^{th} antenna during the k^{th} period of time over the subcarriers in the m^{th} sub-code. In terms of $\mathbf{c}_{im}[k]$, the code structure may be rewritten as in equation 5.46.

$$\mathbf{C} = \left[\mathbf{C}^{\mathcal{T}}[k]\, \mathbf{C}^{\mathcal{T}}[k+1]\, \ldots\, \mathbf{C}^{\mathcal{T}}[k+N_B-1]\right]^{\mathcal{T}} \tag{5.46}$$

$$= \begin{pmatrix} \mathbf{c}_{11}[k] & \mathbf{c}_{21}[k] & \ldots & \mathbf{c}_{N_T1}[k] \\ \mathbf{c}_{12}[k] & \mathbf{c}_{22}[k] & \ldots & \mathbf{c}_{N_T2}[k] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_{1J}[k] & \mathbf{c}_{2J}[k] & \ldots & \mathbf{c}_{N_TJ}[k] \\ \mathbf{c}_{11}[k+1] & \mathbf{c}_{21}[k+1] & \ldots & \mathbf{c}_{N_T1}[k+1] \\ \mathbf{c}_{12}[k+1] & \mathbf{c}_{22}[k+1] & \ldots & \mathbf{c}_{N_T2}[k+1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_{1J}[k+1] & \mathbf{c}_{2J}[k+1] & \ldots & \mathbf{c}_{N_TJ}[k+1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_{11}[k+N_B-1] & \mathbf{c}_{21}[k+N_B-1] & \ldots & \mathbf{c}_{N_T1}[k+N_B-1] \\ \mathbf{c}_{12}[k+N_B-1] & \mathbf{c}_{22}[k+N_B-1] & \ldots & \mathbf{c}_{N_T2}[k+N_B-1] \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{c}_{1J}[k+N_B-1] & \mathbf{c}_{2J}[k+N_B-1] & \ldots & \mathbf{c}_{N_TJ}[k+N_B-1] \end{pmatrix} \tag{5.47}$$

The vectors from code m and time instance k are now combined into a matrix:

$$\mathbf{C}_{km} = [\mathbf{c}_{1m}[k]\, \mathbf{c}_{2m}[k]\, \ldots\, \mathbf{c}_{N_Tm}[k]] \tag{5.48}$$

With this, one can define the m^{th} subcode $\mathbf{C}_m$ with the following structure:

$$\mathbf{C}_m = [\mathbf{C}_{1m}\, \mathbf{C}_{2m}\, \ldots\, \mathbf{C}_{(k+N_B-1)m}] \tag{5.49}$$

In the case where $N_B = 1$, this structure will reduce to the structure for a subcode of an SF code, as given in equation 5.27.

5.3.1 STF Code Design

The concept of spreading LP coded vectors over multiple dimensions, as was used in both TAST and SF codes, may be used to create STF codes. Principally, a vector of symbols is linearly precoded and then placed over all the subcarriers, transmit antennas and time periods. Up to N_T such vectors may be combined using layering to produce a rate N_T STF code.

5.3.1.1 Rate-1 STF Code

Let $\mathbf{s}$ be the size $N_f N_B \times 1$ vector of symbols to be transmitted. $\mathbf{s}$ is broken up into J sub-vectors $\mathbf{s}_m$ of length $N_{fc} N_B$ and linearly precoded to yield J encoded sub-vectors $\mathbf{x}_m$

$$\mathbf{s} = [\mathbf{s}_1^{\mathcal{T}} \ \mathbf{s}_2^{\mathcal{T}} \ \ldots \mathbf{s}_J^{\mathcal{T}}]^{\mathcal{T}} \tag{5.50}$$

$$\mathbf{x}_m = \mathbf{\Theta}_{N_{fc}} \mathbf{s}_m[k] \ \ for \ \ 1 < m < J. \tag{5.51}$$

These encoded subvectors can then be further broken into vectors of length K:

$$\mathbf{x}_m = [\mathbf{x}_{m1}^{\mathcal{T}} \ \mathbf{x}_{m2}^{\mathcal{T}} \ \ldots \mathbf{x}_{mN_T N_B}^{\mathcal{T}}]^{\mathcal{T}}. \tag{5.52}$$

The m^{th} full diversity rate-1 STF code sub-matrix can be given as:

$$\mathbf{C}_m = [\mathbf{C}_{1m} \ \mathbf{C}_{2m} \ \ldots \ \mathbf{C}_{N_B m}], \tag{5.53}$$

where

$$\mathbf{C}_{km} = \begin{pmatrix} \mathbf{x}_{m,(k-1)N_T+1} & \mathbf{0}_K & \ldots & \mathbf{0}_K \\ \mathbf{0}_K & \mathbf{x}_{m,(k-1)N_T+2} & \ldots & \mathbf{0}_K \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_K & \mathbf{0}_K & \ldots & \mathbf{x}_{m,(k-1)N_T+N_T} \end{pmatrix}. \tag{5.54}$$

This code will be able to achieve a diversity order of $N_T N_R K N_B$. As an example, consider the case where $N_T = 2, K = 2, N_B = 2, N_{fs} = 4$. The subcode matrix can be given as [17]:

$$\mathbf{C}_m = \begin{pmatrix} \mathbf{x}_m[1] & 0 & \mathbf{x}_m[5] & 0 \\ \mathbf{x}_m[2] & 0 & \mathbf{x}_m[6] & 0 \\ 0 & \mathbf{x}_m[3] & 0 & \mathbf{x}_m[7] \\ 0 & \mathbf{x}_m[4] & 0 & \mathbf{x}_m[8] \end{pmatrix}, \tag{5.55}$$

where $\mathbf{x}_m = \mathbf{\Theta}_8 \mathbf{s}_m$. An alternative method to create a rate-1 STF code is to create a rate-N_B SF code and then to repeat this code over N_B symbol periods. As an example consider the case

when $N_T = 2, K = 2, M_B = 2, N_{fs} = 4$. The subcode matrix can be given as [17]:

$$\mathbf{C}_m = \begin{pmatrix} \mathbf{x}_m[1] & \phi\mathbf{x}_m[5] & \mathbf{x}_m[1] & \phi\mathbf{x}_m[5] \\ \mathbf{x}_m[2] & \phi\mathbf{x}_m[6] & \mathbf{x}_m[2] & \phi\mathbf{x}_m[6] \\ \phi\mathbf{x}_m[7] & \mathbf{x}_m[3] & \phi\mathbf{x}_m[7] & \mathbf{x}_m[3] \\ \phi\mathbf{x}_m[8] & \mathbf{x}_m[4] & \phi\mathbf{x}_m[8] & \mathbf{x}_m[4] \end{pmatrix}, \tag{5.56}$$

where

$$[x_m[1], x_m[2], x_m[3], x_m[4]] = \mathbf{\Theta}_4[s_m[1], s_m[2], s_m[3], s_m[4]] \tag{5.57}$$

and

$$[x_m[5], x_m[6], x_m[7], x_m[8]] = \mathbf{\Theta}_4[s_m[5], s_m[6], s_m[7], s_m[8]]. \tag{5.58}$$

5.3.1.2 Rate-N_T STF code

Let $\mathbf{s}$ be the size $N_T N_f N_B \times 1$ vector of symbols to be transmitted. $\mathbf{s}$ is broken up into J sub-vectors $\mathbf{s}_m$ of length $N_T N_B N_{fc}$ to be placed in the m^{th} subcode. This vector is then further broken down into N_T subvectors, $\mathbf{s}_{mj}$, of length $N_{fc}N_B$ and linearly precoded to yield $N_T J$ encoded sub-vectors $\mathbf{x}_{mj}$:

$$\mathbf{s} = [\mathbf{s}_1^{\mathcal{T}}\ \mathbf{s}_2^{\mathcal{T}}\ \ldots \mathbf{s}_J^{\mathcal{T}}]^{\mathcal{T}} \tag{5.59}$$

$$\mathbf{s}_m = [\mathbf{s}_{m1}^{\mathcal{T}}\ \mathbf{s}_{2,m2}^{\mathcal{T}}\ \ldots \mathbf{s}_{mN_T}^{\mathcal{T}}]^{\mathcal{T}} \tag{5.60}$$

$$\mathbf{x}_{mj} = \mathbf{\Theta}_{N_{fc}N_B}\mathbf{s}_{mj} \ \ for\ \ 1 < m < N_T J. \tag{5.61}$$

These encoded subvectors can then be split up into $N_T N_B$ vectors of length K:

$$\mathbf{x}_{mj} = [\mathbf{x}_{mj1}\ \mathbf{x}_{mj2}\ \ldots\ \mathbf{x}_{mj(N_T N_B)}]. \tag{5.62}$$

The m^{th} full diversity rate-N_T STF code sub-matrix can be given as [17, 55]:

$$\mathbf{C}_m = [\mathbf{C}_{1m}\ \mathbf{C}_{2m}\ \ldots\ \mathbf{C}_{N_B m}] \tag{5.63}$$

where

$$\mathbf{C}_{km} = \begin{pmatrix} \phi^0\mathbf{x}_{m1((k-1)N_T+1)} & \phi^1\mathbf{x}_{m2((k-1)N_T+1)} & \cdots & \phi^{N_T-1}\mathbf{x}_{mN_T((k-1)N_T+1)} \\ \phi^{N_T-1}\mathbf{x}_{mN_T((k-1)N_T+2)} & \phi^0\mathbf{x}_{m1((k-1)N_T+2)} & \cdots & \phi^{N_T-2}\mathbf{x}_{m(N_T-1)((k-1)N_T+2)} \\ \vdots & \vdots & \ddots & \vdots \\ \phi^2\mathbf{x}_{m2((k-1)N_T+N_T} & \phi^3\mathbf{x}_{m3((k-1)N_T+N_T} & \cdots & \phi^0\mathbf{x}_{m1((k-1)N_T+N_T)} \end{pmatrix}. \tag{5.64}$$

This code will also be able to achieve a diversity order of $N_T N_R K N_B$. As an example consider the case where $N_T = 2, K = 2, N_{fs} = 2, N_B = 2$. The m^{th} subcode matrix can be given as:

$$\mathbf{C}_m = \begin{pmatrix} \mathbf{x}_{m1}[1] & \phi\mathbf{x}_{m2}[1] & \mathbf{x}_{m1}[5] & \phi\mathbf{x}_{m2}[5] \\ \mathbf{x}_{m1}[2] & \phi\mathbf{x}_{m2}[2] & \mathbf{x}_{m1}[6] & \phi\mathbf{x}_{m2}[6] \\ \phi\mathbf{x}_{m2}[3] & \mathbf{x}_{m1}[3] & \phi\mathbf{x}_{m2}[7] & \mathbf{x}_{m1}[7] \\ \phi\mathbf{x}_{m2}[4] & \mathbf{x}_{m1}[4] & \phi\mathbf{x}_{m2}[8] & \mathbf{x}_{m1}[8] \end{pmatrix} \tag{5.65}$$

where $\mathbf{x}_{m1} = \mathbf{\Theta}_8 \mathbf{s}_{m1}$ and $\mathbf{x}_{m2} = \mathbf{\Theta}_8 \mathbf{s}_{m2}$.

5.4 SIMULATION RESULTS

This dissertation uses ST, SF and STF codes and this section provides the result curves for these codes without any FEC coding. Simulations were performed on the MIMO-WiMAX platform developed in [56]. The MIMO-WiMAX parameters chosen for all simulations done are illustrated in Table 5.1. The coding schemes and corresponding conditions in which they

TABLE 5.1: MIMO-WiMAX simulation parameters

MIMO-WiMAX parameters	
Transmit antennas	2
Receive antennas	2
FFT size	128
Number of sub-channels	2
Users per sub-channel	1
Mode	FUSC
Cyclic prefix length	0.25
Maximum Doppler spread	$f_d = 100Hz$
Sampling time	$T_s = 0.8\mu$s
Channel bandwidth	1.25MHz
Transmit filter	Square root raised cosine, $\alpha = 0.5$
Receive filter	Square root raised cosine, $\alpha = 0.5$

were simulated, in all the simulations, are illustrated in Table 5.2. Modulation schemes were adjusted appropriately to maintain the same b/s/Hz. For comparison purposes, it was chosen at 2 b/s/Hz to be able to compare the coding techniques over various channel conditions. In the

case of the SF rate-2 code some simulations were also performed using QPSK to yield a system with a throughput of 4 b/s/Hz.

TABLE 5.2: Coding parameters used in simulation

Code	Modulation	Fading condition	Decoder
ST code	QPSK	Quasi-static fading	Single-symbol ML
Rate-1 SF code	QPSK	Block fading	Sphere-decoding
Rate-2 SF code	BPSK/QPSK	Block fading	Sphere-decoding
Rate-1 STF code	QPSK	Block fading	Sphere-decoding
Rate-2 STF code	BPSK	Block fading	Sphere-decoding

For the purpose of simulation in this dissertation, two different PDPs are used. To illustrate the performance in ideal cases, a simple, equal power, two tap channel is used. The two taps are at 0 μs and 8 μs. The second PDP which is used is the more realistic 20 tap suburban alternative channel [1]. A figure showing this PDP can be seen in Appendix A.

5.4.1 Two tap channel, no channel correlation

The simulation parameters for the first simulation were set up to illustrate the maximum performance of these codes in ideal channel conditions and are summarised in Table 5.3. The simulations were performed on the two tap, equal power, Rayleigh multipath channel to

TABLE 5.3: Simulation 1 parameters

Simulation parameters	
Frequency-Selectivity	Two tap equal power PDP at 0 μs and 8 μs
Time-selectivity	Block-fading conditions with $f_d = 100Hz$
Space-selectivity	None

illustrate the achievable diversity gain for a system with a frequency diversity order of two. For all fading conditions simulated, no time-selectivity was introduced, as the coding techniques assume that the channel remains constant for each OFDM symbol. In the case of quasi-static conditions, the channel remained constant over two OFDM symbols. From Figure 5.5, it can be seen that the STF codes have a steeper slope compared to the SF and ST codes. This illustrates the larger diversity gain exploited by STF codes in block fading conditions. There is a 1 dB loss in performance when QPSK is used instead of BPSK for the rate-2 SF code. The

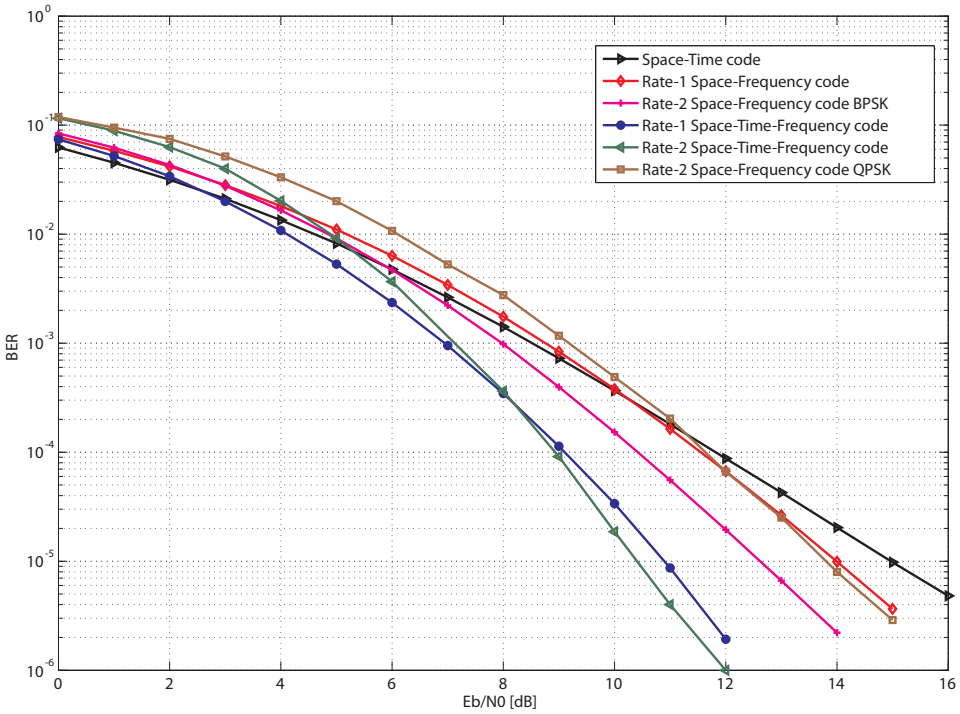

FIGURE 5.5: Results obtained for coding techniques in a two tap equal power PDP at 0 μs and 8 μs with no space-selectivity.

rate-2 SF BPSK code outperforms the rate-1 SF code owing to a higher diversity product [16]. The diversity product consists of an intrinsic diversity product as well as an extrinsic diversity product, which depends on the channel used. Larger delay spreads cause an increase in diversity product for the rate-2 SF BPSK code over the rate-1 SF code. The simulation used a large delay spread (approximately 13 sample delay) which causes the 1.5 dB performance difference. The rate-2 STF code outperforms the rate-1 STF code at high E_b/N_0 only slightly, as these codes have very similar diversity products.

5.4.2 Two tap channel, correlated channel conditions

In this study, simulation parameters were set up to illustrate the performance of these coding techniques in highly correlated channel conditions and are summarised in Table 5.4. The frequency selectivity was kept the same as for the first simulation, to illustrate the effects a highly correlated channel will have on the performance of these codes compared to the uncorrelated conditions. Space-selectivity was introduced at the transmitter and receiver using the method described in Equation 4.30. Figure 5.6 clearly illustrates the performance degradation due to correlated MIMO channels. Compared to Figure 5.5, each curve suffers a

TABLE 5.4: Simulation 2 parameters

Simulation parameters	
Frequency-Selectivity	Two tap equal power PDP at 0 μs and 8 μs
Time-Selectivity	Block-fading conditions with $f_d = 100Hz$
Space-Selectivity	Exponential correlation matrix with $r = 0.7$

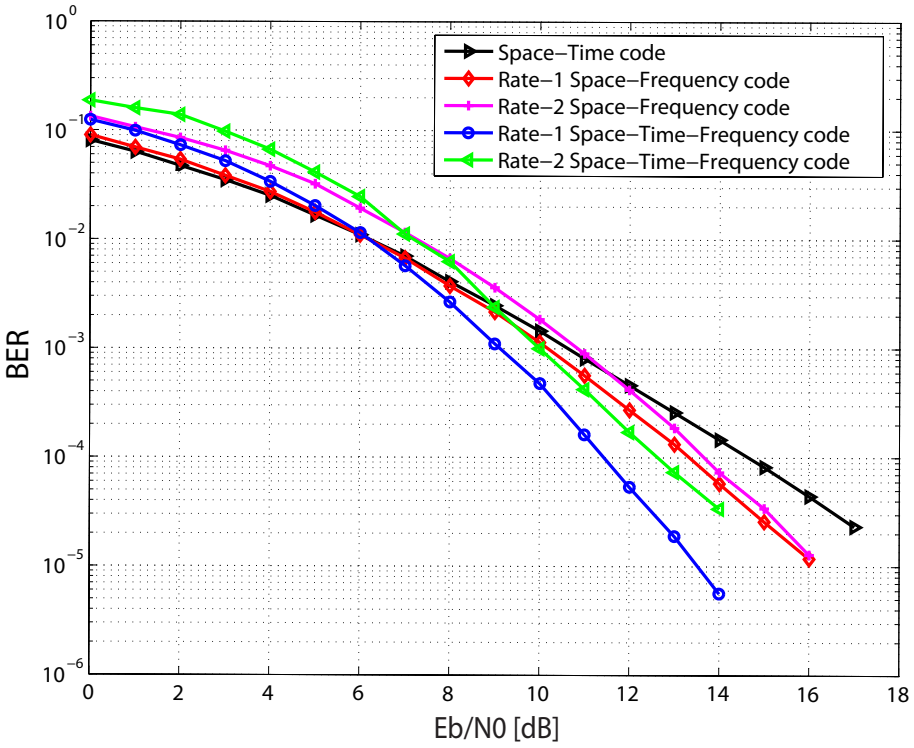

FIGURE 5.6: Results obtained for coding techniques in a two tap equal power PDP at 0 μs and 8 μs using an exponential correlation matrix with $r = 0.7$

performance loss of approximately 3 dB. The results obtained follows the theoretical analysis from [41] which shows that a 70% correlation will result in a 3dB loss compared to uncorrelated conditions. The rate-1 STF code crosses the ST code graph at 6 dB, where in Figure 5.5, the STF code already showed improvement at around 2.5 dB - 3 dB. These results illustrate the degradation in performance of high diversity codes in highly correlated channels at lower E_b/N_0 values. At high E_b/N_0 values, the rate-1 SF and rate-2 SF codes differ by less than 0.5 dB, whereas in Figure 5.5, the rate-2 showed a significant improvement over all E_b/N_0 values. This is due to a lower diversity product, where the extrinsic diversity product is dependent on the channel, which is highly correlated, reducing overall diversity in the channel. The

performance of both the rate-2 SF and rate-2 STF codes were more heavily degraded than the respective rate-1 SF and rate-1 STF codes.

The comparisons of the multi-antenna codes presented for different correlations are illustrated in Figure 5.7. Figures 5.7 illustrate a comparison in E_b/N_0 for different correlations to maintain

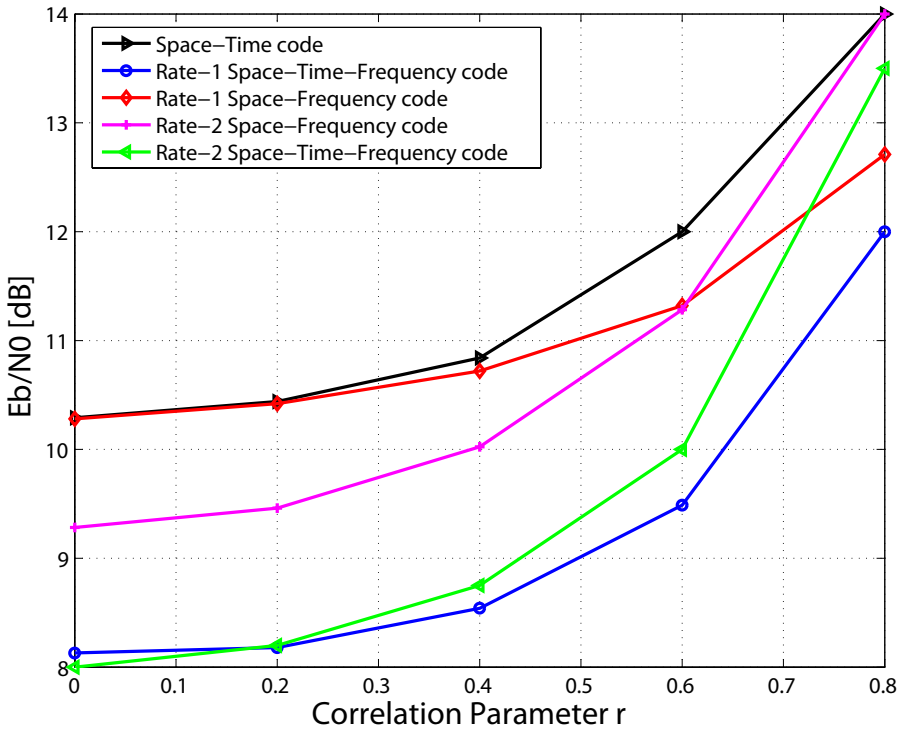

FIGURE 5.7: Performance comparison in two tap equal power PDP for various correlations

a BER of $10^{-3.5}$ in the two-ray equal power PDP. From the figure it can be seen that the rate-1 STF code requires the least amount of power to maintain the BER of $10^{-3.5}$. It also indicates that the rate-2 SF code will perform better than the rate-1 SF code for $r \leq 0.6$.

5.4.3 Suburban Alternative channel, no channel correlation

In this study, the effect, of changing the PDP of the channel, on the relative performance of the codes is analysed. From Figure 5.8, it can be seen that the STF codes have a steeper slope compared to the SF and ST codes. This illustrates the larger diversity gain exploited by STF codes in block fading conditions. There is a 1.5 dB loss in performance at a BER of

10^{-3}, when QPSK is used instead of BPSK for the rate-2 SF code. In contrast to the case on the two tap channel where the rate-2 SF BPSK code outperforms the rate-1 SF code, in this case the rate-1 SF code outperforms the rate-2 SF code. This change in performance is due to the lower extrinsic diversity in the channel. In the two tap channel the rate-2 STF code outperforms the rate-1 STF code at high E_b/N_0 while on the suburban alternative the rate-1 STF code outperforms the rate-2 STF code. This is also due to the lower extrinsic diversity in the channel. From these results one can conclude that the higher rate codes are more sensitive to channel conditions than the rate-1 codes.

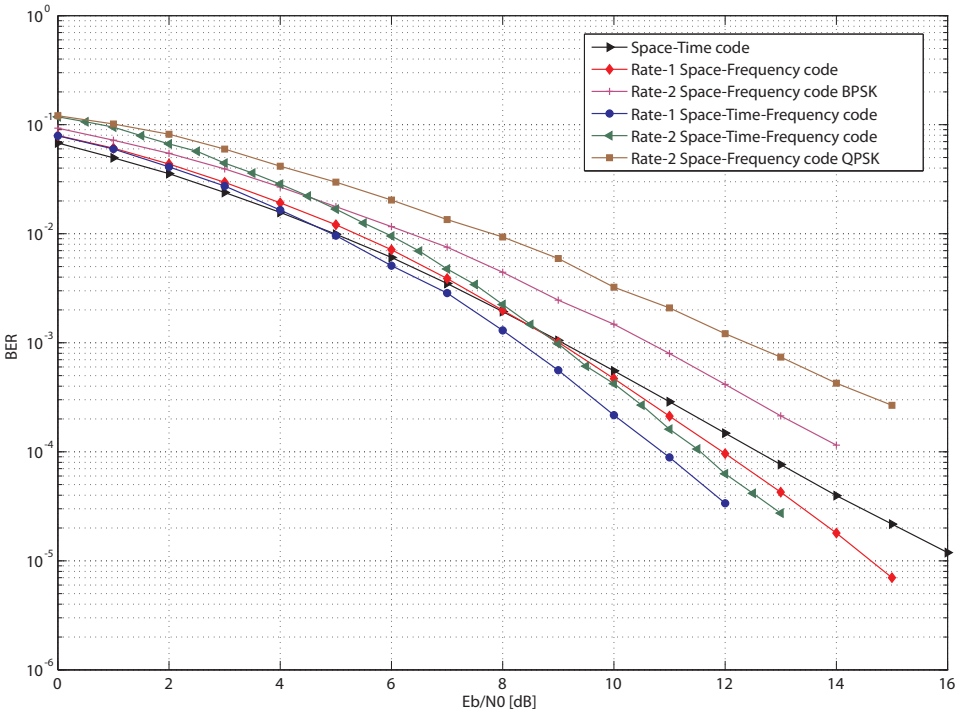

FIGURE 5.8: Results obtained for coding techniques in the 20 tap suburban alternative PDP with no space-selectivity.

5.4.4 Suburban Alternative channel, correlated channel conditions

In this study, the effect of correlation in the channel on the relative performance of the codes is analysed. Figure 5.9 shows the performance of the codes on the suburban alternative channel for various correlation values. For the suburban alternative PDP, the STF code performs best for $r \leq 0.6$, whereas the rate-1 SF code shows the best performance for highly correlated conditions. The rate-2 SF code has the worst performance for all correlated conditions in the suburban alternative PDP. The slope of the rate-2 STF code increases faster than the slope of

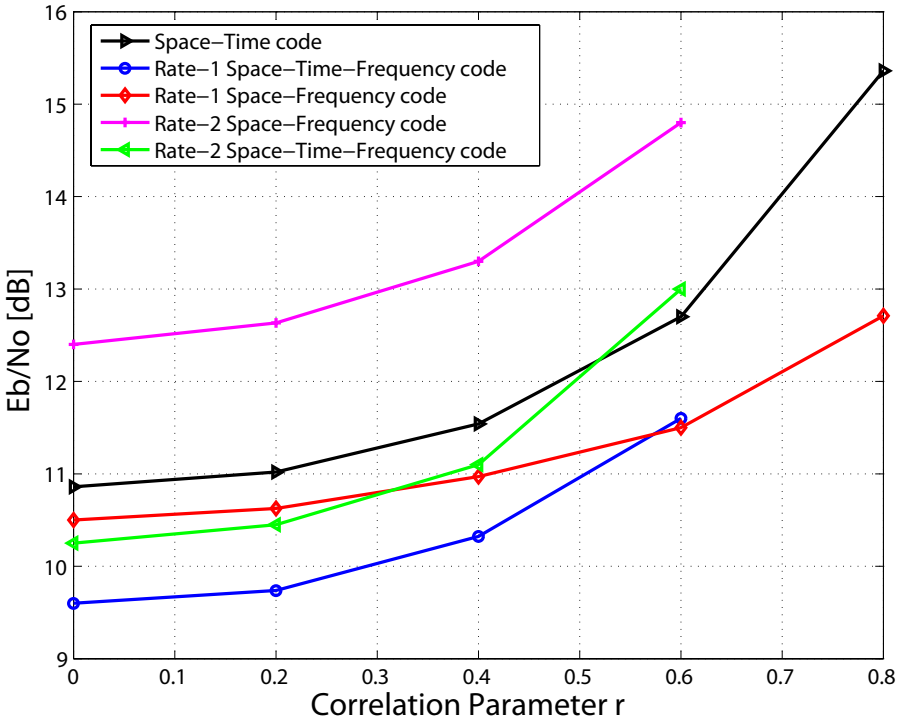

FIGURE 5.9: Performance comparison in a suburban alternative PDP for various correlations

the rate-1 STF code in both Figures 5.7 and 5.9. This indicates that in both ideal and realistic channel PDPs, higher rate codes are more susceptible to highly correlated channel conditions.

5.5 ACHIEVABLE DIVERSITY OF QUANTISED LP CODED STF CODES

The use of LP allows for diversity to be achieved without sacrificing transmission rate. As a result, the requirements on the number of bits at the quantiser is increased. This section derives the limitation on the achievable diversity in a quantised LP system.

5.5.1 Quantisation

Quantisation is the process of approximating a continuous range of values by a set of discrete symbols [57]. In the context of this dissertation, it is taken to mean the digitisation of the real valued baseband received signal with pulse-code modulation (PCM) [58]. Since quantisation is an approximation, a certain amount of error will be introduced by the quantisation process.

This error can be given as:

$$\epsilon = q(x) - x, \tag{5.66}$$

$$\therefore q(x) = x + \epsilon, \tag{5.67}$$

where x is the signal that is quantised, ϵ is the error and $q(x)$ is the quantised signal. Equation (5.67) shows that the quantised signal can be thought of as an unquantised signal to which noise has been added. This noise can generally be regarded as being Gaussian when the quantiser has "high-resolution" [59], i.e. when the number of quantisation levels is large. In [57] it is shown that if a quantiser has N_Q levels (and therefore quantises a signal to $Q_B = \log_2 N_Q$ bits), and the source signal has a range of R_s, and the level spacing of the quantiser (Δ) is chosen as $\Delta = R_s/N_Q$, the SNR can be approximated as:

$$\text{SNR} = C + 6Q_B \text{ dB}, \tag{5.68}$$

where C is a constant. This result is occasionally referred to as the "6-dB-per-bit rule". This results in an error floor depending on the number of bits used in quantisation. Fortunately, for a system employing three bits of quantisation ($Q_B = 3$ bits), the SNR is approximately 18 dB and thus the error floor is only expected after a SNR of 18 dB.

In addition to the noise introduced by quantisation, [60] shows that the quantisation of a time signal affects the precision of a single-tone frequency estimator based on the FFT. The results in [60] show that the variance of the estimated frequency increases dramatically once the number of bits used in quantisation (Q_B) is below four bits. As OFDM is also based on the FFT, a drastic loss in performance is expected in OFDM systems quantised to less than four bits.

5.5.2 Effect of quantisation on LP coded systems

LP is used to encode the data symbols in a manner that extracts the most benefit from the diversity available in the channel. It does this by mapping a set of data symbols to a new set of encoded symbols that are transmitted. An LP coded data vector ($\mathbf{x}$) permits the correct decoding of all of the data symbols ($\mathbf{s}$) encoded by the LP, upon the reception of a single encoded symbol (x_i), thus yielding diversity equal to the rank of the rotation matrix (R_Θ) [52]. Each encoded symbol is therefore required to represent all the information in $\mathbf{s}$. Therefore, if an original data

symbol represented N_{ob} bits per dimension, the number of information bits per dimension that the encoded symbol represents is:

$$N_{cb} = N_{ob} R_{\Theta}. \tag{5.69}$$

The transmission of an N_f sized encoded OFDM block would therefore entail the transmission of $N_f N_{cb}$ bits per dimension over N_f time intervals. Thus, assuming that the information is spread equally over the N_f time intervals, this would require each time interval to contain at least N_{cb} bits per dimension. This would require the digital-to-analogue converters (DACs) and the analogue-to-digital converters (ADCs), used in the system, to quantise the signal per time interval to at least N_{cb} bits. Should the ADCs and the DACs not quantise to N_{cb} bits, then the achievable diversity will be less than the diversity the code was designed for. In other words, if the number of bits of the ADC and the DAC is Q_B, then the maximum achievable diversity (D_{max}) due to linear precoding in a system is given by the following:

$$D_{max} = \frac{Q_B}{N_{ob}}. \tag{5.70}$$

In a MIMO system, a portion of the diversity is obtained from the receive antennas, and only diversity due to rotation is given in (5.70), thus the total diversity D_{total} of the system is given by

$$D_{total} = D_{max} N_R, \tag{5.71}$$

where N_R is the number of receive antennas.

5.5.2.1 Quantising of the received signal

In this dissertation, only the received signal is quantised. The transmitted signal is not quantised except by the limitations of the IEEE-754 *double* format [61] as used by GNU C++[12]. The range of the received signal is mapped to the range of an *unsigned long int* of 64 bits. The received signal is then cast to the *unsigned long int* type. Thus, the lowest value of the signal is then represented by 0 and the highest value by $2^{64} - 1$. Then all but the Q_B most significant bits are discarded, and the result that has been mapped is cast back to type *double* and mapped back to the original range of the received signal. This is done separately for the in-phase and quadrature parts of the received signal. Figure 5.10 shows a diagram of the receiver. The figure shows that the in-phase and quadrature components of the OFDM time signal are quantised separately before being added together for the FFT operation.

[1] http://www.open-std.org/jtc1/sc22/wg21/
[2] http://gcc.gnu.org/

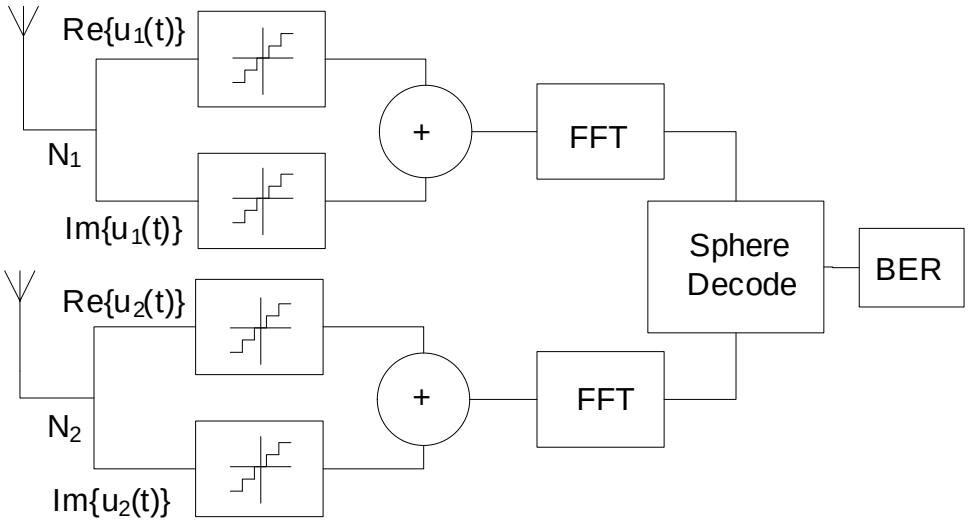

FIGURE 5.10: Block diagram of the implemented baseband receiver.

5.5.3 Ideal two-tap channel simulations

Figure 5.11 shows the BER versus E_B/N_0 plot of the rate-2 STF code, defined by (5.65), for various levels of quantisation. Superimposed on the figure are the theoretical BER plots for QPSK with 6, 8 and 10 orders of diversity obtained from MATLAB's[3] *berfading* function. To facilitate the comparison between the slope of the simulated results and that of the theoretically expected results, the theoretical results were shifted horizontally on the E_B/N_0 axis. In the figure, a noticeable degradation in performance can be observed starting at 5 bits of quantisation ($Q_B = 5$ bits). Equation 5.71 predicts that a system quantised to 5 bits should achieve 10 orders of diversity. By comparing the curve at $Q_B = 5$ bits with the theoretical 10^{th} order diversity curve, it is evident that the asymptotic slopes are very similar. With $Q_B = 4$ bits, Equation 5.71 predicts 8 orders of diversity. Comparing the $Q_B = 4$ curve with the theoretical 8^{th} order diversity curve, it is apparent that the asymptotic slopes are very similar. At 3 bits, Equation 5.71 predicts 6 orders of diversity. However, by comparing the $Q_B = 3$ curve with the theoretical 6^{th} order diversity curve, it appears that the achieved diversity is less than expected. Finally, at 2 bits, the system fails and an error floor of 0.1 is observed. Figure 5.12 shows the BER versus E_B/N_0 plot of the rate 2 SF code, defined by Equation 5.40, for various levels of quantisation. It is evident that the unquantised system achieves less than ten orders of diversity, with the slope indicating that a diversity order of 8 is achieved. With $Q_B = 4$ bits, the code achieves close to three orders of diversity. At $Q_B = 3$ bits, the results follows the two orders

[3] Copyright The Mathworks, Inc., www.mathworks.com

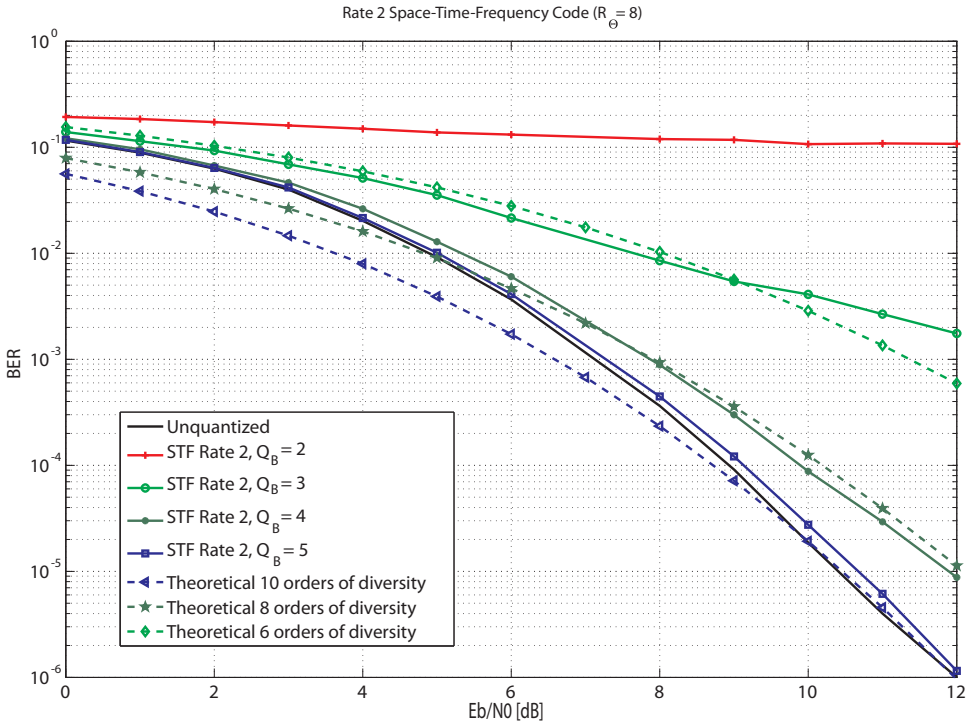

FIGURE 5.11: BER plot of a rate-2 STF Code with $R_\Theta = 8$, with various levels of quantisation. $N_T = N_R = K = 2$

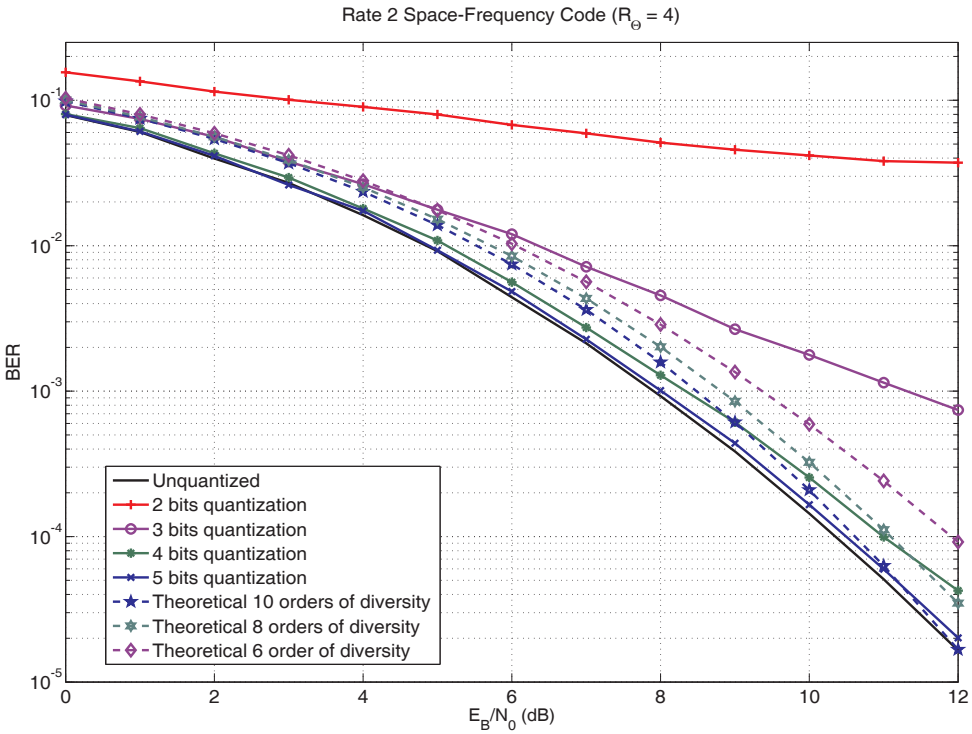

FIGURE 5.12: BER plot of a rate 2 SF Code with $R_\Theta = 4$, with various levels of quantisation. $N_T = N_R = K = 2$

of diversity theoretical results up to an E_B/N_0 ratio of 6dB, beyond which its performance is severely degraded. As was the case for the STF rate 2 code, at 2 bits the system fails with an error floor of approximately 0.03.

Comparing Figures 5.11 and 5.12, it can be seen that the SF code outperforms the STF code when $Q_B < 4$, and that the STF code outperforms the SF code when $Q_B \geq 4$. This is expected, since the STF code has a larger rotation matrix, and would therefore be more severely affected by quantisation than the SF code which has a smaller rotation matrix.

5.5.4 Suburban alternative PDP simulation

In Figure 5.13, it can be seen that an initial order of diversity has been lost due to the channel, but the performance degradation due to quantisation still follows the prediction. A code designed for this channel may be able to extract the additional diversity offered by this channel, but would still be limited by quantisation. Figure 5.14 shows that less than three orders of diversity is achieved by the SF code in the suburban alternative channel. It is evident that neither the SF nor the STF code can fully extract the diversity available in the suburban alternative channel. This may be due to the fact that they were designed for the two tap channel.

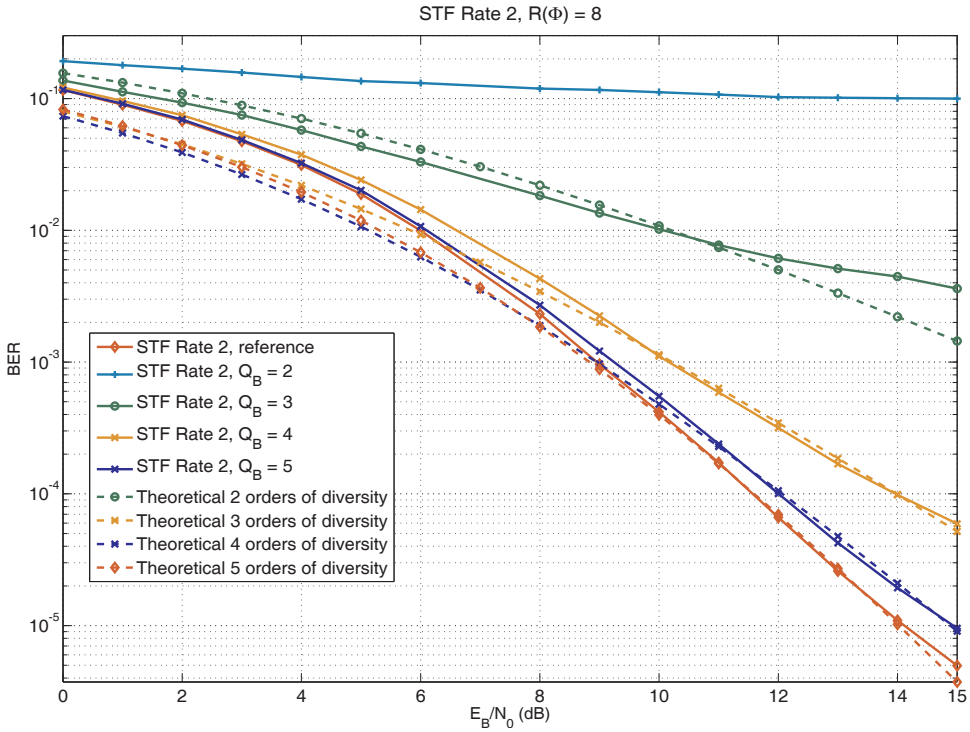

FIGURE 5.13: BER plot of a rate-2 STF Code with $R_\Theta = 8$, with various levels of quantisation. Suburban alternative PDP. $N_T = N_R = K = 2$

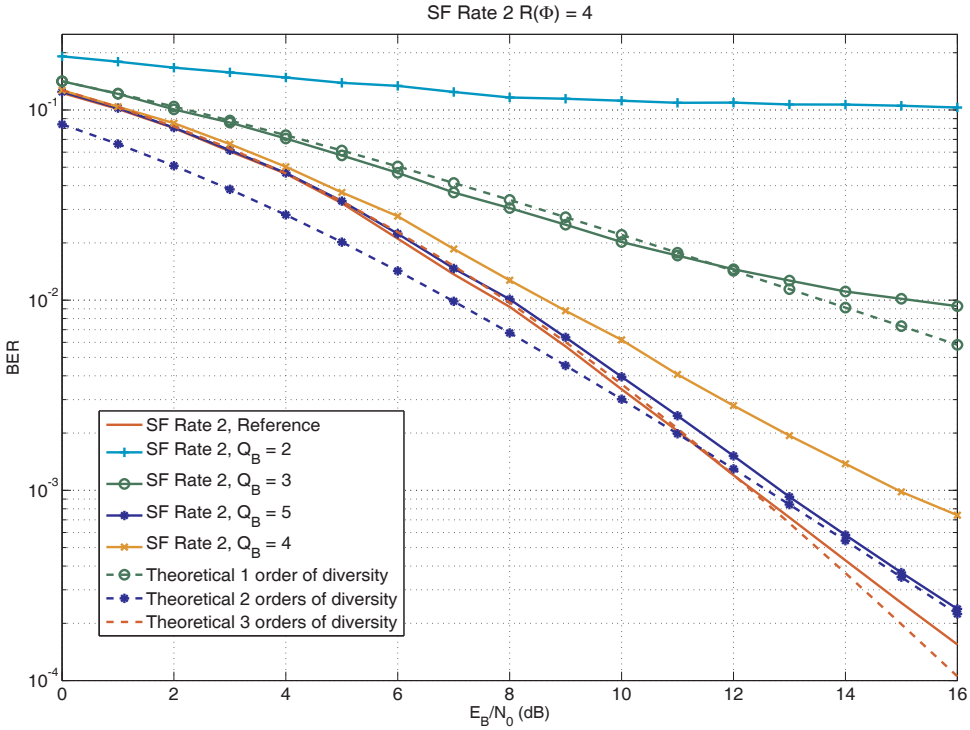

FIGURE 5.14: BER plot of a rate-2 SF Code with $R_\Theta = 4$, with various levels of quantisation. Suburban alternative PDP. $N_T = N_R = K = 2$

Chapter Six

MIMO Detection

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

$$\mathbf{y} = \mathbf{H}\mathbf{x} + \mathbf{n} \tag{6.1}$$

where $\mathbf{H}$ is the equivalent channel matrix, $\mathbf{x}$ is the transmitted data vector and $\mathbf{n}$ is the noise vector. The equivalent channel matrix includes the effects of the channel as well as the rotation matrix used during linear precoding. The detection process requires the receiver to estimate the transmitted sequence based on the received sequence. Normally some parts of the transmitted sequence will be known a priori (pilot symbols) and this knowledge will be used to estimate the channel part of $\mathbf{H}$. This estimation will generally have some error. This dissertation, however, does not focus on channel state estimation and will thus assume that $\mathbf{H}$ is known perfectly at the receiver. Detection thus becomes the process of estimating which possible transmitted sequence is most likely, given the received sequence and the channel state information.

6.1 Maximum A Posteriori Probability MIMO Detection

The probability that a sequence was transmitted, given the received sequence and the channel state information, can be expressed and expanded using Bayes' rule as follows:

$$P(\mathbf{x}|\mathbf{y},\mathbf{H}) = P(\mathbf{y}|\mathbf{x},\mathbf{H})P(\mathbf{x}) \tag{6.2}$$

The detection may be performed one symbol at a time. Assume that the data symbols are drawn from the BPSK modulation alphabet. The optimal detection of a single bit requires the calculation of the MAP that the bit equals a 1 (or a 0). This probability is generally expressed

as a log likelihood ratio (LRR). An LLR for a bit, x_i, is defined as follows:

$$LLR(x_i) = \ln\left(\frac{P(x_i = 1)}{P(x_i = 0)}\right) \tag{6.3}$$

The MAP LLR of a bit takes into account all the available information at the receiver. The optimal detection of a bit in an STF symbol for a MIMO system can thus be expressed by calculating the conditional LLR of the bit given the received sequence and the channel state information (CSI). This a posteriori probability in LLR form for the i^{th} bit will be denoted by Λ_i^p:

$$\Lambda_i^p = LLR(x_i|\mathbf{H}, \mathbf{y}) \tag{6.4}$$

$$= \ln\left(\frac{P(x_i = 1|\mathbf{H}, \mathbf{y})}{P(x_i = 0|\mathbf{H}, \mathbf{y})}\right) \tag{6.5}$$

The zero and one signalling may be trivially replaced by 1 and -1 signalling to match the bit to a BPSK symbol. The above equation may be simplified using Bayes rule as follows:

$$\Lambda_i^p = \ln\left(\frac{P(x_i = 1|\mathbf{H}, \mathbf{y})}{P(x_i = -1|\mathbf{H}, \mathbf{y})}\right) \tag{6.6}$$

$$= \ln\left(\frac{\sum_{\mathbf{x}\in X_i^{+1}} P(\mathbf{y}|\mathbf{x}, \mathbf{H})P(\mathbf{x})}{\sum_{\mathbf{x}\in X_i^{-1}} P(\mathbf{y}|\mathbf{x}, \mathbf{H})P(\mathbf{x})}\right), \tag{6.7}$$

where X_i^s is the set of all possible $\mathbf{x}$ vectors for which $x_i = s$. $P(\mathbf{x})$ represents the a priori probability of the sequence $\mathbf{x}$. Since the noise is Gaussian, this expression can be expanded further:

$$P(\mathbf{y}|\mathbf{x}, \mathbf{H}) = \frac{1}{(\pi N_0)^{M_R}} \exp\left(-\frac{||\mathbf{y} - \mathbf{H}\mathbf{x}||^2}{N_0}\right) \tag{6.8}$$

$$\therefore \Lambda_i^p = \ln\left(\frac{\sum_{\mathbf{x}\in X_i^{+1}} \frac{1}{(\pi N_0)^{M_R}} \exp\left(-\frac{||\mathbf{y}-\mathbf{H}\mathbf{x}||^2}{N_0}\right) P(\mathbf{x})}{\sum_{\mathbf{x}\in X_i^{-1}} \frac{1}{(\pi N_0)^{M_R}} \exp\left(-\frac{||\mathbf{y}-\mathbf{H}\mathbf{x}||^2}{N_0}\right) P(\mathbf{x})}\right) \tag{6.9}$$

$$= \ln\left(\frac{\sum_{\mathbf{x}\in X_i^{+1}} \exp\left(-\frac{||\mathbf{y}-\mathbf{H}\mathbf{x}||^2}{N_0} + \ln P(\mathbf{x})\right)}{\sum_{\mathbf{x}\in X_i^{-1}} \exp\left(-\frac{||\mathbf{y}-\mathbf{H}\mathbf{x}||^2}{N_0} + \ln P(\mathbf{x})\right)}\right) \tag{6.10}$$

$$= \ln\left(\sum_{\mathbf{x}\in X_i^{+1}} \exp\left(-\frac{||\mathbf{y} - \mathbf{H}\mathbf{x}||^2}{N_0} + \ln P(\mathbf{x})\right)\right) - \ln\left(\sum_{\mathbf{x}\in X_i^{-1}} \exp\left(-\frac{||\mathbf{y} - \mathbf{H}\mathbf{x}||^2}{N_0} + \ln P(\mathbf{x})\right)\right). \tag{6.11}$$

The calculation of the LLR for bit x_i thus requires the calculation of the probability of every sequence for which $x_i = 1$ and for which $x_i = -1$. This is clearly a mammoth task and

very computationally expensive. The calculation may be simplified by the use of the max-log approximation:

$$\ln\left(\sum_i e^{x_i}\right) \approx \max_i(x_i) \tag{6.12}$$

as follows:

$$\begin{aligned}\Lambda_i^p \approx & \min_{x\in X_i^1}\left\{\frac{\|\mathbf{y}-\mathbf{Hx}\|^2}{N_0} - \ln P(\mathbf{x})\right\} - \\ & \min_{x\in X_i^{-1}}\left\{\frac{\|\mathbf{y}-\mathbf{Hx}\|^2}{N_0} - \ln P(\mathbf{x})\right\}.\end{aligned} \tag{6.13}$$

The a priori information is generally expressed as a vector of LLRs ($\mathbf{\Lambda^a}$). The a priori probability that a bit is equal to 1 and -1 may be calculated from these LLRs as follows:

$$P(x_i=-1) = \frac{1}{1+\exp(\Lambda_i^a)} \tag{6.14}$$

$$P(x_i=1) = \frac{\exp(\Lambda_i^a)}{1+\Lambda_i^a}. \tag{6.15}$$

The a priori probability of a sequence $\mathbf{x}$ may then be calculated as:

$$P(\mathbf{x}) = \prod_{i:x_i=1} P(x_i=1)\prod_{i:x_i=-1} P(x_i=-1). \tag{6.16}$$

Substituting the equations for the probabilities in terms of the LLRs and simplifying yields:

$$P(\mathbf{x}) = \prod_i \frac{\exp\left(\frac{1}{2}(1+x_i)\Lambda_i^a\right)}{1+\exp(\Lambda_i^a)}. \tag{6.17}$$

From this, one may calculate the a priori term in the max-log-map equation:

$$-\ln P(\mathbf{x}) = -\ln\left(\prod_i \frac{\exp\left(\frac{1}{2}(1+x_i)\Lambda_i^a\right)}{1+\exp(\Lambda_i^a)}\right) \tag{6.18}$$

$$= -\sum_i \frac{1}{2}x_i\Lambda_i^a + \sum_i\left(-\frac{1}{2}\Lambda_i^a + \ln\left(1+\exp(\Lambda_i^a)\right)\right) \tag{6.19}$$

$$= -\mathbf{x}^{\mathcal{T}}\mathbf{\Lambda^a} + \sum_i\left(-\frac{1}{2}\Lambda_i^a + \ln\left(1+\exp(\Lambda_i^a)\right)\right). \tag{6.20}$$

Substituting this into equation 6.13 yields:

$$\begin{aligned}\Lambda_i^p \approx & \min_{x\in X_i^1}\left\{\frac{\|\mathbf{y}-\mathbf{Hx}\|^2}{N_0} - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda^a} + \sum_i\left(-\frac{1}{2}\Lambda_i^a + \ln\left(1+\exp(\Lambda_i^a)\right)\right)\right\} - \\ & \min_{x\in X_i^{-1}}\left\{\frac{\|\mathbf{y}-\mathbf{Hx}\|^2}{N_0} - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda^a} + \sum_i\left(-\frac{1}{2}\Lambda_i^a + \ln\left(1+\exp(\Lambda_i^a)\right)\right)\right\}.\end{aligned} \tag{6.21}$$

Considering that the term:

$$\sum_i \left(-\frac{1}{2}\Lambda_i + \ln\left(1+\exp(\Lambda_i)\right)\right) \tag{6.22}$$

does not depend on $\mathbf{x}$, it will cancel out. This yields:

$$\begin{aligned}\Lambda_i^p \approx & \min_{x\in X_i^1}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{N_0} - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}}\right\} - \\ & \min_{x\in X_i^{-1}}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{N_0} - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}}\right\}.\end{aligned} \tag{6.23}$$

The problem has now been reduced to finding the most likely sequence $\mathbf{x}$ for which the bit, x_i, is equal to 1 and the most likely sequence for which the bit x_i is equal to -1. Calculating Λ_i^p for all the bits in $\mathbf{x}$ can be done by first finding the MAP solution ($\hat{\mathbf{x}}$):

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{N_0} - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}}\right\}. \tag{6.24}$$

The MAP solution provides the hard output of the detector. For each bit in the sequence, the MAP detection algorithm is then run again to find a counter-hypothesis. The counter-hypothesis for the i^{th} bit, $\hat{\mathbf{x}}_i'$, is calculated with the bit in question being fixed at the opposite value of the MAP solution ($-1\times x_i$).

$$\hat{\mathbf{x}}_i' = \arg\min_{\mathbf{x}\in\mathbb{X}, x_i=-x_{MAP}}\left\{\frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{N_0} - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}}\right\} \tag{6.25}$$

The MAP solution and the counter-hypothesis are then substituted into equation 6.23 to calculate Λ_i^p:

$$\begin{aligned}\Lambda_i^p \approx & \hat{x}_i\left(\frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{x}}\|^2}{N_0} - \hat{\mathbf{x}}^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}}\right) - \\ & \hat{x}_i\left(\frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{x}}'_i\|^2}{N_0} - \hat{\mathbf{x}}_i'^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}}\right).\end{aligned} \tag{6.26}$$

This process is summarised in Algorithm 2. The MAP minimisation may be done using any MAP detector. The SD is a solution which is well suited to this problem.

6.2 SPHERE DECODER

The LP based SF and STF codes, described in the previous chapter, require a special detector. While orthogonal ST codes can use simple methods such as MRC, no such simplifications exist for the SF and STF codes. Pure ML detection is computationally very expensive as it requires

Algorithm 2 Soft-Output MIMO Detection

1: **procedure** OBTAIN INITIAL HARD SOLUTION

2:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}\in\mathbb{X}} \left\{ \frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{N_0} - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}} \right\}$$

▷

3: **end procedure**

4: **procedure** CALCULATE SOFT OUTPUTS

5: **for** $i = 1 \rightarrow N$ **do**

6: Calculate i^{th} counter hypothesis:

7:

$$\hat{\mathbf{x}}_i' = \arg\min_{\mathbf{x}\in\mathbb{X}, x_i = -x_{MAP}} \left\{ \frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{N_0} - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}} \right\}$$

8: Calculate i^{th} output LLR:

9:

$$\begin{aligned} \Lambda_i^p \approx\ & \hat{x}_i \left(\frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{x}}\|^2}{N_0} - \hat{\mathbf{x}}^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}} \right) - \\ & \hat{x}_i \left(\frac{\|\mathbf{y}-\mathbf{H}\hat{\mathbf{x}'}_i\|^2}{N_0} - \hat{\mathbf{x}}_i'^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}} \right) \end{aligned} \tag{6.27}$$

10: **end for**

11: **end procedure**

the calculation of the ML metric for every possible transmitted vector. This is unfeasible for most realistic systems. One method that may be used to reduce complexity while maintaining error rate performance is the SD. SD will be used to interchangeably mean both sphere decoding and sphere decoder, where the meaning will be clear from the context. The SD is the generic lattice decoder and produces near optimal ML performance [19]. Many different versions of the SD algorithm have been developed with the aim of reducing complexity [62–66]. The lowest complexity SDs are generally accepted to have a complexity in the order $O(n^3)$ [66–68].

6.2.1 Basic operation of the SD

The ML metric which must be minimised by the detection algorithm is:

$$\gamma(\mathbf{x}) = \frac{1}{N_0}\|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}}, \tag{6.28}$$

Initially, consider the case where the a priori information is zero. The transmitted vector may be viewed as a point in m-dimensional space. One may then view the received vector $\mathbf{y}$ as a point in a new m-dimensional space, which is defined by $\mathbf{H}$. Each possible transmitted vector will then be mapped onto a point in the new vector space. The SD works by creating a hypersphere centred at $\mathbf{y}$ with a specific radius C in the new vector space. The SD algorithm then only considers points within the sphere. Once a valid solution within the sphere is found, the radius is reduced to the distance from the received point to the new solution. This procedure is repeated until no new points can be found within the sphere. The closest point found will then be used as the result. If no valid solutions were found in the initial sphere, then the radius is increased. Thus, if a valid solution is found in the initial radius, the algorithm should produce exactly the same result as the exhaustive ML decoder. If a valid solution is not found, then the radius must be increased and the algorithm performed again.

6.2.2 Complexity reduction

The reason that the SD can achieve a complexity reduction over exhaustive ML decoding is that only points within a sphere, around the received point, are considered. The SD criterion requires the distance between the received vector and a possible solution to be less than a specific radius, C. This metric may be expressed as:

$$\mu(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \leq C^2. \tag{6.29}$$

The procedure that the SD uses to search for points in a sphere relies on the ability of the SD to consider one dimension (one symbol) at a time. This is achieved by rewriting equation 6.29 as follows [69]:

$$\mu(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 = (\mathbf{x} - \hat{\mathbf{x}})^{\dagger}\mathbf{H}^{\dagger}\mathbf{H}(\mathbf{x} - \hat{\mathbf{x}}), \tag{6.30}$$

where $\hat{\mathbf{x}} = (\mathbf{H}^{\dagger}\mathbf{H})^{-1}\mathbf{H}^{\dagger}\mathbf{y}$ is the unconstrained ML solution calculated with the zero forcing (ZF) algorithm. The matrix $\mathbf{H}^{\dagger}\mathbf{H}$ is Hermitian and positive definite and thus has a Cholesky

decomposition $\mathbf{H}^{\dagger}\mathbf{H} = \mathbf{L}^{\dagger}\mathbf{L}$ in which $\mathbf{L}$ is a lower triangular matrix with real, positive diagonal entries. Equation 6.30 then becomes:

$$\mu(\mathbf{x}) = (\mathbf{x} - \hat{\mathbf{x}})^{\dagger}\mathbf{L}^{\dagger}\mathbf{L}(\mathbf{x} - \hat{\mathbf{x}}) \tag{6.31}$$

$$= |l_{11}(x_1 - \hat{x}_1)|^2 + |l_{22}(x_2 - \hat{x}_2) + l_{21}(x_1 - \hat{x}_1)|^2 + \cdots + \left| l_{nn}(x_n - \hat{x}_n) + \sum_{j=1}^{n-1} l_{nj}(x_j - \hat{x}_j) \right|^2. \tag{6.32}$$

The first term in equation 6.32 can be calculated using only the first bit. The second term requires only the first two bits. This pattern continues until the n^{th} bit, where n is the number of bits in the symbol. Since each term in equation 6.32 is positive, the metric will grow monotonically. As a result, when the metric exceeds C^2, the algorithm has determined that all the vectors (that start with the currently considered sequence of bits) lie outside the sphere and may be discarded. The SD process can thus be viewed as a tree-searching algorithm. As an example, consider the hypothetical tree search in Figure 6.1. In the figure, only the first 3 bits

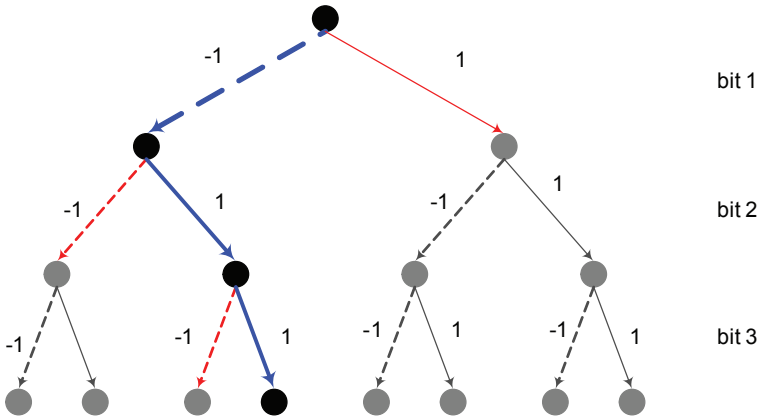

FIGURE 6.1: Diagram of the tree search used in SD.

of the detection are shown. The red lines indicate when the metric associated with a certain bit sequence of bits has grown larger than the radius of the sphere. All the branches in the tree connected to a red branch are shown in grey to indicate that these metrics are never calculated. The blue path shows the sequence of bits that correspond to a symbol inside the sphere. The process starts at the first bit where the metric is calculated for bit 1 = -1. The metric is found to be less than the radius, therefore the process continues. The metric for bit 1 = -1 and bit 2 = -1 is then calculated. This metric is found to be larger than the radius and all paths from bit 2 = -1 are discarded. The metric for bit 1 = -1 and bit 2 = 1 is then calculated. This metric is found to

be inside the radius and thus the process continues until a valid solution (-1,1,1) is found. The radius is now set to the metric for (-1,1,1). The process is repeated, starting at node 1 again. However, all paths from bit 1 = -1 have been calculated. Thus only the branches from bit 1 = 1 need be calculated. However, since the metric from bit 1 = 1 is larger than the new radius, all the paths may be disregarded. The solution has thus been found and only 6 out of a possible 14 branches have been considered, yielding a substantial reduction in complexity compared to the exhaustive search.

The sphere decoder metric in equation 6.29 may be related to the ML metric in equation 6.28 as follows:

$$\gamma(\mathbf{x}) = \frac{1}{N_0}\mu(\mathbf{x}) - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda}^{\mathbf{a}} \tag{6.33}$$

When one assumes that the a priori information is zero (as has been done), then minimising the SD metric will also minimise the ML metric. However, if the algorithm has a priori information available, then modifications need to be made to the SD.

6.2.3 Sphere Decoding with A priori information

The obvious method that one would consider to incorporate a priori information into the SD decoder is to replace the SD metric with the ML metric. The problem with this approach is that a priori information comes in the form of LLRs, which can be both positive and negative. As a result the terms in equation 6.31 will no longer all be positive. Thus, when the metric exceeds C^2, one can no longer assume that the points lie outside the sphere, as the metric may decrease when the next bit is calculated. Overcoming this problem has resulted in two basic approaches. The first relies on a list sphere decoding (LSD) method and the second on translating the the received vector.

6.2.3.1 List Sphere Decoding Method

In the LSD method [70], the SD keeps track of the M best solutions in the tree. In the case where the a priori information is zero, these solutions can then be used in Equation 6.11 or Equation 6.23 to produce soft outputs. The problem with this method is that in order to ensure that a counter-hypothesis exists for each bit (so that the bit's LLR is not infinite), M might have to be very large and the complexity of the LSD algorithm grows exponentially with M.

The extension of this method to include non-zero a priori information is called ITS detection [71]. In ITS detection, the LSD algorithm is run to produce the M best sequences assuming that the a-priori information is zero. The metrics for these sequences are then updated using the a priori information as in Equation 6.28. From this updated metric list the best sequences are then chosen for producing the output. The problem with this method is that unless M is very large, there is a possibility that the best sequence, including the a priori information, was not in the initial list and the MAP solution is thus not found.

6.2.3.2 Translating the received vector

A method which does not suffer from the problems in ITS detection, relies on including the a priori information in the SD metric directly by using a translated vector [72]. The metric, which must be minimised by the SD, is:

$$ML(\mathbf{x}) = \frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{N_0} - \mathbf{x}^{\mathcal{T}}\mathbf{\Lambda}. \tag{6.34}$$

However, the SD is generally able to minimise the metric:

$$ML_{SD}(\mathbf{x}) = \|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2. \tag{6.35}$$

A method must thus be obtained for writing Equation 6.34 in the form of Equation 6.35. Let the translation vector $\tilde{\mathbf{y}}$ be defined as the vector which solves the equation:

$$-\mathbf{\Lambda} = -\frac{2\mathbf{H}^{\mathcal{T}}\tilde{\mathbf{y}}}{N_0}. \tag{6.36}$$

Equation 6.34 then becomes:

$$ML(\mathbf{x}) = \frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2}{N_0} - \frac{2\mathbf{x}^{\mathcal{T}}\mathbf{H}^{\mathcal{T}}\tilde{\mathbf{y}}}{N_0}. \tag{6.37}$$

Since the decoding is a minimisation one can add any term that does not vary with $\mathbf{x}$. Add the constant term:

$$\frac{\tilde{\mathbf{y}}^{\mathcal{T}}\tilde{\mathbf{y}} + 2\mathbf{y}^{\mathcal{T}}\tilde{\mathbf{y}}}{N_0} \tag{6.38}$$

to Equation 6.37:

$$ML(\mathbf{x}) = \frac{\|\mathbf{y}-\mathbf{H}\mathbf{x}\|^2 - 2\mathbf{x}^{\mathcal{T}}\mathbf{H}^{\mathcal{T}}\tilde{\mathbf{y}} + \tilde{\mathbf{y}}^{\mathcal{T}}\tilde{\mathbf{y}} + 2\mathbf{y}^{\mathcal{T}}\tilde{\mathbf{y}}}{N_0} \tag{6.39}$$

$$= \frac{\tilde{\mathbf{y}}^{\mathcal{T}}\tilde{\mathbf{y}} + 2\mathbf{y}^{\mathcal{T}}\tilde{\mathbf{y}} + \mathbf{y}^{\mathcal{T}}\mathbf{y} + \mathbf{x}^{\mathcal{T}}\mathbf{H}^{\mathcal{T}}\mathbf{H}\mathbf{x} - 2\mathbf{x}^{\mathcal{T}}\mathbf{H}^{\mathcal{T}}\mathbf{y} - 2\mathbf{x}^{\mathcal{T}}\mathbf{H}^{\mathcal{T}}\tilde{\mathbf{y}}}{N_0} \tag{6.40}$$

$$= \frac{\|\mathbf{y}+\tilde{\mathbf{y}}-\mathbf{H}\mathbf{x}\|^2}{N_0}. \tag{6.41}$$

The translated received vector $\hat{\mathbf{y}}$ may now be defined as:

$$\hat{\mathbf{y}} = \mathbf{y} + \tilde{\mathbf{y}}. \tag{6.42}$$

The ML metric becomes:

$$ML(\mathbf{x}) = \frac{\|\hat{\mathbf{y}} - \mathbf{H}\mathbf{x}\|^2}{N_0}, \tag{6.43}$$

which is in the form of Equation 6.35. Thus in order to include a priori information for the SD, Equation 6.36 is solved for the translation vector. From this vector the translated received vector is calculated using Equation 6.42. SD is then performed on Equation 6.43.

6.3 HOPFIELD NETWORK

Hopfield networks consist of interconnected neurons [73]. The neurons compute an output $v_i(t)$ from the input signal $u_i(t)$ as follows:

$$v_i(t) = g[\beta u_i(t)] \tag{6.44}$$

where i is the indice of the neuron, β is the gain and $g(x)$ is a smooth monotonically decreasing sigmoid function with the following requirements:

$$\begin{aligned} g(-\infty) &= 0 \\ g(0) &= \frac{1}{2} \\ g(\infty) &= 1. \end{aligned} \tag{6.45}$$

The signal $u_i(t)$ obeys the following equation:

$$\frac{du_i(t)}{dt} = -\frac{u_i(t)}{\tau} + \sum_{j=1}^{N} T_{ij} v_j(t) + I_i \tag{6.46}$$

where τ is a constant that can be set equal to unity, N is the number of neurons in the network and I_i is a constant bias added to the input of a neuron. T_{ij} are elements of a symmetric connectivity matrix that are zero when $i = j$, as each neuron is connected to all the other neurons but not to itself. Thus:

$$\begin{aligned} T_{ij} &= T_{ji} \\ T_{ii} &= 0 \ \forall \ i. \end{aligned} \tag{6.47}$$

Hopfield showed that when Equation 6.47 is met and the neurons are operating in a high gain mode (β is large) the stable states of the network are the local minima of the computational energy of the network (E):

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}v_i v_j - \sum_{i=1}^{N} v_i I_i, \tag{6.48}$$

which may be written in matrix form as:

$$E = -\frac{1}{2}\mathbf{v}^T\mathbf{T}\mathbf{v} - \mathbf{v}^T\mathbf{I}. \tag{6.49}$$

Using a Hopfield network to calculate a minimisation of some kind, such as ML detection, is done by writing the equation to be minimised in such a way as to link the variables in the equation with terms in Equation 6.49. In this way the local minima of the computational energy of the network will match the local minima of the ML equation [74, 75].

6.3.1 Derivation for use of Hopfield network in MIMO detection.

For ML detection, the transmitted vector $\mathbf{x}$ must be found which minimises the following equation:

$$ML(\mathbf{x}) = \|\mathbf{y} - \mathbf{H}\mathbf{x}\|^2 \tag{6.50}$$

$$= (\mathbf{y} - \mathbf{H}\mathbf{x})^{\mathcal{T}}(\mathbf{y} - \mathbf{H}\mathbf{x}) \tag{6.51}$$

This equation can be expanded as follows:

$$ML(\mathbf{x}) = \mathbf{y}^{\mathcal{T}}\mathbf{y} + \mathbf{x}^{\mathcal{T}}\mathbf{H}^{\mathcal{T}}\mathbf{H}\mathbf{x} - 2\mathbf{x}^{\mathcal{T}}\mathbf{H}^{\mathcal{T}}\mathbf{y} \tag{6.52}$$

The first term in the above equation is a constant for all possible transmitted sequences and may thus be removed from the ML equation. In order to match the ML equation to Equation 6.49, the $\mathbf{H}$ matrix will be expanded with summations to yield the following:

$$ML(\mathbf{x}) = -2\sum_{i=0}^{N} x_i\mathbf{H}_\mathbf{i}^{\mathcal{T}}\mathbf{y} + \sum_{i=0}^{N}\sum_{j=0}^{N}\mathbf{H}_\mathbf{i}^{\mathcal{T}}\mathbf{H}_\mathbf{j}x_i x_j \tag{6.53}$$

where $\mathbf{H_i}$ represents the i^{th} column of $\mathbf{H}$, x_i represents the i^{th} transmitted symbol and N represents the number of transmitted symbols. For BPSK type modulation, the values of $\mathbf{x}$ are either 1 or -1. Since the aim is to map the output of the neurons to the transmitted bits, the

definition for the non-linear function defined in 6.45 is changed as follows:

$$\begin{aligned} g(-\infty) &= -1 \\ g(0) &= 0 \\ g(\infty) &= 1. \end{aligned} \tag{6.54}$$

One example of such a function is the hyperbolic tan function. This requirement allows one to add the following term to the ML equation:

$$-\sum_{i=0}^{N} \mathbf{H_i^{\mathcal{T}}} \mathbf{H_i}(x_i^2 - 1). \tag{6.55}$$

This term is created by finding a polynomial with roots equal to -1 and 1. Since x_i is always 1 or -1 this term will thus always be zero and will not affect the ML equation. The $i = j$ case is also separated from the double summation term in Equation 6.53. The equation is then simplified as follows:

$$\begin{aligned} ML(\mathbf{x}) &= -2\sum_{i=0}^{N} x_i \mathbf{H_i^{\mathcal{T}}} \mathbf{y} + \sum_{i=0}^{N}\sum_{\substack{j=0 \\ j\neq i}}^{N} \mathbf{H_i^{\mathcal{T}}} \mathbf{H_j} x_i x_j - \sum_{i=0}^{N} \mathbf{H_i^{\mathcal{T}}} \mathbf{H_i}(x_i^2 - 1) \qquad (6.56) \\ &\quad + \sum_{i=0}^{N} \mathbf{H_i^{\mathcal{T}}} \mathbf{H_i} x_i^2 \\ &= -2\sum_{i=0}^{N} x_i \mathbf{H_i^{\mathcal{T}}} \mathbf{y} + \sum_{i=0}^{N}\sum_{\substack{j=0 \\ j\neq i}}^{N} \mathbf{H_i^{\mathcal{T}}} \mathbf{H_j} x_i x_j - \sum_{i=0}^{N} \mathbf{H_i^{\mathcal{T}}} \mathbf{H_i} x_i^2 + \sum_{i=0}^{N} \mathbf{H_i^{\mathcal{T}}} \mathbf{H_i} \qquad (6.57) \\ &\quad + \sum_{i=0}^{N} \mathbf{H_i^{\mathcal{T}}} \mathbf{H_i} x_i^2. \end{aligned}$$

The third and fifth terms in Equation 6.57 cancel and the fourth term is a constant for all possible $\mathbf{x}$ and can thus be removed. The final ML equation is thus :

$$ML(\mathbf{x}) = \sum_{i=0}^{N}\sum_{\substack{j=0 \\ j\neq i}}^{N} \mathbf{H_i^{\mathcal{T}}} \mathbf{H_i} x_i x_j - 2\sum_{i=0}^{N} x_i \mathbf{H_i^{\mathcal{T}}} \mathbf{y}. \tag{6.58}$$

One can now match the parameters in the ML equation (6.58) to those in the energy equation (6.49). This is done by defining the following:

$$T_{ij} = \begin{cases} -2\mathbf{H_i^{\mathcal{T}}} \mathbf{H_j} & if \;\; i \neq j \\ 0 & if \;\; i = j \end{cases} \tag{6.59}$$

$$\mathbf{I} = 2\mathbf{H}^{\mathcal{T}} \mathbf{y}. \tag{6.60}$$

This definition results in a symmetric $\mathbf{T}$ matrix where $T_{ij} = T_{ji}$. Substituting these definitions into equation 6.58 yields the following:

$$ML(\mathbf{x}) = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}x_i x_j - \sum_{i=1}^{N} x_i I_i \tag{6.61}$$

$$= -\frac{1}{2}\mathbf{x}^T\mathbf{T}\mathbf{x} - \mathbf{x}^T\mathbf{I}, \tag{6.62}$$

which is exactly the same as the energy equation when $\mathbf{x} = \mathbf{v}$. Thus, once the Hopfield network has converged, the transmitted data sequence can be read as the output of the neurons. The equation describing the convergence of the network (Equation 6.46) is in continuous time. A discrete time version for use in a digital system is given below:

$$u_i[t+1] = \mathbf{T}_i\mathbf{v}[t] + I_i \tag{6.63}$$

$$v_i[t+1] = g(\beta u_i[t+1]), \tag{6.64}$$

where $\mathbf{T}_i$ represents the i^{th} row of $\mathbf{T}$. The Hopfield network will thus run several iterations of Equations 6.63 and 6.64 until the values of $\mathbf{v}$ have converged. A diagram showing the Hopfield network can be seen in Figure 6.2. While this derivation was done for BPSK, the

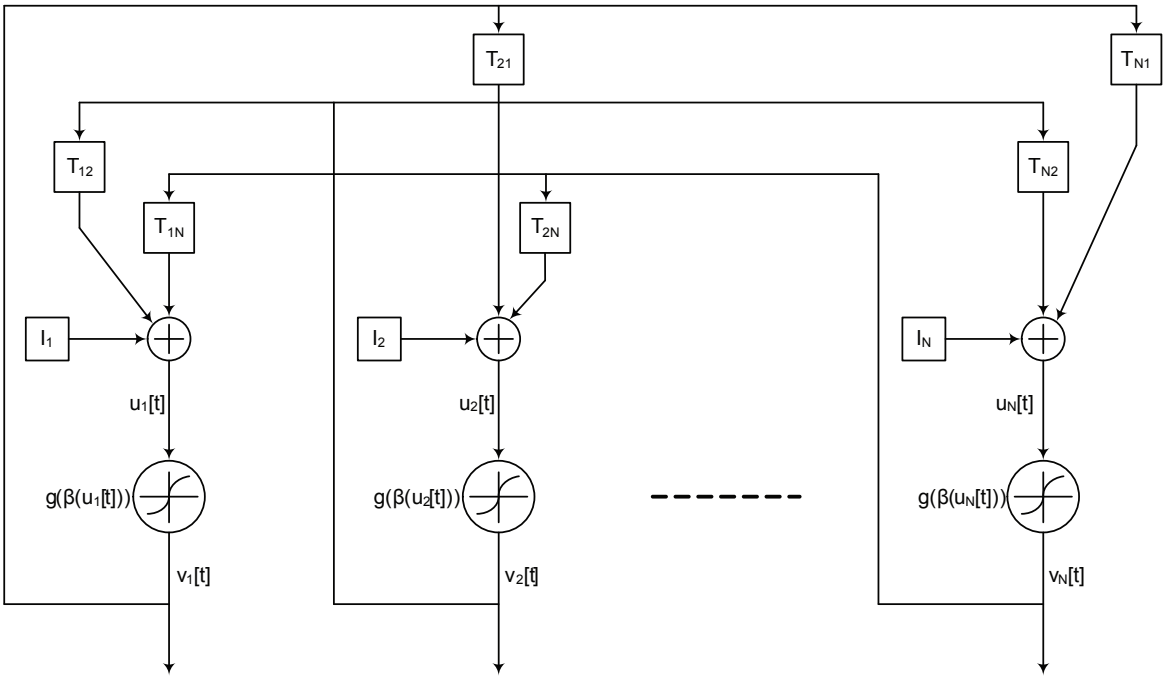

FIGURE 6.2: Diagram of Hopfield network.

same equations hold for QPSK if a complex-to-real conversion is performed. Higher order

modulation techniques can also be used with the method described in [72]. The process to adapt the Hopfield network to use a-priori information is exactly the same as for the SD.

6.3.2 Heuristic improvements to the Hopfield network

The performance of the Hopfield network may be improved by several heuristic methods. The first considers the update procedure of the network. In equation 6.63 all the neurons are updated at the same time (batch update). Better performance can be achieved using sequential update where the neurons are updated one at a time and the new outputs are used for updates of the other neurons. Thus, equation 6.63 is calculated for i from 1 to N and $v_j[t+1]$ is used instead of $v_j[t]$ if $j < i$. The second heuristic allows for the inclusion of a momentum term. Normally $\tau = 1$ (see equation 6.46) which means that $u_i[t]$ does not affect $u_i[t+1]$ directly. A momentum term is added by making τ nonzero and results in the $u_i[t+1]$ depending on $u_i[t]$. This changes Equation 6.63 to:

$$u_i[t+1] = \alpha u[t] + \mathbf{T}_i\mathbf{v}[\mathbf{t}] + \gamma I_i, \tag{6.65}$$

where $\alpha = \left(1 - \frac{1}{\tau}\right)$ and γ are scaling terms which may be varied to improve performance [76]. The third heuristic used to improve performance is called annealing. Annealing has been shown to improve the performance of many kinds of network-based optimisation techniques and specifically Hopfield networks [77, 78]. In an annealing procedure, variable values (scaling factors) are changed from one iteration to the next. The most important parameter to anneal is the gain of the neuron (β). Initially, the gain is set low to allow the neurons to stay in a less determined zone (and to keep the algorithm from getting stuck in a local minima). The gain is then increased with each iteration to force the algorithm to converge. Annealing may also be used on the other scaling factors introduced above. The values of the parameters as used in this dissertation and the annealing procedure are:

$$\gamma[t] = 0.672 + 0.2797 \exp(0.0575t - 0.426) \tag{6.66}$$

$$\beta[t] = 0.0267 + 0.1363 \exp(0.0442t - 1.86) \tag{6.67}$$

$$\alpha[t] = -0.137 + 0.369 \exp(0.131t - 1.863), \tag{6.68}$$

with t being the iteration number. The parameters were obtained and optimised by using a genetic algorithm to minimise the metric:

$$\|\mathbf{y} - \mathbf{H}\tilde{\mathbf{x}}\|, \tag{6.69}$$

where $\tilde{\mathbf{x}}$ is $\mathbf{x}$ with bit $\tilde{x}_i = -x_i$. The optimisation was done in this way, because it closely resembles the required functioning of the Hopfield network in the combined SD-Hopfield detector, which will be explained in the next section. The final update equations are thus:

$$u_i[t+1] = \alpha[t]u[t] + \mathbf{T}_i\mathbf{v}[\mathbf{t}] + \gamma[t]I_i \tag{6.70}$$

$$v_i[t+1] = g(\beta[t]u_i[t+1]). \tag{6.71}$$

The diagram of the final Hopfield network can be seen in Figure 6.3. The iterative operation of the Hopfield network requires the output of the network ($\mathbf{v}$) to be initialised at some value. If nothing is known about the sequence being detected, then the values may be initialised to random values. If, however, information is available, initialising the output close to the correct output will improve the performance and speed up the convergence of the network.

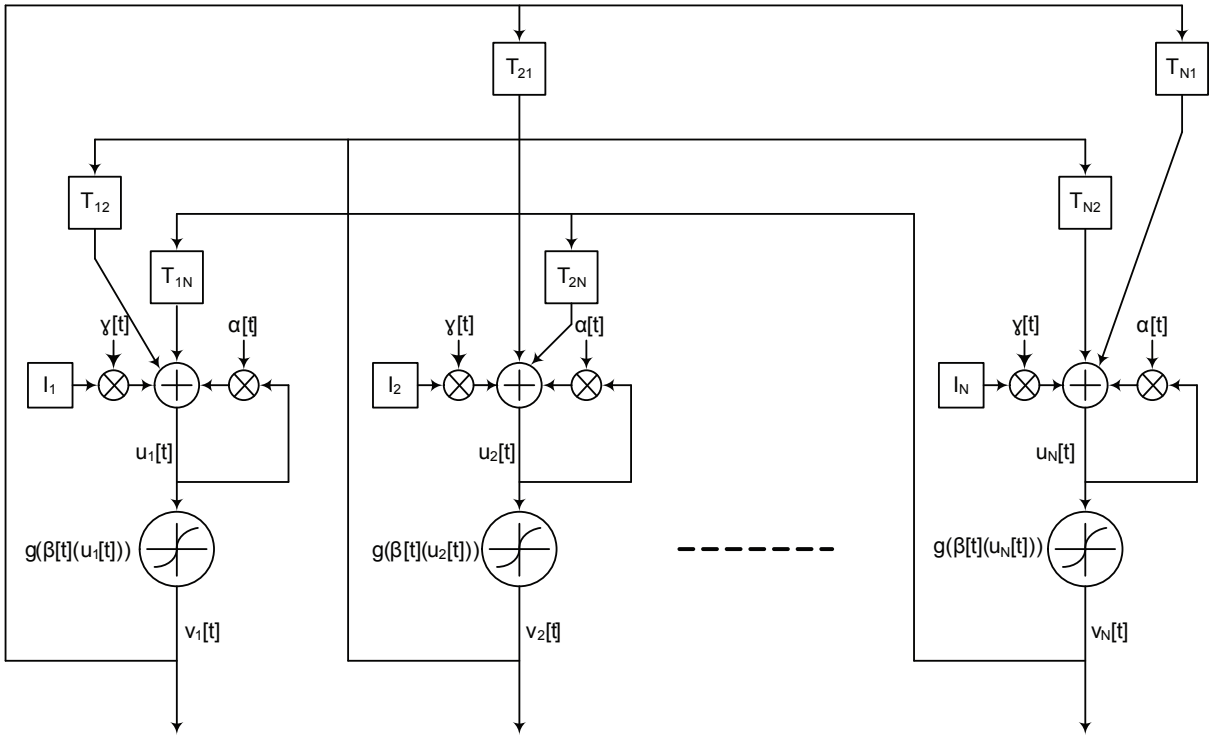

FIGURE 6.3: Diagram of complete Hopfield network including heuristic improvements.

6.4 SD-HOPFIELD DETECTOR

Using the Hopfield network, a reduced complexity soft-output (SO) detector is constructed. Considering Equation 6.23, it can be seen that $N + 1$ ML detection processes are required to produce the soft outputs. In the optimal SO-SD detector, the SD is used for all $N + 1$ detections.

In the SD-Hopfield detector, the SD is used for the first detection, as this will produce a near optimal hard output. The Hopfield network is then used to create the N counter-hypotheses. For the creation of the i^{th} counter-hypothesis, the Hopfield network is initialised at the ML solution ($\hat{\mathbf{x}}$) with the i^{th} bit set to $-\hat{x}_i$. Since the Hopfield network is less complex than the SD, a complexity reduction is achieved. The performance of the system may be improved by employing an iterative structure.

6.4.1 SD-Hopfield turbo detector

The Hopfield network is a suboptimal detector. An iterative joint detector and decoder (turbo) structure can be used to improve the performance of the system. In this itertive system the SD-Hopfield detector is connected in a turbo structure with the decoder of an FEC code. In principle, any FEC code could be used in the turbo structure with the SD-Hopfield decoder. In this dissertation NB-QC-LDPC codes are used. The decoder used is the FFT-BP algorithm [5, 32, 79]. A block diagram of the SD-Hopfield turbo receiver can be seen in Figure 6.4. The

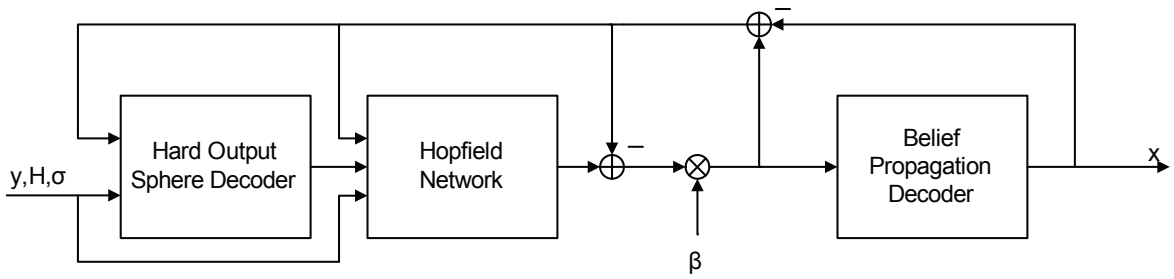

FIGURE 6.4: SD-Hopfield turbo receiver

turbo process works by passing extrinsic information generated by one decoder to the other decoder to be used as a-priori information. In the first iteration the SD decodes the received signal and assumes that the a-priori information is zero. The SD then passes the MAP solution to the Hopfield network, which calculates a counter hypothesis solution for each bit. Using these counter-hypotheses in the max-log-map equation, given in 6.23, produces soft outputs, which are passed to the FFT-BP decoder. The FFT-BP decoder runs a set number of decoding iterations. Extrinsic information is calculated from the output of the FFT-BP decoder, which is then passed back to the SD. The SD then re-decodes the received information using the extrinsic information as a-priori information. The resultant soft outputs then represent the a-posteriori information. From this information the extrinsic information is calculated and passed to the

FFT-BP decoder. This iterative process may be repeated as often as required. The extrinsic information ($\mathbf{\Lambda}^e$) is calculated as:

$$\mathbf{\Lambda}^e = \mathbf{\Lambda}^p - \mathbf{\Lambda}^a. \tag{6.72}$$

Since the soft MIMO detector is sub-optimal, the extrinsic information coming from the Hopfield network will tend to be over optimistic (the LLR values are too large). A simple method to counter this and improve the performance is to introduce a scaling factor. This has been shown to work in many turbo-coded systems. An analysis to find a good scaling factor was done. For this analysis, the SF rate-2 code given in Equation 5.40 was used with the NB-QC-LDPC code, C6, developed in Section 3.4. The result can be seen in Figure 6.5 where the E_b/N_0 required to achieve a BER of 10^{-4} is plotted against different scaling factors. From the figure, one can see that a scaling factor of 0.4 yields the best results. Introducing the scaling factor improves the performance by almost $1dB$.

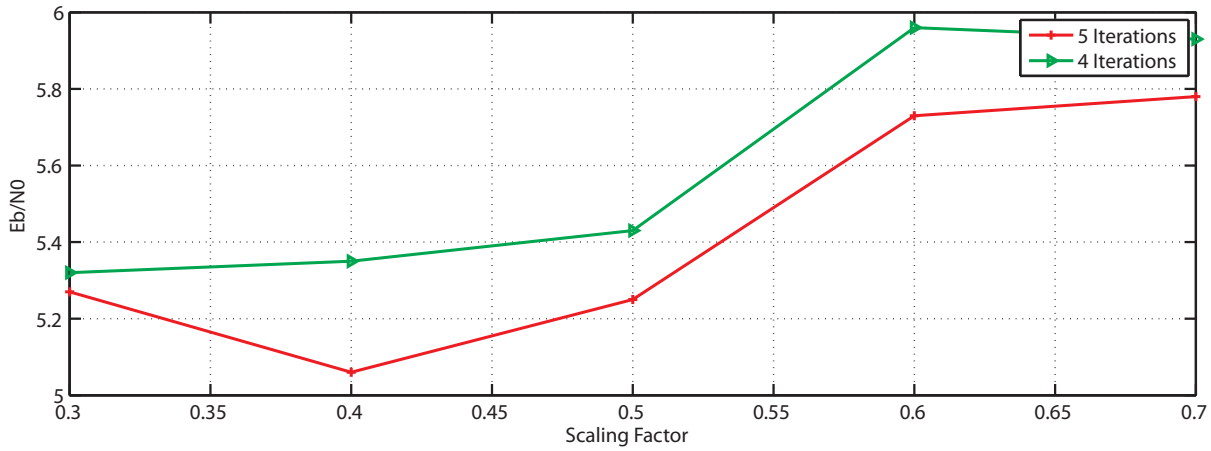

FIGURE 6.5: SNR required to achieve a BER of 10^{-4} for different scaling factors.

6.4.2 Complexity analysis

The complexity of the SD is difficult to provide analytically since it is not always the same and depends greatly on the heuristic methods used to speed up the tree-pruning process. The results provided in literature places the complexity between $O(n^6)$ and $O(n^3)$ in the best case scenarios. The complexity of the Hopfield network is, however, very easy to calculate. The complexity calculation can be broken up into two parts: the number of operations required for

the initialisation process and the number of calculations required per iteration of the network. For this analysis each multiplication or addition will be considered as a floating point operation (flop). The calculation of a transcendental function will be kept separate and referred to as a tflop. These may be calculated using look-up tables.

6.4.2.1 Initialisation Phase

Setting up the connection weights requires the calculation of Equations 6.59 and 6.60. This corresponds to one matrix-by-matrix multiplication and one vector-by-matrix multiplication. This complexity is given in Table 6.1. It should be noted that in the case of the SD-Hopfield

TABLE 6.1: Hopfield Network Initialisation Complexity

Initialisation Complexity		
Description	Operations	Units
Matrix × Matrix	n^3	flop
Matrix × Vector	$2n^2$	flops
Total	$n^3 + 2n^2$	flops

detector, the equations required for the initialisation of the Hopfield network will already have been performed by the SD.

6.4.2.2 Iteration

Each Hopfield iteration requires the calculation of Equation 6.65. This requires: one matrix-by-vector multiplication, two scalar-by-vector multiplications and two vector additions. The annealing of the parameters requires: three transcendental calculations, six multiplications and six additions. An iteration also requires the calculation of the neuron function. This corresponds to one scalar-by-vector multiplication and one vector transcendental calculation. The number of flops required per iteration will be denoted by C_{fiter} and the number of tflops required per iteration will be denoted by C_{titer}. The complexity is given in Table 6.2.

6.4.2.3 Complexity comparison

For the pure SD detection, 50 iterations of the FFT-BP algorithm were run. For the SD-Hopfield detector 10 FFT-BP iterations were run per turbo iteration, resulting in a total of 50 FFT-BP

TABLE 6.2: Hopfield Network Iteration Complexity

Complexity per Iteration		
Description	Operations	Units
Matrix × Vector	$2n^2$	flops
3 - Scalar × Vector	$3n$	flops
2 - Vector + Vector	$2n$	flops
Vector Transcendental	n	tflops
Annealing	3	tflops
Annealing	12	flops
C_{fiter}	$2n^2 + 5n + 12$	flops
C_{titer}	$n + 3$	tflops

iterations. The complexity of the FFT-BP decoding is thus not considered in this analysis. The total complexity of the SD-Hopfield detector can be expressed as:

$$C_{Total} = i_o(C(\text{SD}) + i_i(C_{fiter} + C_{titer}) \quad (6.73)$$

$$= i_o C(\text{SD}) + i_o i_i(C_{fiter} + C_{titer}), \quad (6.74)$$

where i_o is the number of turbo iterations and i_i is the number of internal Hopfield iterations. $C(\text{SD})$ represents the complexity of an SD iteration. Notice that the initialisation complexity of the Hopfield detector has been omitted as the required equations will have been calculated by the SD. The complexity of the optimal SD detector can be represented as:

$$C_{SSD} = (n + 1)C(\text{SD}). \quad (6.75)$$

The specific complexity gain will thus depend on the number of iterations and n. In this dissertation $i_i = 10$, $i_o = 5$ were chosen. The complexity gain can then be expressed as:

$$C_{gain} = C_{SSD} - C_{Total} \quad (6.76)$$

$$= (n + 1)C(\text{SD}) - 5C(\text{SD}) - 50(C_{fiter} \quad (6.77)$$

$$+C_{titer})$$

$$= (n - 4)C(\text{SD}) - 50(C_{fiter} + C_{titer}). \quad (6.78)$$

Considering that the complexity of the SD is cubic in n and the complexity of the Hopfield network is square in n, the gain is heavily dependent on n. For the example of the rate-2 SF code, $n = 16$ and the run-time for the SD-Hopfield detector was approximately five times faster than the optimal SO-SD detector.

6.5 SIMULATION RESULTS

The simulations were performed on the MIMO-WiMAX channel simulator developed in [56]. The sphere decoder used in the simulations was taken from [66]. The simulation parameters can be found in Table 6.3. Simulations were performed on an ideal two-tap channel and a more realistic suburban alternative channel. The suburban alternative PDP in Table 6.3 can be found, and was applied to the simulator in [27]. Whilst the FFT size used is 128, only 48 carriers are used in these simulations, as that corresponds to one WiMAX user. Twelve SF frequency symbols are grouped together and interleaved to form one 48-carrier grouping for a user. This grouping is then combined with a second user, pilot and guard bands to form the 128-carrier OFDM symbol.

TABLE 6.3: MIMO-WiMAX simulation parameters

MIMO-OFDM parameters	
Transmit antennas	2
Receive antennas	2
FFT size	128
Number of sub-channels	2
Users per sub-channel	1
Mode	FUSC
Cyclic prefix length	0.25
Maximum Doppler spread	$f_d = 100Hz$
Sampling time	$T_s = 0.8\mu$s
Channel bandwidth	1.25MHz
Transmit filter	Square root raised cosine, $\alpha = 0.5$
Receive filter	Square root raised cosine, $\alpha = 0.5$
Ideal channel parameters	
Frequency-Selectivity	Two-ray equal power PDP at 0 μs and 8 μs
Time-Selectivity	Slow-fading conditions with $f_d = 100Hz$
Space-Selectivity	None
Realistic channel parameters	
Frequency-Selectivity	Suburban-alternative, 20 tap PDP
Time-Selectivity	Slow-fading conditions with $f_d = 100Hz$

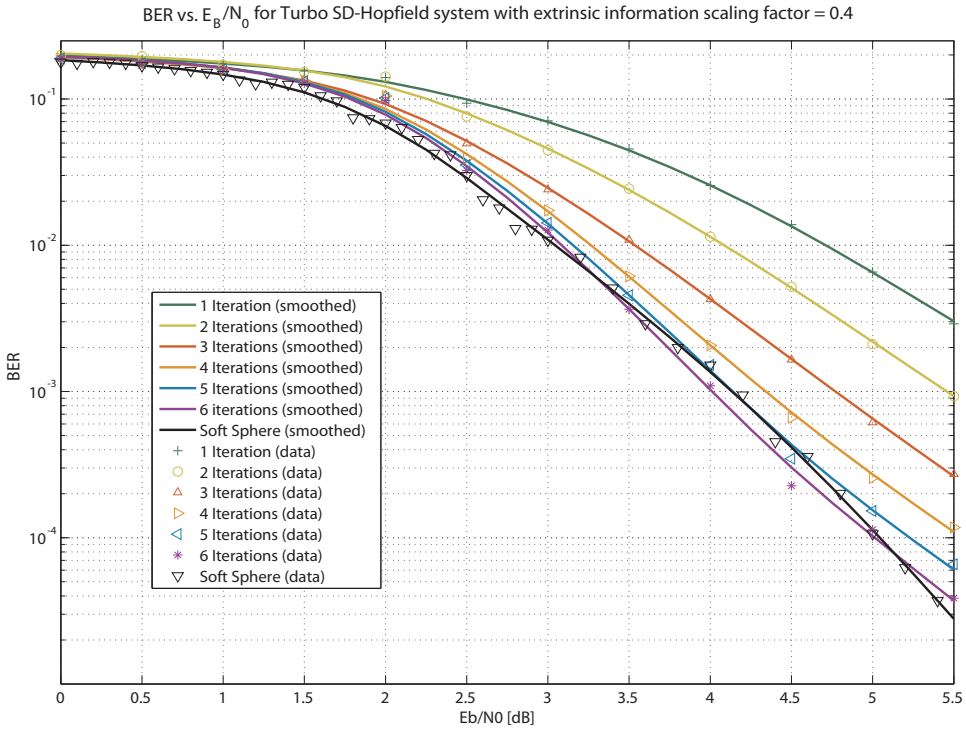

FIGURE 6.6: Performance of SD-Hopfield detector on the rate-2 SF code with the NB-QC-LDPC code on a two tap channel.

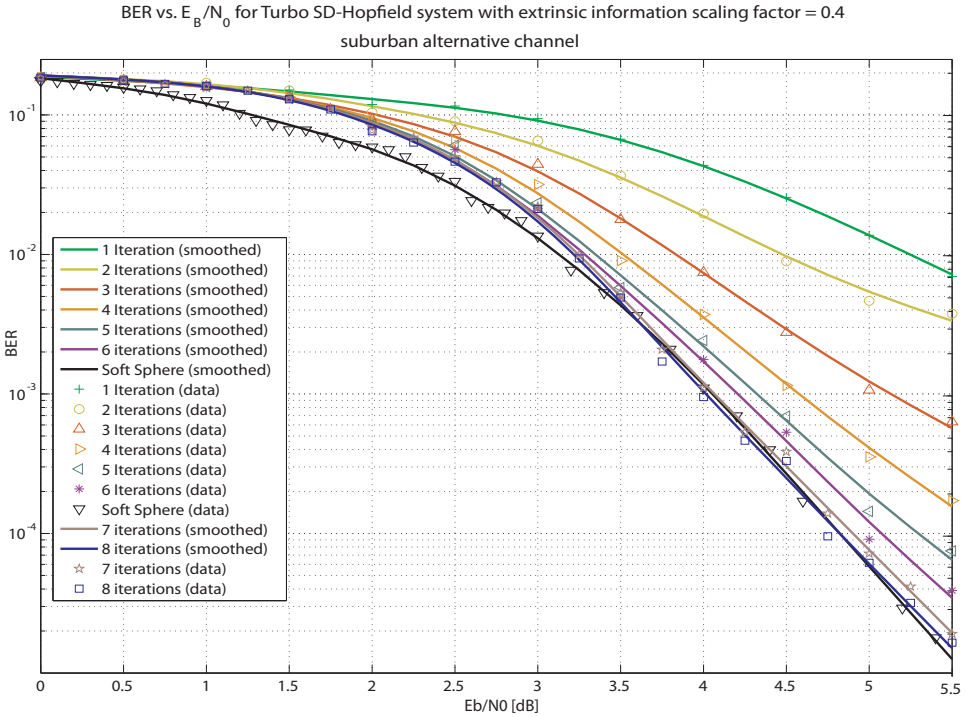

FIGURE 6.7: Performance of SD-Hopfield detector on the rate-2 SF code with the NB-QC-LDPC code on the suburban alternative channel.

From the results in figure 6.6 one can see that the performance of the SD-Hopfield detector is approximately the same as that of the optimal SO-SD detector. The performance of the SD-Hopfield detector improves with each iteration, but from the figure it can be seen that 5 iterations are sufficient to match the performance. Running more iterations allows the SD-Hopfield to beat the pure SD detector. From Figure 6.7 one can see that the SD-Hopfield detector still performs well in realistic channel conditions, although more iterations are required to equal the performance of the optimal SO-SD detector exactly.

CHAPTER SEVEN

THE TURBO NB-QC-LDPC CODED STF-MIMO-OFDM SYSTEM

In this chapter, the building blocks discussed and developed in the previous sections are combined to form a complete system. The system combines the NB-QC-LDPC code with the universal STF codes on a MIMO-OFDM platform. At the receiver an iterative turbo decoder structure is used. The system will thus be called: Turbo NB-QC-LDPC coded STFC-MIMO-OFDM. This chapter will start with a description of the transmitter and the receiver. The performance of the system will then be analysed.

7.1 TRANSMITTER

Figure 7.1 shows a block diagram of the transmitter. The data from the source (**d**) is first encoded using the NB-QC-LDPC code. The encoded data (**e**) is then sent to the STF encoder, which performs LP and layering to produce the codeblock (**C**). This block is then demultiplexed and all the symbol vectors (**x**) sent to the correct transmit antennas. The OFDM symbols are then created (**c**) and transmitted.

7.2 RECEIVER

Figure 7.2 shows a block diagram of the receiver. The signals received at each antenna (**r**) are OFDM demodulated and multiplexed to create a sequence of received symbols (**Y**), which is then sent to the SD. The SD then decodes the received signal assuming no a priori information

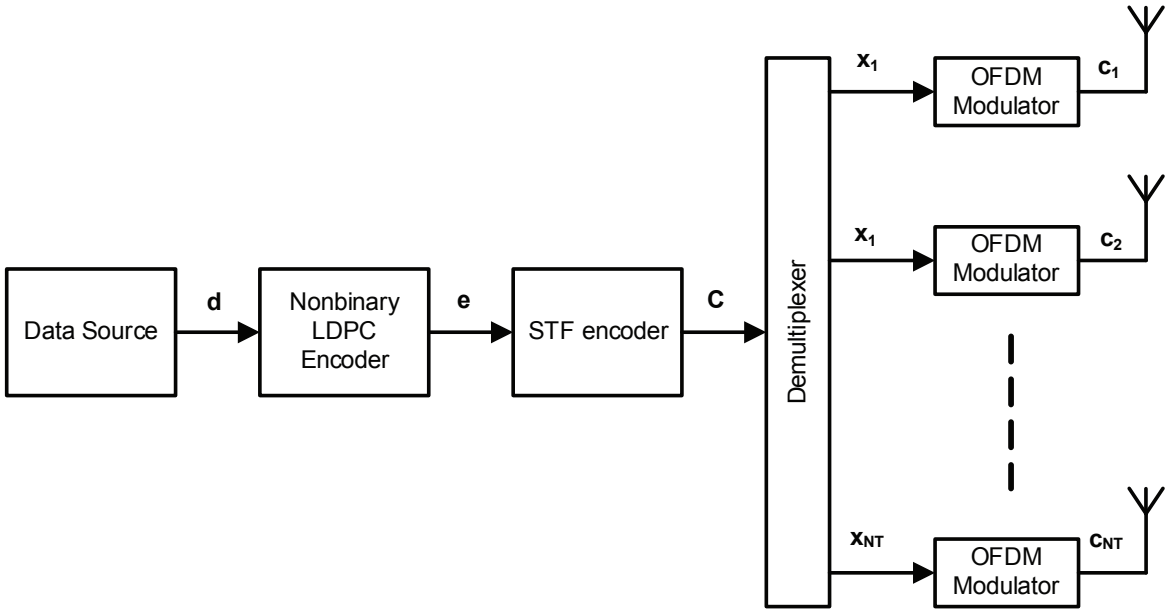

FIGURE 7.1: A block diagram of the transmitter of the Turbo NB-QC-LDPC coded STF-MIMO-OFDM system.

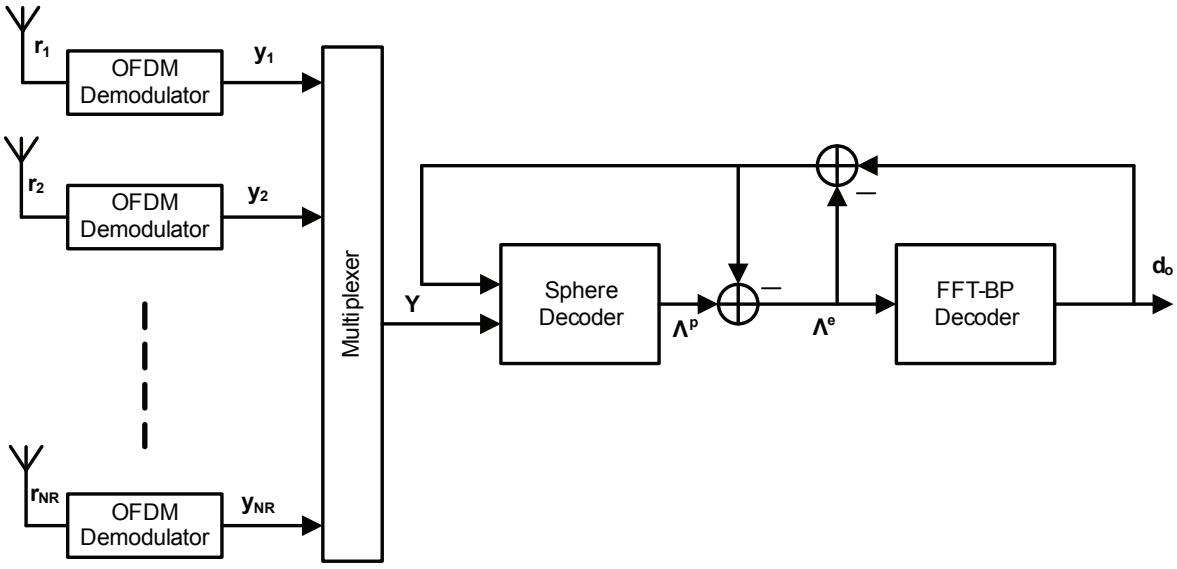

FIGURE 7.2: A block diagram of the receiver of the Turbo NB-QC-LDPC coded STF-MIMO-OFDM system.

and then sends the soft outputs ($\mathbf{\Lambda^p}$) to the FFT-BP decoder. The FFT-BP waits for the entire LDPC codeword to arrive and then decodes the received signal. From the soft output of the FFT-BP decoder, the extrinsic information is calculated and passed back to the SD. Extrinsic

information is calculated in the same manner was described in Chapter 6. The SD then uses the extrinsic information from the BP decoder as a priori information and decodes the received signal again. From the output of the SD the extrinsic information of the SD decoding ($\mathbf{\Lambda^e}$) is extracted and sent to the FFT-BP decoder to be decoded again. The SD and the FFT-BP decoders repeatedly pass extrinsic information back and forth for a set number of iterations, at which time the soft output of the FFT-BP decoder is hard limited and used as the output of the system ($\mathbf{d_o}$).

7.3 SYSTEM ANALYSIS

Using simulations, the performance of the system is analysed. The following aspects of the system are analysed:

1. The effect of the number of turbo iterations and the number of internal FFT-BP decoder iterations on the BER performance.
2. The effect of channel correlation on the BER performance of the system.
3. The effect of quantisation at the receiver on the BER performance of the system.
4. The comparative performance of SF and STF codes in correlated and quantised channel conditions.

The simulations were performed in both an ideal two tap as well as a more realistic 20-tap suburban alternative channel.

7.3.1 Simulation Results and Discussion

The simulations were performed on the channel simulator developed in [56]. The SD used in the simulations was taken from [66]. The simulation parameters can be found in Table 7.1. Simulations were performed on an ideal two tap channel and a more realistic suburban alternative channel. The suburban alternative PDP in Table 7.1 can be found, and was applied to the simulator in [56]. While the FFT size used is 128, of this only 48 carriers are used for the simulations in this dissertation, as that number corresponds to one user. Twelve SF frequency symbols are grouped together and interleaved to form one 48-carrier grouping for a user. This grouping is then combined with a second user, pilot and guard bands to form the 128 carrier OFDM symbol. In this section QPSK was used in all simulations.

TABLE 7.1: MIMO-WiMAX simulation parameters

MIMO-OFDM parameters	
Transmit antennas	2
Receive antennas	2
FFT size	128
Number of sub-channels	2
Users per sub-channel	1
Mode	FUSC
Cyclic prefix length	0.25
Maximum Doppler spread	$f_d = 100Hz$
Sampling time	$T_s = 0.8\mu$s
Channel bandwidth	1.25MHz
Transmit filter	Square root raised cosine, $\alpha = 0.5$
Receive filter	Square root raised cosine, $\alpha = 0.5$
Ideal channel parameters	
Frequency-Selectivity	Two-ray equal power PDP at 0 μs and 8 μs
Time-Selectivity	Slow-fading conditions with $f_d = 100Hz$
Space-Selectivity	None
Realistic channel parameters	
Frequency-Selectivity	Suburban-alternative, 20 tap PDP
Time-Selectivity	Slow-fading conditions with $f_d = 100Hz$

7.4 TURBO SYSTEM RESULTS

7.4.1 Two tap Channel

7.4.1.1 Turbo NB-QC-LDPC coded STF-MIMO-OFDM system, iteration study, no correlation

In this section, the performance of the system with regard to the number of internal FFT-BP and external turbo iterations is investigated. The effect of the number of FFT-BP iterations will be considered first. Figures 7.3, 7.4 and 7.5 show the performance of the rate-1 STF code over a range of turbo iterations for 10, 20 and 30 FFT-BP iterations per turbo iteration, respectively. The simulations were performed on the two tap channel with no correlation in

order to isolate the effects of the number of iterations on the performance of the system, from other complicated channel effects. From these simulations, the E_b/N_0 required to achieve a BER of 10^{-4} for these parameters is summarised in Table 7.2.

TABLE 7.2: FFT-BP Iteration Study, Two Tap Channel

E_b/N_0 required to achieve a BER of 10^{-4}[dB]						
Turbo Iteration number	1	2	3	4	5	6
10 FFT-BP iterations	2.63	2.15	1.93	1.85	1.85	1.8
20 FFT-BP iterations	2.5	2.0	1.86	1.82	1.78	1.75
30 FFT-BP iterations	2.45	2.0	1.84	1.82	1.77	1.77

From the table it can be see that while an increase in FFT-BP iterations will initially improve the performance of the system, the gain achieved decreases when turbo iterations are performed. Specifically, after 3 turbo iterations, the difference between running 10 FFT-BP iterations and running 30 FFT-BP iterations is only 0.09 dB. As a result, the complexity incurred by increasing the number of FFT-BP iterations to more than 10 is unnecessary when more than two turbo iterations are performed. It should be noted that for LDPC codes in general the performance improvement with iteration number is not linear. Thus, increasing the number of FFT-BP iterations to 100 or 1000 might in fact provide substantial gains. However, in this dissertation the focus of the system design is on realistic complexity which is why these cases are not considered.

The gain achieved by increasing the number of turbo iterations will now be considered. Figures 7.3, 7.6 and 7.7 show the performance of the STF rate-1, SF rate-2 and SF rate-1 code respectively. In these simulations, 10 internal FFT-BP iterations are run and the number of turbo iterations varied from 1 to 6. The baseline performance (without the FEC) of these codes can be found in Figure 5.5. From these simulations, the gain achieved for each iteration of the system is presented in Table 7.3. The gain for iteration number i is calculated as the difference in E_b/N_0, between iteration i and $i-1$, required to achieve a BER of 10^{-4}. 0 iterations corresponds to the baseline case. From the table one can see that the gain decreases rapidly with each iteration. After the third iteration, almost all of the gain has been achieved. Thus, considering the complexity incurred, increasing the number of turbo iterations beyond three is unnecessary.

TABLE 7.3: Turbo Iteration Study, Two Tap Channel, BER = 10^{-4}

Gain achieved after each iteration [dB]						
Iteration number	1	2	3	4	5	6
STF rate-1	6.375	0.5	0.225	0.1	0.05	0
SF rate-2	6.3	0.86	0.25	0.16	0.1	0.05
SF rate-1	8.7	0.25	0.075	0.025	0.025	0.025

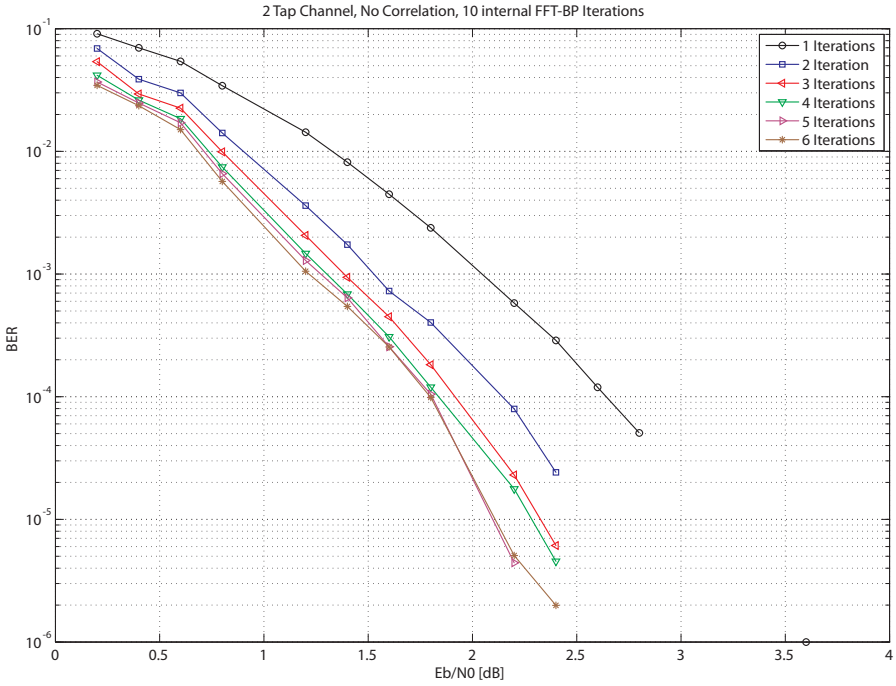

FIGURE 7.3: Performance of the turbo NB-QC-LDPC coded system with the STF rate-1 code on the two tap channel with no correlation and 10 internal FFT-BP iterations for each turbo iteration.

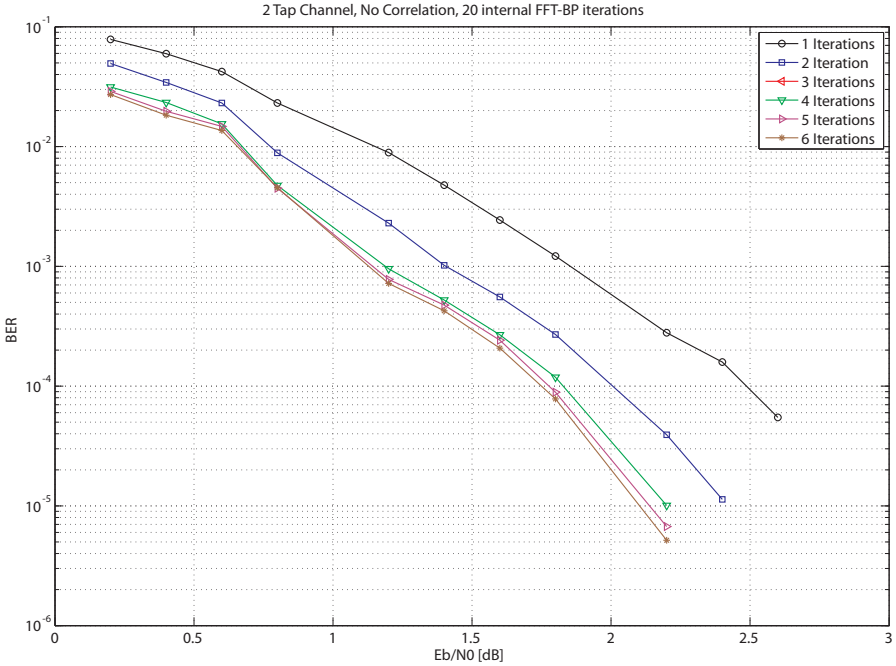

FIGURE 7.4: Performance of the turbo NB-QC-LDPC coded system with the STF rate-1 code on the two tap channel with no correlation and 20 internal FFT-BP iterations for each turbo iteration.

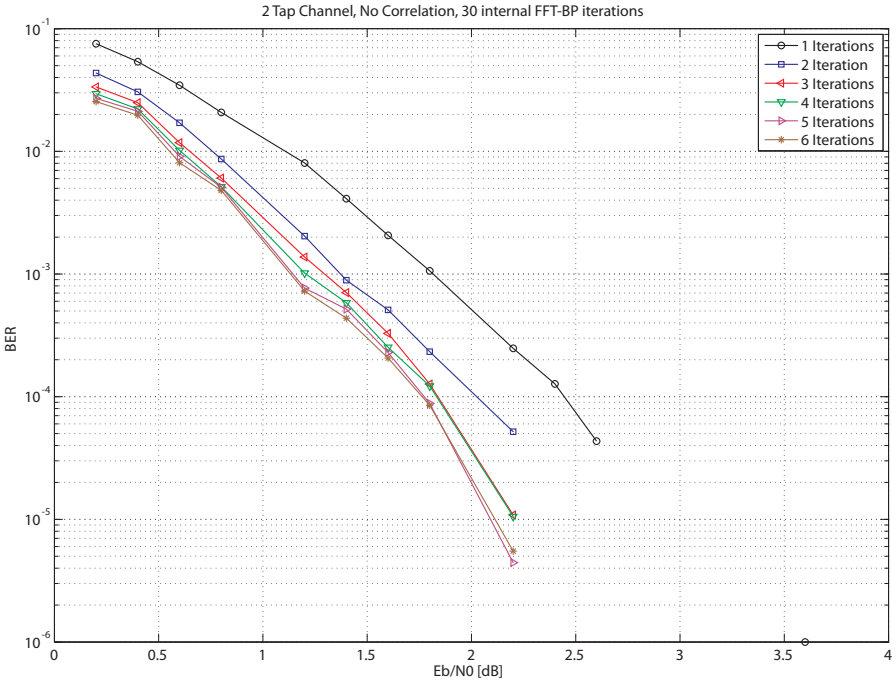

FIGURE 7.5: Performance of the turbo NB-QC-LDPC coded system with the STF rate-1 code on the two tap channel with no correlation and 30 internal FFT-BP iterations for each turbo iteration.

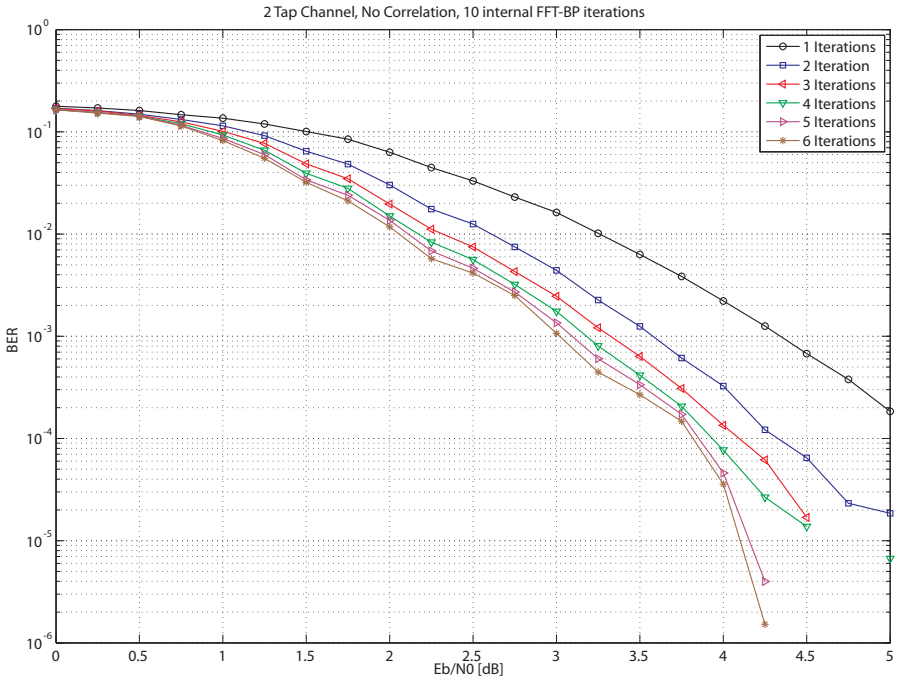

FIGURE 7.6: Performance of the turbo NB-QC-LDPC coded system with the SF rate-2 code on the two tap channel with no correlation and 10 internal FFT-BP iterations for each turbo iteration.

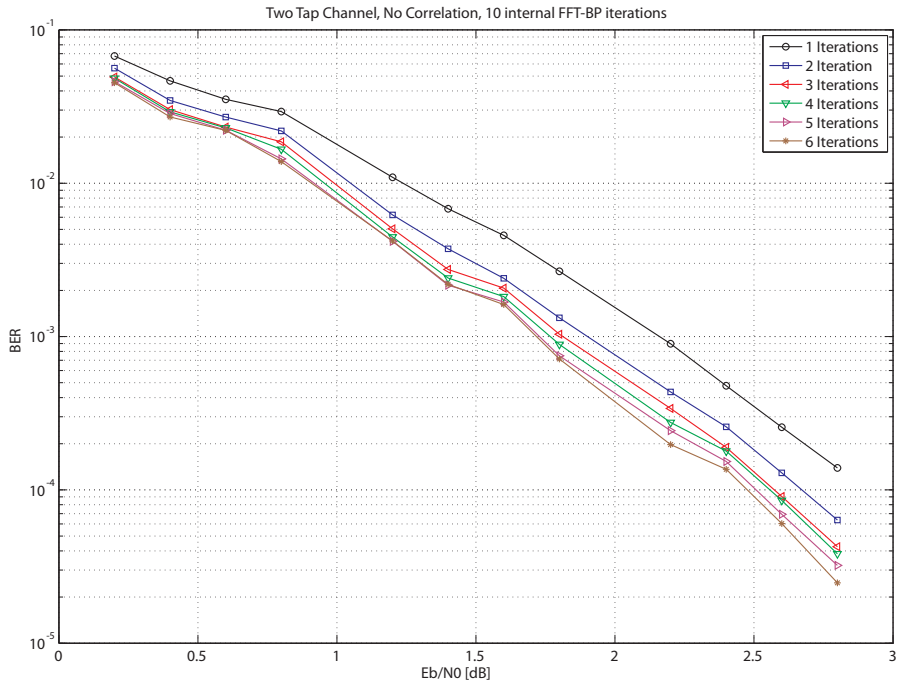

FIGURE 7.7: Performance of the turbo NB-QC-LDPC coded system with the SF rate-1 code on the two tap channel with no correlation and 10 internal FFT-BP iterations for each turbo iteration.

7.4.1.2 Turbo NB-QC-LDPC coded STF-MIMO-OFDM system, correlated channel

In this section the effect of channel correlation on the performance of the system is analysed. The number of FFT-BP iterations is set to 10 for all simulations. In the first part of the analysis, the effect of the number of turbo iterations in a correlated channel with $r = 0.6$ is analysed. Figures 7.8, 7.9 and 7.10 respectively show the performance of the rate-1 STF code, the rate-2 SF code, and the rate-1 SF code on the two tap channel, in correlated channel conditions with a correlation factor of 0.6. Considering the performance of the system after one iteration as the baseline, Table 7.4 shows the gains achieved by running more iterations. From the table it can

TABLE 7.4: Iteration Study in Correlated Channel Conditions, Two Tap Channel, BER = 10^{-3}

Performance gain after each iteration[dB]					
Iteration number	2	3	4	5	6
STF rate-1	0.75	0.25	0.1	0.09	0.07
SF rate-2	0.85	0.3	0.15	0.1	0.15
SF rate-1	0.28	0.06	0.03	0.07	0.04

be seen that the majority of the gain is achieved in the first three iterations. This re-enforces the result obtained for the iteration analysis performed in uncorrelated channel conditions.

In the second part of the analysis, the correlation factor is varied while the number of turbo iterations is kept constant at six. The purpose of this analysis is to quantify the loss experienced by the system for different levels of correlation. While only 3 iterations are required for good performance, six iterations are used in this analysis as this represents the best performance of the system which allows the study to consider the absolute effect of correlation. Figures 7.11, 7.12 and 7.13 show the performance of the rate-1 STF, rate-2 SF and rate-1 SF codes respectively. The loss incurred for each of these codes is summarised in Table 7.5. The loss is calculated as the difference between the E_b/N_0, required to achieve a BER of 10^{-3} as compared to the uncorrelated case. From the table, one can see that the SF rate-1 code is the most robust against channel correlation. In all three cases, the performance losses increase with the correlation factor. The rate at which the loss increases, also increases with the correlation factor.

TABLE 7.5: Correlation Study, Two Tap Channel, BER = 10^{-3}

Performance loss for each correlation factor [dB]				
Correlation	0.2	0.4	0.6	0.8
STF rate-1	0.125	0.555	1.125	2.375
SF rate-2	0.2	0.6	1.375	-
SF rate-1	0.1	0.35	0.65	1.4

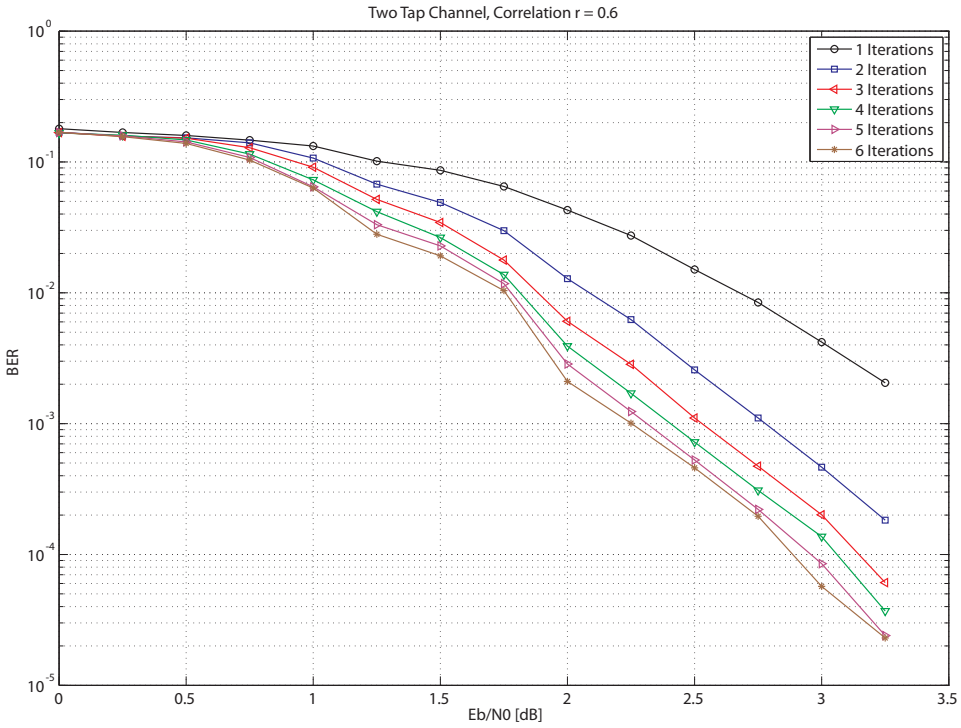

FIGURE 7.8: Performance of the turbo NB-QC-LDPC coded system with the STF rate-1 code on the two tap channel with correlation factor r=0.6 and 10 internal FFT-BP iterations for each turbo iteration.

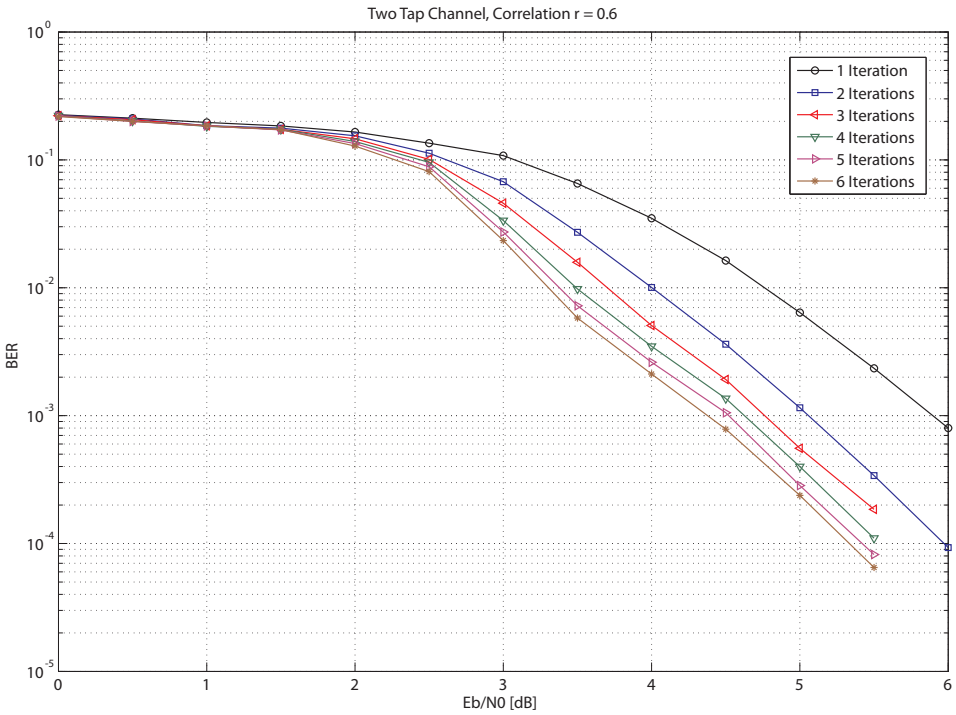

FIGURE 7.9: Performance of the turbo NB-QC-LDPC coded system with the SF rate-2 code on the two tap channel with correlation factor r=0.6 and 10 internal FFT-BP iterations for each turbo iteration.

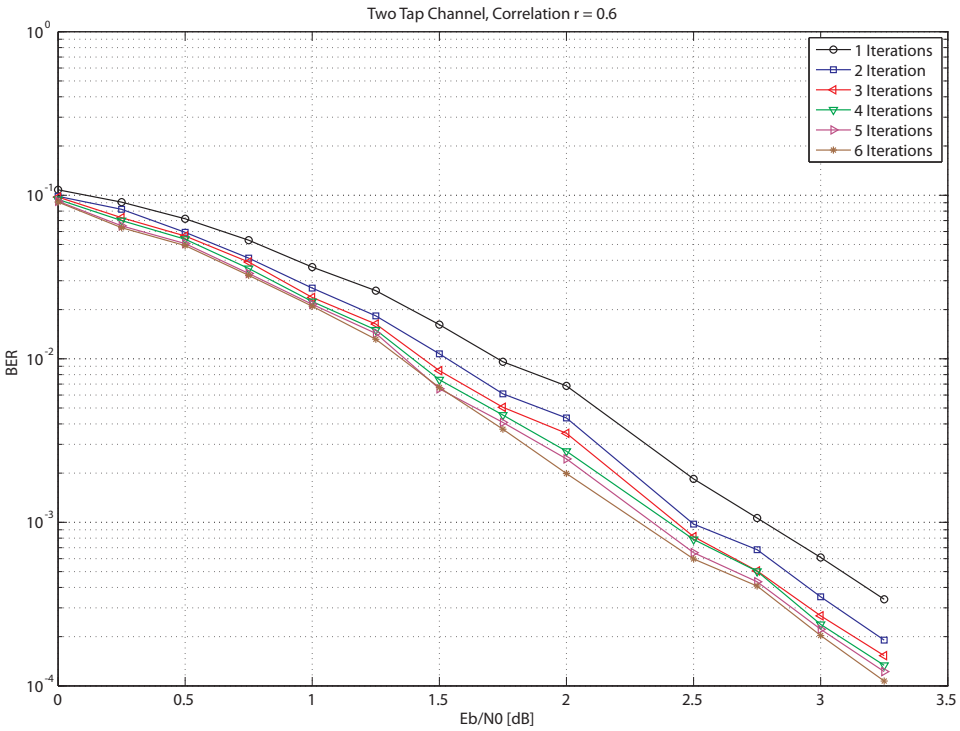

FIGURE 7.10: Performance of the turbo NB-QC-LDPC coded system with the SF rate-1 code on the two tap channel with correlation factor r=0.6 and 10 internal FFT-BP iterations for each turbo iteration.

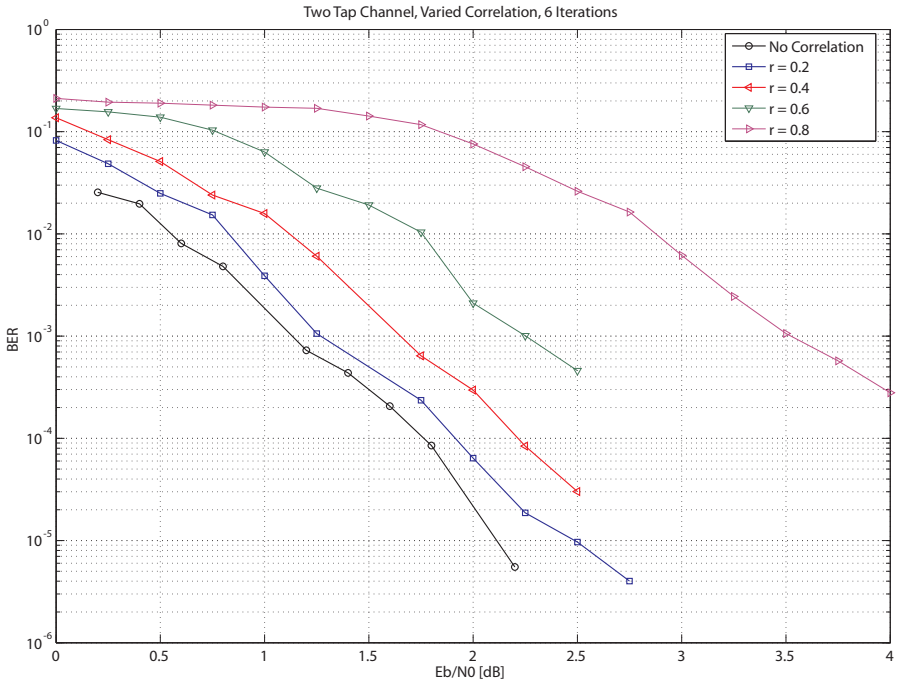

FIGURE 7.11: Performance of turbo NB-QC-LDPC coded system with the rate-1 STF code running 6 turbo iterations on the two tap channel with correlation factors ranging from r=0.0 to r=0.8.

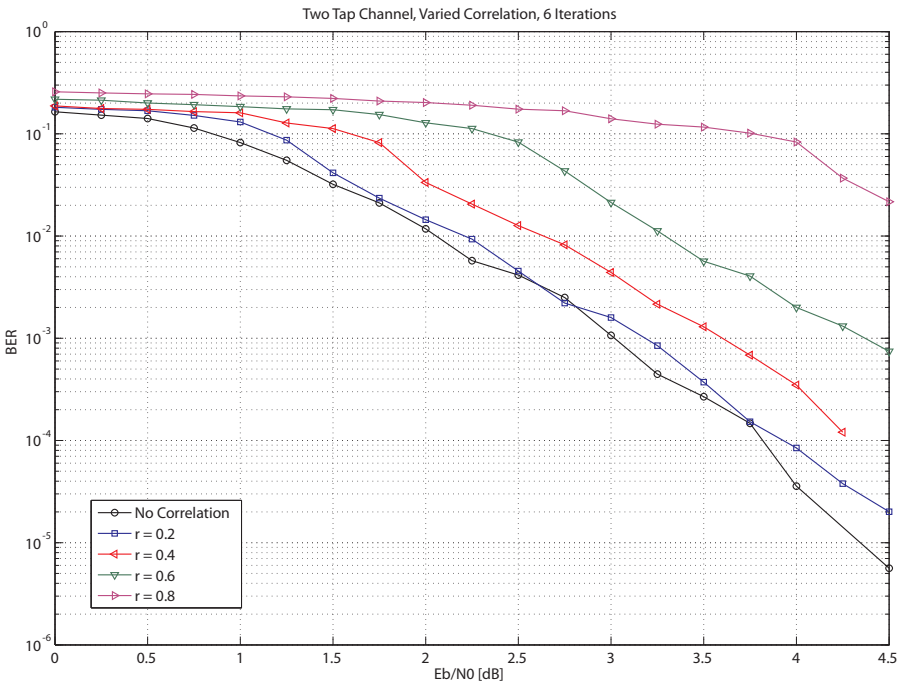

FIGURE 7.12: Performance of turbo NB-QC-LDPC coded system with the rate-2 SF code running 6 turbo iterations on the two tap channel with correlation factors ranging from r=0.0 to r=0.8.

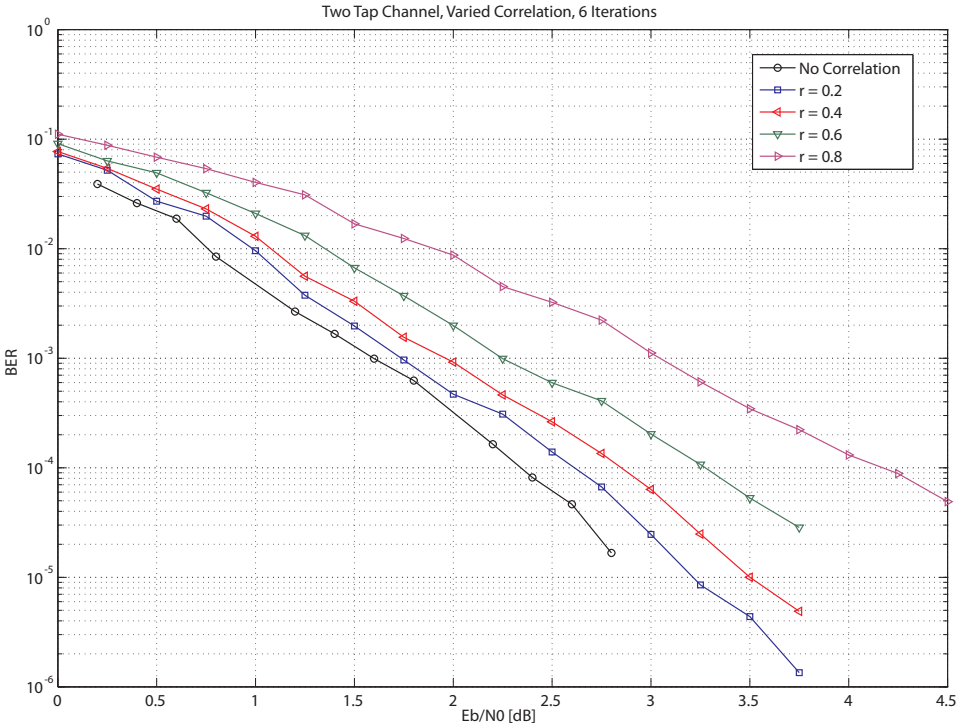

FIGURE 7.13: Performance of turbo NB-QC-LDPC coded system with the rate-1 SF code running 6 turbo iterations on the two tap channel with correlation factors ranging from r=0.0 to r=0.8.

7.4.2 Suburban Alternative Channel

7.4.2.1 Comparison of ST, SF and STF codes without turbo iterations, no channel correlation

In this study, the performance of the different ST, SF and STF codes are compared. The purpose of the study is to determine whether the relative performance of these multi-antenna codes, as discussed in Chapter 5 still hold when an FEC code is applied. For this analysis no turbo iterations are performed. This would correspond to the 1 iteration curves in other figures. Figure 7.14 shows the performance of different ST, SF and STF codes. The number of FFT-BP iterations used is 50.

Comparing the performance of the rate-1 SF and the rate-1 STF code at a BER of 10^{-4} one can see that the STF code outperforms the SF code with 0.2 dB. However, in the uncoded case the STF code outperforms the SF code by 1.1 dB. The slopes of the two curves may also be compared. In the uncoded case, the slope of the STF curve is noticeably steeper than that of the SF curve. However, in the coded case, any difference in slope is marginal. The gain of the STF code over the SF code without the FEC code, is due to the extra time diversity which the code can exploit. However, when the FEC code is added, the FEC code exploits the time diversity. Since the time diversity cannot be exploited twice, the two codes perform approximately the same. Because of a larger SD being required, the decoding of the STF rate-1 code is more complex than that of the SF rate-1 code. This extra decoding complexity is offset by the gain in performance. However, when the FEC code is used there is no gain in performance. There is thus no good reason to use the STF code over the SF code when an external FEC code is employed.

Figures 7.15 and 7.16 show the performance of different ST, SF and STF codes when respectively 8 bit and 5 bit quantisation is employed at the receiver. When comparing the results in the figure with the results in Figure 7.14, one can see no loss in performance.

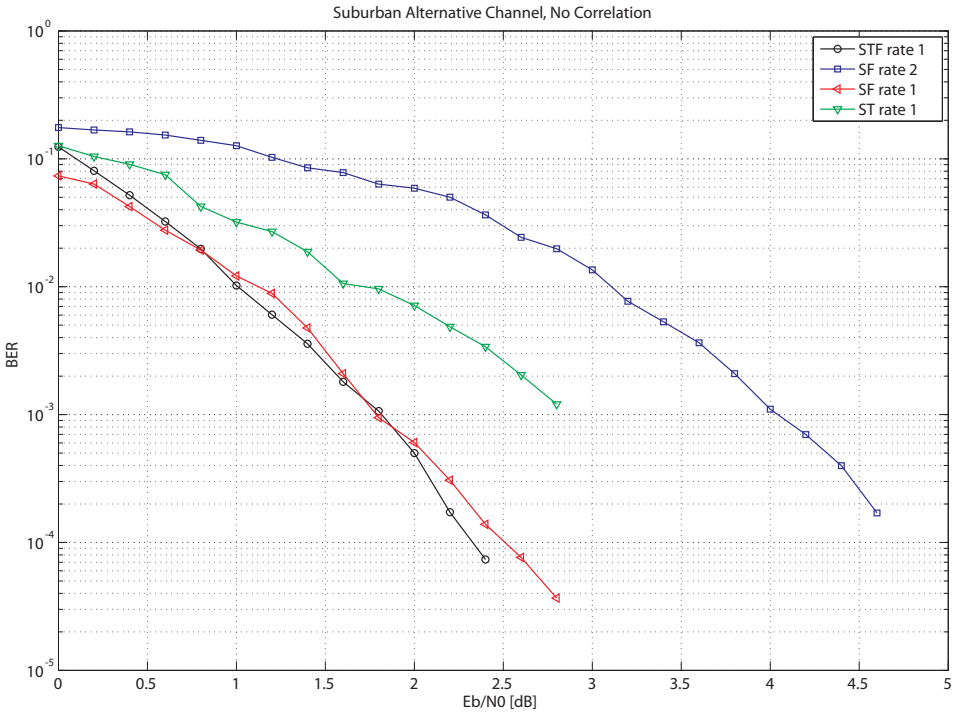

FIGURE 7.14: Performance of the NB-QC-LDPC coded system on the suburban alternative channel with no correlation.

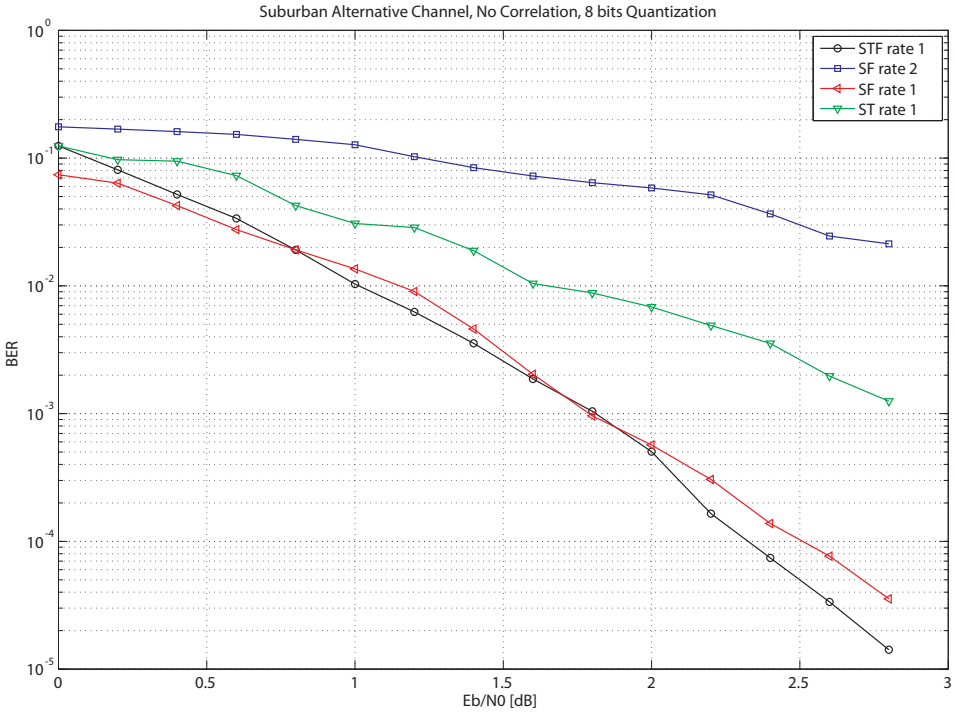

FIGURE 7.15: Performance of the NB-QC-LDPC coded system on the suburban alternative channel with no correlation and 8 bits quantisation.

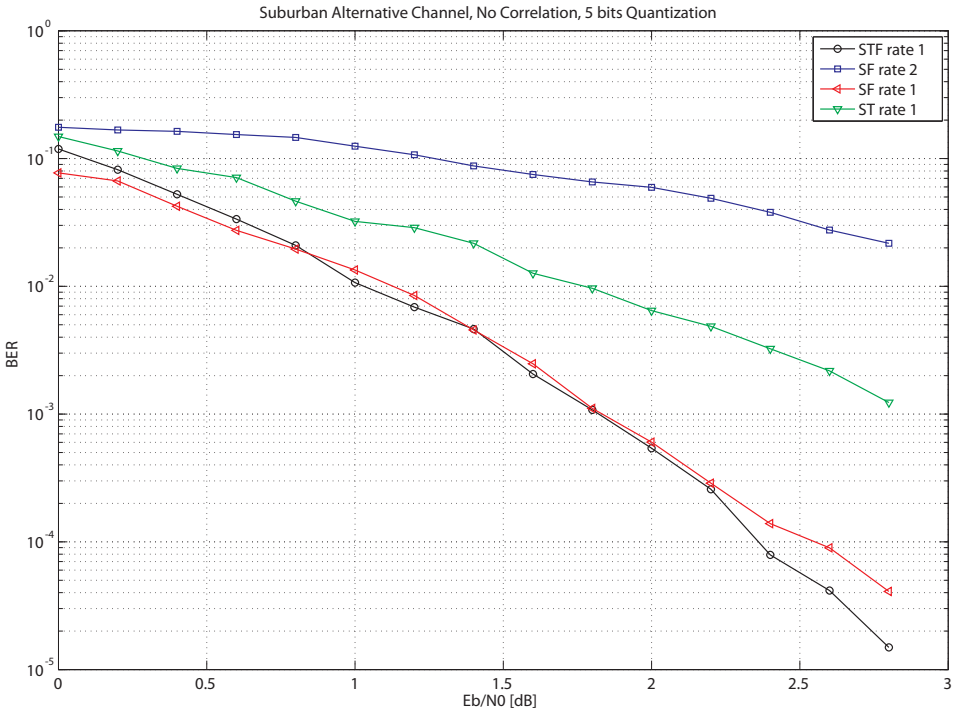

FIGURE 7.16: Performance of the NB-QC-LDPC coded system on the suburban alternative channel with no correlation and 5 bits quantisation.

7.4.2.2 Turbo NB-QC-LDPC coded STF-MIMO-OFDM system, iteration study, no correlation

Figures 7.17, 7.18 and 7.19 show the performance of the rate-1 STF, rate-2 SF and rate-1 SF codes on the 20 tap suburban alternative channel with no correlation. In all three figures the number of internal FFT-BP iterations per turbo iteration is 10. From these figures one can see that similarly to the situation in the ideal two tap channel, the performance of the system after 3 iterations is close to the performance after 6 iterations. This re-enforces the conclusion that only 3 iterations are required in the system. The performance is also very similar to the performance in the ideal 2-tap channel.

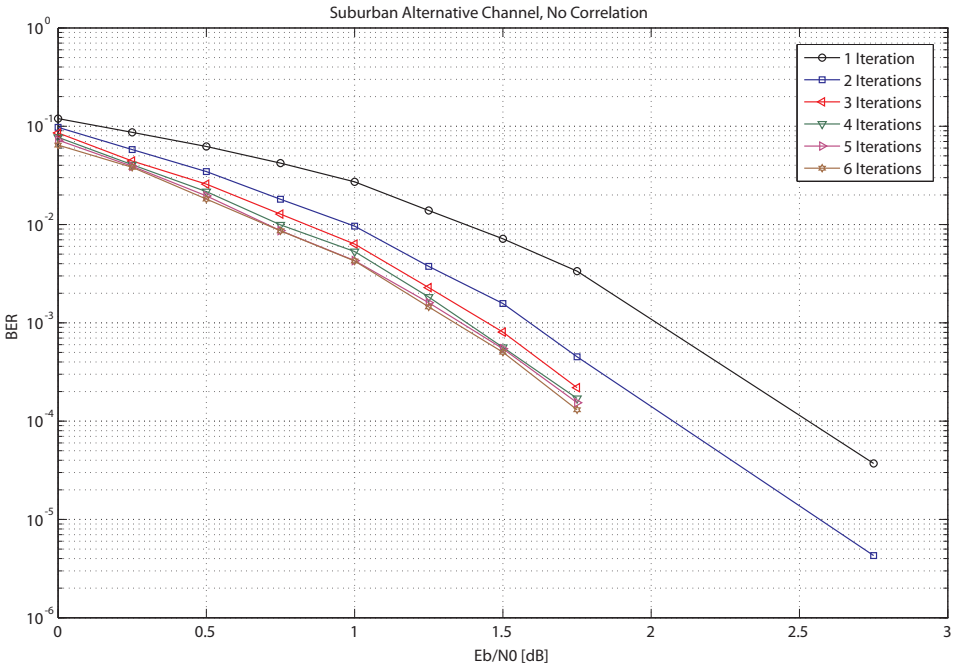

FIGURE 7.17: Performance of turbo NB-QC-LDPC coded system with the rate-1 STF code on the suburban alternative channel with no correlation.

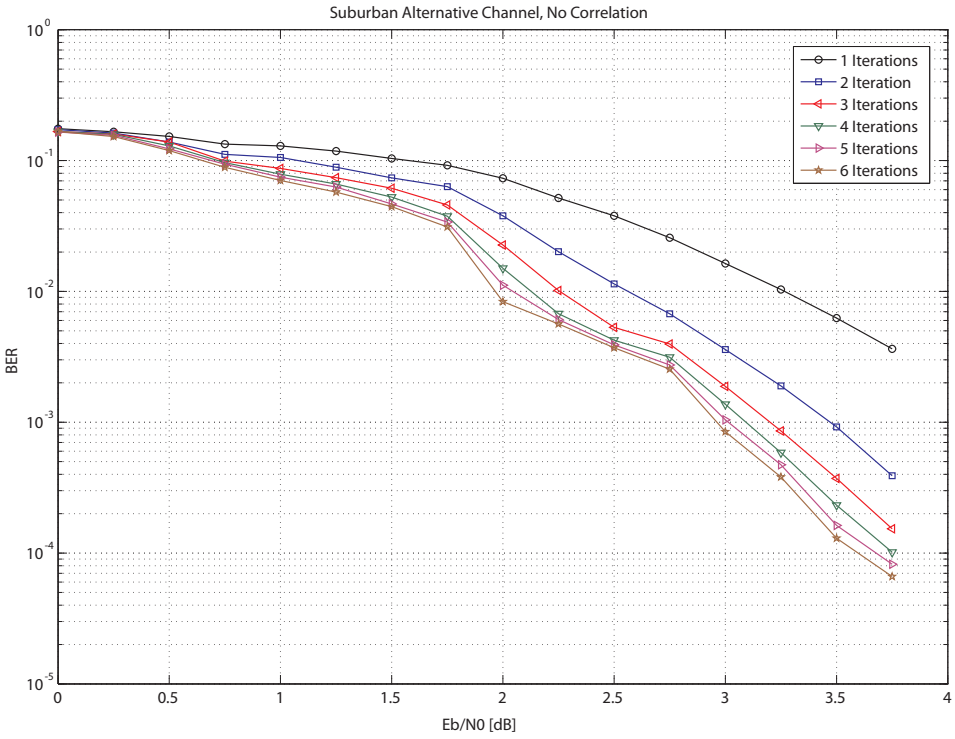

FIGURE 7.18: Performance of turbo NB-QC-LDPC coded system with the rate-2 SF code on the suburban alternative channel with no correlation.

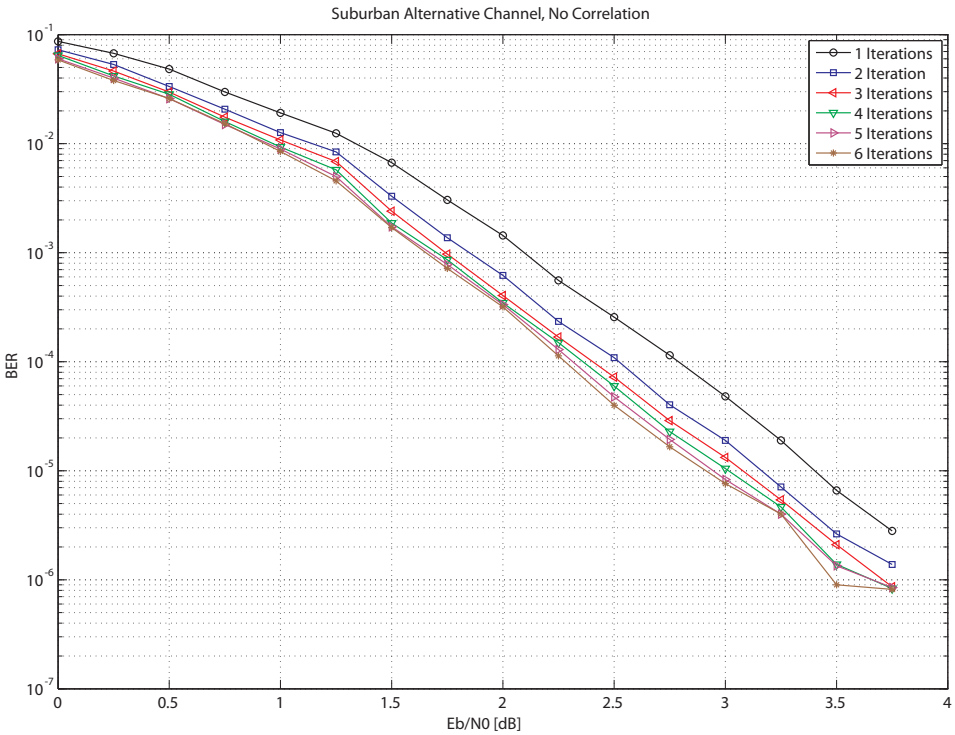

FIGURE 7.19: Performance of turbo NB-QC-LDPC coded system with the rate-1 SF code on the suburban alternative channel with no correlation.

7.4.2.3 Turbo NB-QC-LDPC coded STF-MIMO-OFDM system, correlated channel

Figures 7.20, 7.21 and 7.22 show the performance of the rate-1 STF code, the rate-2 SF code and rate-1 SF code in correlated channel conditions with a correlation factor of 0.6. Figures 7.23, 7.24 and 7.25 show the performance of the rate-1 STF code, the rate-2 SF code and the rate-1 SF code for different correlation factors after 6 turbo iterations. In all cases, the number of FFT-BP iterations are fixed at 10 per turbo iteration. Comparing these figures with the corresponding figures from the two tap channel studies, one can see that except for a very slight degradation in performance, all the results are approximately the same.

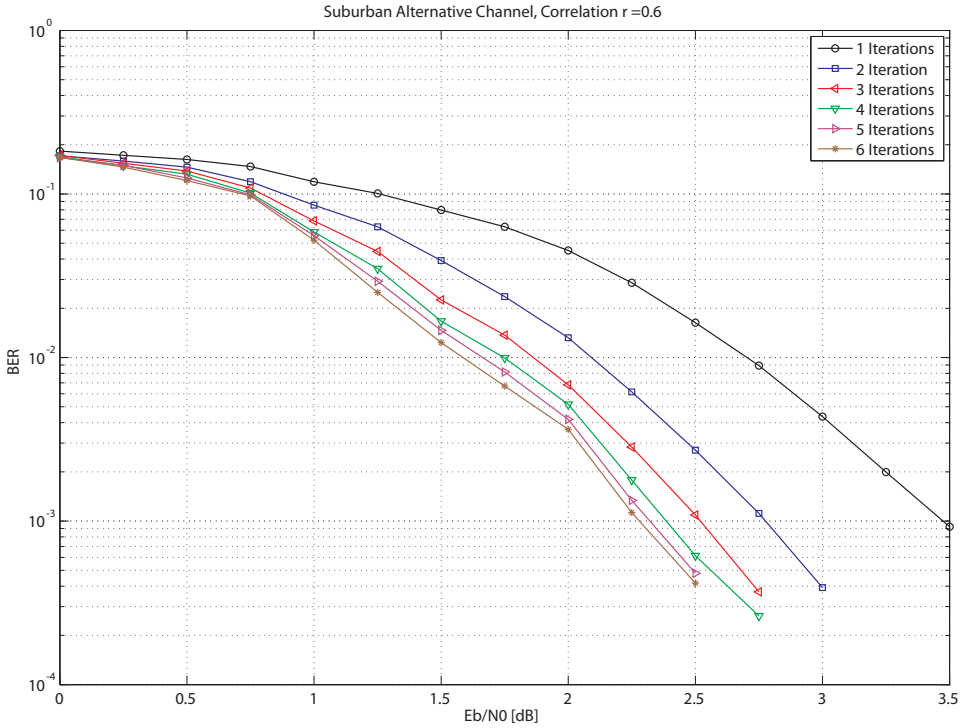

FIGURE 7.20: Performance of turbo NB-QC-LDPC coded system with the rate-1 STF code on the suburban alternative channel with correlation factor r=0.6.

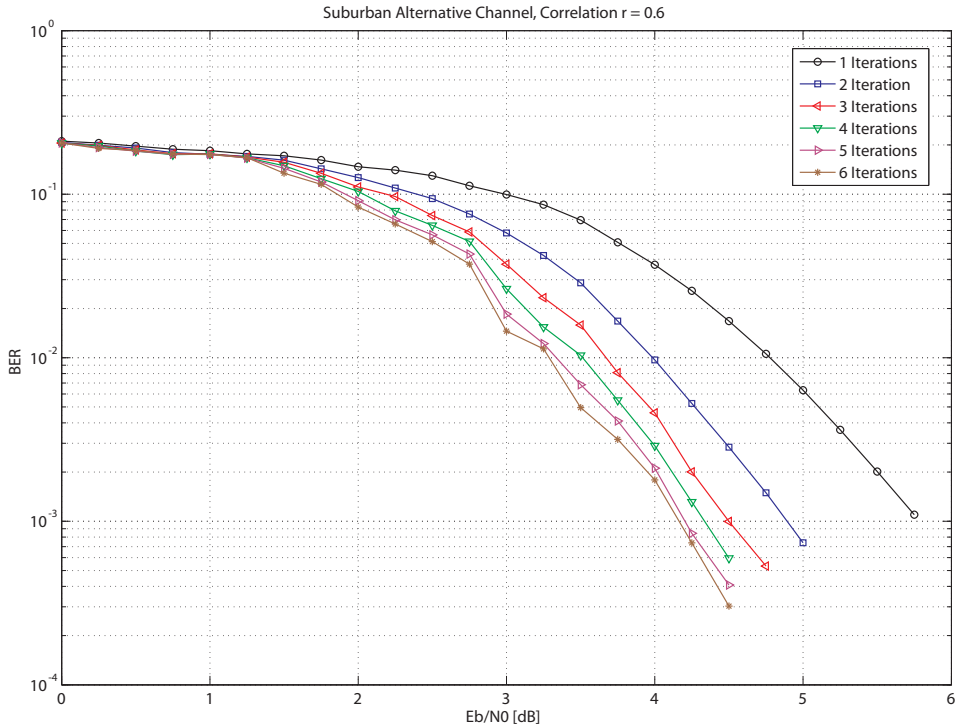

FIGURE 7.21: Performance of turbo NB-QC-LDPC coded system with the rate-2 SF code on the suburban alternative channel with correlation factor r=0.6.

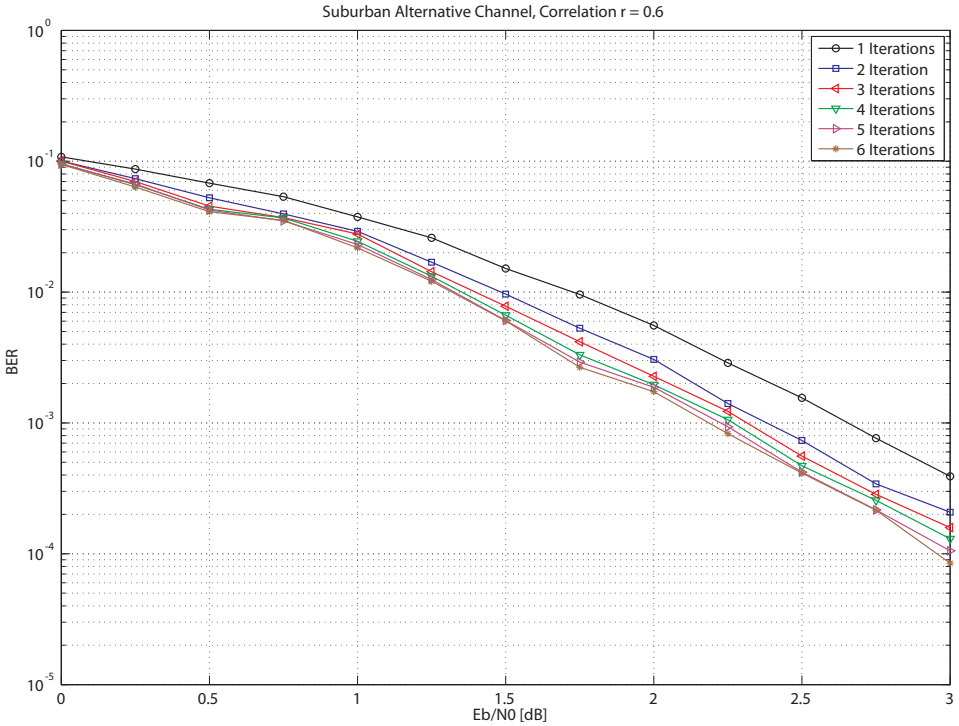

FIGURE 7.22: Performance of turbo NB-QC-LDPC coded system with the rate-1 SF code on the suburban alternative channel with correlation factor r=0.6.

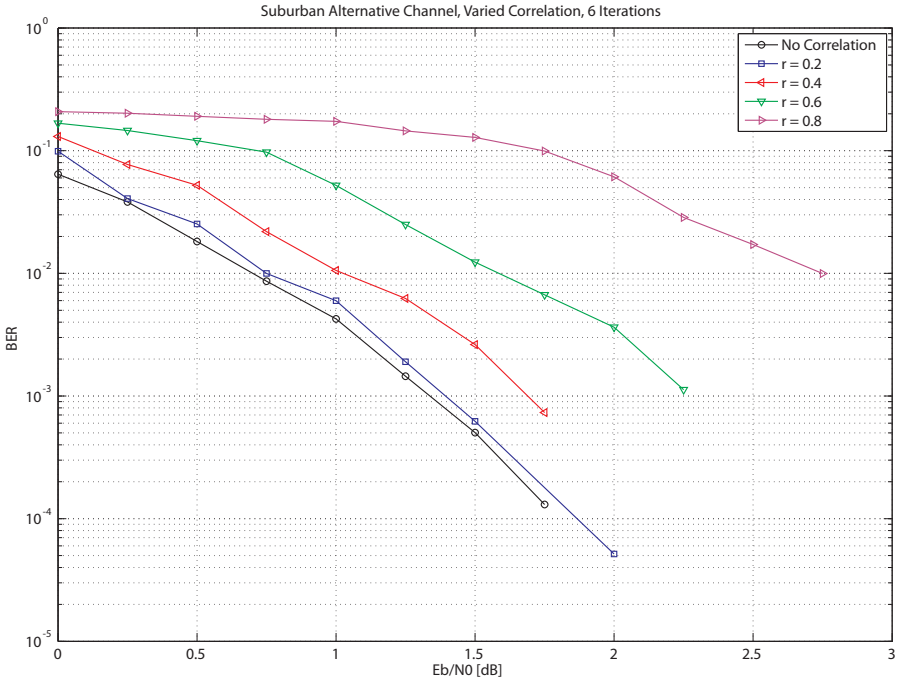

FIGURE 7.23: Performance of turbo NB-QC-LDPC coded system with the rate-1 STF code running 6 turbo iterations on the suburban alternative channel with correlation factors ranging from r=0.0 to r=0.8.

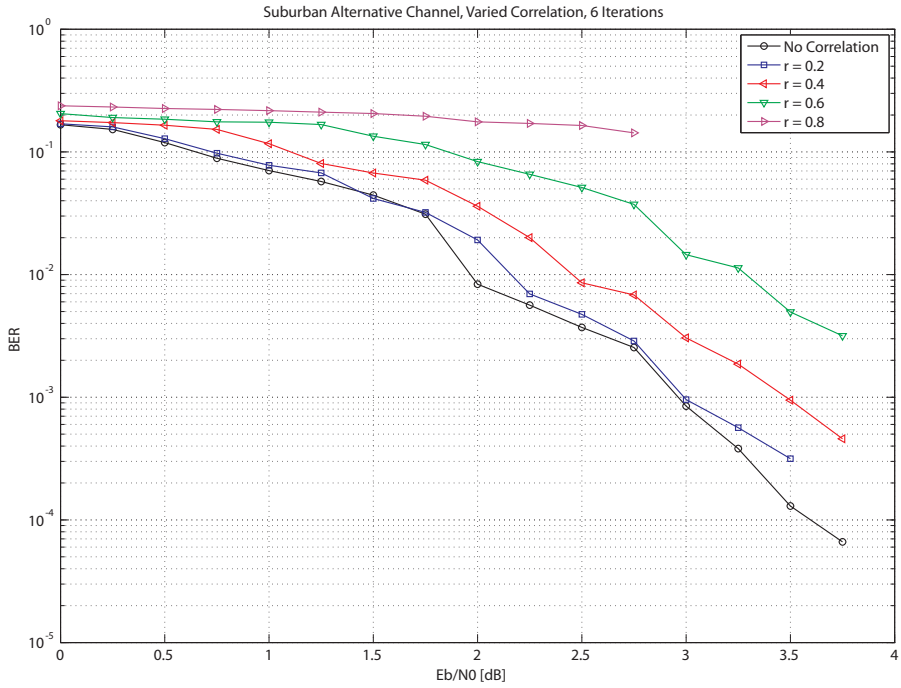

FIGURE 7.24: Performance of turbo NB-QC-LDPC coded system with the rate-2 SF code running 6 turbo iterations on the suburban alternative channel with correlation factors ranging from r=0.0 to r=0.8.

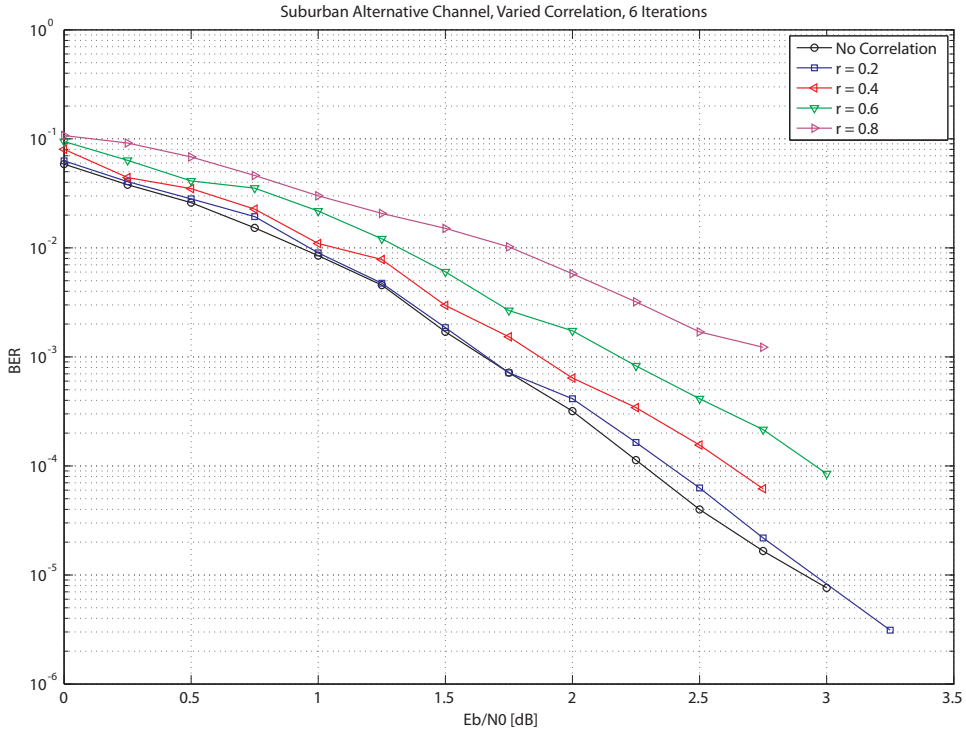

FIGURE 7.25: Performance of turbo NB-QC-LDPC coded system with the rate-1 SF code running 6 iterations on the suburban alternative channel with correlation factors ranging from r=0.0 to r=0.8.

7.5 SUMMARY OF RESULTS

In this section, the results that were obtained in this chapter are summarised. Table 7.6 shows the results of the iteration study that was used to determine how many turbo iterations and how many internal FFT-BP iterations per turbo iteration should be performed. The table also shows the performance of the SF and STF codes in the two tap channel, when there is no channel correlation. Table 7.7 shows the analysis of the system performance in correlated channel conditions. Simulations were also performed in a 20-tap Suburban alternative channel, and the performance results were found to be very similar to those of the 2-tap channel.

TABLE 7.6: Summary of Turbo and FFT-BP Iteration Study, Two Tap Channel

E_b/N_0 required to achieve a BER of 10^{-4}[dB]						
Turbo Iteration number	1	2	3	4	5	6
10 FFT-BP iterations	2.63	2.15	1.93	1.85	1.85	1.8
20 FFT-BP iterations	2.5	2.0	1.86	1.82	1.78	1.75
30 FFT-BP iterations	2.45	2.0	1.84	1.82	1.77	1.77
Iteration gain[dB], 10 FFT-BP iterations						
Turbo Iteration number	1	2	3	4	5	6
STF rate-1	6.375	0.5	0.225	0.1	0.05	0
SF rate-2	6.3	0.86	0.25	0.16	0.1	0.05
SF rate-1	8.7	0.25	0.075	0.025	0.025	0.025

TABLE 7.7: Summary of Analysis in Correlated Channel Conditions, Two Tap Channel

Performance gain after each iteration[dB], correlation = 0.6, BER = 10^{-3}					
Turbo Iteration number	2	3	4	5	6
STF rate-1	0.75	0.25	0.1	0.09	0.07
SF rate-2	0.85	0.3	0.15	0.1	0.15
SF rate-1	0.28	0.06	0.03	0.07	0.04
Performance loss due to each correlation factor [dB], BER = 10^{-3}					
Correlation	0.0	0.2	0.4	0.6	0.8
STF rate-1	0.0	0.125	0.555	1.125	2.375
SF rate-2	0.0	0.2	0.6	1.375	-
SF rate-1	0.0	0.1	0.35	0.65	1.4

7.5.1 Conclusions

Based on the simulation results presented in this chapter, the following list shows the conclusions that were drawn.

- The number of internal FFT-BP iterations, per turbo iteration, should be 10.
- After 3 turbo iterations most of the gain has been achieved. This holds for correlated and uncorrelated conditions in both the 2-tap as well as the 20-tap channel.
- In correlated channel conditions, the performance of the system degrades. The performance loss increases with the correlation factor. The rate at which the performance loss increases, also increases with the correlation factor.
- The SF rate-2 code is the more susceptible to channel correlation than the SF rate-1 code or the STF rate-1 code.
- When a powerful FEC code (such as the NB-QC-LDPC code) is used, an SF code can equal the performance of an STF code at a reduced complexity.

In general, the system performed well over a range of different channel conditions.

Chapter Eight

Conclusions and future research

8.1 Conclusion

In this dissertation a Turbo NB-QC-LDPC coded STF-MIMO-OFDM system was developed.

- In Chapter 2, a basic introduction to FEC codes was provided and terminology and notation which are used throughout the dissertation were presented.
- In Chapter 3, the design of NB-LDPC codes was considered. In order to keep encoding complexity low, a QC code design was chosen. For the sake of realistic complexity and latency, the code length, as well as the number of FFT-BP decoding iterations, was limited. Several design procedures for NB-QC-LDPC codes were implemented and compared. It was found that girth-12 codes performed best. The method used to select the NB elements in the parity check matrix was analysed and it was found that for the field size considered, a random selection procedure performed best. The codes were also analysed with quantisation at the receiver and the effect on the performance was measured.
- In Chapter 4, mobile wireless channels were discussed and the triply selective fading channel model used in the simulations was presented.
- In Chapter 5, multi-antenna coding techniques were discussed. Specifically the ST, SF and STF codes used in the system were presented. The SF and STF codes use LP and layering. It was found that the process of LP places extra requirements on the number of

quantisation bits at the receiver. The limit on the achievable diversity in a quantised LP coded system was mathematically derived. This limit was then verified by simulating SF and STF codes with quantisation at the receiver.

- In Chapter 6, MIMO detection was discussed. The SISO-SD was presented. A reduced complexity SO MIMO detector combining a Hopfield network with an SD in an iterative receiver structure was developed. The performance of the detector was compared with that of the optimal SISO-SD detector. The computational complexity of the SD-Hopfield detector was explicitly calculated and compared with the complexity of the normal SO-SD. It was shown that the Hopfield-SD can achieve the same performance as the optimal SO-SD at a reduced complexity. The SD-Hopfield detector was also shown to provide a convenient method for trading off complexity for performance.

- In Chapter 7, the full Turbo NB-QC-LDPC coded STF-MIMO-OFDM system combining the code developed in Chapter 3 with the MIMO-OFDM system and the STF codes was presented. The turbo structure was explained. The system was analysed with simulations in triply selective block fading conditions on realistic as well as ideal channels. The effect of the number of internal FFT-BP decoding iterations, as well as the number of turbo iterations, on the performance of the system was analysed. The effect of correlation at the antennas, as well as quantisation at the receiver, was analysed with simulations. It was found that the system is robust against channel correlation as well as quantisation at the receiver. It was also shown that in the system, the SF code achieves the same performance as the STF code at less complexity.

8.2 FUTURE RESEARCH

The research presented in this dissertation can be used as a stepping stone for future research. Aspects that deserve attention, which were not considered in this dissertation, include channel state estimation (CSE), adaptive systems, differential multi-antenna coding and higher order modulation schemes.

8.2.1 CSE

In every part of this dissertation it was assumed that perfect CSI was available at the receiver. This is, however, unrealistic as there is always some error associated with the CSE. Research

should thus be done on the effect of imperfect CSI on the performance of the system and all the sub-systems presented in this dissertation. Different CSE algorithms can be implemented and their performance noted. It is also possible to combine the CSE algorithm in an iterative structure with the current iterative joint decoder.

8.2.2 Differential STF coding

The use of differential coding allows the receiver to operate without CSI. No pilot bits are thus required in the signal. This allows for a high rate code to be used without decreasing the effective rate of the system when compared with the uncoded case. Using the system structure as proposed in this dissertation with differential STF codes could thus provide a performance increase over uncoded STF systems without a loss in transmission rate.

8.2.3 Higher order modulation techniques

In the analysis of the proposed system, only QPSK and BPSK were considered. However, for high spectral efficiencies, higher order modulation techniques are required. A performance analysis of the system using higher order modulation techniques in correlated and quantised channel conditions should be performed.

8.2.4 Adaptive systems

Usually adaptive systems change the modulation technique in order to stay in a specific BER range. This, however, tends to result in a discrete jump in the BER experienced by the system. The use of an iterative structure as used in this dissertation allows for a convenient method to trade performance for complexity by increasing and decreasing the number of turbo iterations. This same process could also be use to change the system adaptively in order to maintain a required BER. An adaptive scheme could thus be developed which combines moving between different modulation techniques with changing the number of turbo iterations to yield a smooth transition between modulation schemes.

8.2.5 QC-LDPC code design

In this dissertation, only a limited range of specifications was considered in the code design. Research can be done to compare different design procedures over a greater range of code rates

and field sizes.

APPENDIX A
POWER DELAY PROFILES

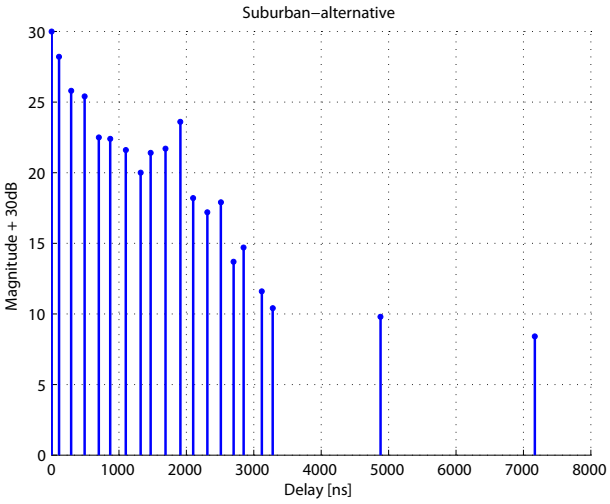

FIGURE A.1: Suburban-alternative power delay profile [1]

REFERENCES

[1] K. Jeong, K. M. Chung, J. C. Kim, J. H. Yu, J. S. Lee, and S. H. Seo, "Multipath channel models for wireless local and metropolitan area networks," in *ICITA '05: Proceedings of the Third International Conference on Information Technology and Applications (ICITA'05) Volume 2*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 295–298.

[2] R. Gallager, *Low Density Parity Check Codes*. Cambridge, Massachusetts: MIT Press, 1963.

[3] D. MacKay and R. Neal, "Near shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, p. 1645, Aug 1996.

[4] M. Davey and D. MacKay, "Low density parity check codes over GF(q)," in *Information Theory Workshop, 1998*, Jun 1998, pp. 70–71.

[5] T. Richardson and R. Urbanke, "The capacity of low-density parity-check codes under message-passing decoding," vol. 47, no. 2, pp. 599–618, Feb 2001.

[6] ——, *Modern Coding Theory*. Cambridge University Press, 2008.

[7] A. Bennatan and D. Burshtein, "Design and analysis of nonbinary LDPC codes for arbitrary discrete-memoryless channels," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 549–583, Feb. 2006.

[8] 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," *IEEE Transactions on Information Theory*, vol. 53, no. 7, pp. 2429–2458, July 2007.

[9] S. Song, B. Zhou, S. Lin, and K. Abdel-Ghaffar, "A Unified Approach to the Construction of Binary and Nonbinary Quasi-Cyclic LDPC Codes Based on Finite Fields," *IEEE Transactions on Communications*, vol. 57, no. 1, pp. 84–93, January 2009.

[10] X. Ge and S.-T. Xia, "Structured non-binary LDPC codes with large girth," *Electronics Letters*, vol. 43, no. 22, 25 2007.

[11] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[12] S. Alamouti, "A simple transmit diversity technique for wireless communications," vol. 16, no. 8, pp. 1451–1458, October 1998.

[13] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Space-time block codes from orthogonal designs," vol. 45, no. 5, pp. 1456–1467, 1999.

[14] M. Damen, K. Abed-Meraim, and J.-C. Belfiore, "Diagonal algebraic space-time block codes," *IEEE Transactions on Information Theory*, vol. 48, no. 3, pp. 628–636, Mar 2002.

[15] H. El Gamal and J. Hammons, A. R., "On the design of algebraic space-time codes for MIMO block-fading channels," *IEEE Transactions on Information Theory*, vol. 49, no. 1, pp. 151–163, 2003.

[16] W. Su, Z. Safar, and K. Liu, "Full-rate full-diversity space-frequency codes with optimum coding advantage," *IEEE Transactions onInformation Theory*, vol. 51, no. 1, pp. 229–249, Jan. 2005.

[17] W. Zhang, X. G. Xia, and P. C. Ching, "High-Rate Full-Diversity Space-Time-Frequency Codes for Broadband MIMO Block-Fading Channels," *IEEE Transactions on Communications*, vol. 55, no. 1, pp. 25–34, 2007.

[18] W. Zhang, X.-G. Xia, and P. Ching, "Universal Space-Frequency Block Coding for MIMO-OFDM Systems," in *Proceedings of 2005 Asia-Pacific Conference on Communications*, Oct. 2005, pp. 227–231.

[19] M. O. Damen, A. Chkeif, and J. C. Belfiore, "Lattice code decoder for space-time codes," *IEEE Communication Letters*, vol. 4, no. 5, pp. 161–163, May 2000.

[20] H. Vikalo, B. Hassibi, and T. Kailath, "Iterative decoding for MIMO channels via modified sphere decoding," *IEEE Transactions on Wireless Communications*, vol. 3, no. 6, pp. 2299–2311, Nov. 2004.

[21] J. Zheng and B. Rao, "LDPC-coded MIMO systems with unknown block fading channels: soft MIMO detector design, channel estimation, and code optimization," *IEEE Transactions on Signal Processing*, vol. 54, no. 4, pp. 1504–1518, April 2006.

[22] J. Zhang and H.-N. Lee, "Performance analysis on LDPC-Coded systems over quasi-static (MIMO) fading channels," *IEEE Transactions on Communications*, vol. 56, no. 12, pp. 2080–2093, December 2008.

[23] J. Hu, H. Zhang, and Y. Yang, "Efficient Detector for LDPC Coded MIMO System," in *Proceedings of 4th International Conference on Wireless Communications, Networking and Mobile Computing, WiCOM '08*, Oct. 2008, pp. 1–4.

[24] O. Alamri, S. Ng, F. Guo, S. Zummo, and L. Hanzo, "Nonbinary LDPC-Coded Sphere-Packed Transmit Diversity," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 5, pp. 3200–3205, Sept. 2008.

[25] M. El-Hajjar, O. Alamri, S. X. Ng, and L. Hanzo, "Turbo Detection of Precoded Sphere Packing Modulation Using Four Transmit Antennas for Differential Space-Time Spreading," *IEEE Transactions on Wireless Communications*, vol. 7, no. 3, pp. 943–952, March 2008.

[26] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo-codes(1)," in *Proceedings of IEEE International Conference on Communications, ICC '93, Geneva*, vol. 2, May 1993, pp. 1064–1070 vol.2.

[27] K. Maré, B. Salmon, and B. Maharaj, "Performance gain of space-time-frequency concatenated LDPC codes," in *Proceedings of IEEE International Conference on Communications ICC '09, Dresden, Germany*, June 2009, pp. CD–ROM.

[28] J. G. Proakis, *Digital Communications*. Fourth edition, New York: McGraw-Hill, 2001.

[29] C. Shannon, "A mathematical theory of communication," *Bell Systems Technical Journal*, vol. 27, pp. 623–656, 1948.

[30] D. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[31] R. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, Sep 1981.

[32] D. Declercq and M. Fossorier, "Decoding Algorithms for Nonbinary LDPC Codes Over GF," *IEEE Transactions on Communications*, vol. 55, no. 4, pp. 633–643, April 2007.

[33] R. Peng and R.-R. Chen, "Application of Nonbinary LDPC Cycle Codes to MIMO Channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 6, pp. 2020–2026, June 2008.

[34] C. Poulliat, M. Fossorier, and D. Declercq, "Design of non binary LDPC codes using their binary image: algebraic properties," in *Proceedings of 2006 IEEE International Symposium on Information Theory*, July 2006, pp. 93–97.

[35] J. K. Kim, S. Balakannan, M. H. Lee, and C. J. Kim, "Low complexity encoding of LDPC codes for high-rate and high-speed communication," in *Proceedings of First International Conference on Distributed Framework and Applications, DFmA '08*, Oct. 2008, pp. 189–193.

[36] Z. Li, L. Chen, L. Zeng, S. Lin, and W. Fong, "Efficient encoding of quasi-cyclic low-density parity-check codes," *IEEE Transactions on Communications*, vol. 54, no. 1, pp. 71–81, Jan. 2006.

[37] L. Zeng, L. Lan, Y. Y. Tai, B. Zhou, S. Lin, and K. Abdel-Ghaffar, "Construction of nonbinary cyclic, quasi-cyclic and regular LDPC codes: a finite geometry approach," *IEEE Transactions on Communications*, vol. 56, no. 3, pp. 378–387, March 2008.

[38] Y.-H. Liu, X.-M. Wang, and J.-H. Ma, "Design of Quasi-Cyclic LDPC Codes Based on Euclidean Geometries," in *Proceedings of International Conference on Advanced Information Networking and Applications, AINA '09*, May 2009, pp. 207–211.

[39] X.-Y. Hu and E. Eleftheriou, "Binary representation of cycle Tanner-graph GF(2b) codes," in *Proceedings of IEEE International Conference on Communications, ICC '04*, vol. 1, June 2004, pp. 528–532 Vol.1.

[40] C. Xiao, J. Wu, S. Y. Leong, Y. R. Zheng, and K. B. Letaief, "A discrete time model for triply selective MIMO Rayleigh fading channels," *IEEE Transactions on Wireless Communications*, vol. 3, no. 5, pp. 1678–1688, Sep 2004.

[41] S. Loyka, "Channel capacity of MIMO architecture using the exponential correlation matrix," *Communications Letters, IEEE*, vol. 5, no. 9, pp. 369–371, Sep 2001.

[42] G. Marsagalia and T. A. Bray, "A convenient method for generating normal variables," *SIAM Rev.*, vol. 6, no. 3, pp. 260–264, July 1964.

[43] L. Staphorst, "Viterbi decoded linear block codes for narrowband and wideband wireless communication over mobile fading channels," *Masters Dissertation, University of Pretoria*, July 2005.

[44] Y. Zheng and C. Xiao, "Improved models for the generation of multiple uncorrelated Rayleigh fading waveforms," *Communications Letters, IEEE*, vol. 6, no. 6, pp. 256–258, Jun 2002.

[45] V. Tarokh, N. Seshadri, and A. Calderbank, "Space-time codes for high data rate wireless communication: performance criterion and code construction," *IEEE Transactions on Information Theory*, vol. 44, no. 2, pp. 744–765, Mar 1998.

[46] V. Tarokh, H. Jafarkhani, and A. R. Calderbank, "Correction to "Space-time block codes from orthogonal designs"," *IEEE Transactions on Information Theory*, vol. 46, no. 1, pp. 314–314, 2000.

[47] H. Jafarkhani, "A quasi-orthogonal space-time block code," *IEEE Transactions on Communications*, vol. 49, no. 1, pp. 1–4, Jan 2001.

[48] W. Su and X.-G. Xia, "Signal constellations for quasi-orthogonal space-time block codes with full diversity," *IEEE Transactions on Information Theory*, vol. 50, no. 10, pp. 2331–2347, Oct. 2004.

[49] J. Boutros, E. Viterbo, C. Rastello, and J. C. Belfiore, "Good lattice constellations for both Rayleigh fading and Gaussian channel," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 502–518, March 1996.

[50] X. Giraud, E. Boutillon, and J. C. Belfiore, "Algebraic tools to build modulation schemes for fading channels," *IEEE Transactions on Information Theory*, vol. 43, no. 3, pp. 938–952, May 1997.

[51] Y. Xin, Z. Wang, and G. B. Giannakis, "Space-time diversity systems based on linear constellation precoding," *IEEE Transactions on Wireless Communications*, vol. 2, no. 2, pp. 294–309, March 2003.

[52] Z. Liu, Y. Xin, and G. B. Giannakis, "Linear constellation precoding for OFDM with maximum multipath diversity and coding gains," *IEEE Transactions on Communications*, vol. 51, no. 3, pp. 416–427, 2003.

[53] E. Viterbo and F. Oggier, *Algebraic Number Theory and Code Design for Rayleigh Fading Channels*. Now publishers inc., 2004.

[54] H. El Gamal and M. Damen, "Universal space-time coding," *IEEE Transactions on Information Theory*, vol. 49, no. 5, pp. 1097–1119, May 2003.

[55] W. Zhang, X.-G. Xia, and P. C. Ching, "High-rate full-diversity space-time-frequency codes for MIMO multipath block-fading channels," in *Proceedings of IEEE Global Telecommunications Conference, GLOBECOM '05*, vol. 3, 2005, p. 5.

[56] K. P. Maré and B. T. Maharaj, "Performance Analysis of Modern Space-Time Codes on a MIMO-WiMAX Platform," in *Proceedings of IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, WIMOB '08*, 2008, pp. 139–144.

[57] R. M. Gray and D. L. Neuhoff, "Quantization," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2325–2383, Oct. 1998.

[58] G. F. Sage, "Performance of Multilevel PCM," *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-2, no. 4-Suppl, pp. 353–361, 1966.

[59] H. Viswanathan and R. Zamir, "On the whiteness of high-resolution quantization errors," *IEEE Transactions on Information Theory*, vol. 47, no. 5, pp. 2029–2038, 2001.

[60] A. Høst-Madsen and P. Händel, "Effects of sampling and quantization on single-tone frequency estimation," *IEEE Transactions on Signal Processing*, vol. 48, no. 3, pp. 650–662, 2000.

[61] *IEEE Standard for Floating-Point Arithmetic*, IEEE Std. 754-2008, 2008.

[62] J. Ahn, H.-N. Lee, and K. Kim, "Schnorr-Euchner sphere decoder with statistical pruning for MIMO systems," in *Proceedings of 6th International Symposium on Wireless Communication Systems, ISWCS '09*, Sept. 2009, pp. 619–623.

[63] H. Artes, "Reducing sphere decoder complexity by elliptical tree pruning," in *Proceedings of IEEE 5th Workshop on Signal Processing Advances in Wireless Communications*, July 2004, pp. 333–337.

[64] A. Chan and I. Lee, "A new reduced-complexity sphere decoder for multiple antenna systems," in *Proceedings of IEEE International Conference on Communications, ICC '02*, vol. 1, 2002, pp. 460–464.

[65] T. Cui, C. Tellambura, and W. Chen, "Reduced complexity sphere decoding using forcing rules," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers*, vol. 1, Nov. 2004, pp. 1218–1221 Vol.1.

[66] Z. Safar, W. Su, and K. J. R. Liu, "A fast sphere decoding algorithm for space-frequency block codes," *EURASIP J. Appl. Signal Process.*, vol. 2006, pp. 148–148, January 2006.

[67] H. Vikalo and B. Hassibi, "On the sphere decoding algorithm I. Expected complexity," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2806–2818, Aug 2005.

[68] ——, "On the sphere decoding algorithm II. Generalizations, second order statistics and applications to communications," *IEEE Transactions on Signal Processing*, vol. 53, no. 8, pp. 2819–2834, Aug 2005.

[69] M. Sellathurai and S. Haykin, *Space-Time Layered Information Processing for Wireless Communication*. Wiley-IEEE Press, April 2009.

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

[71] Y. de Jong and T. Willink, "Iterative tree search detection for MIMO wireless systems," *IEEE Transactions on Communications*, vol. 53, no. 6, pp. 930–935, June 2005.

[72] R. Wang and G. Giannakis, "Approaching MIMO channel capacity with soft detection based on hard sphere decoding," *IEEE Transactions on Communications*, vol. 54, no. 4, pp. 587–590, April 2006.

[73] 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

[74] S. Bang, B. Sheu, and C.-H. Chang, "Maximum likelihood sequence estimation of communication signals by a Hopfield neural network," in *Proceedings of IEEE International Conference on Neural Networks, IEEE World Congress on Computational Intelligence*, vol. 5, Jun-2 Jul 1994, pp. 3369–3374 vol.5.

[75] R. Altes, "Unconstrained minimum mean-square error parameter estimation with Hopfield networks," in *Proceedings of IEEE International Conference on Neural Networks*, Jul 1988, pp. 541–548 vol.2.

[76] S. Abe and A. Gee, "Global convergence of the Hopfield neural network with nonzero diagonal elements," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 42, no. 1, pp. 39–45, Jan 1995.

[77] S. V. Balakrishnan-Ayer, "Solving Combinatorial Optimization Problems Using Neural Networks with Applications in Speech Recognition," Ph.D. dissertation, Trinity College, Cambridge, 1991.

[78] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. Wiley-Interscience, October 2000.

[79] 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.