SPARSE GRAPH CODES ON A MULTI-DIMENSIONAL WCDMA PLATFORM

by

Jacobus David Vlok

Submitted in partial fulfilment of the requirements for the degree

Master of Engineering (Electronic Engineering)

in the

Department of Electrical, Electronic and Computer Engineering

Faculty of Engineering, Built Environment and Information Technology

UNIVERSITY OF PRETORIA

February 20, 2007

Supervisor: Professor L. P. Linde

I have seen that everything human has its limits and end no matter how extensive, noble and excellent; but Your commandment is exceedingly broad and extends without limits into eternity.

Psalm 119.96 Amplified

SUMMARY

SPARSE GRAPH CODES ON A MULTI-DIMENSIONAL WIDEBAND CODE DIVISION MULTIPLE ACCESS PLATFORM

by

Jacobus David Vlok

Supervisor: Professor L.P. Linde

Co-supervisor: L. Staphorst

Department of Electrical, Electronic and Computer Engineering

Master of Engineering (Electronic)

Digital technology has made complex signal processing possible in communication systems and greatly improved the performance and quality of most modern telecommunication systems. The telecommunication industry and specifically mobile wireless telephone and computer networks have shown phenomenal growth in both the number of subscribers and emerging services, resulting in rapid consumption of common resources of which the electromagnetic spectrum is the most important. Technological advances and research in digital communication are necessary to satisfy the growing demand, to fuel the demand and to exploit all the possibilities and business opportunities. Efficient management and distribution of resources facilitated by state-of-the-art algorithms are indispensable in modern communication networks.

The challenge in communication system design is to construct a system that can accurately reproduce the transmitted source message at the receiver. The channel connecting the transmitter and receiver introduces detrimental effects and limits the reliability and speed of information transfer between the source and destination. Typical channel effects encountered in mobile wireless communication systems include path loss between the transmitter and receiver, noise caused by the environment and electronics in the system, and fading caused by multiple paths and movement in the communication channel. In multiple access systems, different users cause interference in each other's signals and adversely affect the system performance.

To ensure reliable communication, methods to overcome channel effects must be devised and implemented in the system. Techniques used to improve system performance and capacity include temporal, frequency, polarisation and spatial diversity. This dissertation is concerned mainly with temporal or time diversity. Channel coding is a temporal diversity scheme and aims to improve the system error performance by adding structured redundancy to the transmitted message. The receiver exploits the redundancy to infer with greater accuracy which message was transmitted, compared with uncoded systems.

Sparse graph codes are channel codes represented as sparse probabilistic graphical models which originated in artificial intelligence theory. These channel codes are described as factor graph structures with bit nodes, representing the transmitted codeword bits, and bit-constrained or check nodes. Each constraint involves only a small number of code bits, resulting in a sparse factor graph with far fewer connections between bit and check nodes than the maximum number of possible connections. Sparse graph codes are iteratively decoded using message passing or belief propagation algorithms. Three classes of iteratively decodable channel codes are considered in this study, including low-density parity-check (LDPC), Turbo and repeat-accumulate (RA) codes.

The modulation platform presented in this dissertation is a spectrally efficient wideband system employing orthogonal complex spreading sequences (CSSs) to spread information sequences over a wider frequency band in multiple modulation dimensions. Special features of these spreading sequences include their constant envelopes and power output, providing communication range or device battery life advantages.

This study shows that multiple layer modulation (MLM) can be used to transmit parallel data streams with improved spectral efficiency compared with single-layer modulation, providing data throughput rates proportional to the number of modulation layers at performances equivalent to single-layer modulation. Alternatively, multiple modulation layers can be used to transmit coded information to achieve improved error performance at throughput rates equivalent to a single layer system.

Keywords: belief propagation, block Turbo codes, channel modelling, complex spreading sequences (CSSs), low-density parity-check (LDPC) codes, multi-dimensional mod-

ulation, repeat-accumulate (RA) codes, sparse graph channel coding, trellis structure, wireless communication

SAMEVATTING

YL GRAFIEKKODES OP 'n MEERVOUDIG-DIMENSIONELE WYEBAND KODE-VERDEELDE MULTI-TOEGANG PLATFORM

deur

Jacobus David Vlok

Studieleier: Professor L.P. Linde

Mede-studieleier: L. Staphorst

Departement Elektriese, Elektroniese en Rekenaar Ingenieurswese

Meester in Ingenieurswese (Elektronies)

Digitale tegnologie het komplekse seinverwerking in kommunikasiestelsels moontlik gemaak en het bygedra tot groot verbeterings in die werkverrigting en kwaliteit van die meeste hedendaagse telekommunikasiestelsels. Die telekommunikasie-industrie, en veral mobiele draadlose telefoon en rekenaar-netwerke, het merkwaardige groei in beide die aantal gebruikers en nuwe dienste getoon, wat gelei het tot die snelle uitputting van gemene hulpbronne, waaronder die frekwensie-spektrum die belangrikste is. Tegnologiese vooruitgang en navorsing in digitale kommunikasie is noodsaaklik om die groeiende aanvraag te bevredig, om aanvraag aan te vuur en om al die moontlikhede en besigheidsgeleenthede te eksploiteer. Effektiewe bestuur en verspreiding van hulpbronne wat deur innoverende algoritmes moontlik gemaak word, is onmisbaar in moderne kommunikasie-netwerke.

Die uitdaging in kommunikasiestelsel-ontwerp is om die stelsel sodanig saam te stel dat die versende boodskap akkuraat by die ontvanger herwin kan word. Die kanaal wat die sender en ontvanger verbind veroorsaak skadelike effekte en plaas 'n beperking op die betroubaarheid en spoed van inligting-oordrag tussen die bron en bestemming. Tipiese kanaaleffekte wat in mobiele draadlose kommunikasiestelsels voorkom sluit in padverlies tussen die sender en ontvanger, ruis wat deur die omgewing en elektronika in die stelsel veroorsaak word, en deining a.g.v. multipad en beweging in die kommunikasiekanaal. Verskillende gebruikers in multi-toegang stelsels veroorsaak steurnisse in

mekaar se seine en die stelsel se werkverrigting word daardeur benadeel.

Metodes om kanaaleffekte te oorkom moet ontwikkel en in die stelsel implementeer word om betroubare kommunikasie te verseker. Tegnieke om stelsel-werkverrigting en kapasiteit te verbeter sluit tyd-, frekwensie-, polarisasie- en ruimtelike diversiteit in. Hierdie verhandeling het hoofsaaklik te make met tyd-diversiteit. Kanaalkodering is 'n tyd-diversiteitskema en poog om stelsel-foutwaarskynlikheid te verbeter deur gestruktureerde oortolligheid in die versende boodskap in te bou. Die ontvanger wend die oortolligheid aan om met groter akkuraatheid af te lei watter boodskap gestuur is, vergeleke met ongekodeerde stelsels.

Yl grafiekkodes is kanaalkodes wat deur yl grafiese waarskynlikheidsmodelle, wat oorspronklik in kunsmatige intelligensieteorie aangewend is, voorgestel kan word. Dié kanaalkodes word deur faktor-grafiekstrukture beskryf met bis-nodi, wat die versende kodewoord-bisse voorstel, en bis-beperkte of kontrole-nodi. Elke beperking betrek slegs 'n klein aantal kodebisse, met die gevolg dat 'n yl faktor-grafiek met baie minder verbindings tussen bis- en beperkings-nodi as wat moontlik is, geskep word. Yl grafiekkodes word iteratief gedekodeer deur boodskap- of sekerheids-propagasie-algoritmes aan te wend. Drie groepe iteratief-dekodeerbare kanaalkodes word in hierdie studie ondersoek, naamlik lae-digtheid pariteit-kontrole (LDPK), Turbo en herhaal-akkumuleer (HA) kodes.

Die modulasieplatform wat in hierdie verhandeling voorgestel word is 'n spektraaleffektiewe wyeband stelsel wat gebruik maak van ortogonale komplekse spreisekwensies (KSS) om datastrome oor 'n wyer frekwensieband in meervoudige dimensies te sprei. Spesiale kenmerke van hierdie spreisekwensies sluit hulle konstante omhullings en drywingsuitsette in, wat voordele i.t.v. kommunikasie-reikwydte en toestel-battery-lewe inhou.

Hierdie studie bewys dat multi-laag-modulasie (MLM) aangewend kan word om parallelle datastrome te versend met verbeterde spektrale effektiwiteit vergeleke met enkel-laag-modulasie, terwyl data deurset-snelhede eweredig aan die aantal modulasielae gehandhaaf word met foutwaarskynlikheid ekwivalent aan enkel-laag-modulasie. Alternatiewelik kan multi-laag-modulasie aangewend word om gekodeerde inligting te versend om verbeterde foutwaarskynlikheidsvlakke te haal teen deurset-snelhede ek-

wivalent aan die van 'n enkellaagstelsel.

Sleutelwoorde: sekerheids-propagasie, blok Turbo kodes, kanaalmodellering, komplekse spreisekwensies (KSS), lae-digtheid pariteit-kontrole (LDPK) kodes, meervoudig-dimensionele modulasie, herhaal-akkumuleer (HA) kodes, yl grafiek kanaalkodering, traliestruktuur, draadlose kommunikasie

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

A-PDF	Amplitude Probability Density Function
AAC	Aperiodic Auto-Correlation
AC	Auto-Correlation
ACC	Aperiodic Cross-Correlation
AM	Amplitude Modulation
AWGN	Additive White Gaussian Noise
BCH	Bose, Ray-Chaudhuri, Hocquenchem
BER	Bit Error Rate
BM	Branch Metric
BPSK	Binary Phase Shift Keying
BTC	Block Turbo Code
CC	Cross Correlation
CDMA	Code Division Multiple Access
CE	Constant Envelope
CLL	Code Lock Loop
CSI	Channel State Information
CSS	Complex Spreading Sequence
DSB	Double Sideband
DSP	Digital Signal Processor
DSSS	Direct Sequence Spread Spectrum
ECC	Error Correcting Coding
EDGE	Enhanced Data Rates for GSM Evolution

FDD	Frequency Division Duplexing
FDMA	Frequency Division Multiple Access
FEC	Forward Error Correction
FFC	Flat Fading Channel
FHSS	Frequency Hopping Spread Spectrum
FM	Frequency Modulation
FZC	Frank-Zadoff-Chu
GCD	Greatest Common Divider
GCL	Generalised Chirp-Like
GPRS	General Packet Radio Service
GSM	Global System for Mobile Communication
HSDPA	High Speed Downlink Packet Access
HSUPA	High Speed Uplink Packet Access
IBP	Iterative Belief Propagation
IEEE	Institute of Electrical and Electronics Engineers
IIR	Infinite Impulse Response
ISI	Inter-Symbol Interference
ITU	International Telecommunication Union
LBC	Linear Block Code
LDPC	Low-Density Parity-Check
LI	Linearly Interpolated
LLR	Log Likelihood Ratio
LOS	Line-Of-Sight
MAP	Maximum A-Posteriori
MD	Multi-Dimensional
MDM	Multi-Dimensional Modulation
ML	Maximum Likelihood
MLM	Multi-Layered-Modulation
MP	Message Passing
MUI	Multi-User Interference

NLOS Non-Line-Of-Sight

OFDM Orthogonal Frequency Division Multiplexing

P/S Parallel-to-Serial
PAC Periodic Auto-Correlation
PCC Periodic Cross-Correlation
PDF Probability Density Function
PM Path Metric
PS Park-Park-Song-Suehiro
PSD Power Spectral Density
PSK Phase Shift Keying

QPSK Quadrature Phase Shift Keying

RA Repeat-Accumulate
RM Reed-Muller
RMS Root Mean Square
RS Reed-Solomon
RU Root-of-Unity
RUF Root-of-Unity Filtered
RV Random Variable

S/P Serial-to-Parallel
SDMA Space Division Multiple Access
SISO Soft-Input Soft-Output
SNR Signal-to-Noise-Ratio
SO Super-Orthogonal
SOVA Soft Output Viterbi Algorithm
SS Spread Spectrum

TC Turbo Code
TDMA Time Division Multiple Access

VA	Viterbi Algorithm
VoIP	Voice over Internet Protocol
WCDMA	Wideband Code Division Multiple Access
WLAN	Wireless Local Area Network
ZC	Zadoff-Chu
ZCC	Zero Cross-Correlation
2D	Two dimensional
4D	Four dimensional

NOMENCLATURE

$\alpha(t)$	Channel fading amplitude (temporal)
$\alpha_I(t)$	Inphase component of $\alpha(t)$
$\alpha_Q(t)$	Quadrature component of $\alpha(t)$
α_n	Channel fading amplitude corresponding to the n^{th} transmitted bit
$a_i(t,\tau)$	Amplification factor of i^{th} multipath signal component
a_k	Chu CSS's k^{th} main element
b_m	Filtered CSS's m^{th} interpolated element
B_C	Coherence bandwidth of channel
$BM_{s,s'}^{x_k}$	Branch metric connecting trellis states s and s' corresponding to bit x_k
BW_{signal}	Bandwidth of transmitted signal
$c(t)$	Complex spreading sequence (temporal)
$c_r(t)$	Real component of $c(t)$
$c_i(t)$	Imaginary component of $c(t)$
$c_{m,i}$	Code word C_m's i^{th} bit
$C_{bit}(m)$	Set of code bit positions participating in the m^{th} check bit
$C_{check}(n)$	Set of check bit positions participating in the n^{th} code bit
C_m	m^{th} valid code word
C_{scale}	Power-equality scaling factor used in channel model
C_{LOS}	LOS power constant used in channel model
$d(t)$	Complex input data stream
$d_i(t)$	Complex data stream's i^{th} MD component sequence
$d_I(t)$	Inphase component of $d(t)$
$d_Q(t)$	Quadrature component of $d(t)$
ε_b	Energy per uncoded bit

f_{bit}	Baseband data bit rate
f_c	Carrier frequency in Hertz
f_d	Instantaneous Doppler shift frequency
f_{samp}	Sampling frequency
f_D	Maximum Doppler spreading frequency
f_{DN}	Normalised Doppler spreading frequency
γ	RA code repetition factor
γ_i	Attenuation factor of i^{th} multipath signal component
g	Gap dimension in LDPC parity check matrix
$\mathbf{G}$	Generator matrix of channel code
$h(t,\tau)$	Time-varying impulse response of multipath channel
$\bar{h}_i$	i^{th} column vector of $\mathbf{H}$
$\mathbf{H}$	Parity check matrix of channel code
I_f	Interpolation factor of Doppler filter
k_η	AWGN scaling factor
k_p	Power delay profile coefficient
k_r	Real auto dot product constant of the CSS $c(t)$
k_i	Imaginary auto dot product constant of the CSS $c(t)$
k_{LU}	Constant of LU-decomposition complexity proportional to w_r
K	Dataword length
K_f	Number of serial bi-quad filters used in the Doppler filter
K_{lin}	Linear Rician factor
K_{Rice}	Rician factor measured in dB
K_{CSS}	Constant determined by the auto dot product of CSS $c(t)$
$\Lambda(x)$	LLR of x
$\Lambda_c(x)$	Channel LLR of x
$\Lambda_A(x)$	A-priori LLR of x
$\Lambda_E(x)$	Extrinsic LLR of x
$\Lambda_I(x)$	Intrinsic LLR of x

L	Length of CSS (number of spreading chips in the sequence)
L_p	Number of paths in the multipath channel model
L_t	Number of paths in the trellis considered during decoding
μ	Expected value of noise variable η
$m_{a,b}$	Message passed from a-layer to b-layer in factor graph
M	Channel code parameter; $M = N - K$
$\eta(t)$	Gaussian distributed noise process
n_s	Number of samples representing one spreading chip
n_{samp}	Number of samples representing one input bit in the channel
N	Codeword length
N_0	Single-sided noise PSD
$\phi(t)$	Channel phase response (temporal)
$\phi_{corr,ab}$	A-periodic correlation between sequences a and b
$\varphi_{corr,ab}$	Periodic correlation between sequences a and b
ϕ_k	Phase angle of k^{th} main element of CSS
ϕ_m	Phase angle of m^{th} interpolated (filtered) CSS
$\Psi_i(t)$	i^{th} MD base vector
$\Pi_{a,b}$	Interleaver translating from a-format to b-format
$p(x)$	Probability of variable x
$p(\mathbf{r}\|\mathbf{t})$	Posterior probability of $\mathbf{t}$
$p(\mathbf{t}\|\mathbf{r})$	Channel transition probability or likelihood function of $\mathbf{t}$
$\mathbf{P}$	Parity sub-matrix of $\mathbf{G}$
$P(t)$	Received power at time instance t
P_{drop}	Power difference between strongest multipath component and the component arriving at τ_{max}, measured in dB
$P_{drop,lin}$	Linear measure of P_{drop}
$P_e(bit)$	Probability of bit error
$P_{s,ave}$	Time-average power of $s(t)$
$P_{s,ins}(t)$	Instantaneous output power of $s(t)$ at time instance t
$P_{s,short}(t)$	Short-time-average output power of $s(t)$ at time instance $t = t_0 + t_s$

P_{LOS}	Power content of LOS signal component
P_{NLOS}	Power content of NLOS signal component
$PM^{x}_{l,k}$	l^{th} path metric corresponding to bit x_k
$PM^{f}_{s,i}$	Forward direction path metric at trellis coordinate (s, i)
$PM^{r}_{s,i}$	Reverse direction path metric at trellis coordinate (s, i)
$q^{t}_{m,n}$	Probability that the transmitted bit is t, given the information obtained from all check nodes, except node z_m
$r(t)$	MD received signal (temporal)
$\mathbf{r}$	Channel-corrupted received code word
r_c	CSS index number of a complete complex sequence
r_{cc}	CSS index number of one component (either real or imaginary) of a complex sequence
r	Integer index of CSS family member
$\bar{r}_m$	m^{th} received code word block
r_n	n^{th} received bit of $\mathbf{r}$
R_c	Channel code rate
σ_τ	RMS delay spread of multipath channel
σ^2	Variance of noise variable η
$s(t)$	Transmitted MD signal (temporal)
$\bar{s}$	Transmitted MD code word block
$S_{hh}(f)$	PSD function of a Doppler spreading process
$S_{m,i}$	m^{th} state at depth i in the LBC trellis
$\theta_i(t, \tau)$	Phase shift experienced by i^{th} multipath signal component
τ	Relative time shift between two sequences
τ_c	Positive discrete correlation time shift
τ_e	Time constant of multipath channel power delay profile
$\tau_i(t)$	Relative time delay of i^{th} multipath component at time instance t
τ_{max}	Maximum excess delay of multipath channel

$\mathbf{t}$	Transmitted code word
t_0	Reference time instance used in power calculations
t_n	n^{th} transmitted bit of $\mathbf{t}$
t_s	Time interval over which power is calculated
T_{samp}	Sampling period
T_C	Chip period
T_S	Symbol period
w_c	Column weight of parity check matrix or carrier frequency in rad/s
$w_{m,n}^t$	Probability that node z_m is satisfied if the transmitted bit is t
w_r	Row weight of parity check matrix
W_L	L^{th} primitive root of unity
$\mathbf{x}$	Information message word before transmission
$\mathbf{y}$	Received and decoded message word
ζ	Cross or interference term present in transmitted MD signal $s(t)$
ζ_n	n^{th} code bit node in a factor graph
z_m	m^{th} check node in a factor graph

CHAPTER 1

INTRODUCTION

1.1 CHAPTER SUMMARY

Communication is the basic prerequisite for interaction and dates back to the beginning of time. Many communication methods and devices have since been developed and improved and with the dawn of computer technology, communication exploded and birthed the information age. Several researchers addressed the communication problem and devised new technologies to transfer information from one entity to another. Shannon [1] stated that the fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.

A general communication system consists of an information source, transmitter, channel, receiver and information destination [1, 2]. The communication channel introduces undesirable effects into the transmitted information and limits the reliability of reception and the speed of communication through the channel. Reliable communication over noisy channels is however possible if the rate of communication through the channel is below the channel capacity threshold.

Digital technology has made complex signal processing in communication systems possible and resulted in the development of sophisticated modulation and source and channel coding techniques. Source coding involves the mapping of the output of an information source into digital symbols in order to transmit and recover the information message at the receiver. Channel coding techniques aim to improve the quality of reception of the transmitted message by adding structured redundancy. The receiver uses the redundancy in the received sequence to decide which sequence was most probably transmitted and the probability of correct reception is thus increased. By using appropriate encoding and decoding systems, the channel capacity limit can be reached, though a trade-off between error performance, coding speed and complexity exists.

The communication system considered in this study is a digital wideband wireless

multi-user access system appropriate for deployment within mobile cellular telephone and computer networks. The technology can however be adapted and used in any digital communication system. The focus of this study is the performance evaluation of a currently popular class of iteratively decodable codes in research circles, namely sparse graph codes, on the wideband communication platform. This chapter provides an overview of telecommunication history, channel coding techniques and the technologies and standards employed in modern telecommunication networks. Finally, the motivation, contribution and layout of the dissertation are given.

1.2 HISTORY OF TELECOMMUNICATION

Telecommunication is the exchange of information over a distance across a communication medium, including free space, air, copper lines, coaxial cable or optical fibre. The most primitive example of communication (excluding smoke and lighthouse signals) is probably the semaphore, the first visual telegraphy system built in 1792 which used beams of wood or flags to signal messages visible over a maximum range of 30 kilometres. More sophisticated telecommunication includes the transfer of data across satellite and mobile phone networks. This section gives a brief overview of telecommunication history and the accompanying technological developments. More detailed historical reviews can be found in [3, 4, 5, 6].

1.2.1 ELECTRICAL TELEGRAPHY

Telegraphy experiments in the 18th century and the invention of the electric battery in 1799 by Volta were followed by the invention of the electrical telegraph in 1837 by Samuel Morse, who also developed one of the first digital variable length source codes, the Morse code. Soon after, several telegraphy companies installed widespread telegraph lines including the first transcontinental cable in 1858.

1.2.2 INVENTION OF TELEPHONY

In 1876, Alexander Bell patented a design for the telephone and founded the Bell Telephone Company which quickly built large telephone networks in North America and Britain. After Bell's patent expired several other telephone companies were founded and interconnection issues arose. Network infrastructure such as central offices were constructed to accommodate the growing customer base and the human-operated switch-

board gave way to electromechanical switches which were eventually replaced by digital signalling and switching technologies.

1.2.3 WIRELESS TELEGRAPHY

Several scientists and researchers, including Oersted, Faraday, Gauss, Maxwell, Tesla, Hertz and Marconi, contributed to the field of wireless telegraphy or radio technology and conducted many experiments. Electromagnetism was discovered in 1820 by Oersted and wireless telegraphy principles explained and demonstrated by Tesla in 1893. Marconi patented a radio telegraph system in 1897 and established commercial wireless communication with transatlantic range in 1901.

1.2.4 COMMERCIAL RADIO AND TELEVISION

Radio transmission technology evolved from spark-gap modulation to two modulation methods invented by Edwin Armstrong, namely amplitude and frequency modulation (AM and FM), along with new technological advances and hardware inventions. The first AM radio station went on air in 1920 and the first FM communication system was demonstrated in 1933. With advances in moving pictures recording technology, the first television system was demonstrated in 1929 and commercial broadcasting commenced in 1936.

1.2.5 COMMUNICATION SATELLITES

The invention of the transistor and integrated circuit (IC) and many other factors, including rocket technology, upper-atmosphere research and defence programs of the world's superpowers contributed to the development of communication satellites. The first satellites were launched in the 1950s and in 1965 commercial satellite communication services were introduced. Further technological advances resulted in the launch of many more satellites, supporting navigational, reconnaissance and meteorological systems and services, all relying on communication technology.

1.2.6 COMPUTER NETWORKS

The idea of centralised computing using a mainframe computing device with remote dumb terminals arose during the 1940s. Computer technology advances and especially the development of packet switching protocols in the 1960s resulted in the ARPANET

(Advanced Research Projects Agency Network), a computer network connecting research institutes. More networks connecting universities and research bodies were developed and merged during the 1980s after the TCP/IP (Transmission Control Protocol/Internet Protocol) suite gained popularity. Many other computer networks sprang up and merged with existing networks and with the development of more data transfer protocols (e.g. SMTP (Simple Mail Transfer Protocol) and HTTP (Hypertext Transfer Protocol)), the Internet came into existence.

Fixed line networks were extended to include wireless networks, making Internet connections available via wireless local area networks (WLANs) implementing standards such as IEEE (Institute of Electrical and Electronics Engineers) 802.11 (WiFi) with data rates currently up to 54 Mbps and IEEE 802.16 (WiMax) at a maximum data rate of 75 Mbps. Technologies such as VoIP (Voice over Internet Protocol) and instant messaging across the Internet emerged, circumventing and threatening legacy voice services and the companies providing them.

1.2.7 MOBILE TELEPHONY

Half-duplex analog radio phone technology which operated without a cellular network dates back to 1946 and is sometimes referred to as zero generation (0G) technology. Other non-cellular mobile phone technologies include cordless and satellite phones. The cellular concept originated in 1947 at Bell Labs, though the first commercial cellular network trials were performed in 1978 after much research and development.

First generation (1G) cellular networks were introduced in the 1980s and supported voice communication only using analog FM transmission and FDMA/FDD (Frequency Division Multiple Access/Frequency Division Duplexing) to accommodate multiple users and channels. Frequency-reuse and handover strategies were also implemented across these networks. AMPS (Advanced Mobile Phone System) in North America, TACS (Total Access Communication System) and NMT (Nordic Mobile Telephony) in Europe were examples of 1G networks.

Second generation (2G) networks evolved from 1G networks in the 1990s and are purely digital networks using digital modulation and TDMA/FDD (Time Division Multiple Access/Frequency Division Duplexing - used in the GSM (Global System for Mobile Communication) and IS-136 cellular standards) or CDMA/FDD (Code Division Mul-

tiple Access/FDD - used in the IS-95 or cdmaOne standard). 2G technology supports voice, text messaging (SMS) and circuit-switched data (up to 9.6 kbps) communications with added digital signal processing benefits such as FEC (Forward Error Correction), encryption and value added services such as caller identification. Standards making higher data speeds possible on the 2G platform were also implemented and are sometimes referred to as 2.5G, e.g. GPRS (General Packet Radio Service) and 2.75G, e.g. EDGE (Enhanced Data rates for GSM Evolution).

Third generation (3G) networks co-exist with 2G networks and refer to the latest network technologies that are capable of providing higher data rates (up to 2 Mbps), using wideband modulation techniques with increased user capacity. Internet access, e-mail, multimedia streaming, video telephony and instant messaging are all services available on 3G mobile devices, including mobile handsets and computers. The two main 3G standards include UMTS (Universal Mobile Telecommunication System) WCDMA (Wideband Code Division Multiple Access) adopted by GSM countries and CDMA2000, a MC-CDMA (Multi Carrier CDMA) technology, adopted by countries using IS-95. Standards that improve the 3G specifications are sometimes referred to as 3.5G, e.g. HSDPA (High Speed Downlink Packet Access) providing a maximum downlink data rate of 14.4 Mbps and uplink rate of 384 kbps, and 3.75G e.g. HSUPA (High Speed Uplink Packet Access) providing a maximum uplink data rate of 5.76 Mbps.

1.2.8 CONVERGING COMMUNICATION NETWORKS

"Fourth generation" (4G) or "Beyond-3G" (B3G) networks are the next planned wireless access technologies that will succeed 3G through the LTE (Long Term Evolution) programme. Even higher data rates (100 Mbps to 1 Gbps) and new services such as mobile video, which will follow with the new bandwidth availability, will be provided. 4G is envisaged as an AIP (all-IP) packet switched network technology as opposed to the mixed circuit and packet switched 3G networks. The multiple access technology being considered for implementation in 4G is OFDM (Orthogonal Frequency Division Multiplexing).

A single device will be able to connect to several networks (2G, 3G and 4G) simultaneously and FMC (Fixed-Mobile Convergence) will provide seamless integration between different networks and standards (including cellular and WLAN communications), aiming to create a global wireless network. The different network generations will probably

continue to co-exist (or at least backward compatibility will be provided) providing a wide range of services and devices depending on customer preference. The IMT-2000 (International Mobile Telecommunications 2000) standard launched by the ITU (International Telecommunication Union) is an example of the effort to create universal compatibility between different networks and operators to enhance inter-connectivity, though many issues will have to be resolved in order to reach true compatibility between networks, services and standards.

1.3 MULTIPLE USER ACCESS TECHNOLOGY

Several techniques have been developed to allow coexistence of different wireless communication systems sharing common resources such as the radio spectrum. The same techniques are used to provide access to several concurrent users in one system. The multiple access techniques mainly used are listed below.

- TDMA where different users are each allocated a separate time slot to transmit or receive data
- FDMA where different users are each allocated a unique carrier frequency or channel on which information is communicated
- CDMA where each user is assigned an orthogonal spreading code and multiple users share the same bandwidth
- SDMA (Space Division Multiple Access) where the communication network is physically configured to increase user capacity e.g. multiple and/or directional antennas are used to split cells into smaller sectors, allowing frequency reuse in the same cell

Multiple access techniques are combined to assign resources economically to different users e.g. in GSM networks, groups of multiple users are assigned to different frequency channels (FDM) where each frequency channel is split up in several time slots to accommodate each user (TDMA). Alternatively, networks can allocate all available resources to all users currently using the network, allowing higher data rates per user during non-peak traffic times. Network signalling and very accurate timing are major design concerns in networks employing these techniques. Duplexing techniques are based on frequency division (FDD) or time division (TDD) to provide separate uplink and downlink channels.

1.3.1 SPREAD SPECTRUM

Spread spectrum (SS) is an important technique forming the basis of CDMA networks. Spreading a signal over a wider frequency band using an orthogonal spreading code has many advantages including interference immunity (processing gain), low transmission spectral power density and allowing multiple users to share the same bandwidth.

FHSS (Frequency Hopping SS) and DSSS (Direct Sequence SS) are the two versions of SS currently in use. FHSS systems spread the transmitted information over a wide frequency band by transmitting a narrow band signal on a constantly changing carrier frequency (according to a frequency hopping pattern known by the intended receiver). DSSS systems spread the information message by modulating the message with a pseudo-random spreading sequence, also known to the receiver.

The characteristics of the spreading codes used in an SS system determine how many users can be accommodated in the system and the quality of communication in the multi-user environment. The pairwise cross correlation between all the spreading codes in the code family should be as close to zero as possible to minimise multi-user interference (MUI). The code length determines the maximum number of users in the system and the spreading factor, though other factors (e.g. implementation issues such as memory and decimal precision) may influence the code length. For optimal performance, all spreading codes used in a CDMA system should be aligned perfectly and hence the need for accurate timing in such systems.

Several SS code families including Kasami, Gold and Walsh [4, 7] have been developed. In this study, families of complex spreading sequences (CSSs) [8, 9, 10] are considered due to their superior correlation properties and signal envelope and spectral characteristics.

1.3.2 WIDEBAND COMMUNICATION

As the number of users requiring wireless data service (including voice) and the data speed per user constantly increase, finite resources (like radio spectrum regulated by communication authorities) are being used up. Higher data rate communications require larger frequency bandwidths or more intelligent and efficient use of available spectrum e.g. using multilevel modulation techniques or orthogonal code division. Spread

spectrum allows more users to share the same frequency band, though each contending user might still experience data speeds obtainable by narrower band communications. New technological innovation and multi-user access techniques are required to address these demands. Unlicensed frequency bands and the open spectrum approach are new paradigms suggested by some authors [11].

1.4 CHANNEL CODING TECHNIQUES

In 1948, Shannon [1] proved that there exist block codes that can achieve an arbitrarily small error probability for transmission across a memoryless channel, if the communication rate does not exceed the capacity of the channel. Shannon, though proving the theory, did not describe these codes or a way to find them. Several researchers invented coding schemes, not only in a search to reach the Shannon limit, but also to improve communication systems by creating practical coding schemes. Two main classes of codes, namely block and convolutional codes, emerged with several decoding algorithms. This section provides a brief overview of some of the best known channel coding techniques, including coded modulation and space-time diversity techniques. Technical details on these and other codes, including encoding and decoding algorithms, can be found in [12].

1.4.1 BLOCK CODES

An (N, K) block code is a coding scheme which encodes K-symbol data words into N-symbol code words by adding $N-K$ redundancy symbols to each data word. Encoding normally involves algebraic manipulation like matrix multiplication with the generator or parity check matrices of the code on a DSP (Digital Signal Processor). Decoding can be accomplished using algebraic search algorithms, syndrome or list decoding or by employing trellis-based (e.g. the Viterbi algorithm [13]) or factor graph algorithms (e.g. belief propagation [14]). Several block coding schemes have been developed and this section discusses three examples of linear block codes, including Hamming (a binary code), Reed-Solomon (a finite field code) and LDPC (an iteratively decodable sparse graph code) codes.

1.4.1.1 HAMMING CODES

Richard Hamming invented a linear error correcting code (ECC) during the 1940s in an effort to improve the reliability of a punch card reader, the input device of a Bell Labs electromechanical computer and published the code in [15]. Hamming codes are single error correcting and double error detecting codes, improving the older, less robust parity and repetition codes.

1.4.1.2 REED-SOLOMON CODES

Reed-Solomon (RS) codes, invented by Irvin Reed and Gustav Solomon in 1960 [16], are finite field polynomial block codes and a subset of the BCH (Bose, Ray-Chaudhuri, Hocquenchem) code family. At the time RS codes were invented, digital technology was not advanced enough to make the codes implementable and a practical decoding algorithm for RS codes was first invented in 1968 by Berlekamp [17]. RS codes are strong burst error correcting codes used in mass storage technology (e.g. CD and DVD) and deep space communications (e.g. the Voyager and Mars Pathfinder missions in conjunction with convolutional codes).

1.4.1.3 LDPC CODES

Low-density parity-check (LDPC) codes were originally invented by Gallager in the early 1960s [18] and were the first capacity-approaching codes. LDPC codes were however not implementable due to technological limitations and were forgotten for over the next 30 years, when during the 1990s several researchers rediscovered these remarkable codes [19]. LDPC codes form a class of linear block codes which can be decoded using iterative algorithms on the code's factor graph. Recent research produced regular and irregular codes [20] and techniques to optimise these codes for different communication channels [14]. Error performance of an LDPC code within 0.0045 dB from the Shannon limit was reported in 2001 [21].

1.4.2 CONVOLUTIONAL CODES

As an alternative to block codes, Peter Elias [22] invented codes with random tree structures implemented relatively easy using shift registers. The encoder is normally described by a constraint length indicating the shift register length and the code rate K/N indicating that K input symbols are encoded into N output symbols. The encoder can perform encoding continually and indefinitely, though strategies are normally

implemented to reset the encoder at regular time intervals to improve performance. Several decoding algorithms have been developed, including sequential decoding (developed by Wozencraft [23]) and maximum likelihood decoding (e.g. the Viterbi algorithm [24]). Convolutional codes can be described as generator polynomials, state and trellis diagrams and can be designed as recursive or non-recursive and systematic or non-systematic codes.

1.4.3 CONCATENATED AND HYBRID CODES

Serial concatenated or product codes were introduced in [25], where the output of one encoder (outer code) is interleaved and fed as input to a following encoder (inner code). Forney [26] introduced a concatenated code in 1966 using an outer RS and inner convolutional code, which is still used in space exploration missions, operating within 2 dB from the Shannon limit.

In 1993, Berrou et al. [27] introduced a parallel concatenated coding scheme known as Turbo codes, consisting of 2 parallel convolutional codes iteratively decoded with error performance within 0.2 dB from the Shannon limit. Turbo codes ignited a new research interest into iteratively decodable codes and several turbo coding schemes were developed, including block turbo codes [28] and several turbo-like codes [29].

1.4.4 CODED MODULATION

Combining channel coding and modulation into a single process was introduced in [30]. Coded modulation basically involves the expansion of the signal constellation of an uncoded communication system to include coding redundancy, while the bandwidth requirement remains unchanged. Effectively the data rate remains constant (comparing the data throughput rate of the uncoded and coded systems) and the Euclidean distance between transmitted symbols is increased providing a coding gain. An example coded modulation technique is trellis coded modulation (TCM) invented by Ungerboeck [31] employing a mapping-by-set-partitioning method to assign codeword bits to symbols. The Viterbi algorithm is used in the decoder by assigning each transmitted symbol to a trellis path.

1.4.5 SPACE TIME CODING

Space time coding is an antenna diversity technique to improve the reliability of wireless data transmission. Two main space time coding techniques are space time trellis codes (STTCs) [32] and space time block codes (STBCs) [33]. The STTC technique involves encoding the data with a trellis-based (convolutional) code and transmitting the encoded signal over multiple antennas and time slots to obtain both coding and diversity gains. The STBC technique provides only a diversity gain by transmitting data blocks over multiple antennas.

1.5 MOTIVATION AND CONTRIBUTION

The telecommunication industry will continue to grow and create new business opportunities. New technological advances and research are necessary to satisfy growing needs, to fuel the growth and to exploit all the possibilities. When developing a new communication system or modulation technique a study must be undertaken to test and evaluate the performance of each subsystem and the system as a whole under different conditions to predict the system's behaviour when deployed as a real system. The study should include extensive simulations and comparisons to theoretical communication systems.

1.5.1 PREVIOUS RESEARCH

This study builds on previous research done on a wideband multidimensional modulation technique presented in [8, 9, 34, 35] not yet implemented in a real system and on recent research on iterative decodable channel coding (or sparse graph coding) techniques including turbo codes [27, 28, 36], LDPC codes [14, 37, 38] and RA (Repeat-Accumulate) codes [29, 39, 40]. The combination and configuration possibilities arising from implementing coding schemes on modulation platforms are virtually endless and this study is limited to only a small fraction of those combinations.

1.5.2 AIM AND ASSUMPTIONS

The main objectives of this study are listed below.

- Investigation of the MD (Multi-Dimensional) communication platform by developing and simulating the basic 4D (Four Dimensional) system in a multi-user

communication environment

- Investigation of the output power and correlation characteristics of the CSSs used on the MD platform
- Development of a reliable channel model to accurately approximate realistic wide-band wireless radio channels
- Investigation and comparison of sparse graph coding schemes in a search for optimal coding performance and data throughput in terms of implementation complexity and error performance by performing extensive simulation studies

The main assumptions made to perform the simulations in this study are included in the list below.

- Perfect synchronisation between all users in the CDMA cell under consideration is maintained. The spreading codes of all users therefore line up perfectly with no relative time shift between any of the users' codes.
- Perfect spreading sequence code lock and carrier phase synchronisation for all users are maintained at all times during simulations.
- The instantaneous fading amplitude and phase angle (collectively known as CSI (Channel State Information)) are perfectly extracted from the channel, when CSI knowledge is required in the simulations.
- A multipath channel with a static power delay profile is assumed.

In a practical communication system the above assumptions are ideal, though subsystems are needed to ensure such conditions. These subsystems will introduce overhead and additional complexity to the basic communication system. Only the basic communication system and coding schemes are considered in this study and any additional complexity not directly incurred by the basic system is ignored when comparing the complexity of different coding schemes.

1.5.3 CONTRIBUTION

The main focus of this study is the error performance evaluation of the three channel coding classes (BTC (Block Turbo Code), LDPC and RA codes) on a multidimensional

wideband communication system in a realistic multi-user multipath environment approximating modern wireless communication system scenarios. The communication platform considered in this study provides an alternative modulation technique to OFDM and similar technologies for application in B3G and 4G communication systems. This section summarises all the contributions made to the field of digital communications through this study.

1.5.3.1 PUBLICATIONS

The publications that form part of this study's contributions and a short summary of the contents of these publications are given below.

- *"Power and Spectrally Efficient Four-Dimensional Super-Orthogonal WCDMA Building Block for Next Generation Wireless Applications"* [41] published in *IEEE Communications Letters*, contains details regarding the design of the basic 4D communication platform and the CSSs used (presented in Chapter 2 of this dissertation).

- *"Performance of a Block-Turbo-Coded Four-Dimensional WCDMA Communication System in AWGN (Additive White Gaussian Noise) and Multipath Fading"* [42] submitted for publication in the *IEE Electronics Letters*, contains details regarding the design of the MD Turbo coding system using constituent block codes (presented in Chapter 4 of this dissertation) and the error performance results of the uncoded and coded 4D communication system (presented in Chapter 5).

- *"Performance of a Quasi-Synchronous Four-Dimensional Super-Orthogonal W-CDMA Modulator for Next Generation Wireless Applications"* [43] accepted for publication in the *South African Journal of Science*, presents uncoded performance results of the 4D communication system with comparisons to M-QAM systems in terms of error performance and data throughput rates.

1.5.3.2 CONTRIBUTION TO MD COMMUNICATION

Although the basic MD modulation technique and platform were presented in [8, 9, 34, 35], this dissertation contributes to the understanding of the 4D building block and CSSs used by presenting a thorough study and comparison of two CSS families with given sequence lengths. The correlation characteristics of the CSSs (and components of the CSSs separately) and their effect on the error performance of the 4D system are

investigated and explained, providing deeper insight into the functioning of CSSs on the MD platform.

1.5.3.3 CONTRIBUTION TO CHANNEL MODELLING

The channel model presented in this dissertation is the well known Clarke model [44, 45]. The model presented in [45] uses a fixed IIR (Infinite Impulse Response) filter to simulate Doppler effects in the transmitted signal waveform. This dissertation presents a channel model by combining the design in [45] with a very accurate Doppler filter design based on the model presented in [46]. The resultant channel model is the Clarke-model with an easily variable Doppler spreading frequency.

1.5.3.4 CONTRIBUTION TO CHANNEL CODING

This study considers existing channel codes, though the contribution to channel coding research lies in the application of these codes on a new MD platform and a coding scheme comparison in terms of performance and implementation complexity. The turbo coding scheme presented in this dissertation is a unique combination of existing algorithms, including a 3D decoder structure based on [28] with SISO (Soft-Input Soft-Output) constituent decoders employing algorithms based on [12, 36] with slight alterations. An alternative trellis expurgation and decoding technique (similar to the technique presented in [45]) is presented. The message passing algorithm used to decode RA codes presented in [40] is implemented with slightly different decoding scheduling.

1.5.3.5 SOFTWARE DEVELOPED

During the course of this study, the complete simulation platform including the basic multidimensional modulation building block, the multipath fading channel simulator and all encoding and decoding modules were developed in *C++* and extensive simulations were performed on several computers and multi-node clusters.

This dissertation presents the first error performance simulation results of the 4D communication system under realistic channel conditions, including uncoded and coded performance.

1.6 LAYOUT OF DISSERTATION

This dissertation includes descriptions of all the components of the communication system being studied, including the basic communication platform, the channel model, the channel codes and the error performance results. Each chapter deals with one of these subjects as follows.

Chapter 2 deals with the wideband multiple user communication system and the spreading sequences used. 2D (Two dimensional) and 4D transmitter and receiver structures are described mathematically and the actual implementations are also considered. The correlation and spectral characteristics of two complex spreading sequence (CSS) families, including special classes of the GCL (Generalised Chirp-Like) and ZCC (Zero Cross Correlation) families, are studied and compared to predict the error performance results in a synchronous CDMA environment.

Chapter 3 focuses on the channel model. In order to evaluate the error performance of the communication system, a realistic channel model must be used to perform simulations. A multi-user, multipath channel simulation model is described which can approximate AWGN and frequency selective fading effects, typically encountered in a wide band wireless communication system. The multipath channel simulator consists of three paths with exponentially decaying power delay profile for each user with each path described in terms of average received power, Rician factor and Doppler spreading frequency. The simulation platform is also validated against mathematical descriptions of the channel.

Chapter 4 considers channel coding and specifically sparse graph codes with code design and parameters chosen to match the multidimensional communication system. An introduction to the basic channel coding objective, channel decoding strategies and concepts pertaining to iterative decoding, e.g. log likelihood ratios (LLR) and SISO decoding, are discussed. Block turbo codes with Reed-Muller (RM) constituent codes and the trellis-based SISO decoding algorithm and encoding and decoding strategies of low-density parity-check (LDPC) and RA codes are also explained in detail. The construction of trellises and factor graphs and the message passing (MP) based algorithms are also discussed.

Chapter 5 focuses on the simulated error performance results of the uncoded MD system employing both GCL and ZCC CSSs and the coded 4D system employing ZCC CSSs. The coded simulation platform is described and theoretical BER (bit error rate) curves, including capacity limit curves, are given. Each coding scheme (described in Chapter 4) implemented on the wideband platform (described in Chapter 2) is evaluated under different channel conditions (described in Chapter 3) and compared.

Chapter 6 contains conclusions drawn from the study and areas of possible future research building on this study are identified. Finally, addenda including more correlation graphs of the CSSs used, a detailed description of the belief propagation algorithm and a list and summary of the *C++* classes developed during this study, are provided.

CHAPTER 2

WCDMA PLATFORM

2.1 CHAPTER SUMMARY

This chapter describes the multidimensional wideband code division multiple access (WCDMA) platform consisting of four dimensional (4D) transmitter and receiver structures introduced in [35]. To understand the 4D system, a 2D system is presented, analysed and extended to form the 4D system. The 4D transmitter transmits four parallel data streams by spreading the data with a complex spreading sequence (CSS) and then modulating the spreaded data onto quadrature carriers.

A $4m$-dimensional transmitter can be created by placing m 4D transmitters in parallel. Alternatively m users, each with a 4D transmitter, can be accommodated in a CDMA environment. The length and correlation characteristics of the CSS family employed determines the maximum value of m. The CSSs are filtered by a constant envelope (CE) root-of-unity (RU) technique [47] to provide the CSSs with a near Nyquist bandwidth and constant complex envelope which is an attractive characteristic resulting in more efficient use of power amplifiers in communication systems [48].

The spreading sequences used are the main factor determining the performance of the system. Two classes of CSSs (constant envelope generalised chirp-like (CE-GCL) and zero cross-correlation (ZCC)) [8, 49] are considered by evaluating and comparing their time, spectral and correlation characteristics. The WCDMA system and CSSs used also depend on perfect synchronisation by assuming that a pilot channel is available in the system.

2.2 TWO DIMENSIONAL SYSTEM

2.2.1 MATHEMATICAL ANALYSIS

In this section a complex 2D SS communication system is evaluated mathematically assuming an ideal channel and by viewing the data and spreading sequences as complex mathematical functions as in [50].

2.2.1.1 TWO DIMENSIONAL TRANSMITTER

The 2D SS transmitter is shown in figure 2.1. A complex data stream $d(t) = d_1(t) + jd_2(t)$ is spread by a complex spreading sequence $c(t) = c_r(t) + jc_i(t)$.

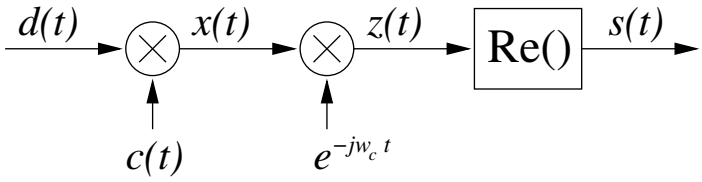

Figure 2.1: Mathematical 2D transmitter structure

The spreaded data stream is given by

$$\begin{aligned} x(t) &= d(t)c(t) = [d_1(t) + jd_2(t)]\,[c_r(t) + jc_i(t)] \\ &= [d_1(t)c_r(t) - d_2(t)c_i(t)] + j\,[d_1(t)c_i(t) + d_2(t)c_r(t)]\,, \end{aligned} \tag{2.1}$$

and the modulated spreaded data stream

$$\begin{aligned} z(t) &= x(t)e^{-jw_c t} \\ &= x(t)\,[cos(w_c t) - jsin(w_c t)] \\ &= [d_1(t)c_r(t) - d_2(t)c_i(t)]\,cos(w_c t) + [d_2(t)c_r(t) + d_1(t)c_i(t)]\,sin(w_c t) \\ &\quad + j\,\{[d_2(t)c_r(t) + d_1(t)c_i(t)]\,cos(w_c t) - [d_1(t)c_r(t) - d_2(t)c_i(t)]\,sin(w_c t)\}\,, \end{aligned} \tag{2.2}$$

with $j = \sqrt{-1}$ the complex operator and w_c the carrier frequency.

The transmitted signal $s(t)$ is then given by

$$\begin{aligned} s(t) &= Re\,\{z(t)\} \\ &= [d_1(t)c_r(t) - d_2(t)c_i(t)]\,cos(w_c t) + [d_2(t)c_r(t) + d_1(t)c_i(t)]\,sin(w_c t). \end{aligned} \tag{2.3}$$

2.2.1.2 TWO DIMENSIONAL RECEIVER

The 2D SS receiver is shown in figure 2.2. Assuming an ideal channel, the received signal $r(t)$ is identical to the transmitted signal $s(t)$ (see figure 2.1).

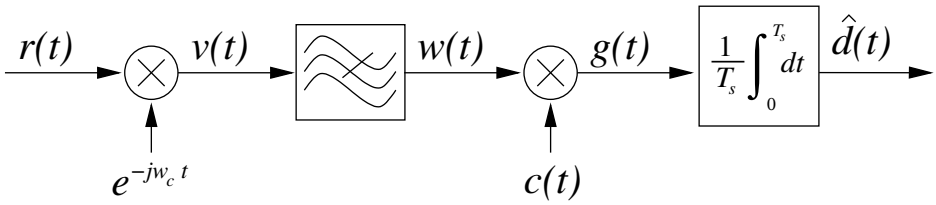

Figure 2.2: Mathematical 2D receiver structure

The received signal is demodulated to obtain

$$\begin{aligned} v(t) &= r(t)e^{-jw_c t} \\ &= r(t)\left[cos(w_c t) - jsin(w_c t)\right] \\ &= \left[d_1(t)c_r(t) - d_2(t)c_i(t)\right]cos^2(w_c t) - j\left[d_2(t)c_r(t) + d_1(t)c_i(t)\right]sin^2(w_c t) \\ &\quad + \left\{\left[d_2(t)c_r(t) + d_1(t)c_i(t)\right] - j\left[d_1(t)c_r(t) - d_2(t)c_i(t)\right]\right\}cos(w_c t)sin(w_c t), \end{aligned} \tag{2.4}$$

and then low-pass filtering $v(t)$ gives

$$w(t) = \frac{1}{2}\left[d_1(t)c_r(t) - d_2(t)c_i(t)\right] - j\frac{1}{2}\left[d_2(t)c_r(t) + d_1(t)c_i(t)\right]. \tag{2.5}$$

To recover the original transmitted data bits from the baseband signal $w(t)$ the dot product operation is performed as shown below.

$$\begin{aligned} \hat{d}(t) &= w(t)\cdot c(t) \\ &= \frac{1}{T_s}\int_0^{T_s} w(t)c(t)\,dt \\ &= \frac{1}{2T_s}\int_0^{T_s}\left[d_1(t)c_r(t) - d_2(t)c_i(t) - jd_2(t)c_r(t) - jd_1(t)c_i(t)\right]\left[c_r(t) + jc_i(t)\right]dt \\ &= \frac{1}{2T_s}\int_0^{T_s}\left[c_r^2(t) + c_i^2(t)\right]\left[d_1(t) - jd_2(t)\right]dt \\ &= K_{css}\left[d_1(t) - jd_2(t)\right] \end{aligned} \tag{2.6}$$

Notice that $d_1(t) - jd_2(t)$ is constant over a symbol period T_s. K_{css} is a constant determined by the auto dot products of the CSS components $c_r(t)$ and $c_i(t)$. In the case of orthonormal components K_{css} will equal unity. The original transmitted symbol $d(t) = d_1(t) + jd_2(t)$ can therefore be obtained from $\hat{d}(t) = d_1(t) - jd_2(t)$ by sampling at the end of each symbol period T_s.

2.2.2 ACTUAL TRANSMITTER AND RECEIVER STRUCTURES

The actual transmitter and receiver structures are based on the mathematical model by mapping the complex functions via orthogonal vectors (including carriers and spreading sequences) into the real world. In the mathematical model complex components of all the input vectors are transmitted in both the real and imaginary planes on both quadrature carriers, though in the real world this is not necessarily the case (see equations 2.2 and 2.3).

2.2.2.1 TWO DIMENSIONAL TRANSMITTER

The 2D transmitter is essentially a QPSK (quadrature phase shift keying) modulator as shown in figure 2.3. The input sequence $d(t)$ is split into two parallel data streams $d_I(t)$ and $d_Q(t)$ by a serial-to-parallel (S/P) converter after which a pulse shaping circuit processes each branch to create pulses $x_I(t)$ and $x_Q(t)$. The two pulses are then modulated onto quadrature carriers creating inphase and quadrature components $z_I(t)$ and $z_Q(t)$ respectively. The sum of the latter two components is the transmitted signal $s(t)$ carrying 2 data bits per channel use (assuming unbalanced or dual-channel transmission).

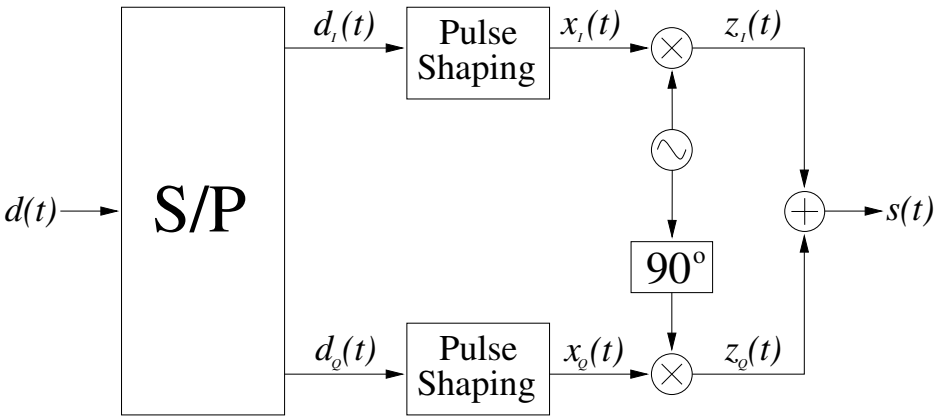

Figure 2.3: Basic dual-channel QPSK transmitter structure

The 2D SS transmitter shown in figure 2.4 is formed by replacing each pulse shaping circuit in figure 2.3 with a SS multiplier. The complex data signal $d(t) = d_I(t) + jd_Q(t)$ is transmitted by setting $d_1(t) = d_I(t)$ and $d_2(t) = d_Q(t)$ for dual-channel transmission or $d_1(t) = d_2(t) = d_I(t) = d_Q(t)$ for balanced transmission. Continuing from equation

2.3 and assuming balanced transmission, the transmitted symbol can be expressed as:

$$\begin{aligned} s(t) &= d_1\left[c_r(t) - c_i(t)\right] cos(w_c t) + d_1\left[c_r(t) + c_i(t)\right] sin(w_c t) \\ &= d_1(t)c_1(t)cos(w_c t) + d_1(t)c_2(t)sin(w_c t), \end{aligned} \tag{2.7}$$

by defining orthogonal spreading codes (see [50]):

$$c_1(t) = c_r(t) - c_i(t) \tag{2.8}$$

$$c_2(t) = c_r(t) + c_i(t). \tag{2.9}$$

For dual-channel transmission the transmitted signal is

$$s(t) = d_1(t)c_1(t)cos(w_c t) + d_2(t)c_2(t)sin(w_c t). \tag{2.10}$$

In general $c_1(t)$ and $c_2(t)$ need not be defined as in equations 2.8 and 2.9; the correlation characteristics of the spreading sequences used determine whether and how well the SS communication system will work.

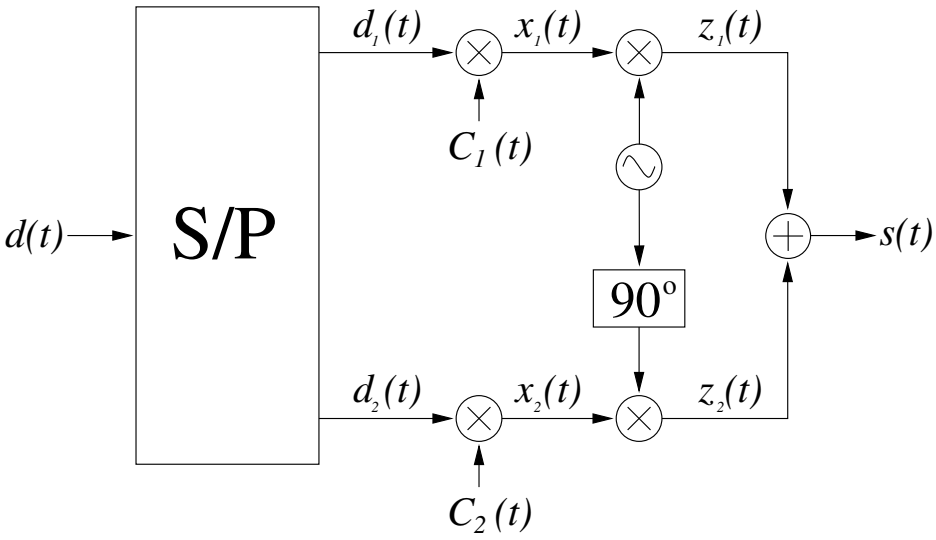

Figure 2.4: Dual-channel two dimensional SS transmitter structure

2.2.2.2 TWO DIMENSIONAL RECEIVER

The 2D receiver structure is a correlation-type demodulator [7] shown in figure 2.5. Assuming no channel effects, the received signal $r(t)$ equals the transmitted signal $s(t)$ given in equation 2.10.

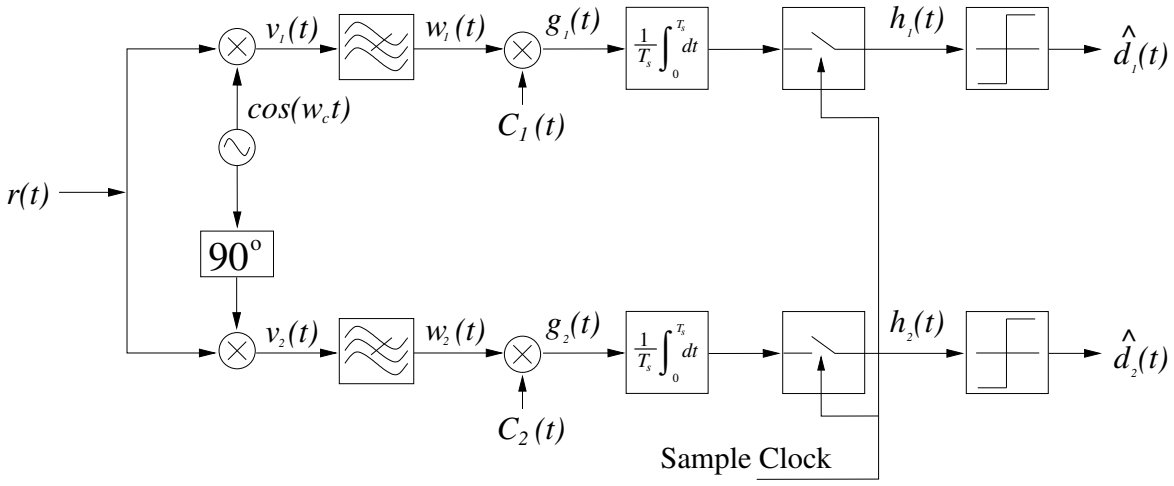

Figure 2.5: Two dimensional correlation-type SS receiver structure

The received signal is coherently demodulated and with perfect phase lock on the transmitted carrier frequency w_c, the following signals are obtained:

$$\begin{aligned} v_1(t) &= r(t)cos(w_c t) \qquad (2.11)\\ &= d_1(t)c_1(t)cos^2(w_c t) + d_2(t)c_2(t)sin(w_c t)cos(w_c t) \\ v_2(t) &= r(t)sin(w_c t) \qquad (2.12)\\ &= d_1(t)c_1(t)cos(w_c t)sin(w_c t) + d_2(t)c_2(t)sin^2(w_c t), \end{aligned}$$

which after low-pass filtering become

$$\begin{aligned} w_1(t) &= \frac{1}{2}d_1(t)c_1(t) \qquad (2.13)\\ w_2(t) &= \frac{1}{2}d_2(t)c_2(t). \qquad (2.14) \end{aligned}$$

The dot product operation is then performed via multiply, integrate and dump circuits to obtain after every symbol period

$$\begin{aligned} h_1(t) &= \frac{1}{T_s}\int_0^{T_s} w_1(t)c_1(t)\,dt \qquad (2.15)\\ &= K_{css,1}d_1(t) \\ h_2(t) &= \frac{1}{T_s}\int_0^{T_s} w_2(t)c_2(t)\,dt \qquad (2.16)\\ &= K_{css,2}d_2(t), \end{aligned}$$

with $K_{css,1}$ and $K_{css,2}$ respectively inphase and quadrature constants depending on the auto dot products of the spreading sequences $c_1(t)$ and $c_2(t)$. The hard limited outputs $\hat{d}_1(t)$ and $\hat{d}_2(t)$ are then obtained by conventional threshold decision devices.

2.3 FOUR DIMENSIONAL COMMUNICATION SYSTEM

The four dimensional communication system is closely related to the two dimensional system. In fact, the 4D system is equivalent to two 2D systems operating concurrently and in this section the 4D system will be evaluated accordingly.

2.3.1 MATHEMATICAL ANALYSIS

2.3.1.1 FOUR DIMENSIONAL TRANSMITTER

By placing two 2D transmitters (see figure 2.1) in parallel, a 4D transmitter is obtained as shown in figure 2.6.

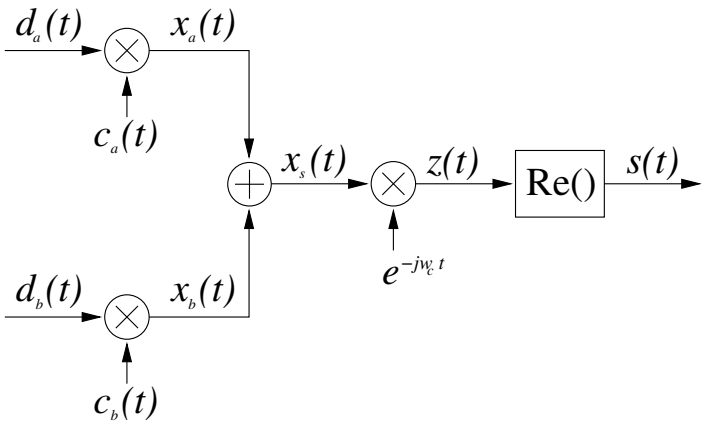

Figure 2.6: Mathematical 4D transmitter structure formed by placing two 2D transmitters in parallel

The input to the 4D transmitter is two complex data signals $d_a(t) = d_1(t) + jd_2(t)$ and $d_b(t) = d_3(t) + jd_4(t)$ containing 4 data bits. Two complex spreading sequences $c_a(t) = c_{ar}(t) + jc_{ai}(t)$ and $c_b(t) = c_{br}(t) + jc_{bi}(t)$ are used to spread the input signals $d_a(t)$ and $d_b(t)$ respectively. The transmitted signal can be written as:

$$\begin{aligned} s(t) &= Re\left\{[x_a(t) + x_b(t)]\, e^{-jw_c t}\right\} \\ &= [d_1(t)c_{ar}(t) - d_2(t)c_{ai}(t) + d_3(t)c_{br}(t) - d_4(t)c_{bi}]\, cos(w_c t) \\ &\quad + [d_2(t)c_{ar}(t) + d_1(t)c_{ai}(t) + d_4(t)c_{br}(t) + d_3(t)c_{bi}]\, sin(w_c t). \end{aligned} \tag{2.17}$$

The transmitted signal can be demodulated using a complex correlation-type receiver similar to the structure in figure 2.5. The main disadvantage of using the 4D transmitter

in figure 2.6 is the fact that 2 CSSs are required. By making a small adjustment an alternative 4D transmitter is formed that only requires a single CSS.

2.3.1.2 ALTERNATIVE FOUR DIMENSIONAL TRANSMITTER

The alternative 4D transmitter is shown in figure 2.7 where the modulator is split up such that the real and imaginary components of e^{-jw_ct} are respectively multiplied onto the real and imaginary branches.

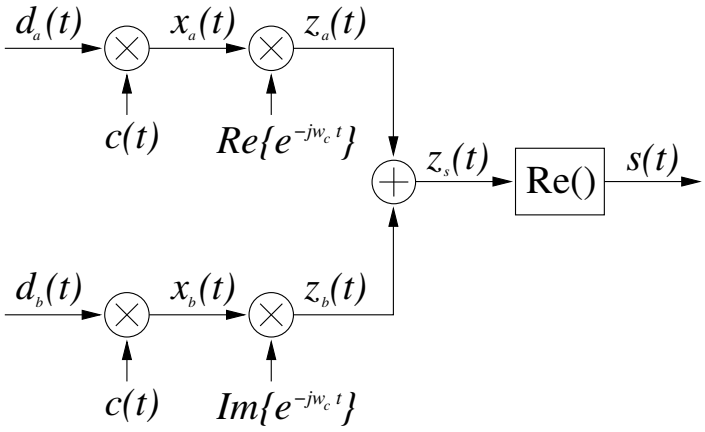

Figure 2.7: Alternative mathematical 4D transmitter structure

The complex input data signals $d_a(t) = d_1(t) + jd_2(t)$ and $d_b(t) = d_3(t) + jd_4(t)$ are multiplied with the CSS $c(t) = c_r(t) + jc_i(t)$ to obtain

$$x_a(t) = [d_1(t)c_r(t) - d_2(t)c_i(t)] + j\,[d_2(t)c_r(t) + d_1(t)c_i(t)] \quad (2.18)$$

$$x_b(t) = [d_3(t)c_r(t) - d_4(t)c_i(t)] + j\,[d_4(t)c_r(t) + d_3(t)c_i(t)]\,, \quad (2.19)$$

and after modulation

$$\begin{aligned} z_a(t) &= x_a(t)cos(w_ct) \quad (2.20)\\ &= [d_1(t)c_r(t) - d_2(t)c_i(t)]\,cos(w_ct) + j\,[d_2(t)c_r(t) + d_1(t)c_i(t)]\,cos(w_ct)\\ z_b(t) &= -x_b(t)sin(w_ct) \quad (2.21)\\ &= -[d_3(t)c_r(t) - d_4(t)c_i(t)]\,sin(w_ct) - j\,[d_4(t)c_r(t) + d_3(t)c_i(t)]\,sin(w_ct). \end{aligned}$$

The transmitted signal $s(t)$ is then

$$\begin{aligned} s(t) &= Re\,\{z_a(t) + z_b(t)\} \quad (2.22)\\ &= [d_1(t)c_r(t) - d_2(t)c_i(t)]\,cos(w_ct) - [d_3(t)c_r(t) - d_4(t)c_i(t)]\,sin(w_ct). \end{aligned}$$

2.3.1.3 FOUR DIMENSIONAL RECEIVER

To demodulate and despread the 4D signal transmitted by the structure in figure 2.7, the receiver configuration shown in figure 2.8 is used.

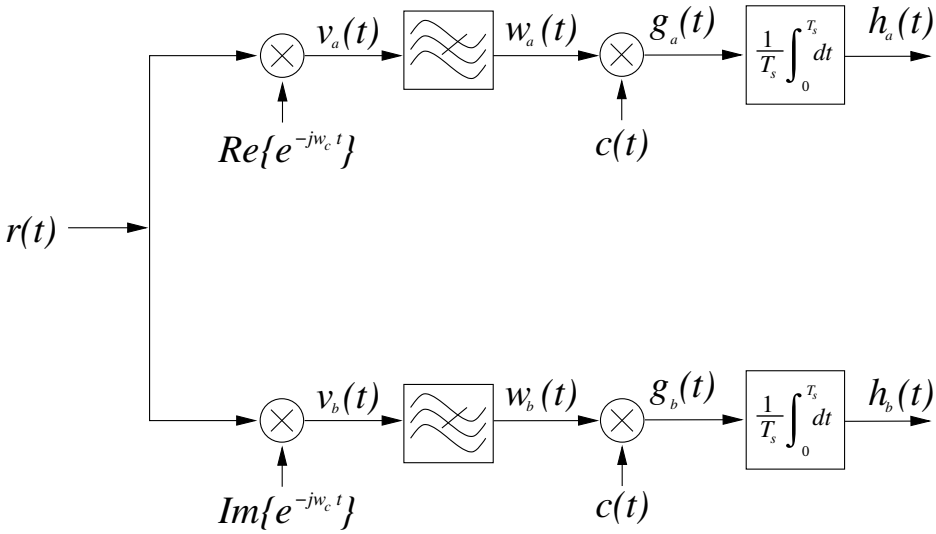

Figure 2.8: Mathematical 4D receiver structure

Assuming an ideal channel, the received signal $r(t)$ is equal to the transmitted signal $s(t)$ given in equation 2.22. After demodulation the signals

$$\begin{aligned} v_a(t) &= r(t)Re\left\{e^{-jw_c t}\right\} \qquad (2.23)\\ &= \left[d_1(t)c_r(t) - d_2(t)c_i(t)\right]cos^2(w_c t) - \left[d_3(t)c_r(t) - d_4(t)c_i(t)\right]sin(w_c t)cos(w_c t)\\ v_b(t) &= r(t)Im\left\{e^{-jw_c t}\right\} \qquad (2.24)\\ &= \left\{\left[d_1(t)c_r(t) - d_2(t)c_i(t)\right]cos(w_c t)sin(w_c t) - \left[d_3(t)c_r(t) - d_4(t)c_i(t)\right]sin^2(w_c t)\right\}, \end{aligned}$$

are obtained and after filtering

$$\begin{aligned} w_a(t) &= \frac{1}{2}\left[d_1(t)c_r(t) - d_2(t)c_i(t)\right] \qquad (2.25)\\ w_b(t) &= \frac{1}{2}\left[-d_3(t)c_r(t) + d_4(t)c_i(t)\right]. \qquad (2.26) \end{aligned}$$

The dot product operations are described by

$$\begin{aligned} h_a(t) &= w_a(t)\cdot c(t) \qquad (2.27)\\ &= \frac{1}{2T_s}\int_0^{T_s}\left[d_1(t)c_r(t) - d_2(t)c_i(t)\right]\left[c_r(t) + jc_i(t)\right]dt\\ &= \frac{1}{2T_s}\int_0^{T_s}\left\{\left[d_1(t)c_r^2(t) - jd_2(t)c_i^2(t)\right] - c_r(t)c_i(t)\left[d_2(t) - jd_1(t)\right]\right\}dt \end{aligned}$$

$$\begin{aligned} h_b(t) &= w_b(t) \cdot c(t) \\ &= \frac{1}{2T_s}\int_0^{T_s} \left[-d_3(t)c_r(t) + d_4(t)c_i(t)\right]\left[c_r(t) + jc_i(t)\right] dt \\ &= \frac{1}{2T_s}\int_0^{T_s} \left\{\left[-d_3(t)c_r^2(t) + jd_4(t)c_i^2(t)\right] + c_r(t)c_i(t)\left[d_4(t) - jd_3(t)\right]\right\} dt, \end{aligned} \tag{2.28}$$

and if $\int_0^{T_s} c_r(t)c_i(t)\,dt = 0$, which is valid only for CSSs with orthogonal complex components, the above equations can be simplified to

$$\begin{aligned} h_a(t) &= \frac{1}{2T_s}\int_0^{T_s} \left[d_1(t)c_r^2(t) - jd_2(t)c_i^2(t)\right] dt \\ &= k_r d_1(t) - jk_i d_2(t) \end{aligned} \tag{2.29}$$

$$\begin{aligned} h_b(t) &= \frac{1}{2T_s}\int_0^{T_s} \left[-d_3(t)c_r^2(t) + jd_4(t)c_i^2(t)\right] dt \\ &= -k_r d_3(t) + jk_i d_4(t), \end{aligned} \tag{2.30}$$

with k_r and k_i the real and imaginary auto dot products of the CSS $c(t) = c_r(t) + jc_i(t)$ respectively. By employing further decision circuitry the original transmitted signals $d_a(t)$ and $d_b(t)$ can be recovered from $h_a(t)$ and $h_b(t)$.

2.3.2 ACTUAL TRANSMITTER AND RECEIVER STRUCTURES

2.3.2.1 FOUR DIMENSIONAL TRANSMITTER

As in the 2D case (see section 2.2.2.1), the 4D transmitter is essentially a QPSK modulator (see figure 2.3) with each pulse shaping circuit replaced by a 2D SS module as shown in figure 2.9. The input sequence $d(t)$ is split up into four parallel data streams $d_1(t)$ to $d_4(t)$ by an S/P converter after which the upper two streams ($d_1(t)$ and $d_2(t)$) and the lower two streams ($d_3(t)$ and $d_4(t)$) are processed identically but separately. Two orthogonal spreading codes are used to spread each stream group as follows: data streams $d_1(t)$ and $d_2(t)$ are respectively multiplied by orthogonal spreading codes $c_r(t)$ and $c_i(t)$ to obtain $x_1(t)$ and $x_2(t)$ which are summed to obtain the inphase component $y_1(t)$. The quadrature component $y_2(t)$ is obtained likewise from data streams $d_3(t)$ and $d_4(t)$. After modulating the inphase and quadrature components onto quadrature carriers, the modulated components $z_1(t)$ and $z_2(t)$ are summed to produce the transmitted signal

$$\begin{aligned} s(t) &= \left[d_1(t)c_r(t) + d_2(t)c_i(t)\right] cos(w_c t) \\ &\quad + \left[d_3(t)c_r(t) + d_4(t)c_i(t)\right] sin(w_c t), \end{aligned} \tag{2.31}$$

carrying 4 data bits per channel use.

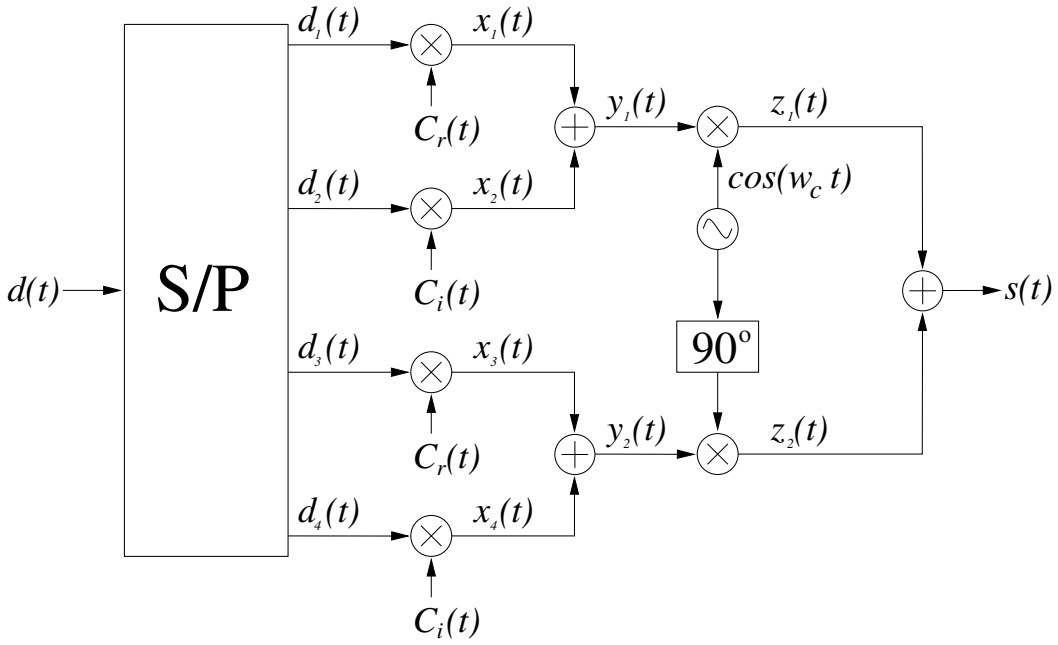

Figure 2.9: Four dimensional DSSS transmitter structure

2.3.2.2 FOUR DIMENSIONAL RECEIVER

As in the 2D case (see section 2.2.2.2) the 4D receiver is a correlation-type demodulator [7] as shown in figure 2.10. Assuming that $r(t) = s(t)$ given in equation 2.31 and following the same argument as for the analysis of the 4D receiver in section 2.3.1, the signals after carrier synchronisation are

$$\begin{aligned} v_1(t) &= [d_1(t)c_r(t) + d_2(t)c_i(t)]\, cos^2(w_c t) \\ &\quad + [d_3(t)c_r(t) + d_4(t)c_i(t)]\, sin(w_c t)cos(w_c t) \qquad (2.32) \\ v_2(t) &= [d_1(t)c_r(t) + d_2(t)c_i(t)]\, cos(w_c t)sin(w_c t) \\ &\quad + [d_3(t)c_r(t) + d_4(t)c_i(t)]\, sin^2(w_c t), \qquad (2.33) \end{aligned}$$

and after low-pass filtering

$$\begin{aligned} w_1(t) &= \frac{1}{2}\,[d_1(t)c_r(t) + d_2(t)c_i(t)] \qquad (2.34) \\ w_2(t) &= \frac{1}{2}\,[d_3(t)c_r(t) + d_4(t)c_i(t)]\,. \qquad (2.35) \end{aligned}$$

After performing the functions of despreading, integrating and sampling, the signals

$$\begin{aligned} h_1(t) &= k_r d_1(t) \qquad (2.36) \\ h_2(t) &= k_i d_2(t) \qquad (2.37) \end{aligned}$$

$$h_3(t) = k_r d_3(t) \tag{2.38}$$
$$h_4(t) = k_i d_4(t), \tag{2.39}$$

are obtained (assuming perfectly orthogonal CSSs) from which the original data signals are extracted by using a decision element.

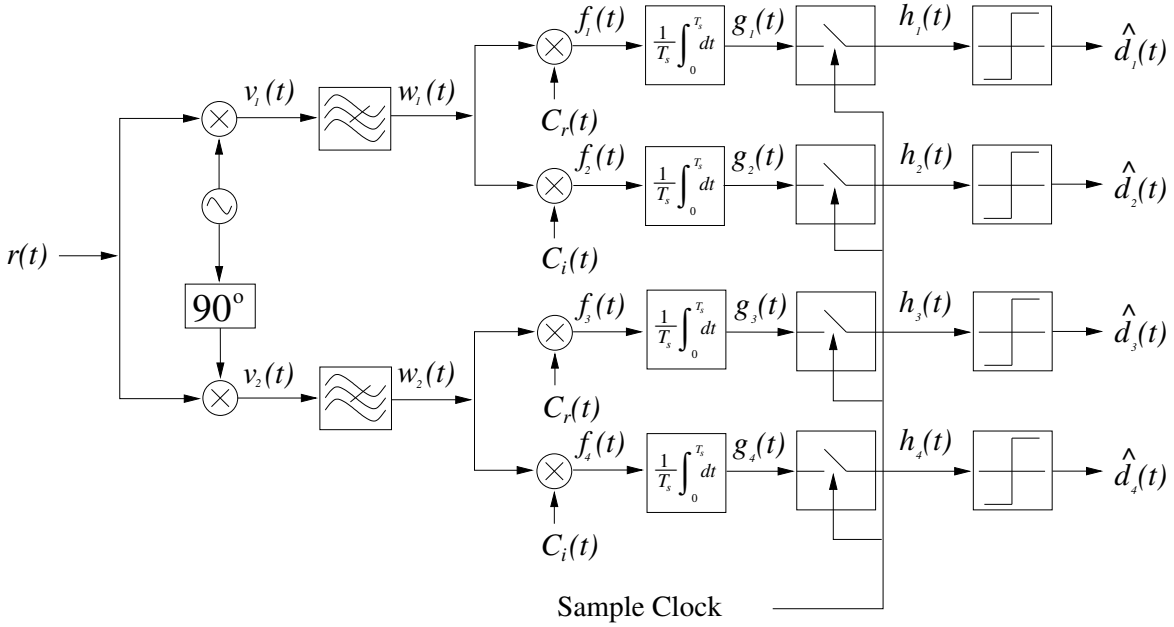

Figure 2.10: Four dimensional DSSS receiver structure

The performance of both the 2D and 4D WCDMA communication systems presented in this section depends primarily on the correlation characteristics of the CSSs. The base vectors of the multidimensional space are the CSSs and it is therefore important to employ CSSs with excellent cross correlation characteristics between CSSs (for the 2D case) and between real and imaginary components of CSSs (for the 4D case).

2.4 COMPLEX SPREADING SEQUENCES

The use of complex spreading sequences (CSSs) has become an alternative to traditional orthogonal spreading sequences in multi-user communication systems due to the superior correlation characteristics of CSSs. The real and imaginary components of a CSS are typically transmitted via quadrature carriers to remove correlation interference between components of a sequence and therefore a single CSS can be used to spread two independent data streams in a dual-channel QPSK system.

Several classes of CSSs have been developed, although only Zadoff-Chu (ZC), CE-GCL (a ZC derivative) and ZCC (a PS derivative) CSSs will be considered in this study. The CE-GCL and ZCC classes are respectively derived from ZC and PS sequences using a linear and non-linear RU filtering technique. The former case will be used to illustrate the technique and the performance results (see Chapter 5) will mainly focus on communication systems employing the latter class of sequences. This section contains mathematical descriptions, correlation and spectral characteristics of ZC, GCL and ZCC CSSs and reference to literature is made where neccesary.

2.4.1 ZADOFF-CHU SEQUENCES

Polyphase codes of perfect square lengths exhibiting a periodic auto-correlation (PAC) function equal to zero except for a single maximum value at $\tau = 0$ were presented by Frank and Zadoff in [51]. These codes were extended to include codes of arbitrary length by Chu in [10], although anticipated by Zadoff in an earlier patent (see [52]). Frank-Zadoff-Chu (FZC) sequences are defined as:

$$a_k = \begin{cases} W_L^{\frac{k^2}{2}+qk} & for\ L\ even \\ W_L^{\frac{k(k+1)}{2}+qk} & for\ L\ odd \end{cases} \tag{2.40}$$

with the sequence length L and the sequence element index number (or chip number) $k = 0, 1, 2, \ldots, L-1$. W_L is the L^{th} primitive root of unity defined as

$$W_L = \mathrm{e}^{-j2\pi r/L} \tag{2.41}$$

with r an integer relatively prime to L, meaning that $GCD(L, r) = 1$. Additional linear phase shifts of the form $e^{j2\pi qk/L}$ can be introduced into the code by setting q to a non-zero integer value without affecting the code's correlation characteristics [10]. By setting q to a constant integer, r corresponds to a single member in the sequence family whose size equals the number of unique relatively prime sets (L, r).

By setting $q = 0$ the sequences originally presented by Chu [10] defined by

$$a_k = \begin{cases} exp\left\{\frac{jr\pi k^2}{L}\right\} & for\ L\ even \\ exp\left\{\frac{jr\pi k(k+1)}{L}\right\} & for\ L\ odd \end{cases} \tag{2.42}$$

are obtained. All Chu CSS family members start and end in the complex coordinate $(1, 0)$, making each CSS perfectly symmetric. The real and imaginary components and

the phase transition diagram of a non-interpolated baseband Chu sequence with $L = 63$ and $r = 1$ is shown in figure 2.11 and the corresponding complex envelope is shown in figure 2.12.

It is clear from figure 2.12 that all sequence elements lie on the complex unity circle (i.e. all sequence elements are roots of unity) with large phase jumps between subsequent elements. The sequence consists only of a limited number of unique complex pairs (the number of unique complex pairs equals $(L + 1)/4$ for uneven L).

2.4.1.1 APERIODIC CORRELATION CHARACTERISTICS

The aperiodic correlation between two discrete time signals $s_a[lT_{samp}]$ and $s_b[lT_{samp}]$ is calculated by computing partial dot products between the two signals by shifting them relative to each other by an amount $\tau = lT_{samp}$ in time (l is an integer time shift parameter and T_{samp} is the sample spacing). The aperiodic correlation is then defined as:

$$\phi_{corr,ab}[\tau] = \begin{cases} \sum_{k=0}^{L-1+l} s_a[(k-l)T_{samp}]s_b^*[kT_{samp}] & -L+1 \leq l < 0 \\ \sum_{k=0}^{L-1-l} s_a[kT_{samp}]s_b^*[(k+l)T_{samp}] & 0 \leq l \leq L-1 \\ 0 & |l| > L \end{cases} \tag{2.43}$$

where L is the length of both sequences s_a and s_b and s_b^* is the complex conjugate of s_b. The length of $\phi_{corr,ab}$ is $(2L - 1)$ samples.

The AAC (aperiodic auto-correlation) and ACC (aperiodic cross-correlation) functions of a Chu sequence $c_1 = r_1 + j(i_1)$ (and the complex components separately) with $L = 63$ and $r = 1$ are shown in figure 2.13. Using both the real (r_1) and imaginary (i_1) components of the sequence c_1, results in a single correlation spike at $l = 0$ with small side lobes as shown in figure 2.13(a). If the real and imaginary components are used separately (see figure 2.13(b) and (c) respectively) the spike at $l = 0$ is reduced significantly (approximately 50 % compared to the complex case in figure 2.13(a)) with side lobes approximately the same as in the complex case. The correlation between the real and imaginary components (figure 2.13(d)) shows small lobes approximately the same amplitude as for the complex case.

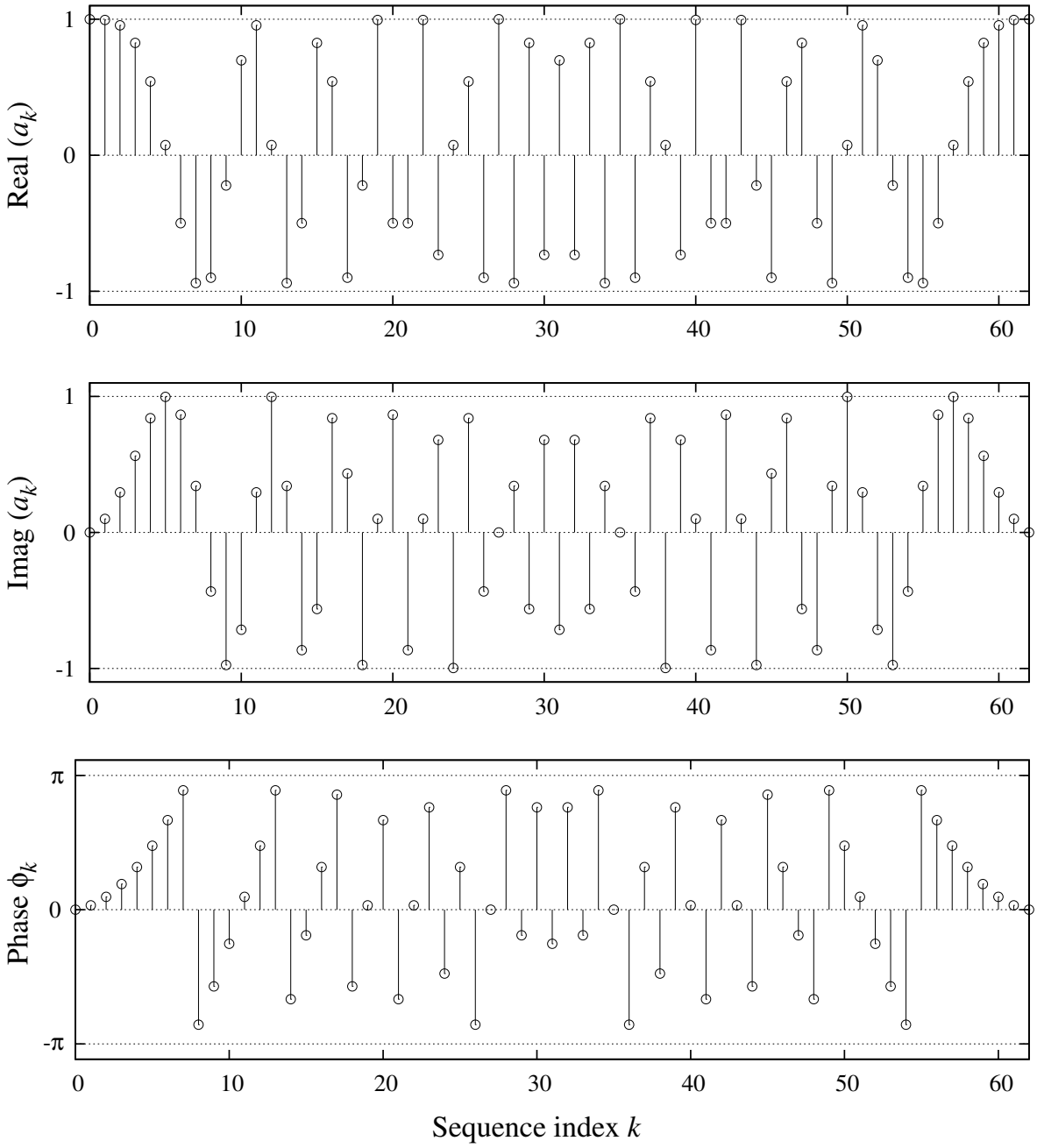

Figure 2.11: The figure shows the primary roots-of-unity of an unfiltered Chu CSS for $L = 63$ and $r = 1$ projected on the real and imaginary axes and the corresponding phase angles

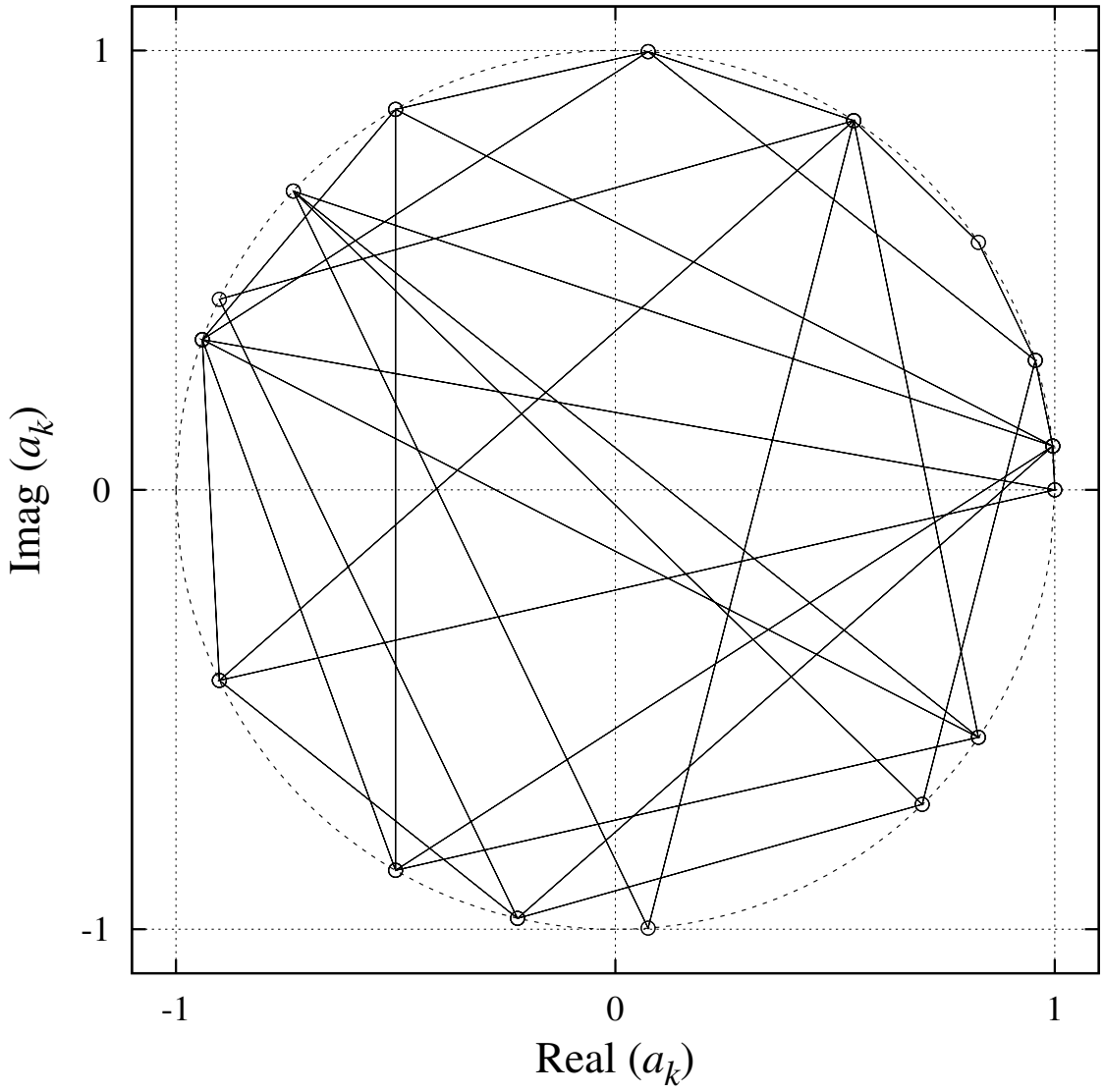

Figure 2.12: Complex envelope of an unfiltered non-interpolated Chu CSS for $L = 63$ and $r = 1$

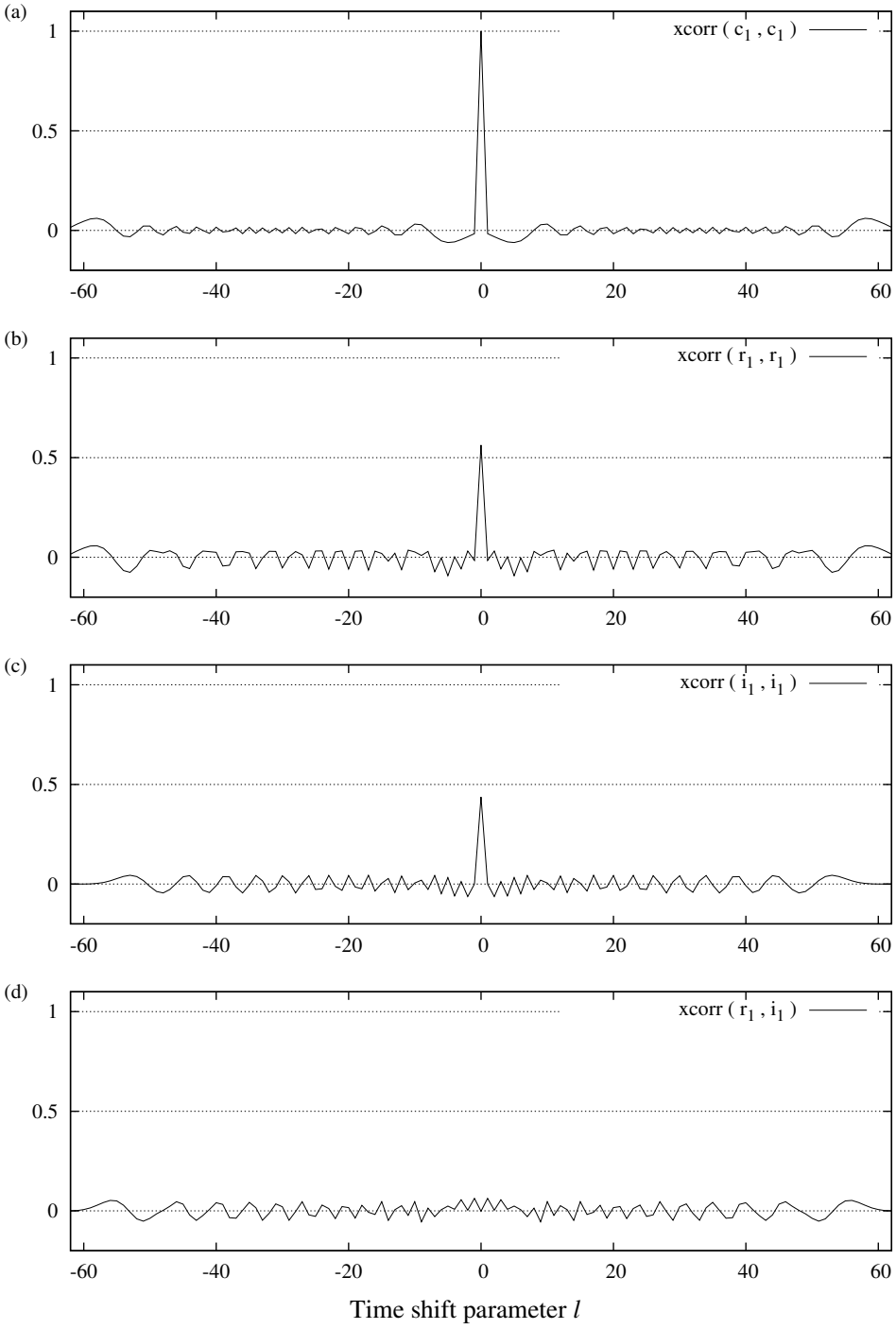

Figure 2.13: The figure shows the AAC of a length $L = 63$ Chu (a) complex sequence (b) real sequence only (c) imaginary sequence only and (d) ACC between real and imaginary sequences using only the primary sequence values (1 sample/chip).

2.4.1.2 PERIODIC CORRELATION CHARACTERISTICS

The aperiodic correlation function only considers the correlation given over one period of a sequence. The periodic correlation function considers correlation where the sequence repeats in time and is defined as [50]:

$$\varphi_{corr,ab}[\tau] = \sum_{k=0}^{L-1} s_a[kT_s]s_b^*[(k+l)T_s \ mod \ LT_s], \tag{2.44}$$

where the index $[k+l]$ is computed modulo L. The PAC and PCC (periodic cross-correlation) functions of a Chu sequence (and the complex components) with $L = 63$ and $r = 1$ is shown in figure 2.14, where figure 2.14(a) exhibits a periodic auto-correlation function equal to zero except for a single maximum value at $l = 0$.

Both the aperiodic and periodic correlation functions are important in the design of a CDMA communication system. The aperiodic correlation characteristics predict the multi-user performance and the amount of degradation that will be experienced by adding more users to the system assuming that all sequences are perfectly aligned in time. The periodic correlation characteristics determine how well sequence synchronisation can be implemented in the system.

Assigning a complete complex sequence to a single data stream (or user) in a CDMA environment will therefore result in better performance than by using separate components of a complex sequence per data stream, though transmitting a complex sequence is equivalent to transmitting two component sequences.

2.4.1.3 SPECTRAL CHARACTERISTICS

The frequency spectrum of a length-63 Chu CSS is shown in figure 2.15. The unspreaded input data speed is assumed to be 1000 symbols per second and each chip is represented by 8 equal-amplitude samples. The unfiltered sequence displays a sinc-like (resulting from the sample-and-hold sequence) DSB (double sideband) spectrum with zero-to-zero bandwidth of 126 kHz.

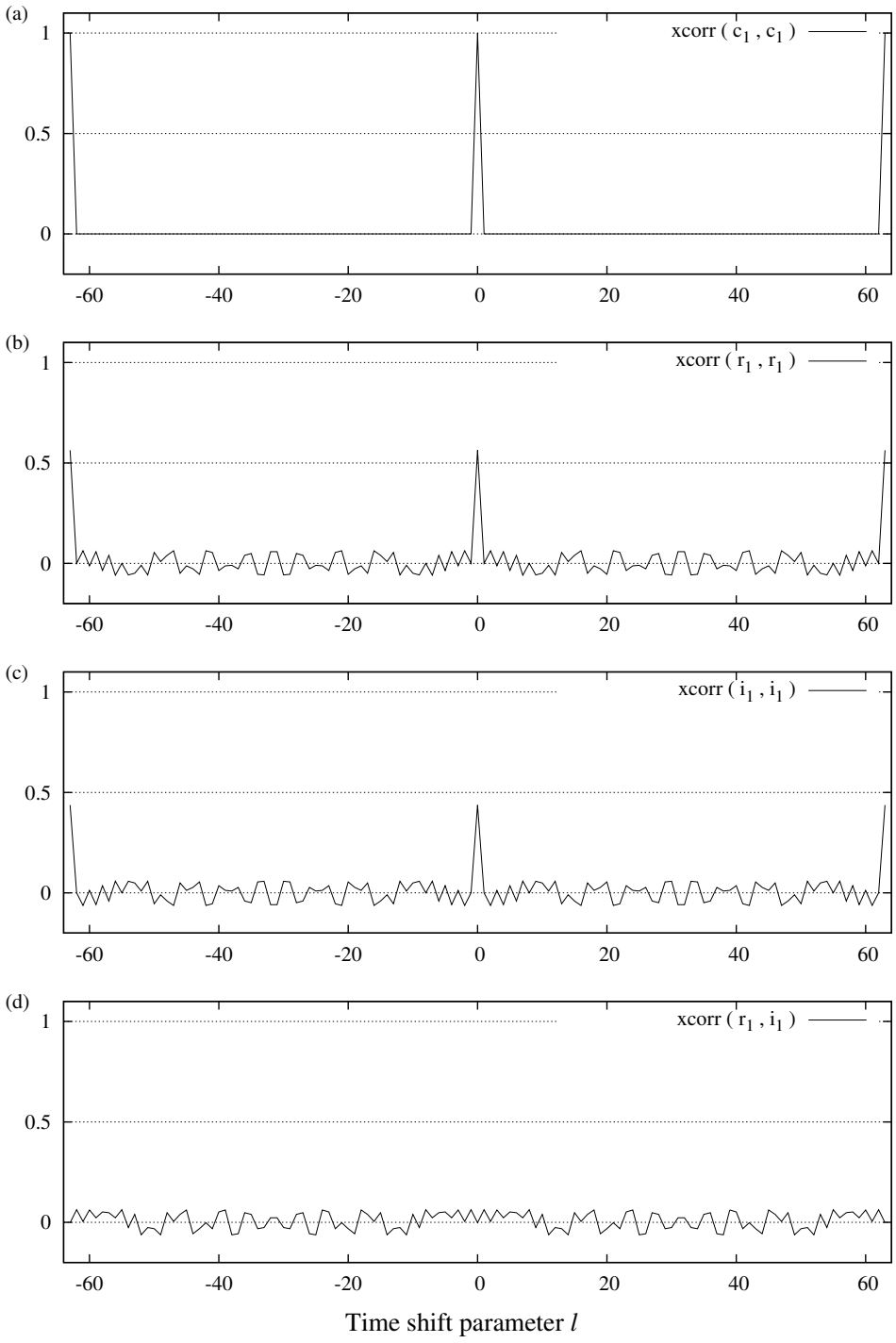

Figure 2.14: The figure shows the PAC of a length $L = 63$ Chu (a) complex sequence (b) real sequence only (c) imaginary sequence only and (d) the PCC between real and imaginary sequences using only the primary sequence values (1 sample/chip).

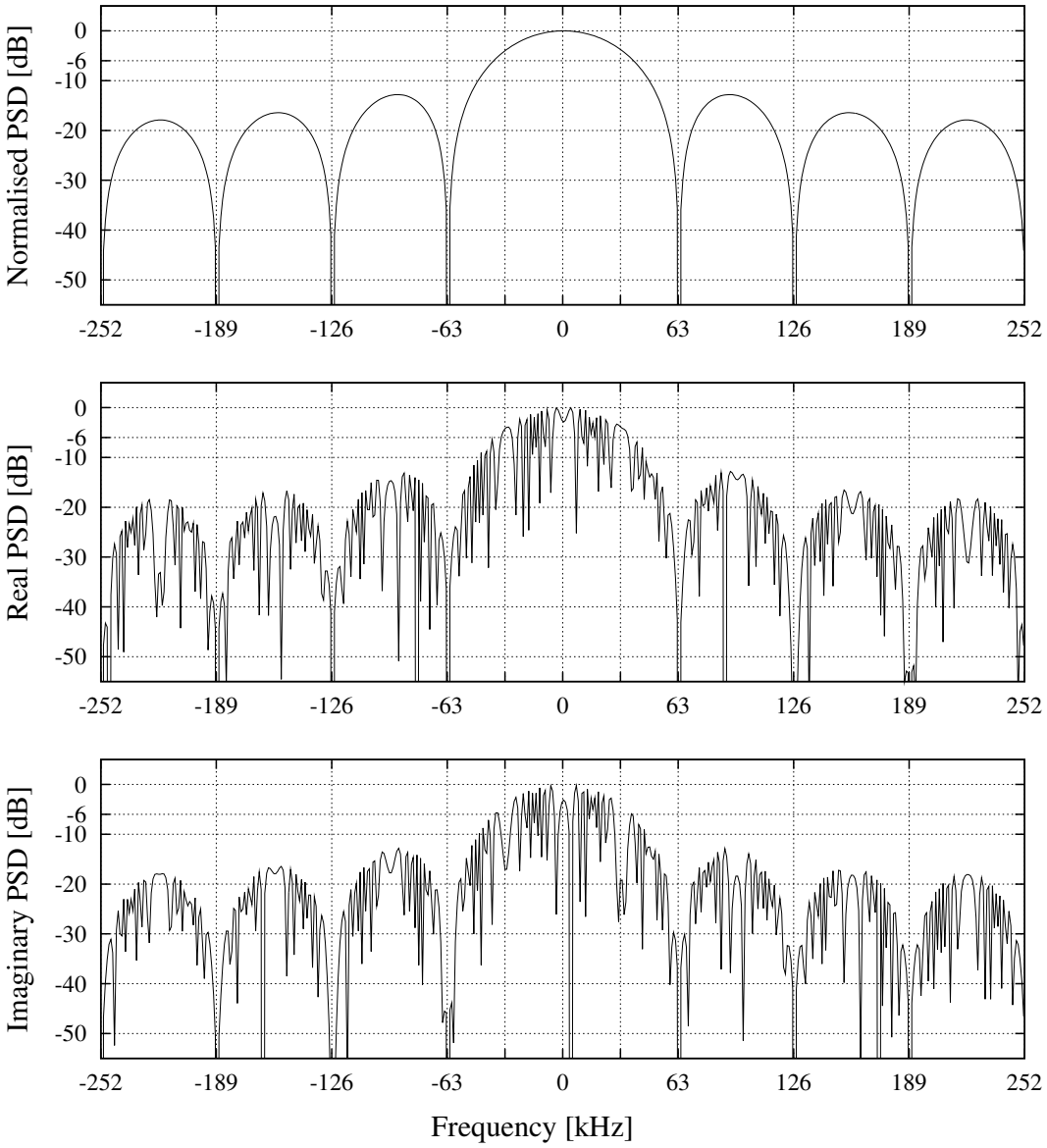

Figure 2.15: The figure shows the absolute value of the unified spectrum of a length-63 Chu CSS with 8 equal-amplitude samples/chip with the real and imaginary frequency components for a data transmission speed of 1000 symbols/s.

2.4.2 GENERALISED CHIRP-LIKE SEQUENCES

Constant envelope (CE) generalised chirp-like (GCL) complex spreading sequences (CSSs) are obtained by filtering Chu sequences with a root-of-unity (RU) filter in order to achieve band-limited sequences satisfying the minimum Nyquist bandwidth [8, 47]. The filtering involves phase interpolation between the main Chu CSS elements $a_k = e^{j\phi_k}$ (with ϕ_k the phase angle) defined in equation 2.42 and shown in figure 2.12 on the complex unit circle. The filtered sequence is defined as

$$b_m = exp\left\{j\left(\phi_k + i\left[\frac{\phi_{k+1} - \phi_k}{n_s}\right]\right)\right\}, \tag{2.45}$$

with the index $m = kn_s + i$, the number of samples per chip n_s, indexing parameters $i = 0, \ldots, n_s - 1$ and $k = 0, \ldots, L - 1$, with L the unfiltered CSS length.

The shortest path between consecutive Chu elements a_k and a_{k+1} on the complex unit circle is always chosen such that the maximum phase jump $(\phi_{k+1} - \phi_k)$ is limited to π. The GCL sequence ($L = 63$, $r = 1$) is shown in figure 2.16 and the corresponding constant complex envelope is shown in figure 2.17.

2.4.2.1 CORRELATION CHARACTERISTICS

The AAC of a GCL CSS (and the complex components) with $L = 63$ and $r = 1$ is shown in figure 2.18. As in section 2.4.1 (figure 2.13) the AAC shows a maximum value at $\tau = 0$ with small side lobes. Other correlation figures (including the PAC, ACC and PCC between different GCL sequences) are shown in Addendum A.

2.4.2.2 SPECTRAL CHARACTERISTICS

The frequency spectrum of a length-63 GCL CSS is shown in figure 2.19. The unspreaded input data speed is assumed to be 1000 symbols per second and each chip is represented by 8 samples. The filtered sequence displays a DSB spectrum with a -6 dB bandwidth of approximately 63 kHz. The Nyquist frequency at 31.5 kHz is also shown, which is approximately the -6 dB cutoff frequency for the sequence shown.

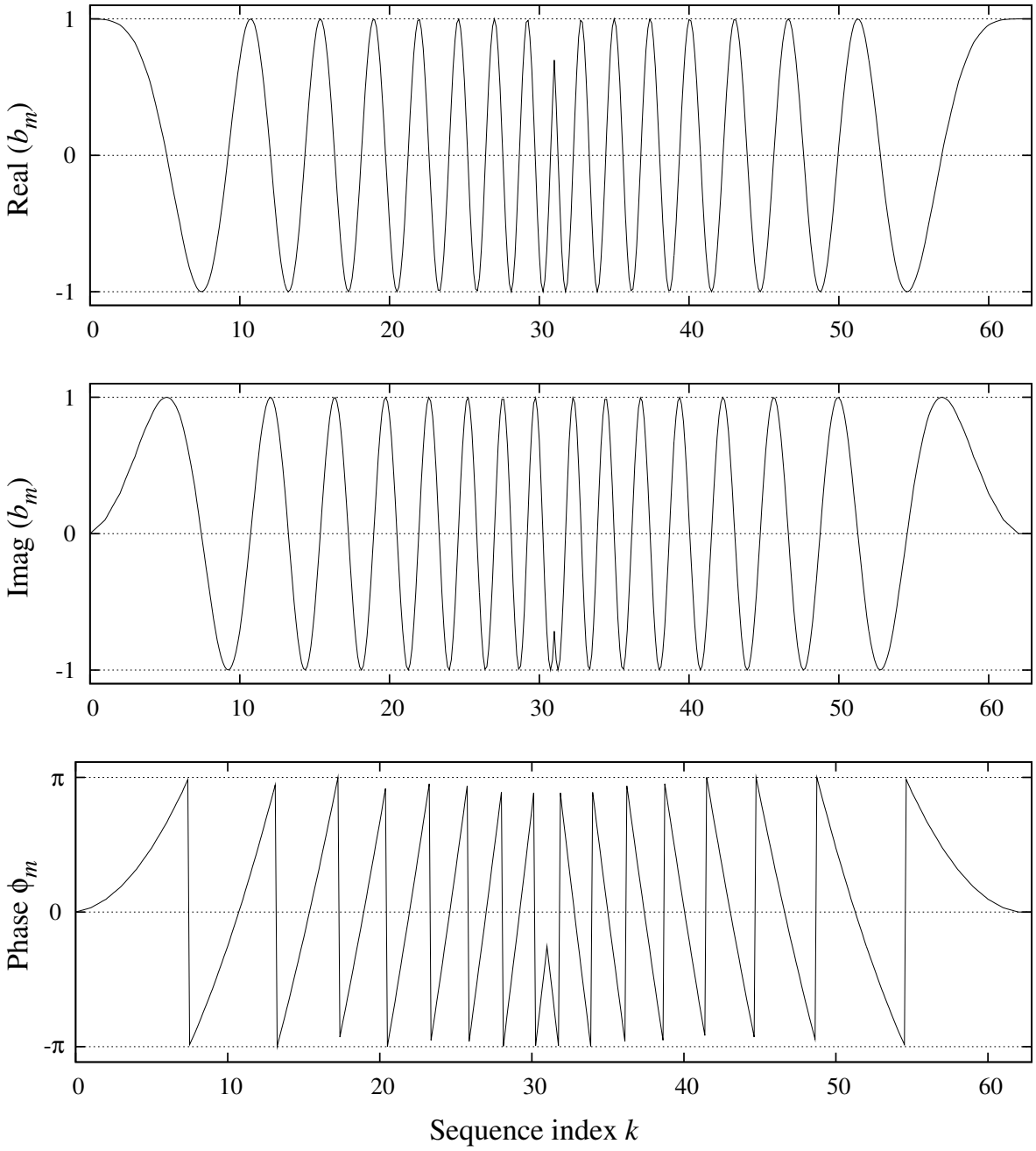

Figure 2.16: The figure shows the root-of-unity filtered (RUF) GCL CSS for $L = 63$ and $r = 1$ projected on the real and imaginary axes and the corresponding phase angles

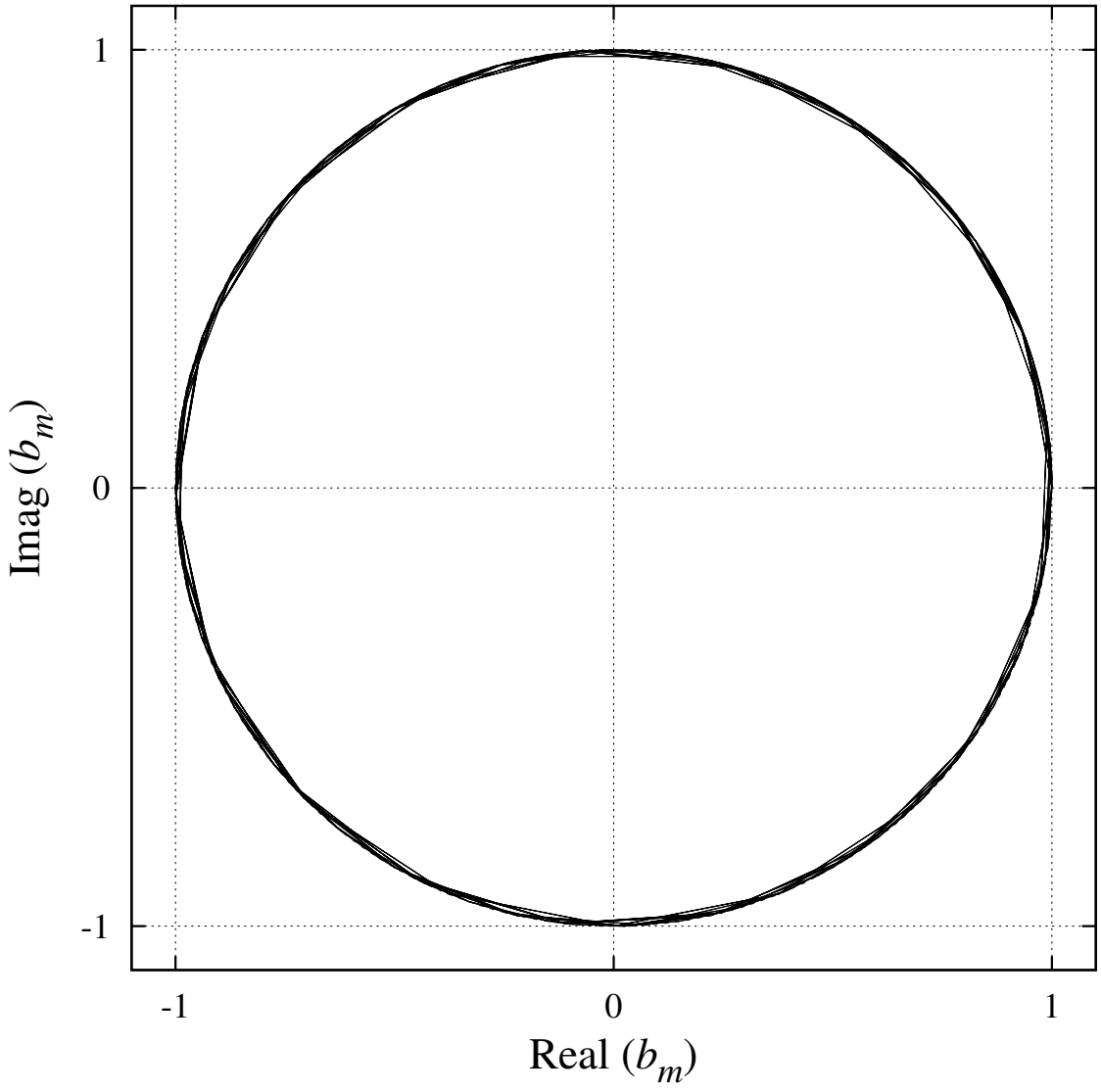

Figure 2.17: Complex envelope of the root-of-unity filtered GCL CSS for $L = 63$, $r = 1$ and 8 interpolated samples per chip

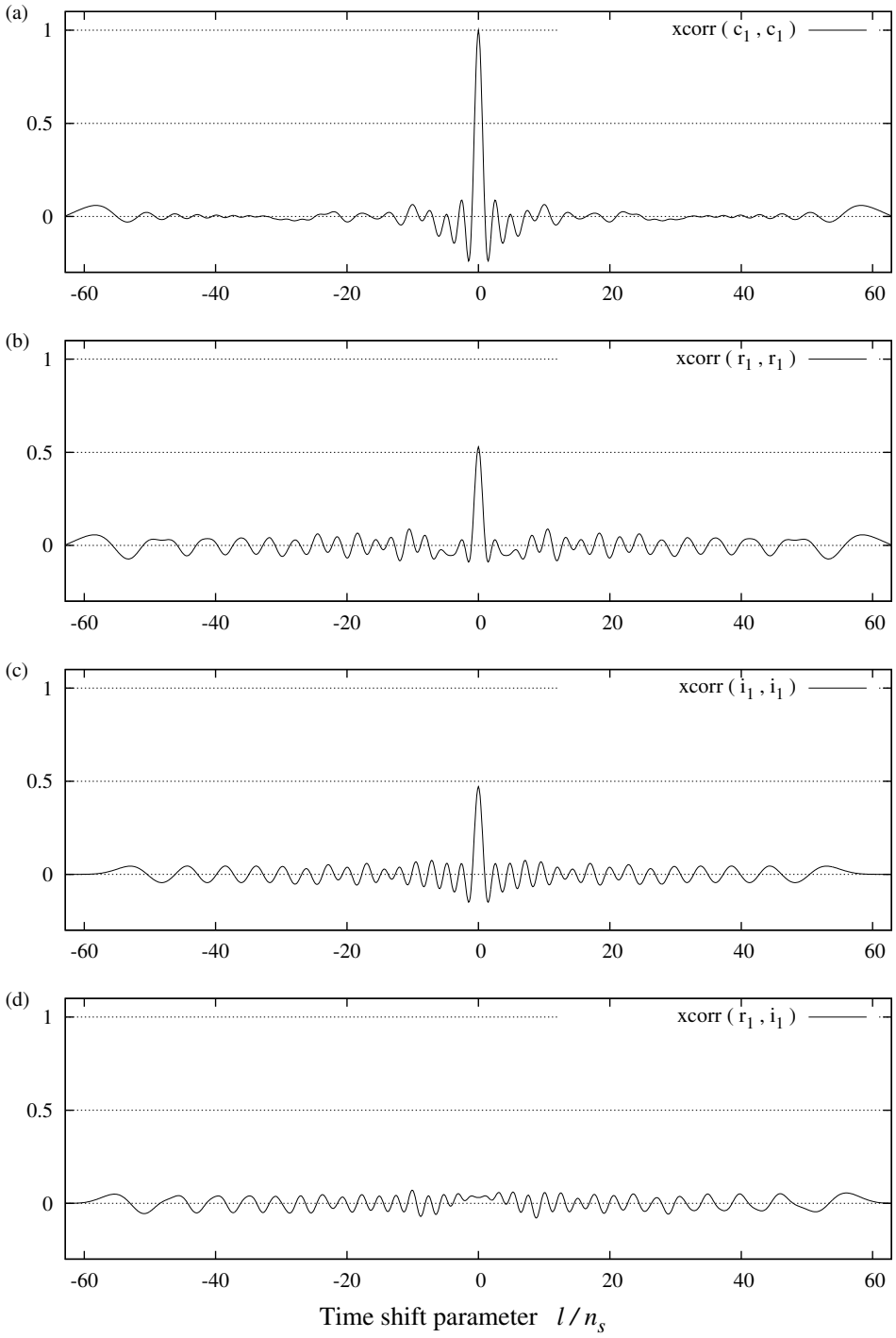

Figure 2.18: The figure shows the AAC of a length $L = 63$ GCL (a) complex sequence (b) real sequence only (c) imaginary sequence only and (d) ACC between real and imaginary sequences.

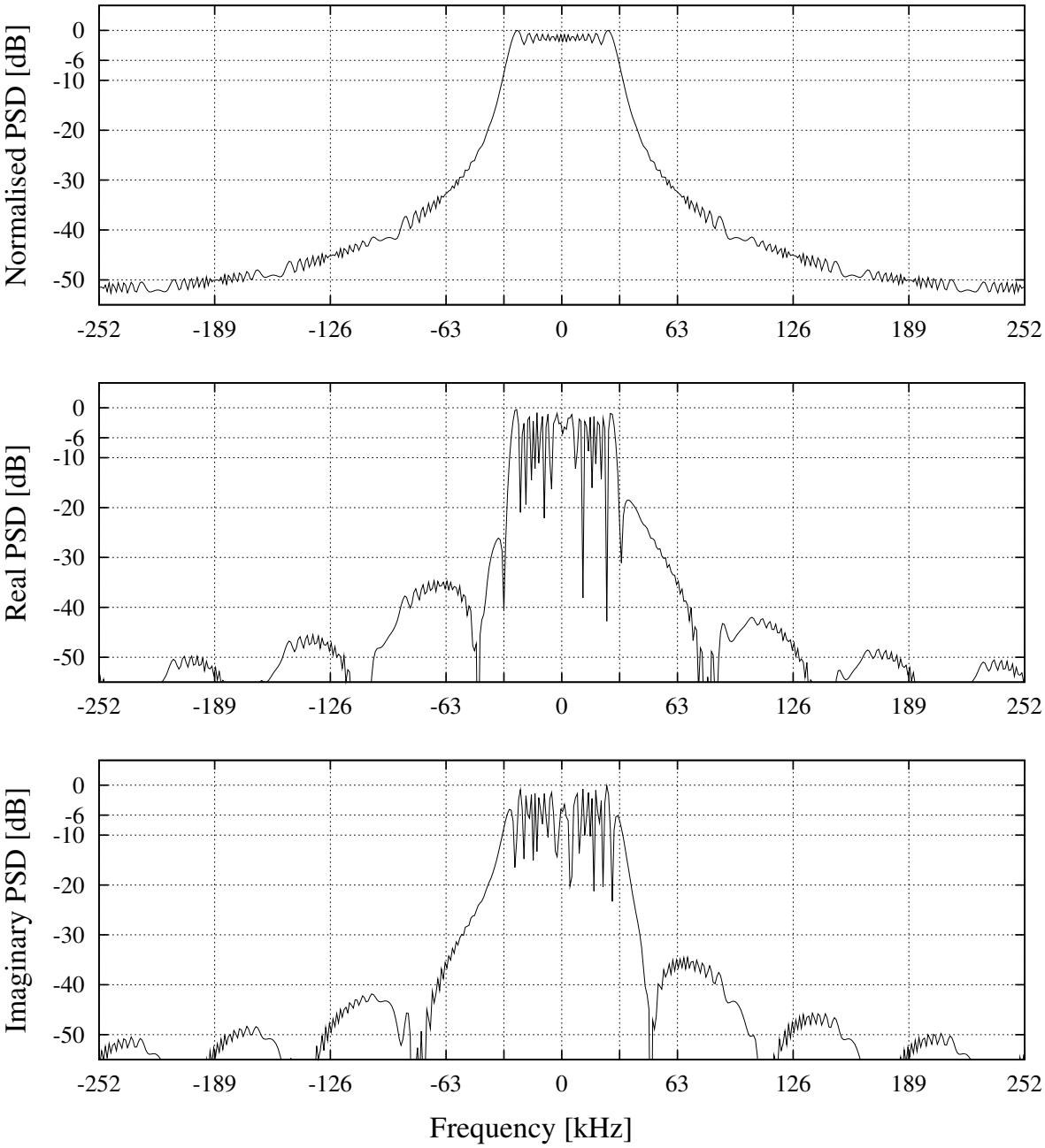

Figure 2.19: The figure shows the absolute value of the unified spectrum of a length-63 GCL CSS with 8 samples/chip with the real and imaginary frequency components for a data transmission speed of 1000 symbols/s. The Nyquist frequency $f_{Nyq} = 31.5$ kHz is also indicated.

2.4.3 ZERO CROSS CORRELATION SEQUENCES

As CE-GCL-CSSs are derived from ZC sequences using a linear CE-RU filtering technique, ZCC CSSs are derived from Park-Park-Song-Suehiro (PS) CSSs using a non-linear CE-RU filtering technique described in patent [34]. PS sequences are presented in [53] and details concerning generation of PS sequences including time and correlation graphs can be found in [9]. ZCC sequences are described similar to GCL sequences defined in equation 2.45 in section 2.4.2 and the same symbols will be used here to signify ZCC CSSs.

Length-63 sequences will again be considered with only 7 CSSs in the family. Shown in figure 2.20 is the real and imaginary components and the phase diagram of the first code ($r = 1$) in the family. These sequences also exhibit a constant envelope as shown in figure 2.17.

2.4.3.1 CORRELATION CHARACTERISTICS

The AAC functions of ZCC sequence $r = 1$ (and the complex components) with $L = 63$ is shown in figure 2.21. Figure 2.21(a) shows a maximum auto-correlation value at $\tau = 0$ and 6 large correlation spikes on either side of $\tau = 0$, which can be ascribed to the periodic time characteristic of the sequence (see figure 2.20). Multiple auto correlation peaks are a trade-off for zero cross correlation between different CSSs (see figures 2.18, 2.21 and the PAC, ACC and PCC figures given in Addendum A).

2.4.3.2 SPECTRAL CHARACTERISTICS

The frequency spectrum of all length-63 ZCC CSSs are shown in figure 2.22. The unspreaded input data speed is assumed to be 1000 symbols per second and each chip is represented by 8 samples. The filtered sequence displays a DSB spectrum of approximately 63 kHz. The Nyquist frequency at 31.5 kHz is also shown and it is clear that ZCC sequences do not fit the Nyquist bandwidth as close as GCL sequences (see figure 2.19). CE-ZCC-CSSs are less chirp-like (ZCC CSSs are more random as can be seen by comparing figures 2.16 and 2.20) and hence exhibit a less smooth spectrum than GCL CSSs.

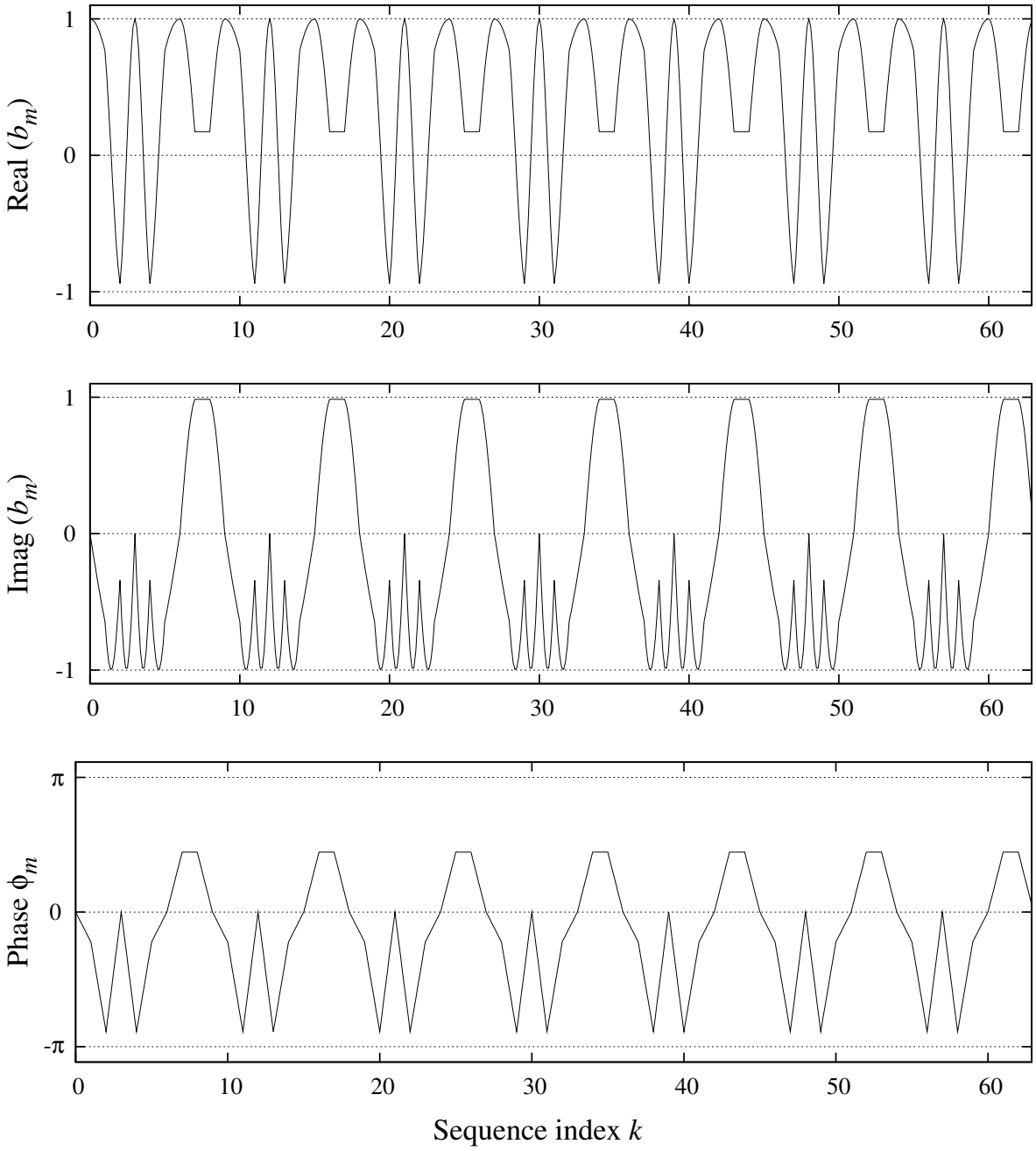

Figure 2.20: The figure shows the first code ($r = 1$) of the ZCC CSS family for $L = 63$ projected on the real and imaginary axes and the corresponding phase angles

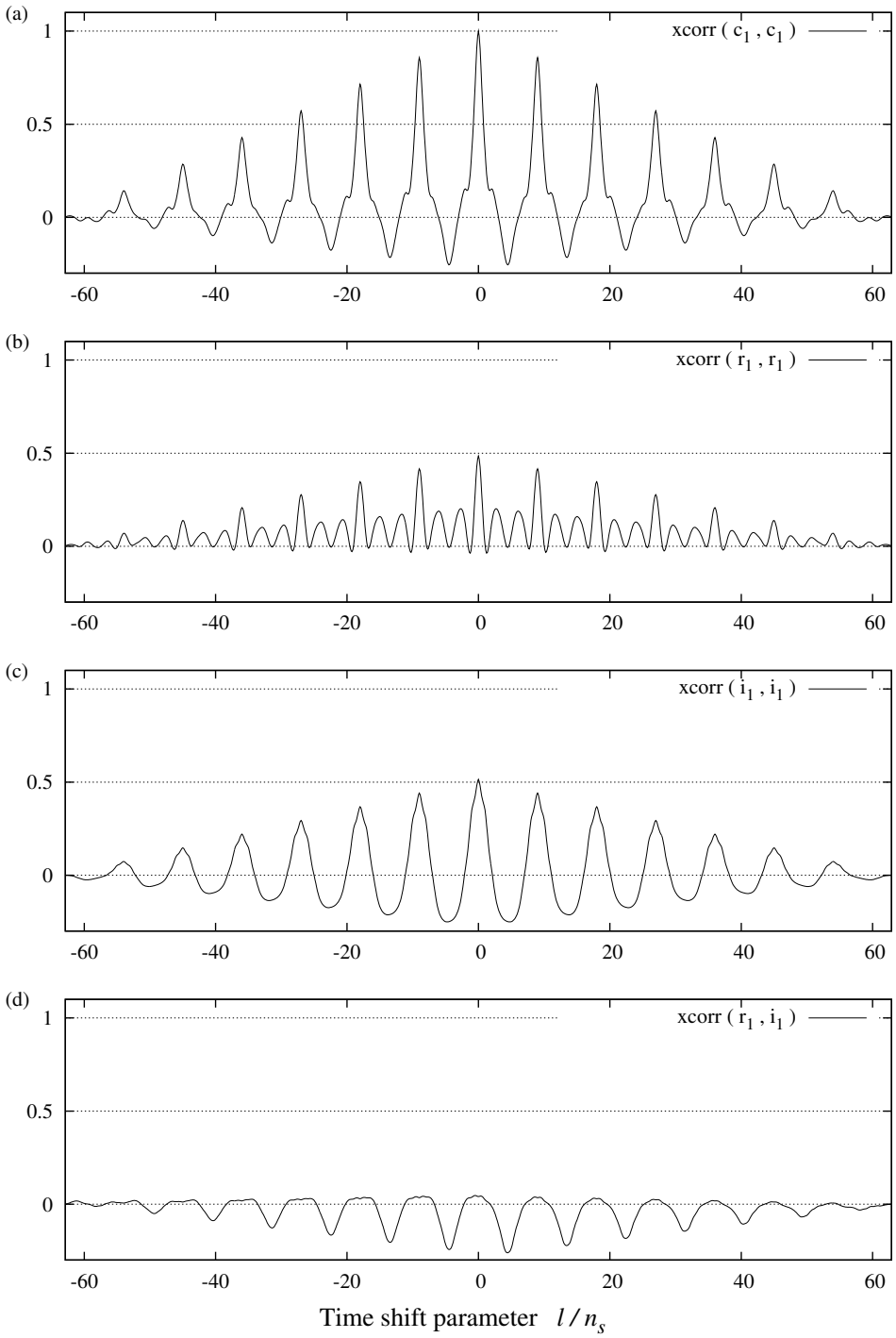

Figure 2.21: The figure shows the AAC of code $r = 1$, length-63 ZCC (a) complex sequence (b) real sequence only (c) imaginary sequence only and (d) ACC between real and imaginary sequences.

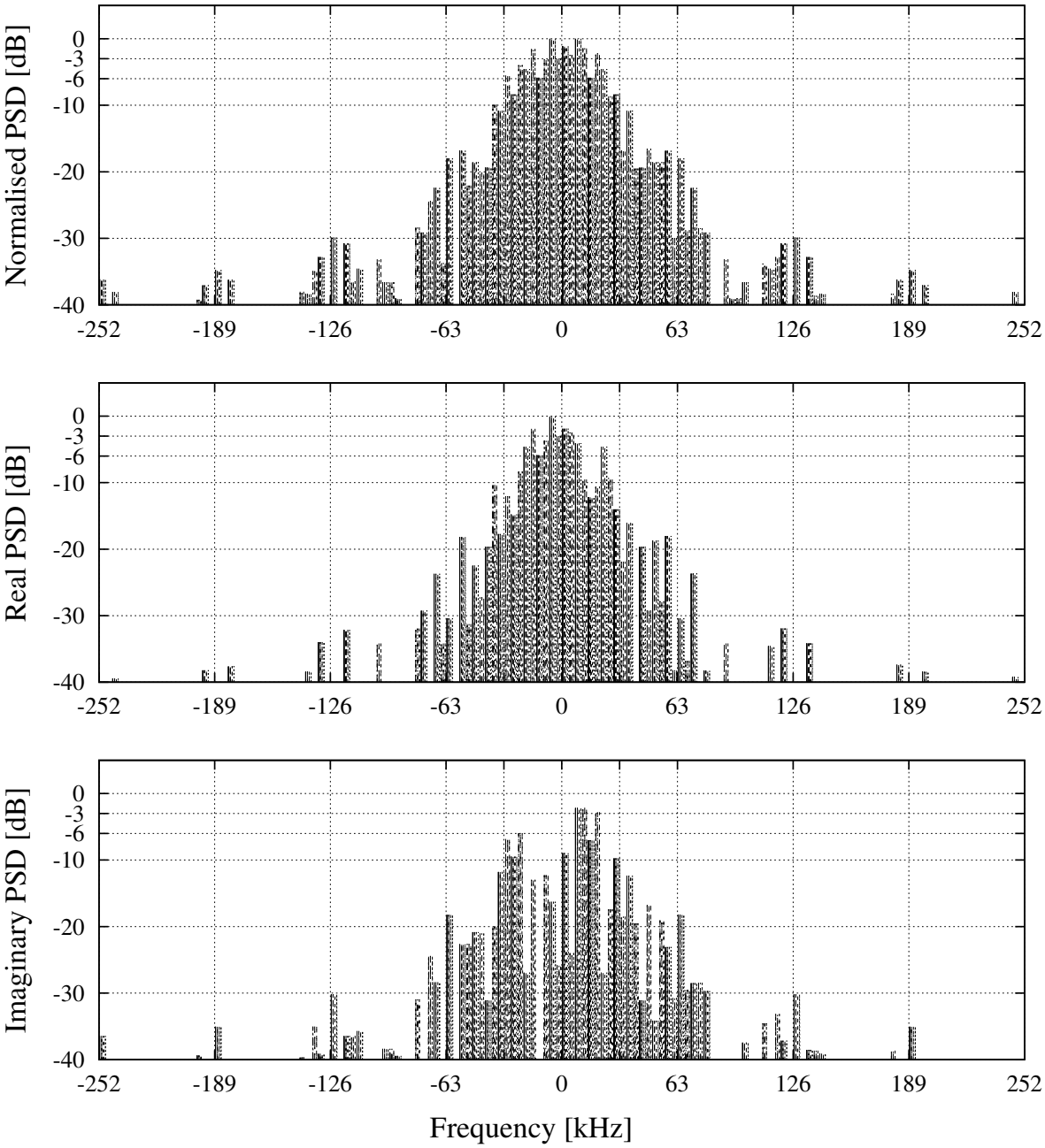

Figure 2.22: The figure shows the absolute value of the unified spectrum of all 7 length-63 ZCC CSS with 8 samples/chip with the real and imaginary frequency components for a data transmission speed of 1000 symbols/s. The Nyquist frequency $f_{Nyq} = 31.5$ kHz is also indicated.

2.5 EVALUATION OF CSSs IN A SYNCHRONOUS CDMA ENVIRONMENT

Due to the use of CSSs with non-zero periodic correlation characteristics (e.g. the PAC of ZCC CSSs as shown in figure A.7(a) exhibits 6 large correlation spikes on either side of the $\tau = 0$ spike) synchronisation in the CDMA communication system is of utmost importance. Loss of perfect or at least near-perfect (within a few chips) synchronisation will cause ambiguities in the demodulator and separation of different users' data will become difficult or impossible. The periodic correlation characteristics are also important in the design of the code lock loop (CLL) in the receiver (CLLs fall outside the scope of this dissertation and more information can be found in [50]). The presence of a pilot channel is therefore mandatory, ensuring quasi-synchronous operation.

Perfect synchronisation requires signalling or pilot channels to align the data of all users in the CDMA cell or environment and intelligence in the transceivers to remove or reduce multipath effects (multiple and varying time delayed versions of the LOS (line-of-sight) or main received signal). Assuming perfect synchronisation (as is done in this dissertation) the most important factor determining performance in a multi-user environment (ignoring all other issues involved in synchronisation mentioned) is the a-periodic correlation values at $\tau = 0$ of the spreading sequences employed. This section focuses on correlation values obtained from perfectly synchronised CSSs, including CE-GCL and ZCC CSSs and their power efficiency.

2.5.1 BASIS VECTORS OF MD SPACE

The basis vector set used in multi-dimensional modulation (MDM) determines the error performance and power efficiency of the multi-dimensional (MD) communication system. From figure 2.9 the base vectors of the 4D system can be expressed as

$$\Psi_1(t) = c_r(t)cos(w_c t) \quad (2.46)$$

$$\Psi_2(t) = c_i(t)cos(w_c t) \quad (2.47)$$

$$\Psi_3(t) = c_r(t)sin(w_c t) \quad (2.48)$$

$$\Psi_4(t) = c_i(t)sin(w_c t), \quad (2.49)$$

with $c_r(t)$ and $c_i(t)$ the real and imaginary components of the CSS respectively and w_c the carrier frequency in rad/s. In this section the AAC and ACC values of the

4D base vector set at time shift $\tau = 0$ are evaluated and compared to illustrate or verify orthogonality. The parameter τ indicates the relative time shift between two sequences including positive and negative shifts, though in this section only positive shifts are considered and is indicated by τ_c (to indicate the progress of the discrete time convolution or dot product process). The complex envelope and power characteristics of the base vector set are also investigated.

2.5.2 GENERALISED CHIRP-LIKE SEQUENCES

2.5.2.1 BASIS VECTOR FUNCTIONS

The basis functions $\Psi_i; i = 1, 2, 3, 4$, of a length $L = 63$ GCL CSS with $r = 1$ are shown in figure 2.23 (using 32 samples per chip and a carrier frequency 8 times higher than the base band data speed; $f_c = 8f_{bit}$).

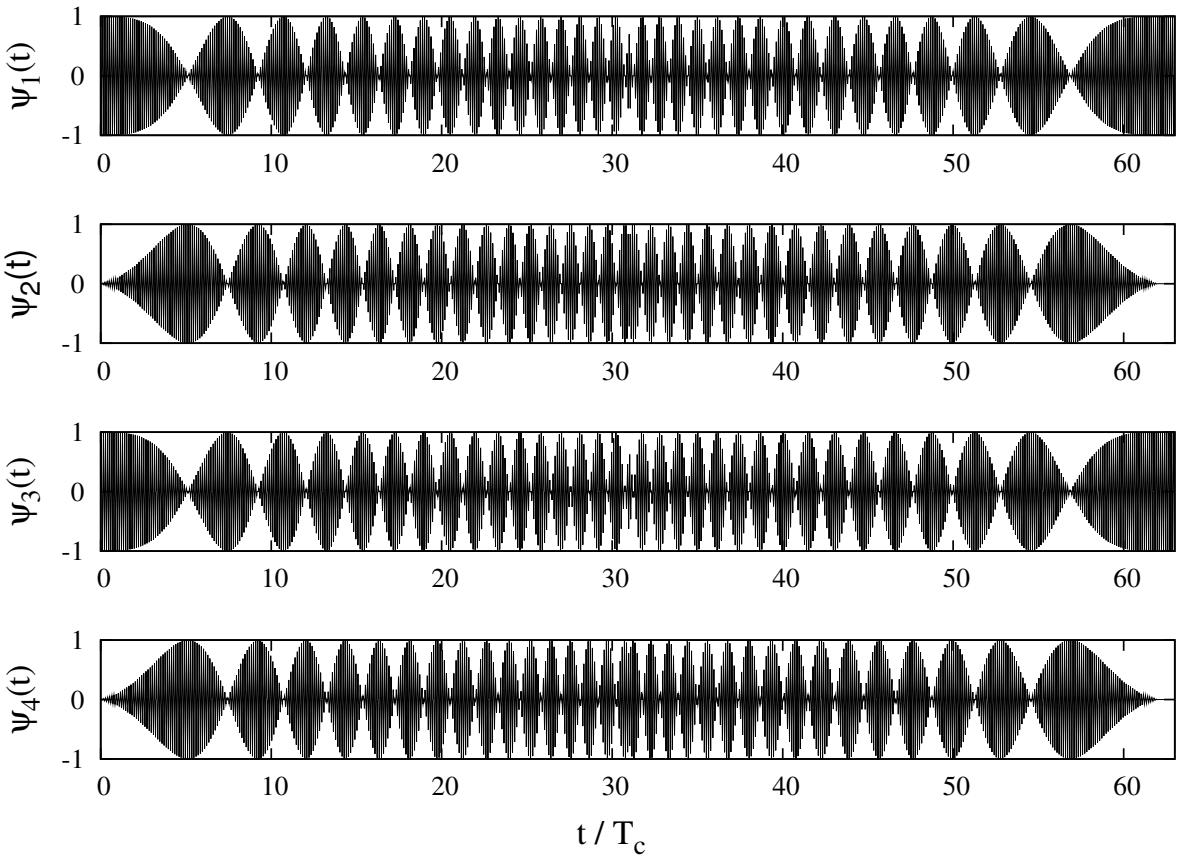

Figure 2.23: Length-63 4D-WCDMA GCL ($r = 1$) basis functions with 32 samples per chip

2.5.2.2 POWER EFFICIENCY AND COMPLEX MD ENVELOPE

The time characteristics and orthogonality between the base vectors used, determine the power efficiency of the MD system. The output signal $s(t)$ of the 4D modulator in figure 2.9 can be written (by combining equation 2.31 with equations 2.46 to 2.49) as:

$$s(t) = [d_1(t)\Psi_1(t)] + [d_2(t)\Psi_2(t)] + [d_3(t)\Psi_3(t)] + [d_4(t)\Psi_4(t)]\,, \tag{2.50}$$

with $d_i(t) = \pm 1V; i = 1, 2, 3, 4$. The signal $s(t)$ therefore has average power:

$$\begin{aligned} P_{s,ave} &= \frac{1}{T_s}\int_{t_0}^{t_0+T_s} |s(t)|^2\, dt \\ &= \frac{1}{T_s}\int_{t_0}^{t_0+T_s} \left\{\Psi_1^2(t) + \Psi_2^2(t) + \Psi_3^2(t) + \Psi_4^2(t)\right\} dt + \zeta, \end{aligned} \tag{2.51}$$

with the cross term

$$\begin{aligned} \zeta &= \frac{2}{T_s}\int_{t_0}^{t_0+T_s} \{\Psi_1(t)\Psi_2(t) + \Psi_1(t)\Psi_3(t) + \Psi_1(t)\Psi_4(t) \\ &\quad + \Psi_2(t)\Psi_3(t) + \Psi_2(t)\Psi_4(t) + \Psi_3(t)\Psi_4(t)\}\, dt \\ &= \frac{2}{T_s}\int_{t_0}^{t_0+T_s} \{\Psi_1(t)\Psi_2(t) + \Psi_3(t)\Psi_4(t)\}\, dt, \end{aligned} \tag{2.52}$$

where only the two cross dot products $[\Psi_1 \cdot \Psi_2]$ and $[\Psi_3 \cdot \Psi_4]$ feature in the power equation, due to quadrature modulation. The two aforementioned cross dot products are negligible or approaching zero, assuming perfectly synchronous conditions and perfect orthogonality between all base vectors over every symbol period T_s.

It is also constructive to consider the instantaneous output power of the 4D modulator. The short-time averaged power at time instance $t = t_0 + t_s$ is given by:

$$P_{s,short}(t) = \frac{1}{t_s}\int_{t_0}^{t_0+t_s} |s(t)|^2\, dt, \tag{2.53}$$

where t_s is the time interval over which the power is calculated. The instantaneous output power is then obtained in the limit:

$$P_{s,ins}(t) = \lim_{t_s \to 0} \frac{1}{t_s}\int_{t_0}^{t_0+t_s} |s(t)|^2\, dt. \tag{2.54}$$

If the base vector set has the special characteristic that orthogonality between different base vectors is achieved not only over a complete symbol period T_s, but also over a time fraction less than the symbol period ($t_s < T_s$), the instantaneous output (or short-time average) power can be approximated as:

$$P_{s,ins}(t) \approx \Psi_1^2(t) + \Psi_2^2(t) + \Psi_3^2(t) + \Psi_4^2(t). \tag{2.55}$$

Figure 2.24 shows power related graphs for GCL CSSs including the graph of $P_{s,short}(t)$ (given in equation 2.53) with t_s the time interval of 10 CSS chips (at any time instance on the graph, the short-time averaged power output over the 10 most recent CSS chips are given) and the graph of the approximation of $P_{s,ins}(t)$ (given in equation 2.55) showing a constant value ($P_{s,ins}(t) = 1 \; \forall \; t$), illustrating the point-by-point orthogonality of the basis functions.

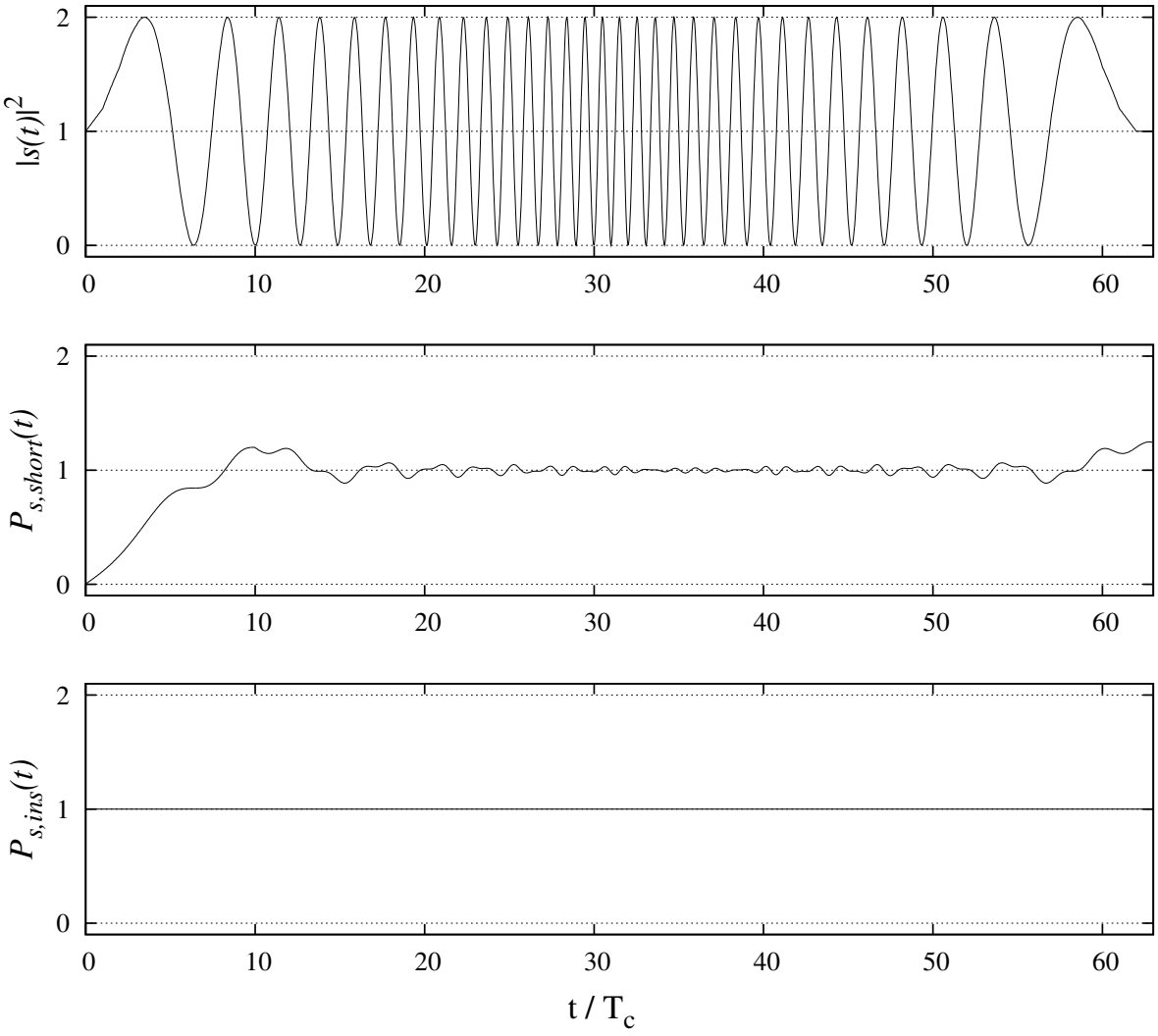

Figure 2.24: The figure shows graphs related to the output power of the 4D modulator employing a length-63 GCL ($r = 1$) CSS.

By plotting the quadrature vectors ($\Psi_1(t)$ versus $\Psi_2(t)$ on the cosine carrier and $\Psi_3(t)$ versus $\Psi_4(t)$ on the sine carrier) the complex envelope (or manifold) of the transmitted 4D signal is obtained in two dimensions over time, displaying the power characteristics of the signal. The complex manifold of 6 chips in a length 63 GCL CSS is shown in figure 2.25 illustrating a constant complex envelope and near constant instantaneous power. The orthogonal quadrature carriers can also be seen in figure 2.25.

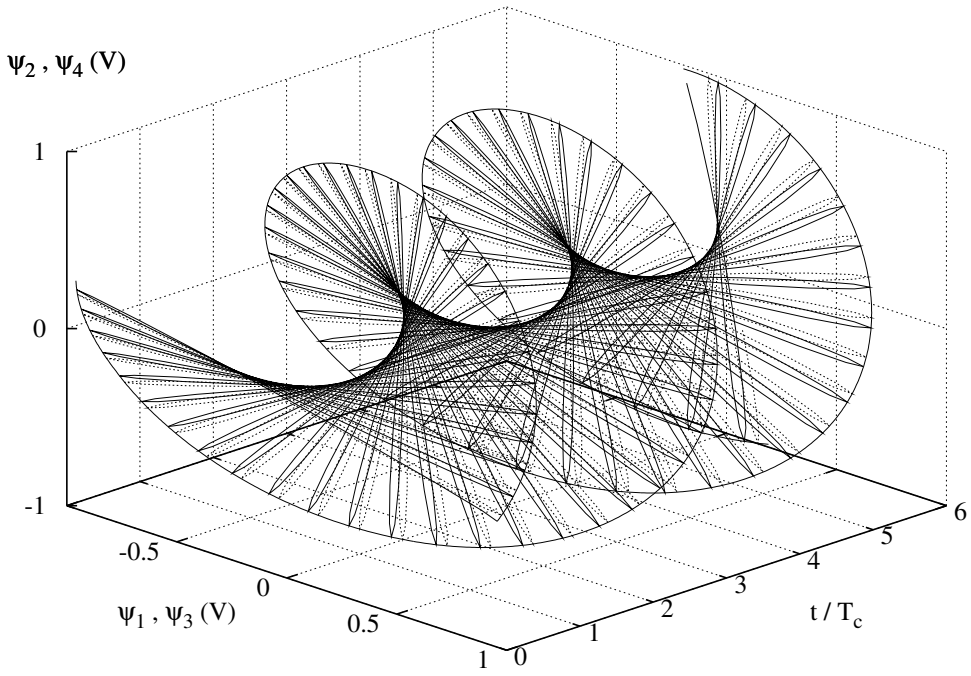

Figure 2.25: Complex manifold of a length-63 GCL CSS ($r = 1$) showing 6 chips in the sequence and 128 samples per chip

2.5.2.3 CROSS DOT PRODUCTS

The normalised accumulating dot product functions between basis vector $\Psi_1(t)$ and all other basis vectors ($\Psi_2(t)$,$\Psi_3(t)$ and $\Psi_4(t)$) of a length $L = 63$ DSB CSS ($r = 1$) are shown in figure 2.26. The dot products correspond to the output signals $g_i(t); i = 1, 2, 3, 4$ of the integrate module in the 4D correlation-type receiver shown in figure 2.10, where the correlation $(\Psi_1(t); \Psi_1(t))$ matches $g_1(t)$ (the other correlation functions contribute to interference signals in $g_1(t)$). The cross correlation between quadrature carriers is understandably zero as shown in figure 2.26, while a non-zero disturbance signal $(\Psi_1(t); \Psi_2(t))$ is also present. The value at $\tau_c/T_c = 63$ CSS chips in figure 2.26 (equivalent to the value at $\tau = 0$ in the correlation graphs in section 2.4.2) is the decisive factor (assuming perfect synchronisation) as to whether the received bit will be demodulated as a $+1$ or -1.

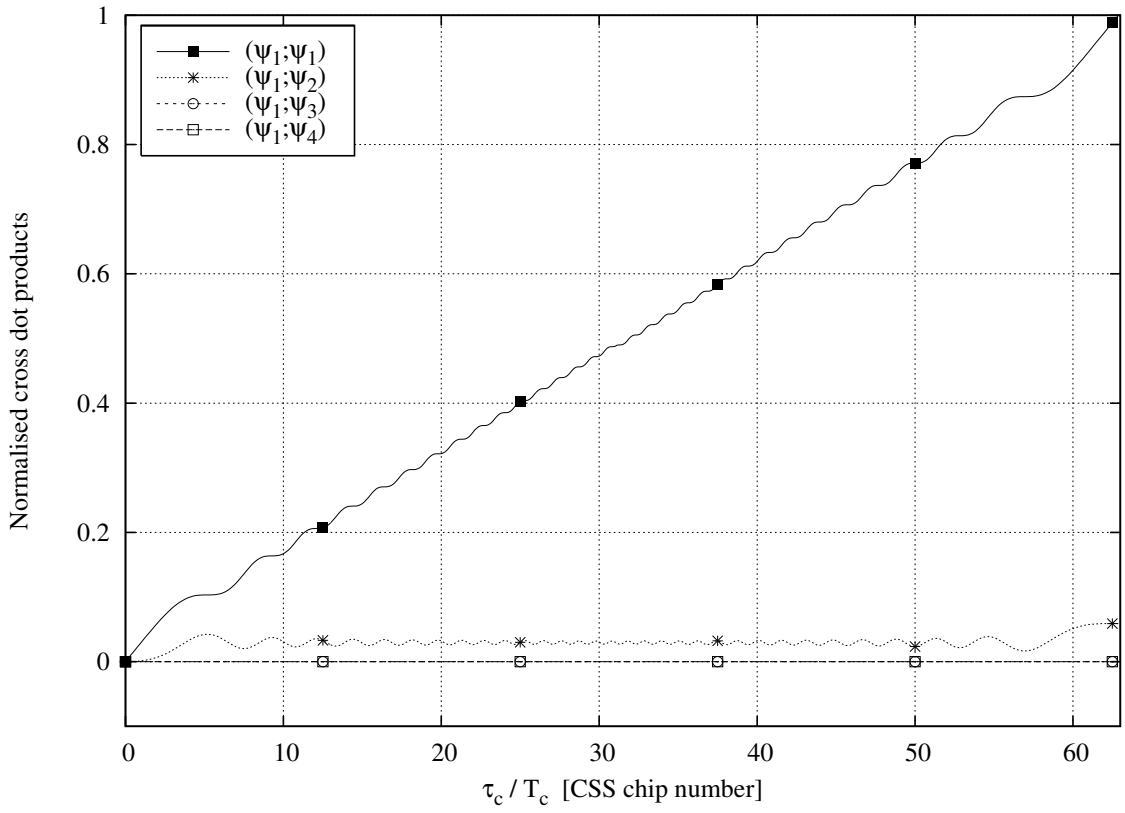

Figure 2.26: Auto and cross dot products between basis vectors of a length-63 GCL ($r = 1$) CSS

2.5.2.4 CORRELATION COMPARISON OF COMPLEX SEQUENCES

To predict the performance of the synchronous CDMA system, the correlation values at $\tau = 0$ (or $\tau_c/Tc = 63$ CSS chips) must be investigated. Figure 2.27 shows the normalised correlation values between all complex sequences (by performing a complex correlation function using both real and imaginary sequences for every complex sequence) of a length $L = 63$ GCL CSS family. Seven of the best sequences [54], numbered from 1 to 7, in the family (with prime index numbers $1, 2, 5, 11, 13, 17$ and 19) are used for comparison purposes. It is clear from the figure that all AC values are approximately equal and the CC values are relatively large which will hinder correct demodulation in the receiver (see Addendum A table A.3 for the exact values).

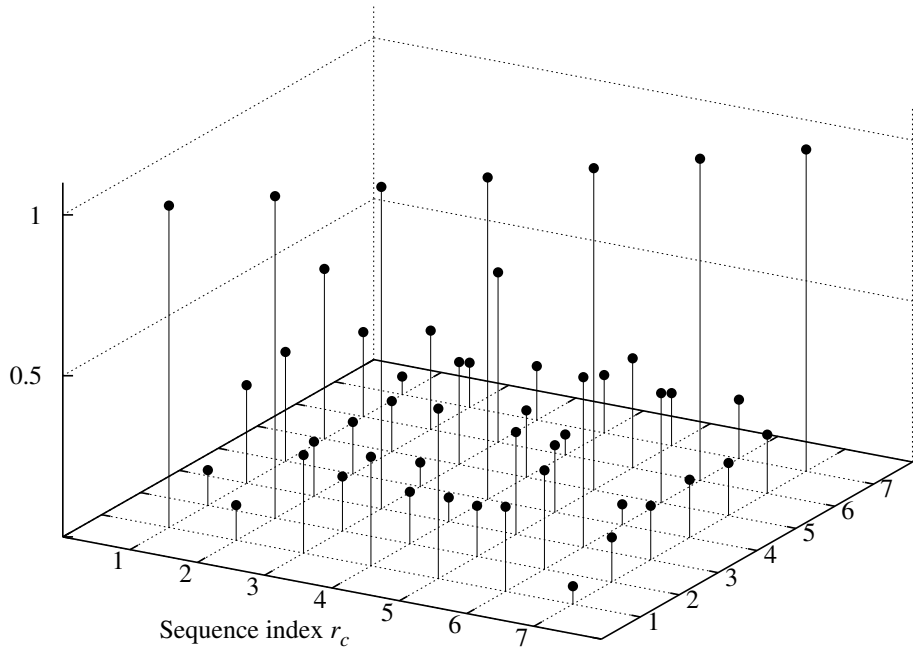

Figure 2.27: Complex aperiodic correlation comparison of the 7 best CSSs of a length $L = 63$ GCL family. The x- and y-axes show the code indexes and the z-axis the normalised correlation amplitudes at $\tau = 0$.

2.5.2.5 CORRELATION COMPARISON OF COMPONENTS OF COMPLEX SEQUENCES

Figure 2.28 shows the correlation values between all the components of the same GCL CSSs considered in figure 2.27 at $\tau = 0$. Sequence indexes 1 and 2 (figure 2.28) are the real and imaginary components of sequence 1 in figure 2.27 respectively. The indexing pattern is followed up to indexes 13 and 14. In a 4D system as presented in section 2.3 the component correlation values are more important than the complex correlation values. Figure 2.28 shows varying AC values (normalised with respect to the complex AC values in figure 2.27) larger than the disturbance CC values, though it can be expected that a 4D system employing the complex components of GCL CSSs will perform worse than a system employing the complete CSSs. Tables A.4 and A.5 in Addendum A provide the exact correlation values.

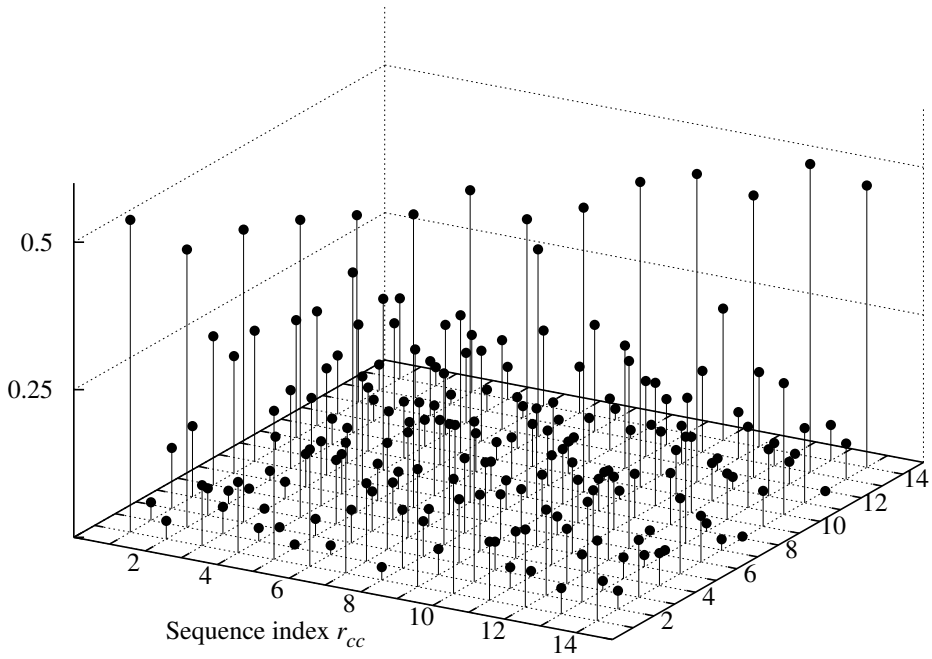

Figure 2.28: Complex aperiodic correlation comparison of the complex components of the 7 best CSSs of a length $L = 63$ GCL family (totalling 14 component codes). The x- and y-axes show the code indexes and the z-axis the normalised correlation amplitudes at $\tau = 0$.

2.5.3 ZERO CROSS CORRELATION SEQUENCES

2.5.3.1 BASIS VECTOR FUNCTIONS

The basis functions $\Psi_i; i = 1, 2, 3, 4$, of a length $L = 63$ SO (super-orthogonal) ZCC CSS are shown in figure 2.29 (using 32 samples per chip and a carrier frequency 8 times higher than the base band data speed; $f_c = 8f_{bit}$).

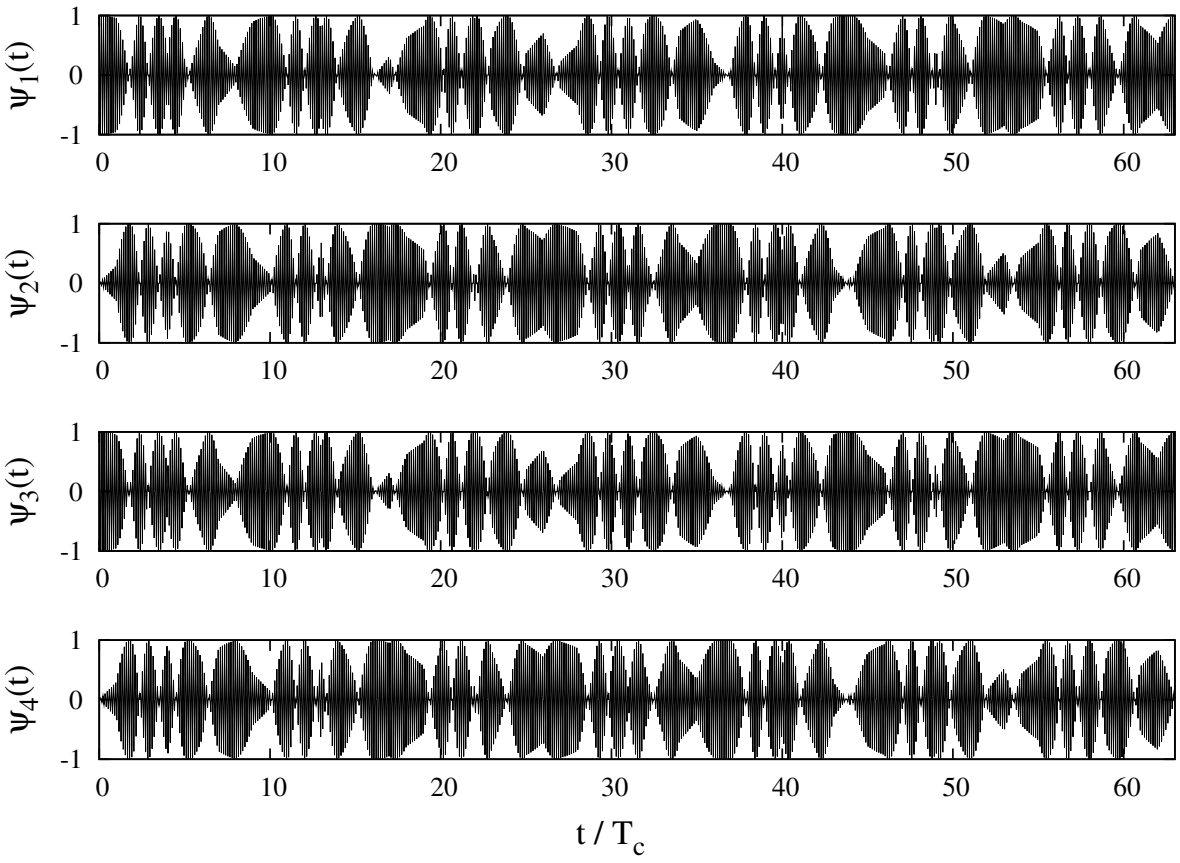

Figure 2.29: Length-63 SO 4D-WCDMA ZCC (code $r = 5$) basis functions with 32 samples per chip

2.5.3.2 POWER EFFICIENCY

Figure 2.30 shows power related graphs for the ZCC CSS (code $r = 5$) and is the ZCC equivalent of figure 2.24.

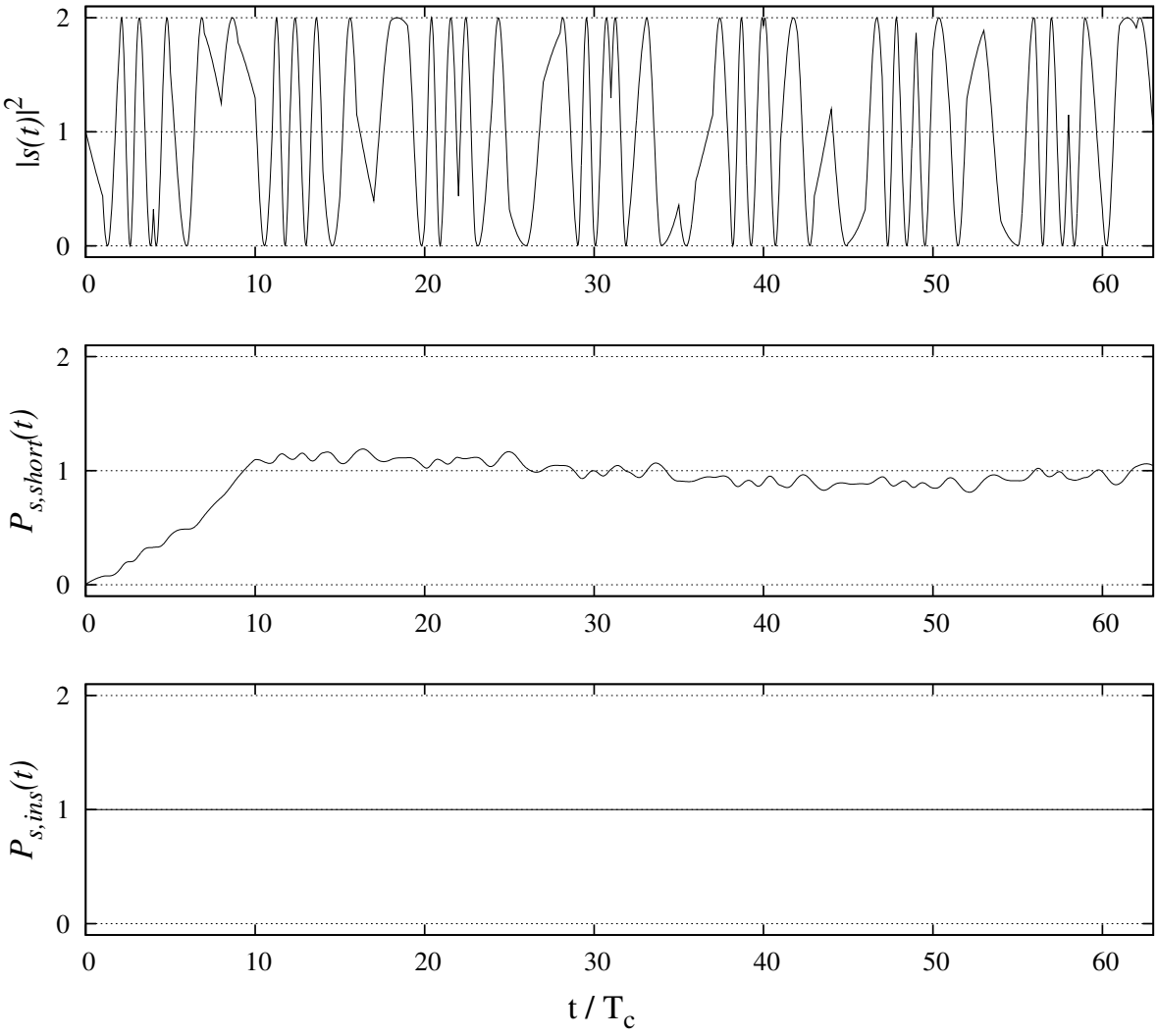

Figure 2.30: The figure shows graphs related to the output power of the 4D modulator employing a length-63 ZCC (code $r = 5$) CSS.

2.5.3.3 COMPLEX MD ENVELOPE

The complex envelope of an $L = 63$ ZCC CSS is shown in figure 2.31. Although the envelope is still constant, the complex rotation direction is not necessarily constant and signs of frequency modulation is also evident (also see figure 2.29).

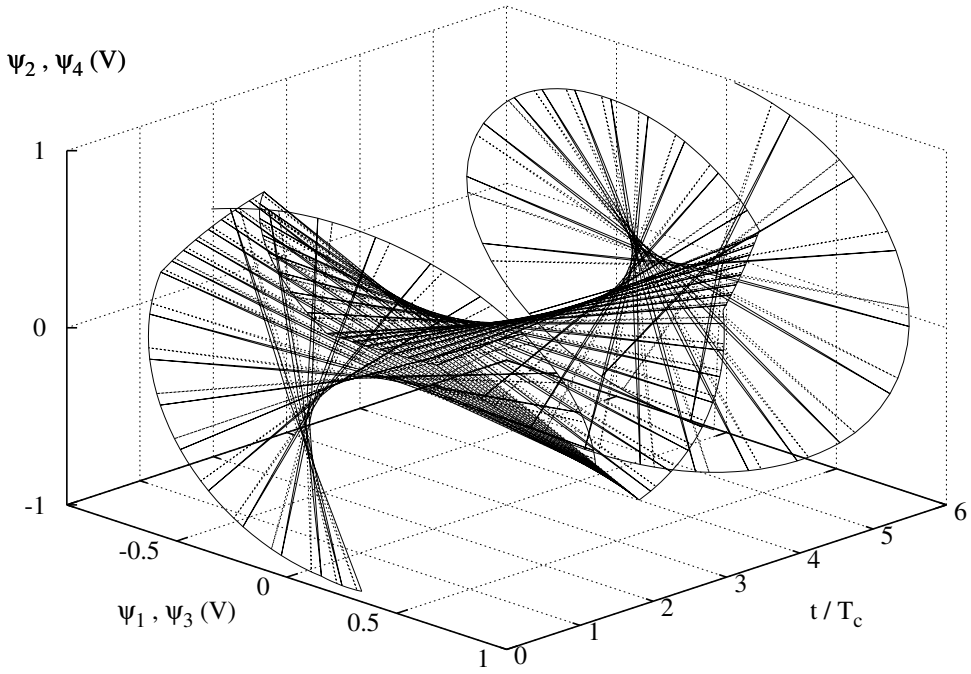

Figure 2.31: Complex manifold of length $L = 63$ ZCC (code $r = 5$) CSS showing 6 chips in the sequence and 32 samples per chip

2.5.3.4 CROSS DOT PRODUCTS

As for the GCL cross dot products shown in figure 2.26, the normalised accumulating dot product functions between basis vector $\Psi_1(t)$ and all other basis vectors ($\Psi_2(t)$,$\Psi_3(t)$ and $\Psi_4(t)$) of a length $L = 63$ ZCC CSS (code $r = 5$) are shown in figure 2.32. Notice the near zero value of the disturbance signal $(\Psi_1(t); \Psi_2(t))$ at $\tau_c/T_c = 63$ CSS chips.

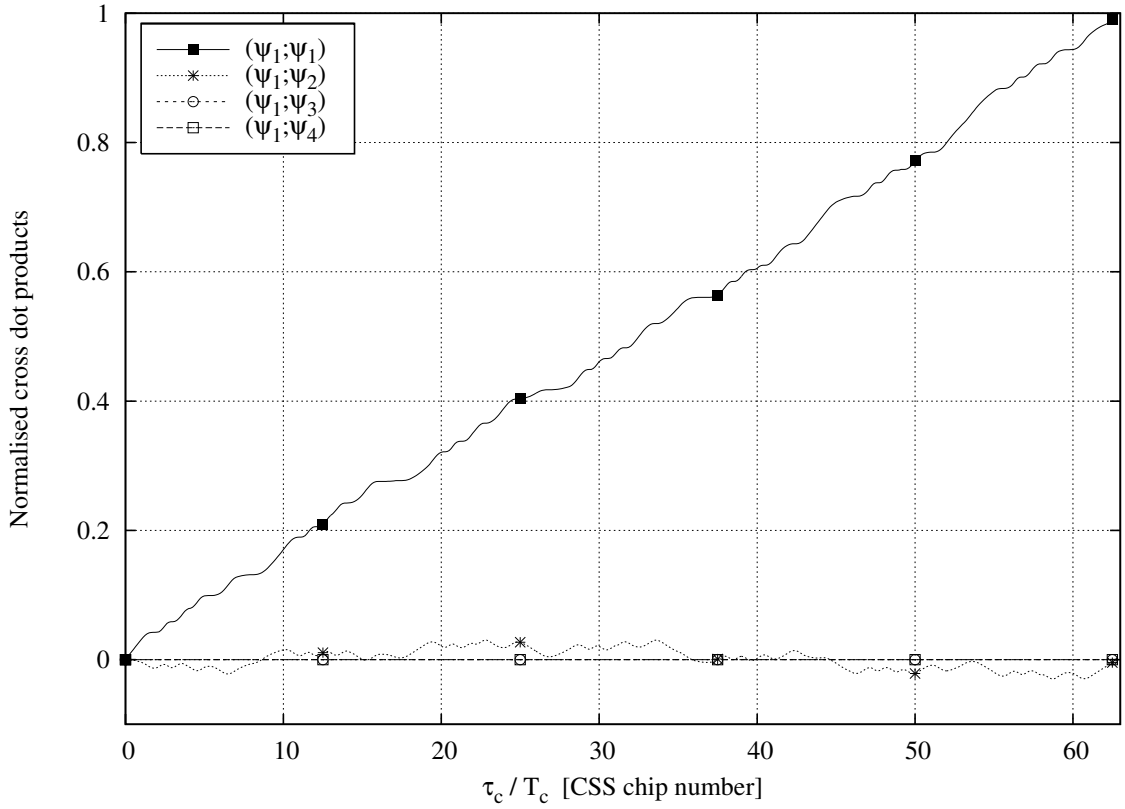

Figure 2.32: Auto and cross dot products between basis vectors of a length-63 ZCC (code $r = 5$) CSS

2.5.3.5 CORRELATION COMPARISON OF COMPLEX SEQUENCES

Similar to the GCL comparison graph in figure 2.27, the normalised complex correlation values at $\tau = 0$ (or $\tau_c/T_c = 63$ CSS chips) of a length $L = 63$ ZCC CSS family is shown in figure 2.33. The AC values are equal and all CC values are zero (ZCC codes are thus perfect complementary sequences [55]) from which excellent multi-user performance can be predicted (see table A.6 in Addendum A for the exact correlation values).

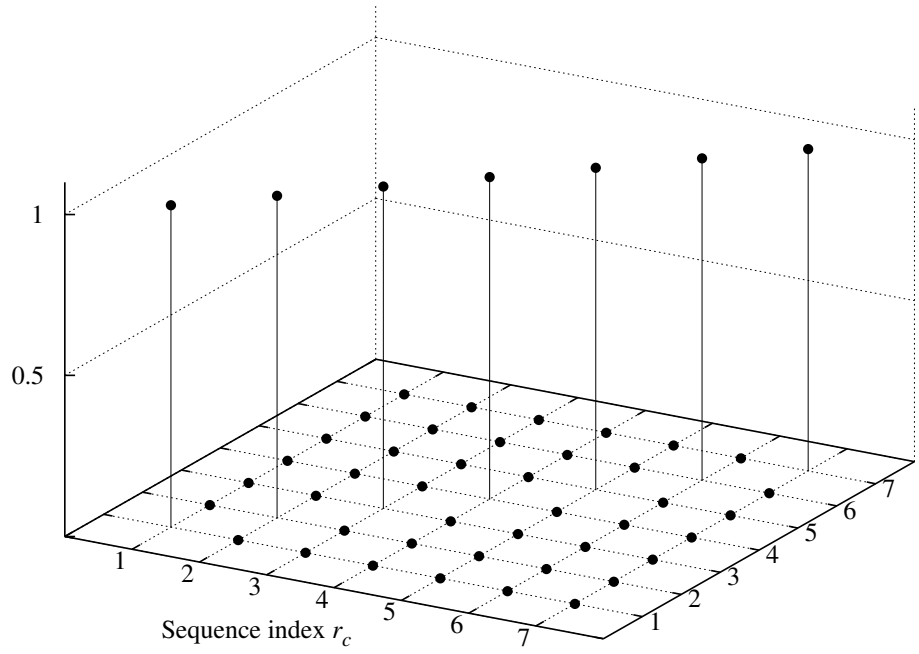

Figure 2.33: Complex aperiodic correlation comparison of a length $L = 63$ ZCC CSS family containing 7 complex spreading codes. The x- and y-axes show the code indexes and the z-axis the normalised correlation amplitudes at $\tau = 0$.

2.5.3.6 CORRELATION COMPARISON OF COMPONENTS OF COMPLEX SEQUENCES

Shown in figure 2.34 are the component correlation values (the sequence indexing is the same as for the GCL comparison shown in figure 2.28; sequences 1 and 2 (figure 2.34) are the real and imaginary components respectively of the CSS with index 1 in figure 2.33). The AC values are constant and most of the CC values are zero, except for certain patterns as shown in figure 2.34 (also see table A.7 and A.8 in Addendum A). The non-zero CC values can be ascribed to the similarity of the temporal ZCC components and the fact that each code is used without its complex counterpart (the codes are designed as complex codes and using these codes singularly, the "complex protection" is lost). By choosing the correct subset of codes, the same performance can be expected, using components of CSSs, as with complete CSSs.

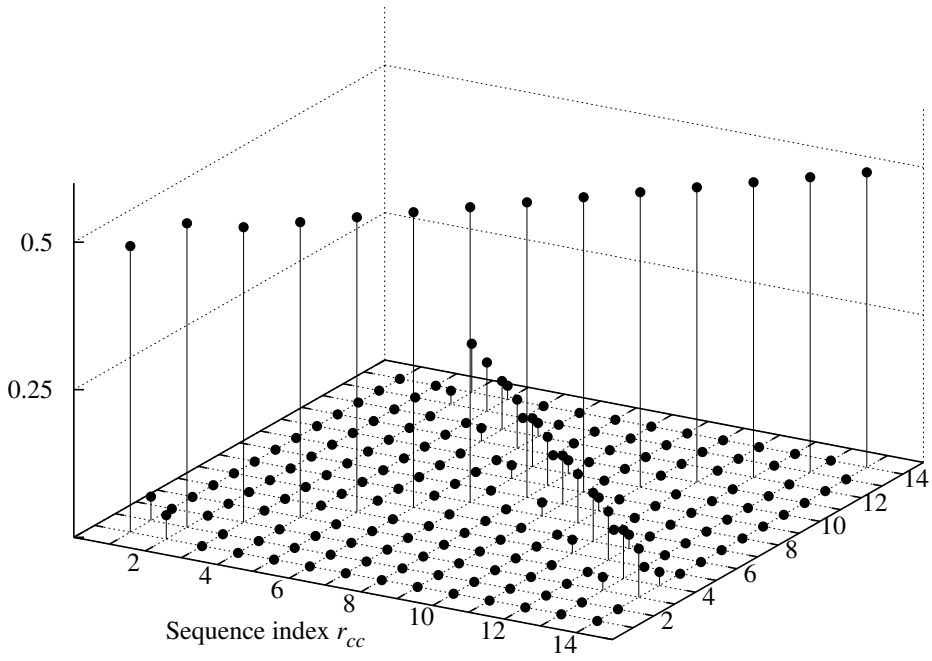

Figure 2.34: Complex aperiodic correlation comparison of the complex components of the 7 CSSs of a length $L = 63$ ZCC family (totalling 14 component codes). The x- and y-axes show the code indexes and the z-axis the normalised correlation amplitudes at $\tau = 0$.

CHAPTER 3

CHANNEL SIMULATION MODELS

3.1 CHAPTER SUMMARY

This chapter explains channel effects with a focus on small-scale effects encountered in a wide band mobile wireless communication system, including AWGN and frequency selective fading. A channel model based on Clarke's model presented in [45, 56] is developed as part of the simulation platform to test and evaluate the performance of the WCDMA communication system. A more detailed study on channel effects, including large-scale effects, can be found for example in [5]. The channel effects simulated by the model are evaluated and validated against theoretical models in terms of time, spectral and probability density functions. The uncoded simulation platform structure and setup including a multi-user fading simulator are also presented.

3.2 THE GAUSSIAN CHANNEL

Probably the most popular and simplest real-input, real-output channel model is the AWGN channel [4, 14]. Several different electromagnetic sources (with different probability distributions) including thermal noise in the channel, electronics in the communication system and other interfering communication systems contribute to create a Gaussian distributed noise variable η defined by the PDF (probability density function):

$$p(\eta|\mu,\sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(\eta-\mu)^2}{2\sigma^2}\right], \tag{3.1}$$

where μ and σ^2 are the expected value and variance of the noise variable η respectively. The noise causes variations in the signal amplitude and phase and subsequently introduces errors in the transmitted signal and limits the error performance or quality of the received signal.

3.3 SMALL-SCALE FADING EFFECTS

The two main small-scale fading effects are multipath delay spread and Doppler spread. The characteristics of both the channel and transmitted signal determine which channel effects are experienced in the communication system. Multipath delay spread is a time dispersion phenomenon which can cause frequency selective fading while Doppler spread is a frequency dispersion phenomenon causing time selective fading [5].

3.3.1 MULTIPATH PROPAGATION

The communication path between transmitter and receiver in a mobile environment is constantly changing due to relative motion in the radio propagation path. Multipath effects due to reflection, diffraction and scattering cause a transmitted signal to arrive in different variations in time, amplitude and phase at the receiver. Rapid changes in signal amplitude over a small time interval and time dispersion is caused, resulting in signal distortion and inter-symbol interference (ISI). Time dispersion can cause either flat or frequency selective fading in the transmitted signal.

3.3.1.1 FLAT FADING

Flat fading or amplitude varying channels are narrowband channels where the radio channel has a constant average gain and linear phase response over a larger bandwidth than the bandwidth of the transmitted signal. The transmitted signal's spectral shape is received unaltered, though the received signal power varies with time, due to gain fluctuations caused by multipath effects. A signal with bandwidth BW_{signal} and symbol period $T_S = BW_{signal}^{-1}$ will experience flat fading if $BW_{signal} \ll B_C$ and $T_S \gg \sigma_\tau$ where B_C and σ_τ are the channel's coherence bandwidth and RMS (root mean square) delay spread respectively [5].

3.3.1.2 FREQUENCY SELECTIVE FADING

Frequency selective channels are wideband channels where the radio channel has a constant average gain and linear phase response over a smaller bandwidth than the bandwidth of the transmitted signal. The transmitted signal's spectral shape is altered because the channel's gain is different for different frequency components in the signal. Frequency selective fading is caused by multipath delays in the channel and results in ISI. A signal with bandwidth BW_{signal} and symbol period $T_S = BW_{signal}^{-1}$ will experience

frequency selective fading if $BW_{signal} > B_C$ and $T_S < \sigma_\tau$ where B_C and σ_τ are the channel's coherence bandwidth and RMS delay spread respectively [5].

3.3.2 DOPPLER SPREAD

A Doppler shift or frequency modulation of the carrier frequency is caused by relative movement between the transmitter and receiver. The instantaneous shift in frequency f_d [Hz] of a single transmitted signal of wavelength λ [m] experienced by the receiver assuming a static transmitter is [5]:

$$f_d = \frac{v}{\lambda} cos(\theta), \tag{3.2}$$

if the receiver travels at v [m/s] and the angle of arrival of the received signal relative to the direction of movement of the receiver is θ. Each multipath signal can experience a unique Doppler shift and with moving objects in the communication path (especially if these objects move faster than the relative motion between the transmitter and receiver) a time varying Doppler shift is induced on each multipath signal. A Doppler spread hence occurs with varying Doppler shifts up to a maximum spread frequency, causing amplitude variations in time in the received signal.

The PSD (power spectral density) of the output of a Doppler spreading process is given by [5, 46]:

$$S_{hh}(f) = \begin{cases} \frac{1}{\pi\sqrt{f_D^2 - f^2}} & \text{for } |\mathrm{f}| < \mathrm{f_D} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

where f_D is the maximum Doppler spread frequency. Figure 3.1 shows the PSD of a Doppler spreading process with $f_D = 100$ Hz. Depending on how fast the channel impulse response changes, either fast or slow fading can occur.

3.3.2.1 FAST FADING

Fast fading occurs when the channel impulse response changes rapidly within a symbol period of the transmitted signal. Frequency dispersion causes time selective fading and distortion of the signal. A signal with bandwidth BW_{signal} and symbol period $T_S = BW_{signal}^{-1}$ will experience fast fading if $T_S > T_C$ and $BW_{signal} < f_D$ where T_C and f_D are the channel's coherence time and maximum Doppler spread respectively [5].

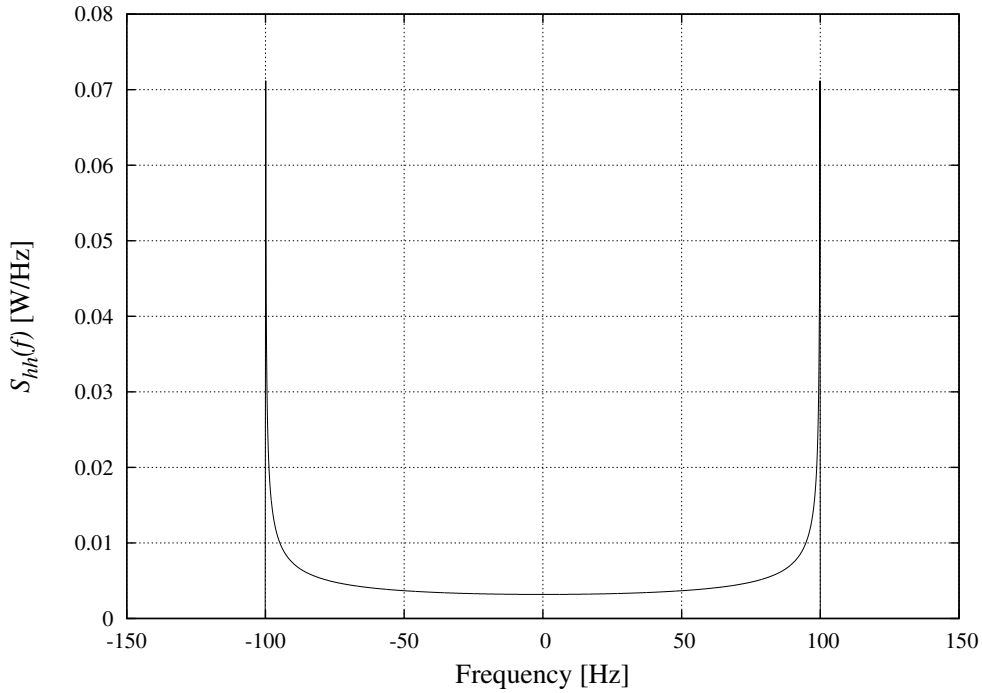

Figure 3.1: Doppler spectrum for $f_D = 100$ Hz

3.3.2.2 SLOW FADING

Slow fading occurs when the channel impulse response changes much slower than the transmitted signal and the channel can be assumed static over a period larger than the symbol period of the transmitted signal. A signal with bandwidth BW_{signal} and symbol period $T_S = BW_{signal}^{-1}$ will experience slow fading if $T_S \ll T_C$ and $BW_{signal} \gg f_D$ where T_C and f_D are the channel's coherence time and maximum Doppler spread respectively [5].

3.4 MULTIPATH CHANNEL RESPONSE

The impulse response of a radio channel is a channel characterisation describing multipath propagation through the channel. A multipath channel can be modelled as a linear filter with time varying impulse response $h(t, \tau)$ where t represents time variations due to relative motion between the transmitter and receiver and τ represents the multipath delay for a given time instance t.

To model a radio channel, the multipath delay axis τ is divided into L_d delay segments or excess delay bins, where all multipath components arriving within the time

span of a single bin is represented as a single multipath component. If only baseband transmission is assumed, the channel impulse response can be expressed as [5]:

$$h_b(t,\tau) = \sum_{i=1}^{L_d} a_i(t,\tau)\exp\left[j\theta_i(t,\tau)\right]\delta(\tau - \tau_i(t)), \tag{3.4}$$

where $a_i(t,\tau)$ and $\tau_i(t)$ are the i^{th} multipath component's amplitude and time delay at time t. The phase term $\theta_i(t,\tau)$ in equation 3.4 represents the phase shift of the i^{th} multipath component. By assuming that the channel is time invariant, equation 3.4 can be simplified to [5]:

$$h_b(\tau) = \sum_{i=1}^{L_d} a_i\exp(j\theta_i)\delta(\tau - \tau_i). \tag{3.5}$$

3.4.1 POWER DELAY PROFILES

Power delay profiles are used in the simulation of digital communication systems in multipath fading channels and can be derived from the channel impulse response. A time-invariant power delay profile can be constructed by averaging an ensemble of delay profiles obtained by making several measurements of $|h_b(t,\tau)|^2$ defined in equation 3.4. The time-invariant power delay profile is then given by [5, 45]:

$$P(\tau) = \overline{|h_b(t,\tau)|^2} = \sum_{i=1}^{L_d} P(\tau_i)\delta(\tau - \tau_i), \tag{3.6}$$

where $P(\tau_i) = \overline{a_i}^2$ is the average power in the i^{th} multipath component and defined by a power delay profile model, such as an exponentially decaying model.

3.5 RAYLEIGH AND RICIAN FADING

Radio waves in a multipath channel can propagate via a LOS path and/or a number of NLOS (non-line-of-sight) paths. The amplitude and phase distribution of the channel affected received signal depend on the ratio of power defined by the Rician factor

$$K_{Rice} = \frac{P_{LOS}}{P_{NLOS}} \tag{3.7}$$

where P_{LOS} is the average power in the LOS component and P_{NLOS} is the average power in the NLOS components.

3.5.1 RAYLEIGH DISTRIBUTION

Rayleigh fading (typical in indoor or urban environments) occurs when multiple indirect (NLOS) paths between the transmitter and receiver exist with no distinct dominant (LOS) path. The Rician factor is therefore $K_{Rice} = 0 = -\infty$ dB. The Rayleigh A-PDF (amplitude PDF) is given by [5, 7]:

$$p(r|\sigma^2) = \begin{cases} \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2}\right) & \text{if } (0 \leq \text{r} < \infty) \\ 0 & \text{if } (\text{r} < 0) \end{cases} \tag{3.8}$$

where σ^2 is the average power of the received signal before envelope detection and the phase PDF is given by [45]:

$$p(\phi) = \frac{1}{2\pi} \quad \text{for } (-\pi \leq \phi \leq \pi), \tag{3.9}$$

assuming an input signal with constant phase.

3.5.2 RICIAN DISTRIBUTION

Rician fading (typical in outdoor environments with an unobstructed LOS path) occurs when a distinct LOS path and a number of NLOS paths exist between the transmitter and receiver. The Rician factor lies anywhere between $K_{Rice} = -\infty$ dB (a Rayleigh fading channel) and $K = \infty$ dB (an AWGN channel). The Rician A-PDF (Amplitude Probability Density Function) can be expressed as [5]:

$$p(r|\sigma^2, K_{lin}) = \begin{cases} \frac{r}{\sigma^2}\exp\left(-\frac{r^2}{2\sigma^2} - K_{lin}\right) I_0\left(\frac{r\sqrt{2K_{lin}}}{\sigma}\right) & \text{if } (0 \leq \text{r} < \infty) \\ 0 & \text{if } (\text{r} < 0) \end{cases} \tag{3.10}$$

where K_{lin} is the linear Rician factor defined in equation 3.7 (see also equation 3.21) and $I_0(\cdot)$ is the modified Bessel function of the first kind and zero-order. The Rician phase PDF is defined based on the equation given in [45] as:

$$p(\phi) = \frac{1}{2\pi}\exp\left(-K_{lin}\right)\left\{1 + \kappa_\phi\sqrt{\pi K_{lin}}\exp\left(\kappa_\phi^2 K_{lin}\right)\left[1 + \text{erf}\left(\kappa_\phi\sqrt{K_{lin}}\right)\right]\right\}, \tag{3.11}$$

with $\kappa_\phi = \cos(\phi)$ and $-\pi \leq \phi \leq \pi$, with ϕ the phase response of the channel at a given time instance.

3.6 CHANNEL MODELLING

3.6.1 AWGN CHANNEL MODEL

To simulate AWGN channel effects, the setup shown in figure 3.2 is used. The stochastic noise process $\eta(t)$ is generated by taking uniformly distributed samples between 0 and 1 and applying a probability distribution transformation algorithm to obtain the Gaussian amplitude distribution. To approximate a random process on a deterministic simulation platform, algorithms such as the Wichmann-Hill [57] algorithm is used to ensure characteristics such as long run length of the pseudo random generated sequence (the built-in random number generator functions in programming languages generate sequences with insufficient repetition length). The Marsaglia-Bray [58] algorithm is used to transform uniform samples to Gaussian distributed samples.

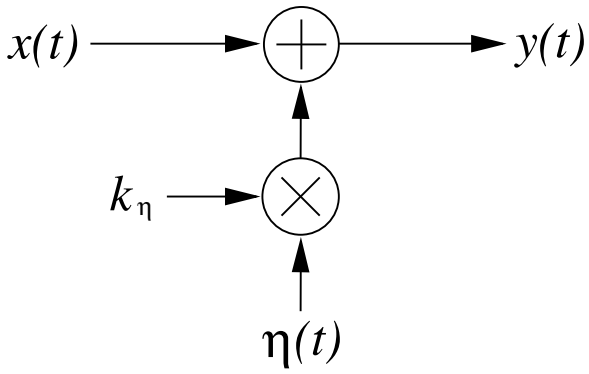

Figure 3.2: Theoretic AWGN model configuration

3.6.1.1 NOISE SCALING

To evaluate a communication system under noisy channel conditions, different SNRs (signal-to-noise-ratios) must be considered. By assuming that the input signal $x(t)$ and the noise signal $\eta(t)$ in figure 3.2 have both zero averages, the SNR in the output signal $y(t)$ is altered by scaling the variance of $\eta(t)$ by the noise scaling factor k_η defined in [14, 45]:

$$k_\eta^2 = \frac{\sigma_{x(t)}^2 f_{samp}}{2\left(\frac{\varepsilon_b}{N_0}\right) f_{bit}} \tag{3.12}$$

where $\sigma_{x(t)}^2$ is the variance of the input signal $x(t)$, f_{samp} is the sampling frequency, f_{bit} is the uncoded bit rate and $\frac{\varepsilon_b}{N_0}$ is the linear rate-compensated SNR per source bit (ε_b is the energy per uncoded bit and N_0 is the single-sided noise PSD). Since the value k_η is

a power scaling factor, the squared value k_η^2 is used in equation 3.12 to produce scaled noise samples with variance $k_\eta^2\sigma_{\eta(t)}^2$, where $\sigma_{\eta(t)}^2$ is the variance of the noise process $\eta(t)$. By assuming baseband transmission and unity variance in the input signal $x(t)$, equation 3.12 can be written as:

$$k_\eta^2 = \frac{n_{samp}}{2\left(\frac{\varepsilon_b}{N_0}\right)R_c} \tag{3.13}$$

where n_{samp} is a constant defining the number of samples used to represent a single input bit in the channel (after encoding and modulation), $\frac{\varepsilon_b}{N_0}$ is the linear SNR per bit and R_c is the code rate, assuming a channel coding scheme is employed in the communication system.

3.6.2 CLARKE'S FLAT FADING CHANNEL MODEL

Clarke's flat fading model [44] is a widely used statistical multipath fading channel model based on electromagnetic scattering [5]. The model is based on the fact that an electromagnetic field $E(t)$ consists of an inphase ($E_I(t)$) and quadrature ($E_Q(t)$) component [59] and that both quadrature components are Gaussian random processes. The envelope of the received E-field is therefore given by

$$|E(t)| = \sqrt{[E_I(t)]^2 + [E_Q(t)]^2}, \tag{3.14}$$

exhibiting a Rayleigh PDF [7].

3.6.2.1 COMPLEX CHANNEL MODEL

Since complex signals are employed in the communication system considered in this study, a complex channel model based on the model presented in [45] is used as part of the simulation platform. Assume a complex signal $x(t) = x_I(t) + jx_Q(t)$ is transmitted through a flat fading channel introducing a fading amplitude $\alpha(t) = \alpha_I(t) + j\alpha_Q(t)$ in the output signal $y(t) = y_I(t) + jy_Q(t)$. The channel output is then described as

$$y(t) = C_{scale}[\alpha(t) + C_{LOS}]x(t), \tag{3.15}$$

where C_{scale} is a power scaling factor to ensure equality in power between the input and output signals of the channel and C_{LOS} is a constant introducing a LOS component in order to simulate Rician channels. Continuing from equation 3.15, the channel output

can further be written in quadrature components as:

$$y_I(t) = C_{scale}\left\{[\alpha_I(t) + C_{LOS}]x_I(t) - \alpha_Q(t)x_Q(t)\right\} \tag{3.16}$$

$$y_Q(t) = C_{scale}\left\{[\alpha_I(t) + C_{LOS}]x_Q(t) + \alpha_Q(t)x_I(t)\right\}. \tag{3.17}$$

3.6.2.2 POWER SCALING

The power scaling factor C_{scale} can be calculated from either equations 3.16 or 3.17. Assuming that the fading amplitude components $\alpha_I(t)$ and $\alpha_Q(t)$ are both zero-mean unity-variance processes, equation 3.16 can be written in terms of power transfer as:

$$P_{y_I(t)} = (C_{scale})^2\left\{[1 + (C_{LOS})^2]P_{x_I(t)} + P_{x_Q(t)}\right\}, \tag{3.18}$$

from which C_{scale} is calculated as [45]:

$$C_{scale} = \frac{1}{\sqrt{2 + (C_{LOS})^2}} \tag{3.19}$$

Equation 3.19 is derived by equating the power in $y(t)$ to $x(t)$ and assuming that the power in the inphase component equals the power in the quadrature component, which can be written as $P_{y_I(t)} = P_{x_I(t)} = P_{y_Q(t)} = P_{x_Q(t)}$.

3.6.2.3 RICIAN FACTOR

To calculate the Rician factor, equation 3.18 is written in terms of LOS and NLOS components as:

$$P_{y_I(t)} = \underbrace{(C_{scale})^2(C_{LOS})^2 P_{x_Q(t)}}_{LOS} + \underbrace{(C_{scale})^2\left(P_{x_I(t)} + P_{x_Q(t)}\right)}_{NLOS} \tag{3.20}$$

and using equation 3.7 the Rician factor can be expressed as [45]:

$$K_{Rice} = 20\log_{10}\left(\frac{C_{LOS}}{\sqrt{2}}\right) \tag{3.21}$$

3.6.2.4 INPHASE CHANNEL MODEL

The inphase channel model corresponding to equation 3.16 is shown in figure 3.3. The quadrature model can similarly be obtained from equation 3.17. As shown in figure 3.3, two statistically independent Gaussian random sources, G_I and G_Q generate zero-mean unity-variance outputs which are filtered by Doppler filters DF_I and DF_Q to produce the fading amplitude components $\alpha_I(t)$ and $\alpha_Q(t)$ respectively.

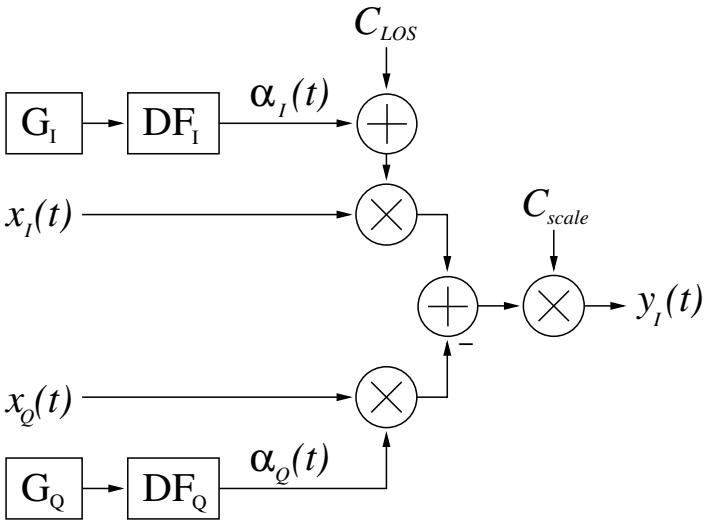

Figure 3.3: Inphase complex flat fading channel model

3.6.2.5 DOPPLER SPREAD SPECTRAL SHAPING FILTER

Filtering the random Gaussian variables produced by the sources G_I and G_Q in figure 3.3 with Doppler filters introduces a Doppler spreading effect into the input signal of the channel model. The design of a Doppler filter with spectral shape defined in equation 3.3 with variable cutoff frequency (to simulate different Doppler spreading frequencies) poses two main challenges [46] discussed below.

- The sampled channel waveform is bandlimited to a discrete frequency $f_D T_{samp}$, where f_D is the maximum Doppler frequency and T_{samp} is the sampling interval. Typical values for f_D in a mobile wireless environment include 33, 67 and 100 Hz (for a 900 MHz carrier and mobile speeds of 40, 80 and 120 km/h), while the value for T_{samp} in this study is approximately $2\mu s$. The maximum value of the discrete frequency $f_D T_{samp}$ is therefore 2×10^{-4} requiring an extremely narrowband discrete filter with very sharp cutoff rate.

- The spectral shape of the Doppler filter defined in equation 3.3 is irrational and cannot be constructed using classic filter design methods.

To address these issues the Doppler spectral shaping filter is based on the design presented in [46], where a fixed IIR filter, used to approximate the Doppler spectral shape, is followed by an interpolator to change the cutoff frequency of the filter.

The IIR filter of order $2K_f$ is constructed by cascading K_f bi-quad filters to obtain the filter's Z-domain transfer function [46]:

$$H(z) = A_f \prod_{k=1}^{K_f} \frac{1 + b_{k,1}z^{-1} + b_{k,2}z^{-2}}{1 + a_{k,1}z^{-1} + a_{k,2}z^{-2}} \tag{3.22}$$

where A_f is a scaling factor ensuring unity amplification and $b_{k,1}, b_{k,2}, a_{k,1}$ and $a_{k,2}$ are the filter coefficients. The magnitude response of a Doppler filter of order 8 is shown in figure 3.4, where the frequency is normalised relative to the Nyquist frequency. The normalised Doppler cutoff frequency of the fixed filter is chosen as $f_{DN} = 0.1$ rad/s.

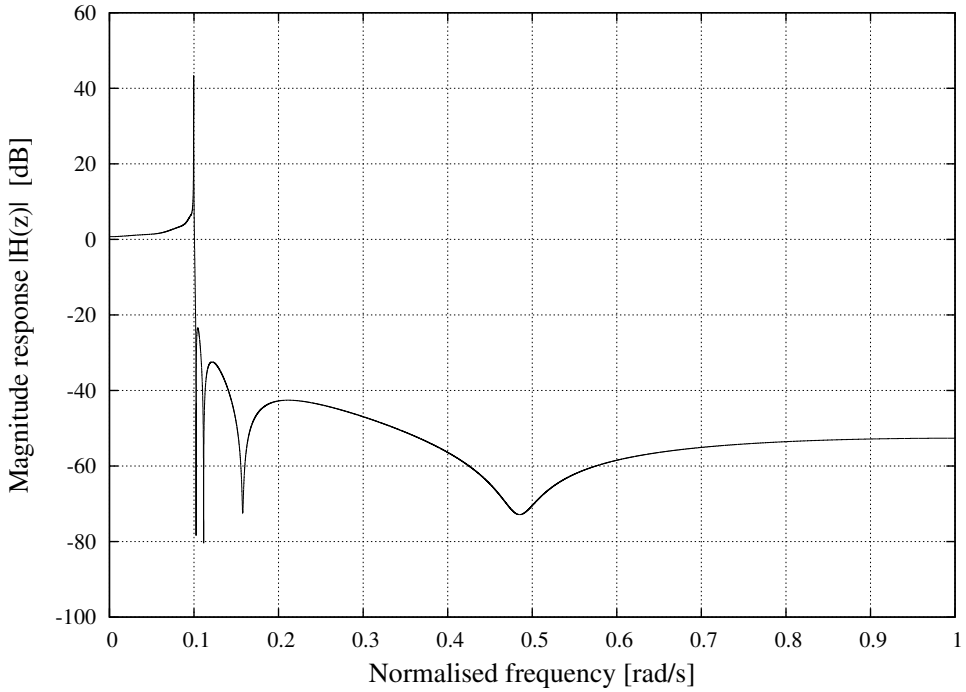

Figure 3.4: Normalised frequency magnitude response of an 8^{th} order IIR Doppler filter

In order to simulate different Doppler rates, the fixed Doppler filter with $f_{DN} = 0.1$ rad/s is used in conjunction with an interpolator to effectively scale the cutoff frequency to the desired Doppler rate. The desired cutoff or Doppler frequency f_D is related to the normalised Nyquist frequency f_{DN}, the sampling frequency f_{samp} and an interpolation factor I_f as:

$$f_D = \frac{f_{DN} f_{samp}}{2(I_f + 1)} \tag{3.23}$$

3.6.2.6 CHANNEL STATE INFORMATION

A signal transmitted through a fading channel will undergo amplitude scaling and phase shift due to the fading amplitude $\alpha(t)$ and phase $\phi(t)$ of the channel. Channel state information (CSI) is defined or carried by these two channel parameters, $\alpha(t)$ and $\phi(t)$.

Receivers in communication systems that employ synchronisation schemes and/or coherent demodulation need accurate estimates of $\phi(t)$, while knowledge of $\alpha(t)$ can improve error performance. It is therefore necessary to extract CSI from the channel at the receiver, either by analysing the received data signal or using a scheme that adds signalling information to the data stream.

In this study perfect CSI at the receiver is assumed by extracting $\alpha(t)$ and $\phi(t)$ directly from the channel. The instantaneous fading amplitude is calculated as [45]:

$$\alpha_{ins}(t) = \frac{\beta_{ins}(t)}{\beta_{avg}} = \sqrt{\frac{[\alpha_I(t) + C_{LOS}]^2 + [\alpha_Q(t)]^2}{2 + [C_{LOS}]^2}} \tag{3.24}$$

where $\beta_{ins}(t)$ and β_{avg} are the magnitudes of the instantaneous and average amplitude variations experienced by signals transmitted through the channel. The instantaneous phase response of the channel is the angle between the inphase and quadrature components of the fading amplitude and is calculated using the equation [45]:

$$\phi_{ins}(t) = -\arctan\left(\frac{\alpha_Q(t)}{\alpha_I(t) + C_{LOS}}\right) \tag{3.25}$$

3.6.3 MULTIPATH CHANNEL MODEL

A multipath channel can be simulated by placing several statistically independent flat fading channel (FFC) models in parallel. Each flat fading model represents either a single communication path through the channel, consisting of a bundle of waves or a combination of communication paths represented as a single path in the channel model. Each path is delayed and scaled independently as required by the power delay profile to create a frequency selective fading channel. A multipath channel model based on the design presented in [45] with $L_p = 3$ paths, as is used in this study, is shown in figure 3.5.

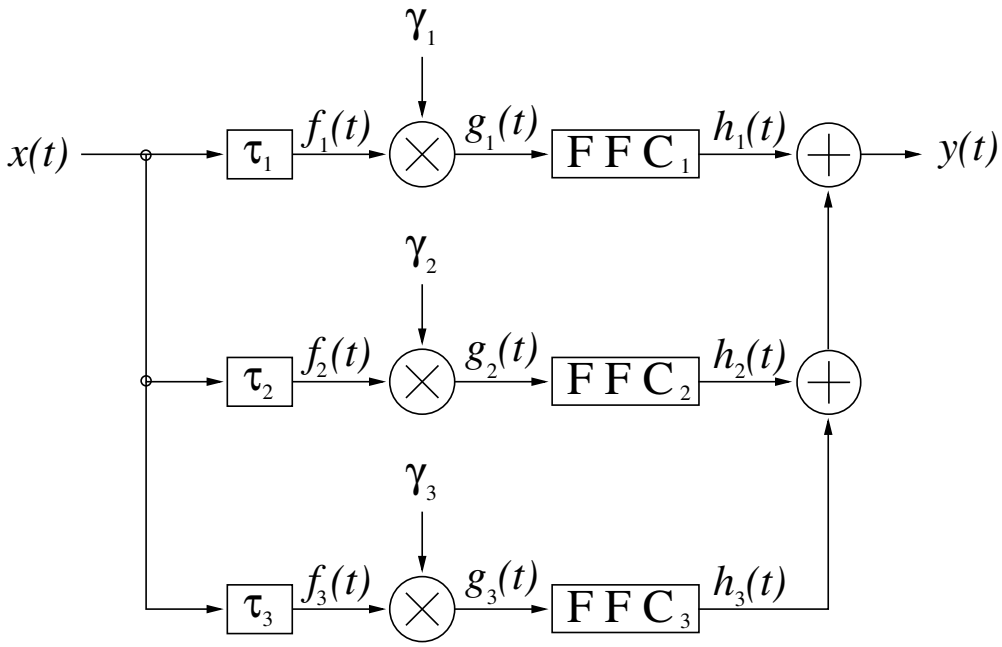

Figure 3.5: Multipath fading channel simulator with 3 communication paths

3.6.3.1 FUNCTIONING OF FADING SIMULATOR

All the time dependent signals shown in figure 3.5 can be treated as complex signals, where each component of a given complex signal experiences the same time delay and power scaling while passing through the channel. The transmitted signal $x(t)$ propagates via $L_p = 3$ paths (numbered as $i = 1, 2, 3$) through the channel, where each copy of $x(t)$ is delayed a time amount τ_i. The time-delayed signal $f_i(t)$ is then scaled (attenuated) by a factor γ_i defined by the power delay profile and passed through a flat fading channel (as described in section 3.6.2). Note that a unique Doppler spreading frequency and Rician factor can be defined for each path. The faded signals $h_i(t)$ are summed to produce the channel simulator output $y(t)$. By ensuring simulator power input-output equality (see section 3.6.3.2), AWGN effects can be introduced by connecting the output of figure 3.5 to the input of figure 3.2.

3.6.3.2 POWER SCALING

The multipath channel simulator should not change the power of the transmitted signal $x(t)$. To ensure that the power in the input and output signals are equal ($P_{x(t)} = P_{y(t)}$) the scaling factors γ_i with $i = 1, 2, 3$ are subjected to the condition [45]:

$$\sum_{i=1}^{3} (\gamma_i)^2 = 1. \tag{3.26}$$

3.6.3.3 POWER DELAY PROFILES

Exponentially decaying power delay profile models are commonly used in the simulation of multipath fading channels [5, 45] and are also used in this study. The power delay profile of an exponentially decaying model has the general form:

$$P(\tau) = k_p \exp\left(-\frac{\tau}{\tau_e}\right) \tag{3.27}$$

where k_p and τ_e are constants. The time constant τ_e can be calculated as:

$$\tau_e = -\frac{\tau_{max}}{\ln(P_{drop,lin})} \tag{3.28}$$

where τ_{max} is the maximum excess delay (the time delay during which multipath energy falls to a certain level below the maximum received energy [5]). $P_{drop,lin}$ is the linear measure of the amount of power difference between the strongest arriving multipath component and the component arriving τ_{max} seconds later. A typical value for $P_{drop,lin}$ is 0.001 ($P_{drop} = -30$ dB).

3.6.4 CALCULATION OF POWER DELAY PROFILE

The calculation of the power delay profile to simulate the frequency selective fading channel used in this study is considered in this section. A signal will experience frequency selective fading (see section 3.3.1.2) if $BW_{signal} > B_C$ and $T_S < \sigma_\tau$ where B_C and σ_τ are the channel's coherence bandwidth and RMS delay spread, respectively.

3.6.4.1 SIGNAL BANDWIDTH

If transmission in a wideband communication system (using RU filtered CSSs) occurs at 1000 symbols per second with 63 chips per symbol, the Nyquist bandwidth is $BW_{Nyq,signal} = 31.5$ kHz. The transmitted DSB modulated bandwidth is therefore $BW_{signal} = 63$ kHz and the transmitted symbol period T_S is equal to the chip period $T_c = 15.87$ μs. See section 6.6.1 for a critical evaluation on the choice of the data throughput rate.

Table 3.1: Multipath channel simulator parameter values

Parameter	User 1	User 2	User 3	User 4	User 5
P_1 [dB]	-0.13942	-0.35727	-0.15194	-1.27541	-0.57568
K_1 [dB]	9	9	9	9	9
$f_{D,1}$ [Hz]	100	67	33	100	67
τ_1 [μs]	0	20	40	60	80
P_2 [dB]	-15.13942	-11.35727	-15.15194	-6.27541	-9.57568
K_2 [dB]	0	0	0	0	0
$f_{D,2}$ [Hz]	67	33	67	33	100
τ_2 [μs]	150	130	190	110	170
P_3 [dB]	-30.13942	-22.35727	-24.15194	-17.27541	-18.57568
K_3 [dB]	-100	-100	-100	-100	-100
$f_{D,3}$ [Hz]	33	100	100	67	33
τ_3 [μs]	300	240	280	220	260
σ_τ [μs]	27.43	32.86	29.63	29.16	34.48
B_C [Hz]	7291.62	6087.14	6748.93	6859.05	5801.18

3.6.4.2 TIME CONSTANT

The power delay profile's time constant τ_e effectively determines the RMS delay spread σ_τ of the channel. The channel parameter values ensuring a frequency selective channel simulator for the communication system considered in this study are $\tau_{max} = 300$ μs and $P_{drop} = -30$ dB. Using equation 3.28 the time constant is calculated to be $\tau_e = 43.429$ μs.

3.6.4.3 PATH POWER AND DELAY

The power delay profile is constructed using equation 3.27. Since the first arriving multipath component is always the strongest in an exponentially decaying model and all other multipath components' excess delays τ_i are time delays relative to the main arriving component, the constant k_p can be calculated as $k_p = \exp\left(-\frac{\tau}{\tau_e}\right)$ with $\tau = 1$ μs. After calculating all path powers, a scaling factor is used to ensure that equation 3.26 is satisfied. The scaling factor is the sum of all path powers since $(\gamma_i)^2 = P(\tau_i)$ of the i^{th} multipath, assuming that the input signal $x(t)$ of the channel simulator shown in figure 3.5 has unity power.

3.6.4.4 MULTIPATH CHANNEL SIMULATOR

Table 3.1 shows the multipath channel simulator parameter values for 5 users with 3 statistically independent paths per user, similar to the simulator setup presented in [45, 56]. Each user's multipath channel has an exponentially decaying power delay profile and is a frequency selective fading channel as indicated by the RMS delay spread (σ_τ) and channel coherence bandwidth (B_C) values shown in table 3.1.

3.6.4.5 RICIAN FACTOR AND DOPPLER SPREAD

The Rician factor of the i^{th} path is chosen such that the main arriving multipath component ($i = 1$) of each user has a strong LOS component ($K_1 = 9$ dB). The second path has $K_2 = 0$ dB for each user and the third path is effectively a Rayleigh fading path ($K_3 = -100$ dB). The Doppler spreading frequencies $f_{D,i}$ are assigned randomly from the set $\{33; 67; 100\}$ Hz corresponding to typical mobile speeds (respectively $\{40; 80; 120\}$ km/h) assuming a 900 MHz carrier frequency. The channel has slow fading characteristics under all circumstances (see section 3.3.2.2).

3.7 SIMULATION PLATFORM

The uncoded simulation platform used in this study is shown in figure 3.6, combining the 4D modulation platform and all the channel models discussed in this chapter. A single user's antipodal data input signal $x(t)$ is modulated by the 4D transmitter (figure 2.9). The modulated signal then passes through the multipath fading channel simulator (figure 3.5) after which all CDMA users' faded signals are added together. The resultant combined signal is corrupted with AWGN effects using the AWGN generator shown in figure 3.2 and demodulated by the 4D receiver (figure 2.10) using the CSI obtained from the multipath fading channel using equations 3.24 and 3.25.

The received signal $y(t)$ is not only an approximation to $x(t)$, but also contains interference components of the other CDMA users apart from fading and AWGN channel effects. The 4D transmitter and multipath fading channel simulator shown in figure 3.6 must therefore be implemented for each CDMA user, though the AWGN generator and 4D receiver including CSI estimation are implemented for only one user. Channel encoding and decoding are likewise implemented for only one user, since the coding schemes considered in this study do not alter the statistics of the transmitted signal.

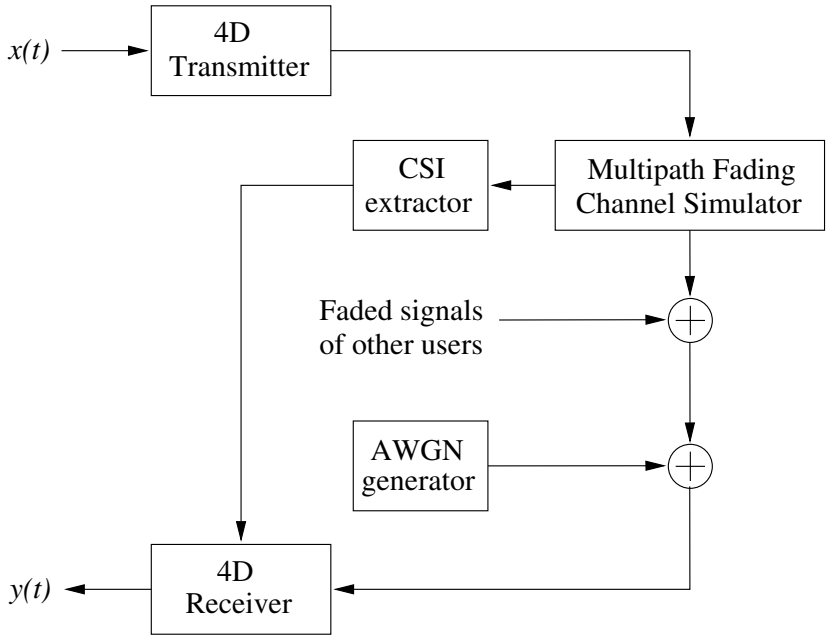

Figure 3.6: Uncoded simulation platform

3.8 CHANNEL MODEL VALIDATION

In order to obtain meaningful simulation results, the channel model must exhibit the same characteristics as defined by theory. The channel model is verified by plotting and comparing time signals, PSDs, amplitude and phase PDFs and BER graphs. In this section all these graphs are considered, with the exception of BER analysis, which is done only in Chapter 5.

3.8.1 AWGN CHANNEL

3.8.1.1 TIME SIGNALS

Baseband BPSK (binary phase shift keying) data signals, each having a data rate of 1000 bps, with different ε_b/N_0 values are shown in figure 3.7.

3.8.1.2 PROBABILITY DENSITY FUNCTION

The A-PDF of a zero mean, unity variance AWGN process $\eta(t)$ is shown in figure 3.8. The theoretical PDF defined in equation 3.1 with $\mu = 0$ and $\sigma^2 = 1$ is also shown to indicate that the measured PDF of the simulation model is nearly identical to theory.

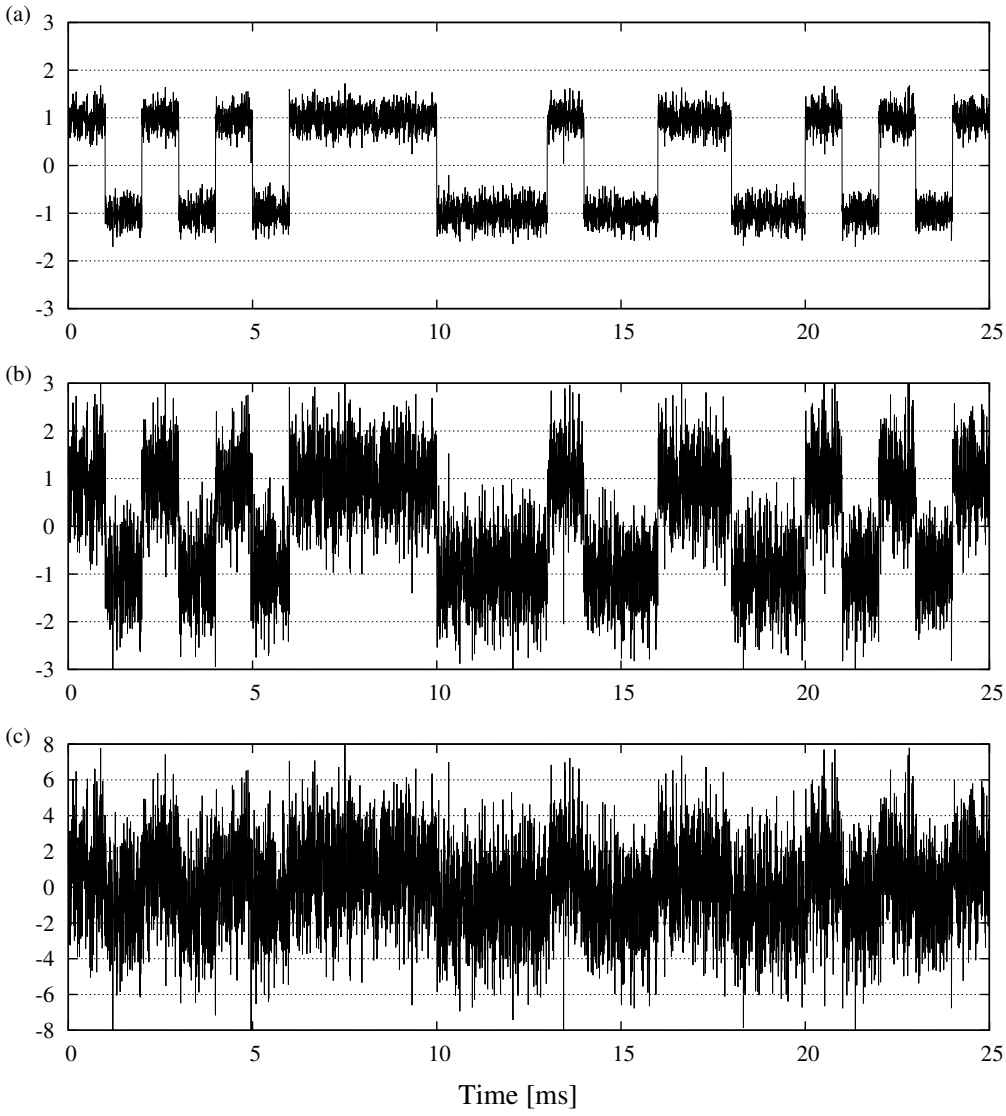

Figure 3.7: Temporal displays of an AWGN affected BPSK data signal with $\frac{\varepsilon_b}{N_0}$ equal to (a) 10 dB, (b) 0 dB and (c) -10 dB. The y-axes indicate the amplitude measured in Volt.

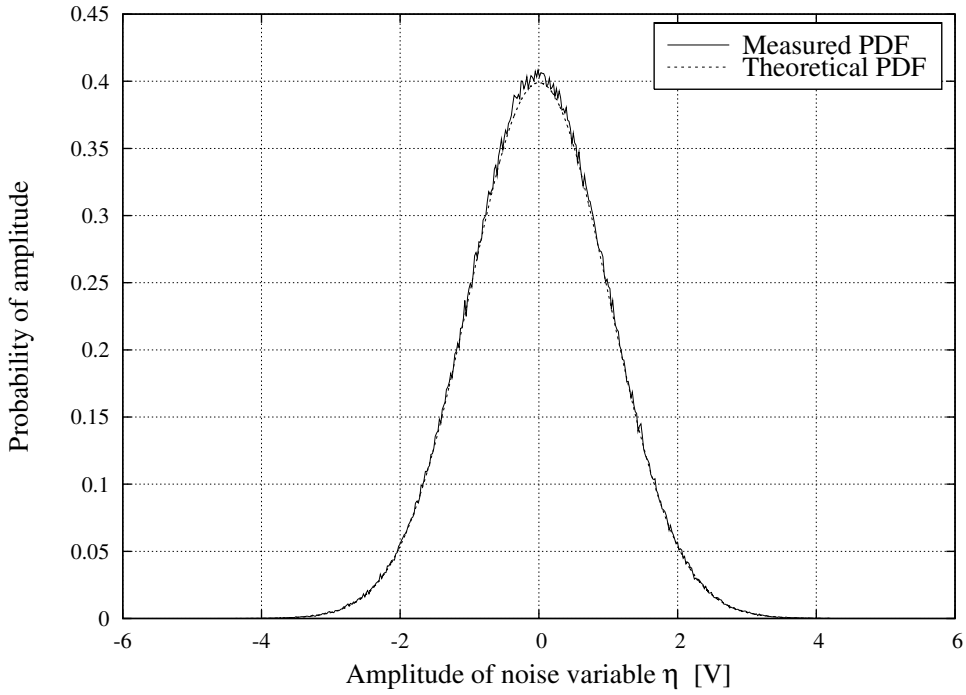

Figure 3.8: Amplitude PDF of a zero mean, unity variance AWGN process

3.8.2 FLAT FADING CHANNEL

3.8.2.1 TIME SIGNALS

A baseband BPSK data signal, with data rate 1000 bps, affected by different flat fading channel conditions, is shown in figure 3.9. The figure shows the received inphase signal after passing through a flat fading channel simulator (partially shown in figure 3.3) with a Doppler spread $f_D = 33$ Hz, varying Rician factors K and no AWGN. Figure 3.9 shows 2 intertwined envelope signals caused by the faded inphase and quadrature input signals' ($x_I(t)$ and $x_Q(t)$ respectively) contributions to both simulator output signals $y_I(t)$ and $y_Q(t)$ (see figure 3.3).

3.8.2.2 AMPLITUDE PROBABILITY DENSITY FUNCTIONS

The flat fading channel's fading amplitude (see equation 3.24) is the channel's instantaneous amplitude response and has a Rician distribution defined by equation 3.10. Figures 3.10 and 3.11 respectively show the theoretical and measured PDFs of the channel's fading amplitude for different Rician factors. The $K = -100$ dB Rice distribution is approximately equivalent to a Rayleigh distribution (see equation 3.8).

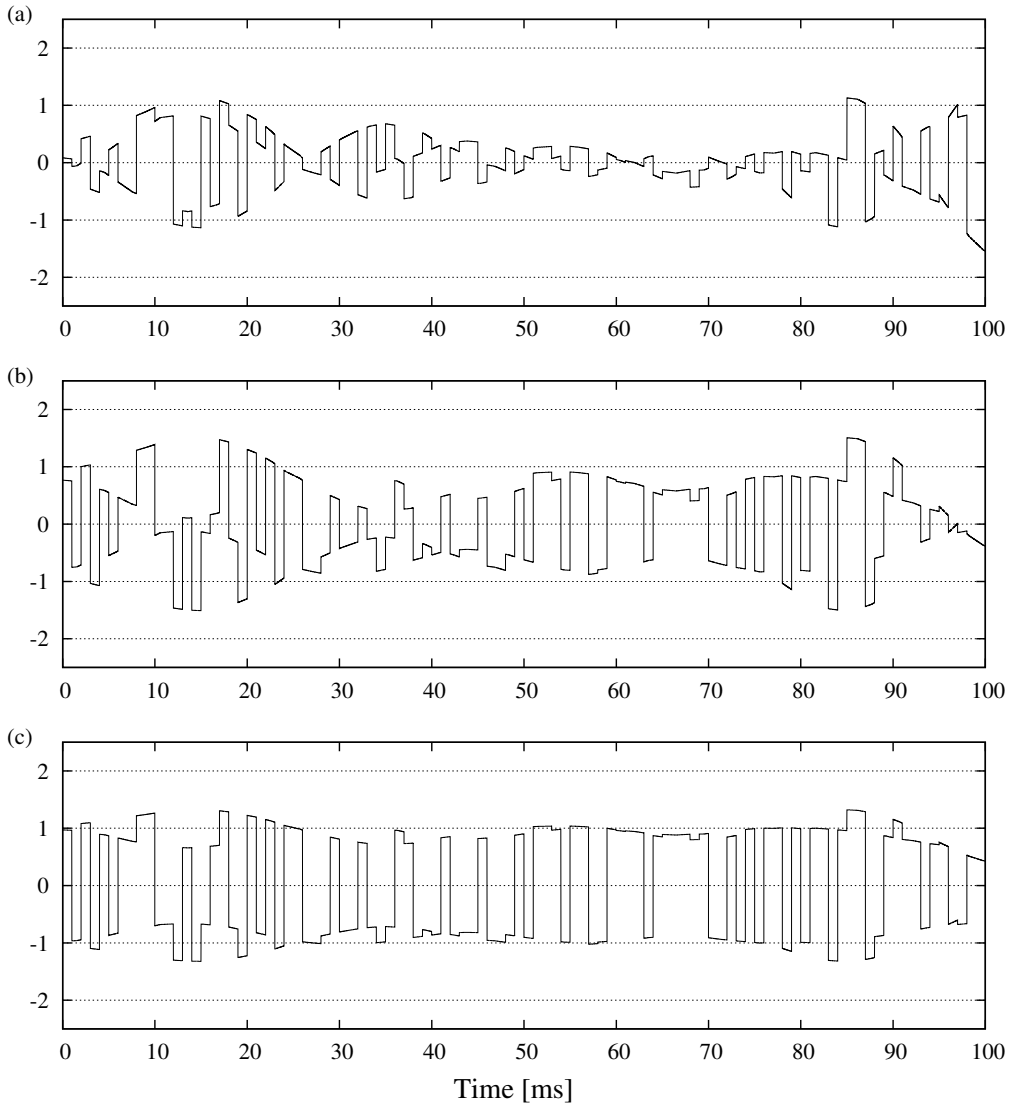

Figure 3.9: Temporal displays of a BPSK signal affected by a 33 Hz flat fading channel with (a) $K_{Rice} = -100$ dB, (b) $K_{Rice} = 0$ dB and (c) $K_{Rice} = 9$ dB. The y-axes indicate the amplitude measured in Volt.

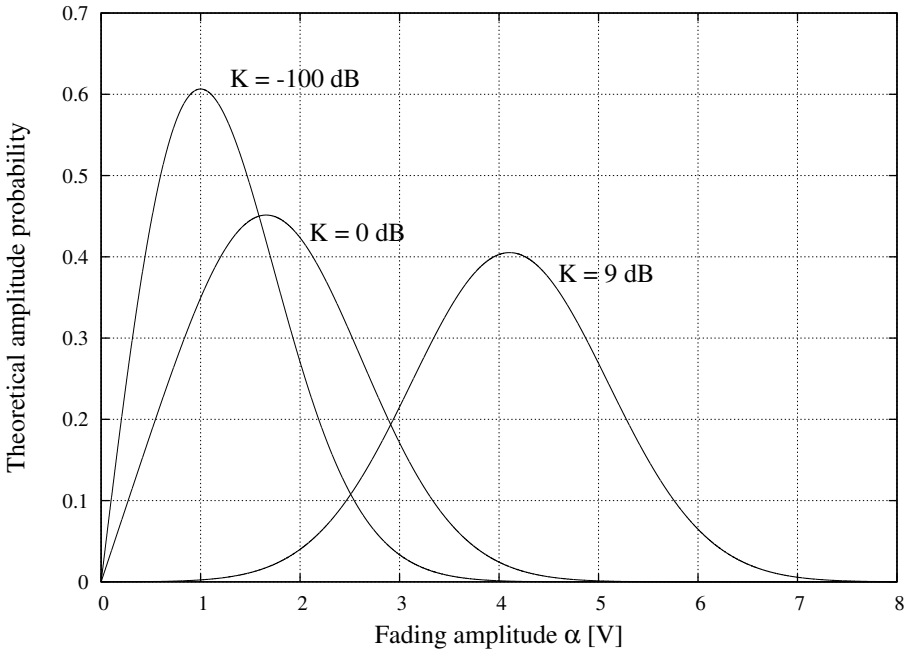

Figure 3.10: Theoretical PDFs of the fading amplitude of the channel simulator for different Rician factor values

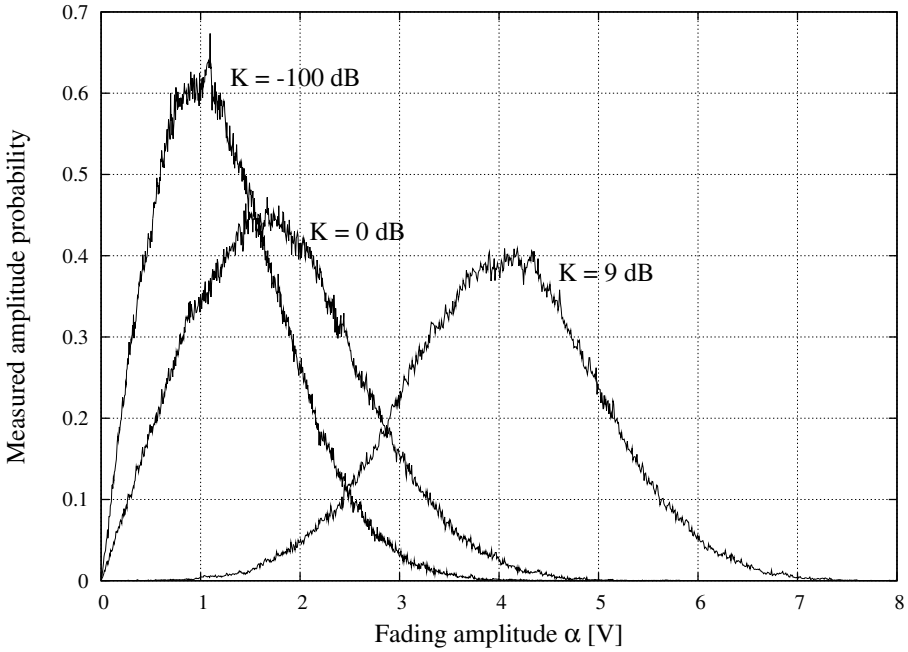

Figure 3.11: Measured PDFs of the fading amplitude of the channel simulator for different Rician factor values

3.8.2.3 PHASE PROBABILITY DENSITY FUNCTIONS

The flat fading channel's phase response (see equation 3.25) has a distribution defined by equation 3.11 (see section 3.5.2). Figure 3.12 shows the theoretical PDFs of the channel's phase response for different Rician factors. The $K = -100$ dB Rician phase distribution is approximately equivalent to a Rayleigh phase distribution (defined by equation 3.9, section 3.5.1). Figure 3.13 shows the measured phase PDFs of the channel simulator, indicating the near theoretical behaviour.

The phase distribution of the flat fading channel simulator is explained as follows. Assume an input signal into the channel with constant phase. The phase response is the phase change experienced by the signal while passing through the channel. If the input signal has a constant zero phase, the phase of the output signal will equal the phase response of the channel. Such cases are illustrated in figures 3.12 and 3.13. For the $K_{Rice} = -100$ dB Rician case (i.e. Rayleigh fading) the channel response is uniform (see equation 3.9) on the interval $[-\pi; \pi]$. The Rayleigh faded signal therefore loses its original phase due to the absence of a LOS signal component. As the LOS signal strength increases, the phase of the faded signal obtains a narrower distribution around the input phase.

3.8.2.4 DOPPLER FILTERS

The frequency impulse response of the flat fading channel simulator (partially shown in figure 3.3) is shown in figure 3.14 for different Doppler spreading frequencies. A 1 Watt DC signal is used as input to the channel simulator and the output displays a Doppler spectrum similar to the spectrum shown in figure 3.1 for $f_D = 100$ Hz. The channel simulator employs a 6^{th} order IIR Doppler filter (see section 3.6.2.5).

3.8.3 MULTIPATH CHANNEL

This section considers the multipath channel model as defined for user 1 in table 3.1.

3.8.3.1 POWER SPECTRAL DENSITY

The PSDs normalised to the transmitted 4D signal (see figure 2.9) employing CE-GCL CSSs (see section 2.4.2) are considered in this section. The PSD of the input and output signals of the multipath channel model, respectively indicated as $x(t)$ and $y(t)$ in figure 3.5, are shown in figure 3.15. The PSD of each of the output signals ($h_1(t)$,

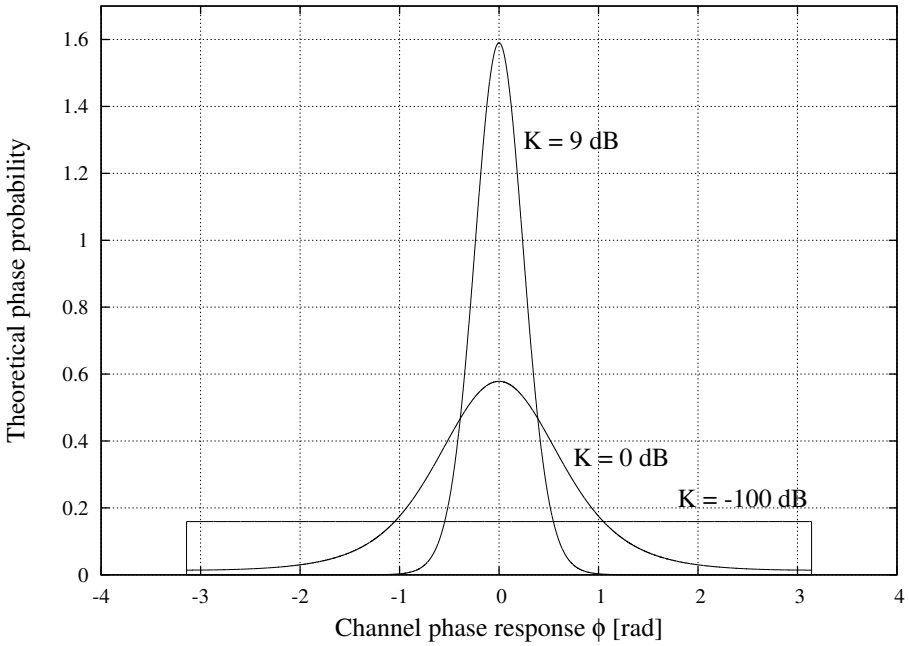

Figure 3.12: Theoretical PDFs of the channel simulator phase response for different Rician factor values

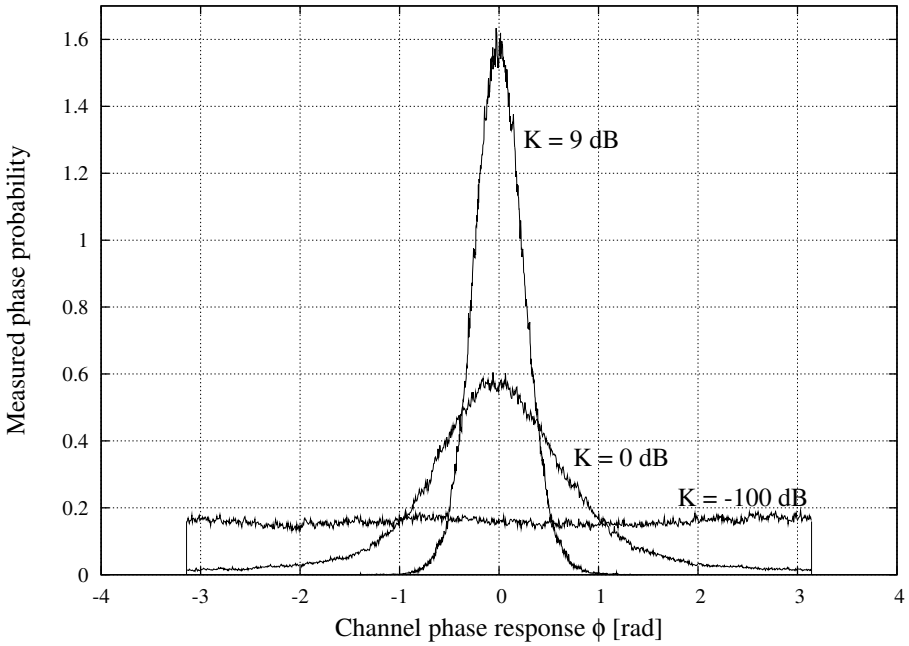

Figure 3.13: Measured PDFs of the channel simulator phase response for different Rician factor values

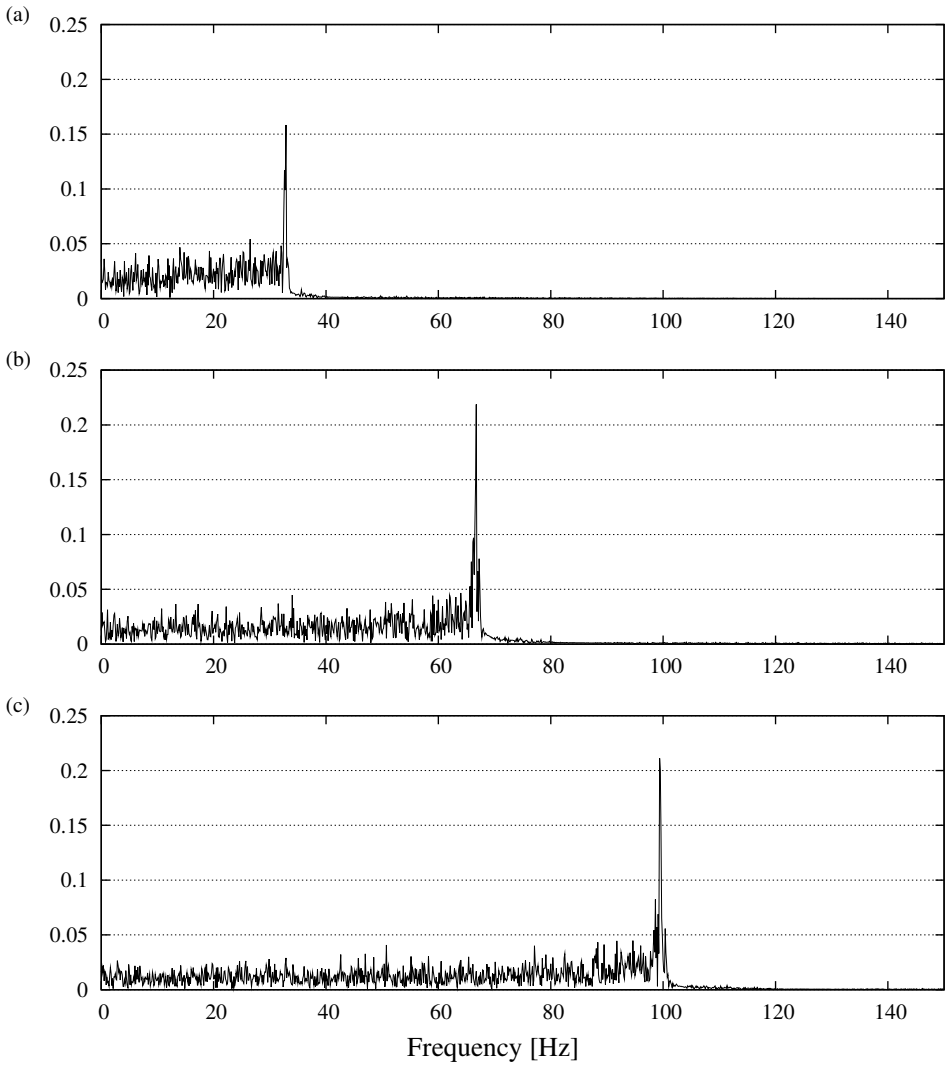

Figure 3.14: The figure shows the frequency impulse PSD response of the flat fading channel simulator (employing a sixth order IIR Doppler filter) output for (a) 33 Hz, (b) 67 Hz and (c) 100 Hz Doppler spreading frequencies. The y-axes indicate the single-sided PSD measured in W/Hz.

$h_2(t)$ and $h_3(t)$) of each of the flat fading modules (FFC_1 to FFC_3) shown in figure 3.5, are shown in figure 3.16. The path powers shown in table 3.1 correspond to the PSDs of each multipath signal in figure 3.16, though the Doppler spreading effects are not clearly visible on the PSD graphs (the signal bandwidth is 63 kHz and the Doppler frequencies are only 33, 67 and 100 Hz).

3.8.3.2 TEMPORAL GRAPHS

Doppler spreading is a frequency dispersion phenomenon causing variations in the envelopes of signals propagating through Doppler spreaded channels over time (see section 3.3). The temporal amplitude variations of the signals normalised to the transmitted 4D signal employing CE-GCL CSSs are considered in this section (the signals correspond to the PSD graphs discussed in section 3.8.3.1 above). The amplitude variations of the real input and output signals of the multipath channel model are shown in figure 3.17 and the output of each path is shown in figure 3.18. The instantaneous received power fluctuates according to the Doppler frequency, though the average received power obeys the values given in table 3.1.

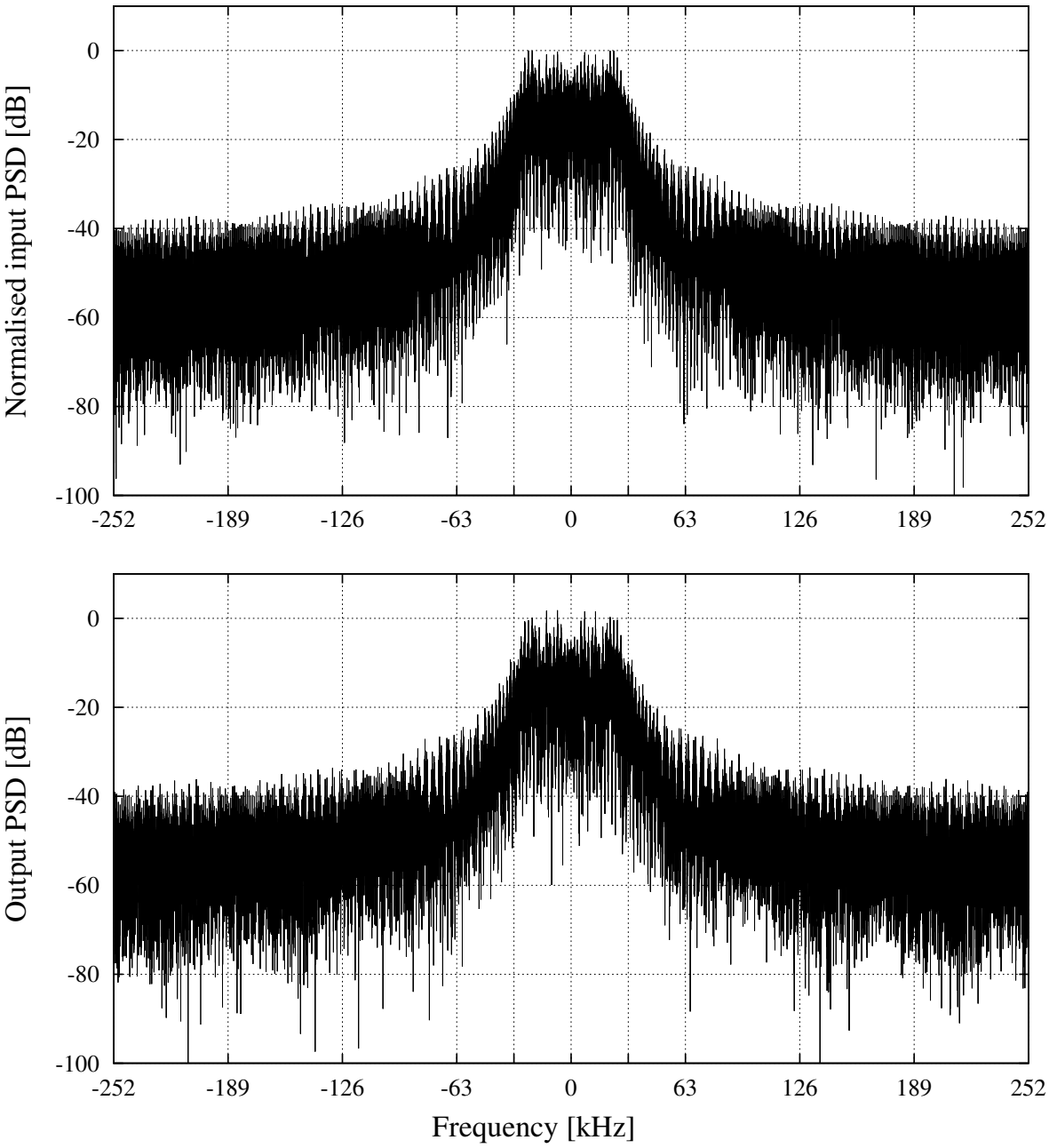

Figure 3.15: Measured PSD of the input and output signals of the multi-path channel model for user 1. The y-axes indicate the double-sided PSD measured in W/Hz.

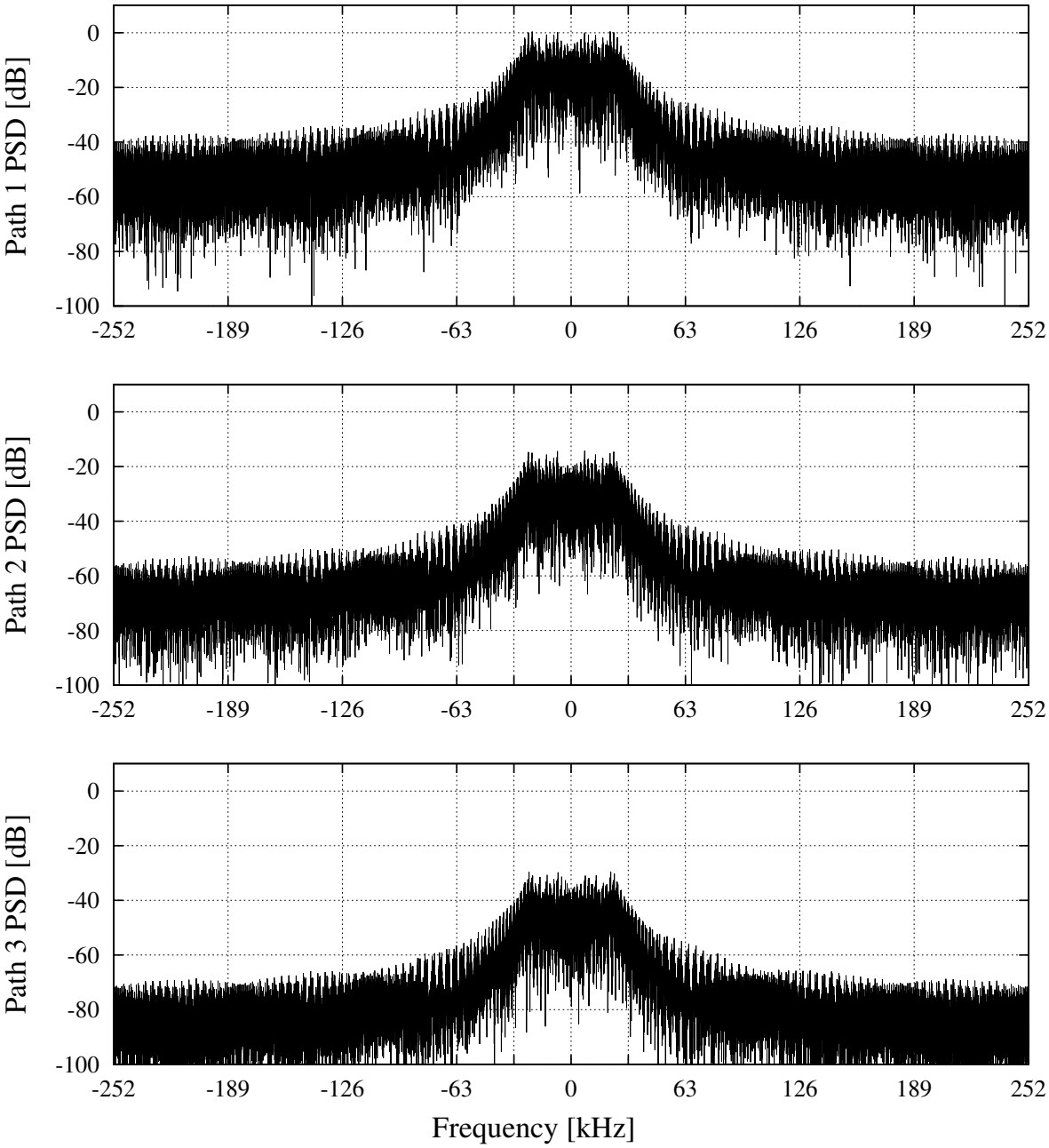

Figure 3.16: Measured PSD of the flat faded signals at the output of each of the paths of the multipath channel model for user 1. The y-axes indicate the double-sided PSD measured in W/Hz.

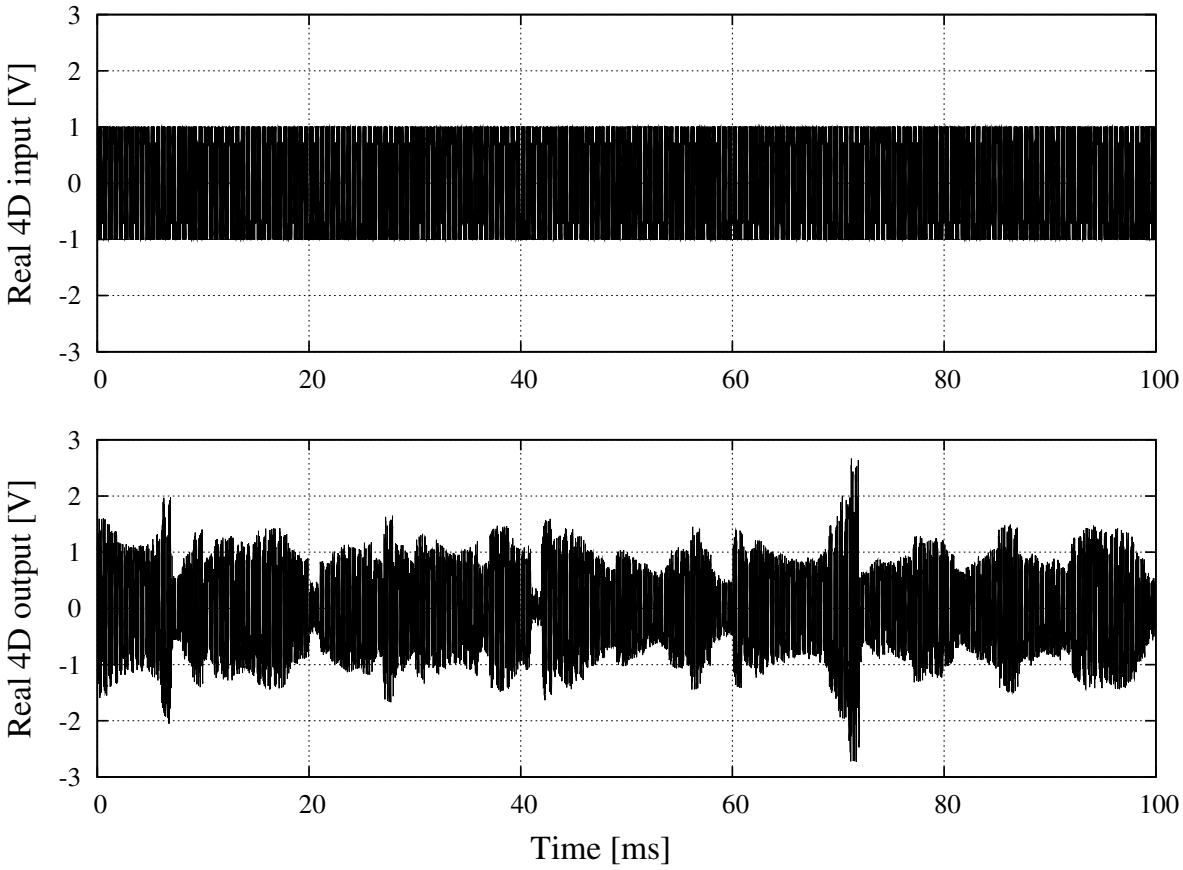

Figure 3.17: Measured temporal amplitude variations of the input and output signals of the multipath channel model for user 1

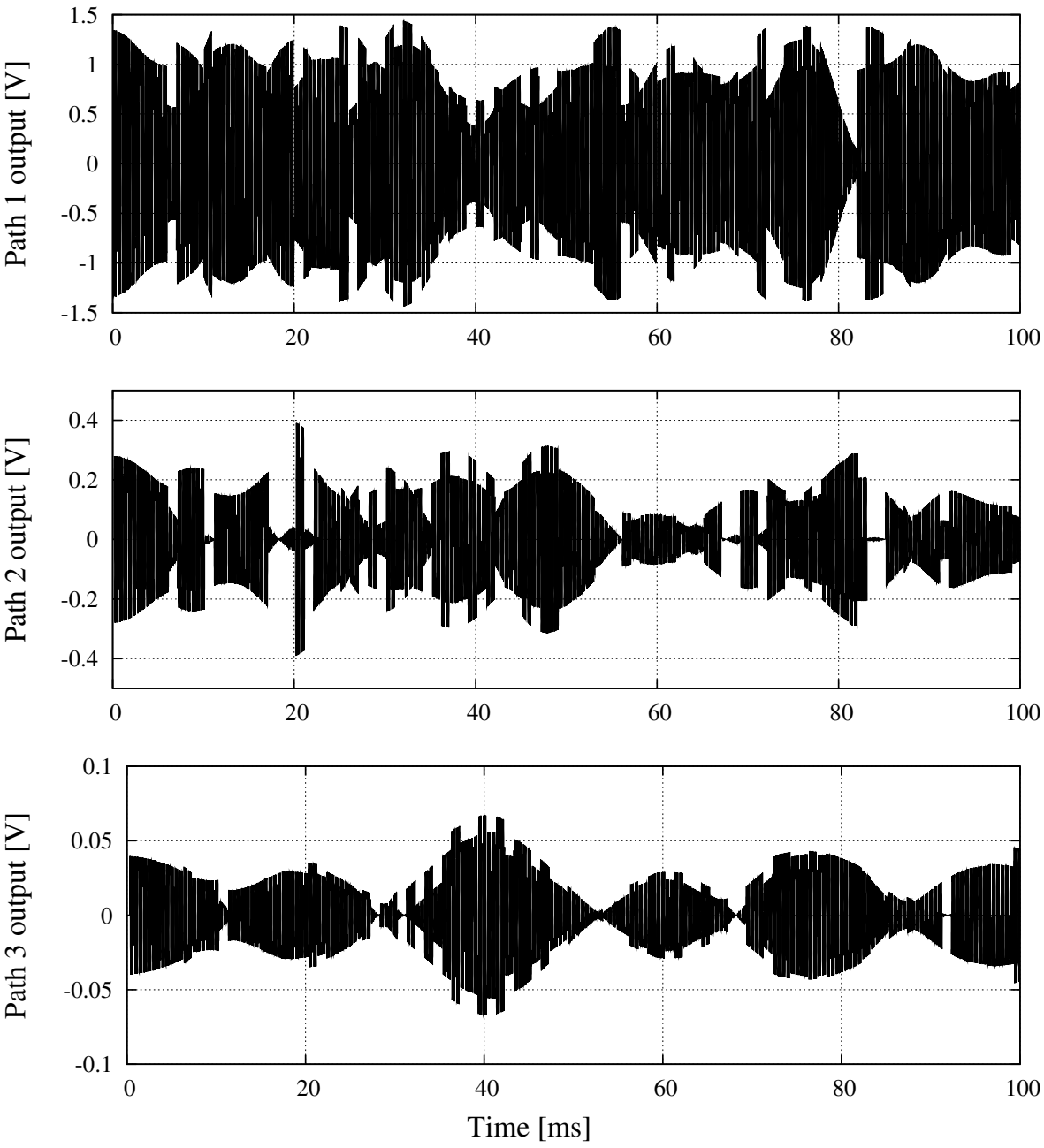

Figure 3.18: Measured temporal amplitude variations of the flat faded signals at the output of each of the paths of the multipath channel model for user 1. Paths 1, 2 and 3 have respectively Doppler spreads of 100, 67 and 33 Hz.

CHAPTER 4

SPARSE GRAPH CODING

4.1 CHAPTER SUMMARY

Sparse graph codes are channel codes represented as sparse probabilistic graphical models which are extensively used in Bayesian belief network theory. Linear codes can be represented as graphs with two sets of nodes; a set representing the transmitted bits and a set representing constraints placed on the transmitted bits. In a sparse graph each constraint involves only a small number of variables or bits in the graph, resulting in a graph with far fewer edges than the maximum number of possible edges. Graph codes can be decoded using Pearl's iterative belief propagation (IBP) or message passing (MP) algorithm [12] commonly used in artificial intelligence.

The state-of-the-art error-correcting codes are based on sparse graphs with error performance approaching the Shannon limit. This chapter considers three classes of codes developed for evaluation on the WCDMA platform, namely turbo codes, low-density parity-check (LDPC) codes and repeat-accumulate (RA) codes. Although turbo codes were originally designed using two parallel constituent convolutional codes [27], linear block codes (LBCs) are used as constituent codes in this study. The block turbo code is iteratively decoded using a SOVA (soft-output Viterbi algorithm) based decoder. The LDPC and RA codes are decoded using different versions of the MP algorithm. The iterative turbo decoding algorithm [27, 60], the VA and Pearl's IBP algorithm are all special cases of the MP algorithm [61, 62].

This chapter focuses on all the encoding and decoding algorithms, including the MD turbo encoder, iterative SOVA-based decoder and MP decoders adapted for the LDPC and RA codes. All coding systems are adapted to work on the 4D communication platform discussed in Chapter 2. For the turbo code, 3D encoder and decoder structures are used to implement the coding and modulation schemes together. For the LDPC and RA codes, simple S/P (serial-to-parallel) and P/S (parallel-to-serial) converters are used to adapt the coding schemes for implementation on the 4D modulation platform.

4.2 THE CODING OBJECTIVE

The fundamental problem of communication is to reproduce a transmitted message at the output of the receiver with highest probability of correct decision [1]. Coding theory is concerned with the creation of practical encoding and decoding systems [14] around a noisy channel to improve communication and to ultimately provide error free transmission through the channel. A simplified baseband digital communication system is shown in figure 4.1 to illustrate basic coding concepts.

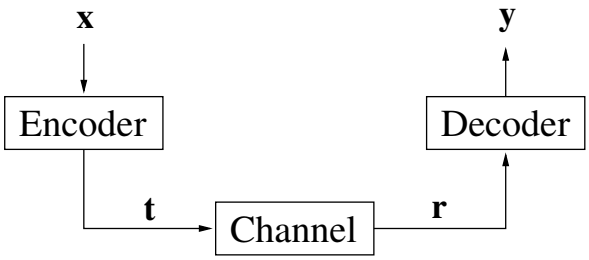

Figure 4.1: A simplified baseband digital communication system with channel coding

An information message word $\mathbf{x}$ emanating from a digital source is encoded into a codeword $\mathbf{t}$ and is transmitted across a channel. The channel corrupted received word $\mathbf{r}$ is then decoded into $\mathbf{y}$, an approximation to $\mathbf{x}$. The accuracy of approximation depends on channel conditions. An important statistical measure referring to figure 4.1 is the posterior probability of the transmitted codeword $\mathbf{t}$. Using Bayes' theorem [7, 14] the posterior probability of $\mathbf{t}$ is given by:

$$p(\mathbf{t}|\mathbf{r}) = \frac{p(\mathbf{r}|\mathbf{t})p(\mathbf{t})}{p(\mathbf{r})} \tag{4.1}$$

where $p(\mathbf{r}|\mathbf{t})$ is the likelihood function (or channel transition probability) of the codeword $\mathbf{t}$ and $p(\mathbf{t})$ is the prior probability (usually uniform over all valid codewords) of the codeword $\mathbf{t}$. The probability of the received codeword $p(\mathbf{r})$ is obtained by summing the likelihoods of all codewords [14]:

$$p(\mathbf{r}) = \sum_{\mathbf{t}} p(\mathbf{r}|\mathbf{t})p(\mathbf{t}), \tag{4.2}$$

which is a constant value if all codewords are equally probable, following from the fact that $p(\mathbf{t})$ is constant if the prior probabilities of the codewords are uniformly distributed.

4.2.1 CHANNEL DECODING

Though it is important to design or choose the correct channel code according to the channel and required error level, several different decoding algorithms can be used in combination with the same encoder. The choice of decoding algorithm is a trade-off design choice between decoding time, complexity and error performance. In most coding systems much more effort and processing power are needed at the decoder than at the encoder.

The two main classes of decoding algorithms are codeword decoding and bitwise decoding [14] algorithms. Codeword decoding is concerned with inferring which codeword $\mathbf{t}$ was transmitted, given the received corrupted word $\mathbf{r}$, while bitwise decoding is concerned with inferring the probability of each bit in the transmitted codeword $\mathbf{t}$. Bitwise decoding algorithms attempt to minimise the bit error rate, while codeword decoding algorithms attempt to find the most likely group or sequence of bits in stead of single bits transmitted [63].

Codeword decoding algorithms include the maximum likelihood (ML) decoder and the maximum a-posteriori (MAP) codeword decoder. The basic requirement of a codeword decoder is to choose as output the correct codeword $\mathbf{t}$ from the known set of codewords, after receiving the channel corrupted codeword $\mathbf{r}$.

4.2.1.1 ML CODEWORD DECODING

The ML decoder selects as output the coded sequence $\mathbf{t}$ that maximises the likelihood function of the codeword $\mathbf{t}$, given as a separable function on memoryless channels as [7, 14]:

$$p(\mathbf{r}|\mathbf{t}) = \prod_{n=1}^{N} p(r_n|t_n), \tag{4.3}$$

where N is the codeword length (number of bits) and r_n and t_n are the n^{th} received and transmitted bits of the codeword respectively.

4.2.1.2 MAP CODEWORD DECODING

Another way of decoding is to identify the most probable codeword $\mathbf{t}$, given the received codeword $\mathbf{r}$, which is achieved by selecting $\mathbf{t}$ that maximises the posterior probability $p(\mathbf{t}|\mathbf{r})$ given in equation 4.1. From equations 4.1, 4.2 and 4.3 it is clear that MAP codeword decoding is equivalent to ML codeword decoding if the prior probabilities of all

codewords are uniformly distributed (i.e. equiprobable).

Codeword decoding can be achieved algebraically by listing all codewords and their probabilities and finding the codeword that maximises either equation 4.1 or 4.3. Applying this algebraic decoder on a binary (N, K) linear code will require exponential decoding time of order 2^K. By exploiting the structure in the channel code, more efficient decoding algorithms can be applied to reduce decoding time and complexity. The min-sum algorithm (e.g. Viterbi) is one such algorithm that can be used to decode linear codes.

4.2.1.3 BITWISE DECODING

Iterative decoding techniques require implementations of bitwise decoding algorithms, where each decoding iteration attempts to improve the probability value (or confidence level) of each bit. The decoded sequence is therefore not necessarily a valid codeword, even though each decoded bit is the best estimate of the corresponding transmitted bit. The exact solution of bitwise decoding for a given received bit is obtained by marginalising probabilities over all other bits in the received codeword. This implies finding the maximum posterior probability of a received bit by not using the probability information of the same bit. Using equation 4.1, the posterior probability of the n^{th} received bit is [14]:

$$p(t_n|\mathbf{r}) = \sum_{\{t_{n'}:n' \neq n\}} p(\mathbf{t}|\mathbf{r}). \tag{4.4}$$

Bitwise decoding is mainly performed using a sum-product variant, including the MAP, BCJR [60], forward-backward and BP algorithms. Note that MAP can either be used to describe codeword or bitwise decoding algorithms, though the MAP-algorithm usually refers to a bitwise decoding algorithm in coding theory. This chapter henceforth only considers bitwise decoding algorithms, although certain bitwise algorithms are codeword decoding algorithms adapted to provide bit probability information as output.

4.2.2 LOG LIKELIHOOD RATIOS

The concept of log likelihood ratios (LLR) is important and widely used in iterative decoding techniques, especially in the decoding of turbo codes. Passing of information between component decoders is significantly simplified due to the simplification of probability calculations, e.g. multiplication becomes addition.

The LLR of an antipodal data bit $x_k; k = 1, \ldots, K$, denoted as $\Lambda(x_k)$ is defined as the natural logarithm of the probability ratio of the bit taking on its two possible values [12, 63]:

$$\Lambda(x_k) = \ln\left[\frac{p(x_k = +1)}{p(x_k = -1)}\right] \tag{4.5}$$

The sign of $\Lambda(x_k)$ indicates whether the data bit x_k is more likely a $+1$ or a -1, and the absolute value of $\Lambda(x_k)$ provides a confidence level of correctness of the value suggested by the sign.

The probabilities $p(x_k = +1)$ and $p(x_k = -1)$ can be calculated using equation 4.5 as follows [63]:

$$e^{\Lambda(x_k)} = \frac{p(x_k = +1)}{1 - p(x_k = +1)} \tag{4.6}$$

therefore

$$p(x_k = +1) = \frac{1}{1 + e^{-\Lambda(x_k)}} \tag{4.7}$$

and

$$p(x_k = -1) = \frac{1}{1 + e^{+\Lambda(x_k)}} \tag{4.8}$$

4.2.2.1 POSTERIOR LLR

Conditional probability ratios can also be expressed in terms of LLRs. The posterior LLR of the data bit x_k is given by [12, 63]:

$$\Lambda(x_k|\mathbf{r}) = \ln\left[\frac{p(x_k = +1|\mathbf{r})}{p(x_k = -1|\mathbf{r})}\right] \tag{4.9}$$

with $p(x_k = \pm 1|\mathbf{r})$ the posterior probability of the data bit x_k, which can be expressed in similar form as equations 4.7 and 4.8 by replacing $\Lambda(x_k)$ with $\Lambda(x_k|\mathbf{r})$.

4.2.2.2 CHANNEL LLR

Another important LLR used in iterative decoding is the channel LLR Λ_c, defined as [12, 63]:

$$\Lambda_c(r_n|t_n) = \ln\left[\frac{p(r_n|t_n = +1)}{p(r_n|t_n = -1)}\right] \tag{4.10}$$

with $p(r_n|t_n)$ the likelihood of the n^{th} code bit after transmission through an AWGN or fading channel, defined in equation 3.1 and rewritten as:

$$p(r_n|t_n) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left[-\frac{(r_n - \alpha_n t_n)^2}{2\sigma^2}\right] \tag{4.11}$$

with α_n the fading amplitude ($\alpha_n = 1$ for AWGN only). For BPSK transmission the transmitted bit takes on antipodal values, i.e. $t_n = \pm 1$. By substituting equation 4.11 into equation 4.10, the channel LLR is simplified to:

$$\Lambda_c(r_n|t_n) = \frac{4}{N_0}\alpha_n r_n, \tag{4.12}$$

with the single-sided noise PSD $N_0 = 2\sigma^2$ [W/Hz].

4.3 TWO DIMENSIONAL TURBO CODES

Turbo codes were first presented in [27] as iteratively decodable concatenated codes consisting of two parallel convolutional codes. Iteratively decodable codes require soft-output decoding algorithms to feed soft probability values from one constituent decoder to the input of another to improve decoding decisions and ultimately error performance. Turbo codes can be decoded either by a decoder consisting of separate soft-input soft-output (SISO) component decoders or by a factor graph decoding algorithm. Factor graph decoding of turbo codes is discussed in [14] and will not be considered in this study.

This section considers the general 2D turbo encoder and decoder structures, based on the structures given in [12, 27].

4.3.1 2D TURBO ENCODER

The input to the general 2D turbo encoder in figure 4.2 is a number of data bits or a data block represented as $\bar{x}$. Encoders 1 and 2 are typically identical systematic convolutional or block code encoder structures, though any code combination is possible, resulting in a hybrid turbo code. The output of encoders 1 and 2 are code or parity output bits only. Code $\bar{c}_1$ is obtained by encoding $\bar{x}$ directly, while $\bar{c}_2$ is obtained by encoding $\Pi_{1,2}\bar{x}$, an interleaved version of $\bar{x}$. The simplest interleaving strategy is to represent $\bar{x}$ as a 2D matrix and encoding the rows and columns of $\bar{x}$ using encoders 1 and 2 respectively. The output $[\bar{c}_0\ \bar{c}_1\ \bar{c}_2]$ is then obtained by P/S conversion. The overall code rate of the encoder is $R_c = \frac{1}{3}$, if both encoders 1 and 2 are rate-half encoders.

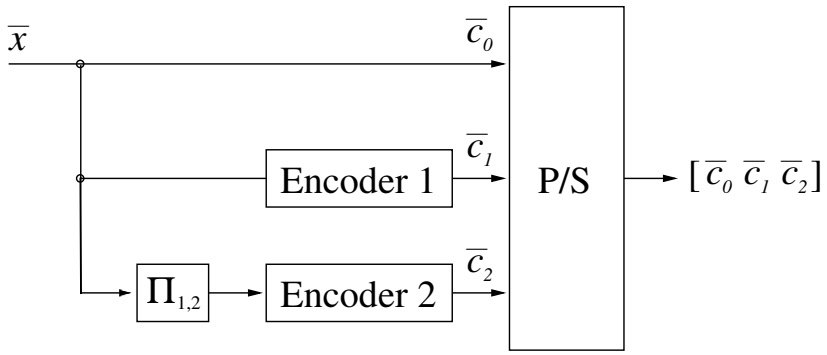

Figure 4.2: Two dimensional systematic turbo encoder structure

4.3.2 2D TURBO DECODER

The decoder shown in figure 4.3 is an iterative decoder, taking as input the received codeword block $[\tilde{c}_0\ \tilde{c}_1\ \tilde{c}_2]$.

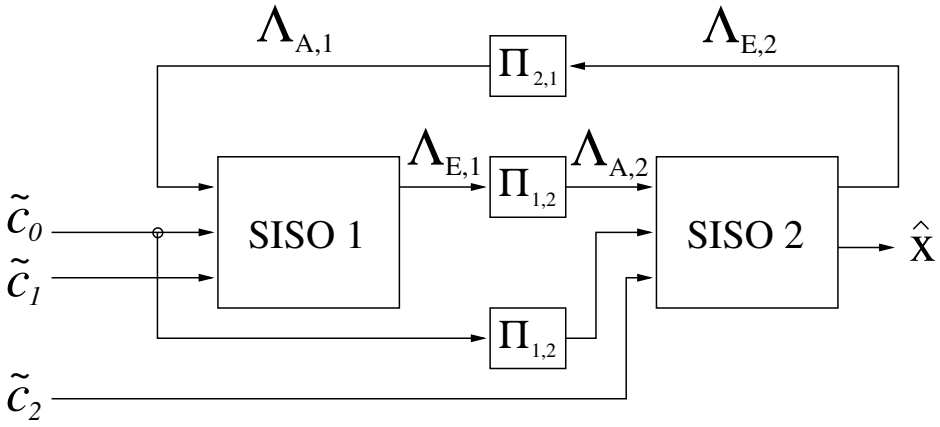

Figure 4.3: Two dimensional turbo decoder structure

Each SISO module takes as input a complete codeword and prior log likelihood ratio (LLR) information ($\mathbf{\Lambda_{A,1}}$ or $\mathbf{\Lambda_{A,2}}$) of the data bits. Each SISO module provides as output extrinsic LLR information ($\mathbf{\Lambda_{E,1}}$ or $\mathbf{\Lambda_{E,2}}$) of the data bits, which become the prior LLRs after interleaving. SISO 2 also provides the estimated soft output data $\hat{\mathbf{x}} = \Lambda(x_k|\mathbf{r})$, being the estimated transmitted data after hard limiting the output.

Although each SISO module processes a single codeword at a time, an entire codeword block must be processed before interleaving can be performed. The $\Pi_{1,2}$ interleaver shown in figure 4.3 rearranges the information processable by SISO 1 to information

processable by SISO 2, while $\Pi_{2,1}$ performs the inverse function of $\Pi_{1,2}$. Typically $\Pi_{1,2}$ is a row-to-column converter and $\Pi_{2,1}$ is a column-to-row converter.

4.3.2.1 INTRINSIC AND EXTRINSIC INFORMATION

The aim of the turbo decoder is to compute a reliability value (posterior probability) for each information bit, which should become more accurate as the number of decoding iterations increase. To improve the correctness of decoding decisions, each component decoder needs information which does not originate from itself. Therefore each component decoder has to provide information to the other component decoder which the other decoder does not have or cannot infer. The posterior probability of each information bit (see equation 4.9) contains intrinsic ($\mathbf{\Lambda}_I$) and extrinsic ($\mathbf{\Lambda_E}$) information, and can be expressed as:

$$\Lambda(x_k|\mathbf{r}) = \Lambda_I(x_k) + \Lambda_E(x_k). \tag{4.13}$$

The intrinsic information of each data bit x_k consists of the prior LLR of the information bit ($\Lambda_A(x_k)$) and the channel LLR defined in equation 4.10. Equation 4.13 can therefore be written as [12]:

$$\Lambda(x_k|\mathbf{r}) = \Lambda_A(x_k) + \Lambda_c(r_k|t_k) + \Lambda_E(x_k). \tag{4.14}$$

Note that only information bits have intrinsic and extrinsic LLRs, but the posterior LLR $\Lambda(x_k|\mathbf{r})$ depends on every received bit $r_n; n = 1, \ldots, N$. Equation 4.14 therefore relates the data bit x_k and the received bit r_k, though the entire received codeword $\mathbf{r}$ influences $\Lambda(x_k|\mathbf{r})$.

The extrinsic LLR can therefore be calculated from equation 4.14 as:

$$\Lambda_E(x_k) = \Lambda(x_k|\mathbf{r}) - \Lambda_A(x_k) - \Lambda_c(r_k|t_k). \tag{4.15}$$

Each SISO decoder uses a soft-output algorithm to calculate $\Lambda(x_k|\mathbf{r})$. The prior LLR $\Lambda_A(x_k)$ is obtained by interleaving $\Lambda_E(x_k)$ provided by the other SISO decoder and the channel LLR $\Lambda_c(r_k|t_k)$ is calculated using equation 4.12.

For the first decoding iteration, the extrinsic LLR from the decoder module SISO 2 (see figure 4.3), denoted as $\mathbf{\Lambda_{E,2}}$, is set to zero and therefore the prior probability $\mathbf{\Lambda_{A,1}}$ available to the decoder SISO 1 is also zero for all bits. The first decoder therefore assumes equally likely values for the input bits for the first iteration.

4.4 THREE DIMENSIONAL TURBO CODES

By placing more component codes in parallel, a higher dimension turbo code can be constructed. To match the 4D modulation platform, a 3D block turbo code (BTC) [28] with $(N, K) = (2K, K)$ constituent linear block codes is considered in this study, creating a $(4K^3, K^3)$ code.

4.4.1 3D TURBO ENCODER

The encoder structure of the 3D block turbo coding system is shown in figure 4.4.

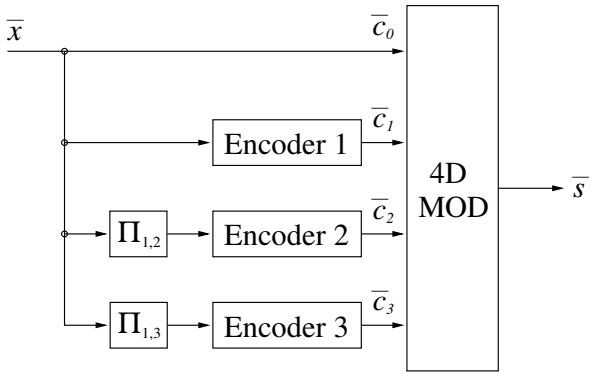

Figure 4.4: Three dimensional systematic turbo encoder structure with 4D modulator

The functioning of the 3D encoder shown in figure 4.4 is exactly the same as for the 2D encoder shown in figure 4.2. The only difference is the additional constituent encoder module denoted as encoder 3. The P/S module is replaced by the 4D modulator shown in figure 2.9 by connecting the 4 parallel lines $d_1(t)$ to $d_4(t)$ directly to the 4 output lines ($\bar{c}_0$, $\bar{c}_1$, $\bar{c}_2$ and $\bar{c}_3$) of the 3D encoder and merging the 4D modulator into the module denoted as *"4D MOD"* in figure 4.4. The output $\bar{s} = [\bar{c}_0\ \bar{c}_1\ \bar{c}_2\ \bar{c}_3]$ corresponds to the 4D modulator output $s(t)$.

The simplest interleaving strategy for the 3D encoder is to represent $\bar{x}$ as a 3D cube and encoding the rows, columns and depth of $\bar{x}$ using encoders 1, 2 and 3 respectively. The interleaver $\Pi_{1,2}$ rearranges the 3D cube from a row encoding to a column encoding configuration and $\Pi_{1,3}$ rearranges the 3D cube from a row encoding to a depth encoding configuration. The interleaving process is a rotation of the data cube (see figure 4.5) in 3D space. The code rate of the encoder shown in figure 4.4 is $R_c = \frac{1}{4}$, assuming all

constituent encoders are rate-half encoders.

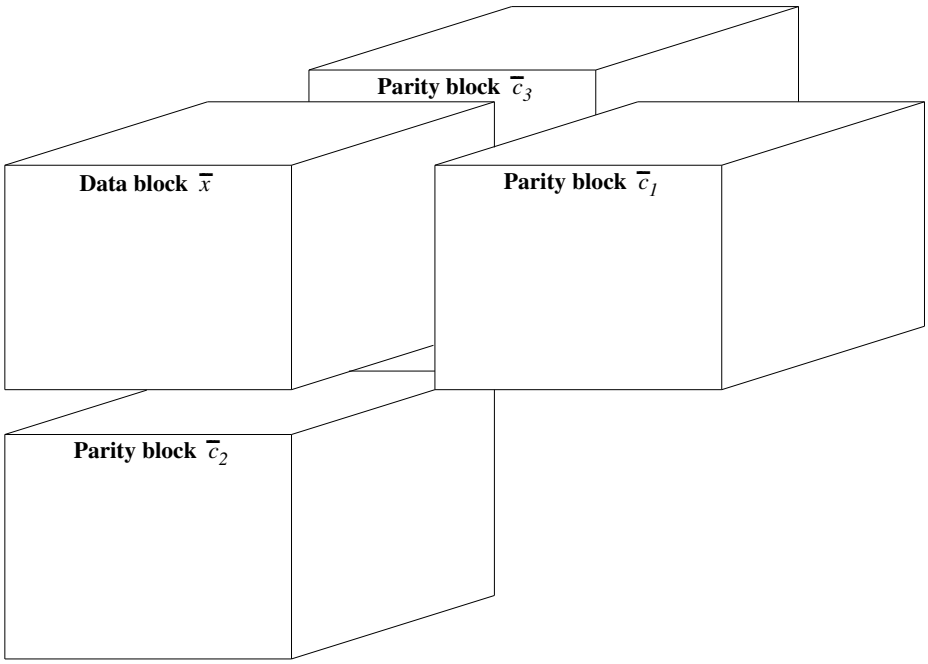

Figure 4.5: Cubes of the 3D block turbo code, where the parity blocks $\bar{c}_1$, $\bar{c}_2$ and $\bar{c}_3$ represent row, column and depth redundancy respectively

4.4.2 3D TURBO DECODER

The 3D turbo decoder shown in figure 4.6 is based on the structure presented in [28] and functions similar to the 2D turbo decoder shown in figure 4.3. The input to the 3D decoder is the 4D demodulator output $\hat{d} = [\tilde{c}_0\ \tilde{c}_1\ \tilde{c}_2\ \tilde{c}_3]$ (see figure 2.10).

Each SISO module takes as input a complete codeword and prior log likelihood ratio (LLR) information ($\mathbf{\Lambda_{A,1}}$, $\mathbf{\Lambda_{A,2}}$ and $\mathbf{\Lambda_{A,3}}$) of the data bits. Each SISO module provides as output extrinsic LLR information ($\mathbf{\Lambda_{E,1}}$, $\mathbf{\Lambda_{E,2}}$ and $\mathbf{\Lambda_{E,3}}$) to every other SISO module, which become the prior LLRs after summing and interleaving as shown in figure 4.6. The output of SISO module m is a 3D cube $\Lambda_{E,m}; m = 1, 2, 3$, containing the extrinsic log-likelihood ratio of each data bit x_k, computed using the equation (see equation 4.15):

$$\Lambda_{E,m}(x_k) = \Lambda(x_k|\bar{r}_m) - \Lambda_{A,m}(x_k) - \Lambda_c(\bar{r}_m|t_k), \tag{4.16}$$

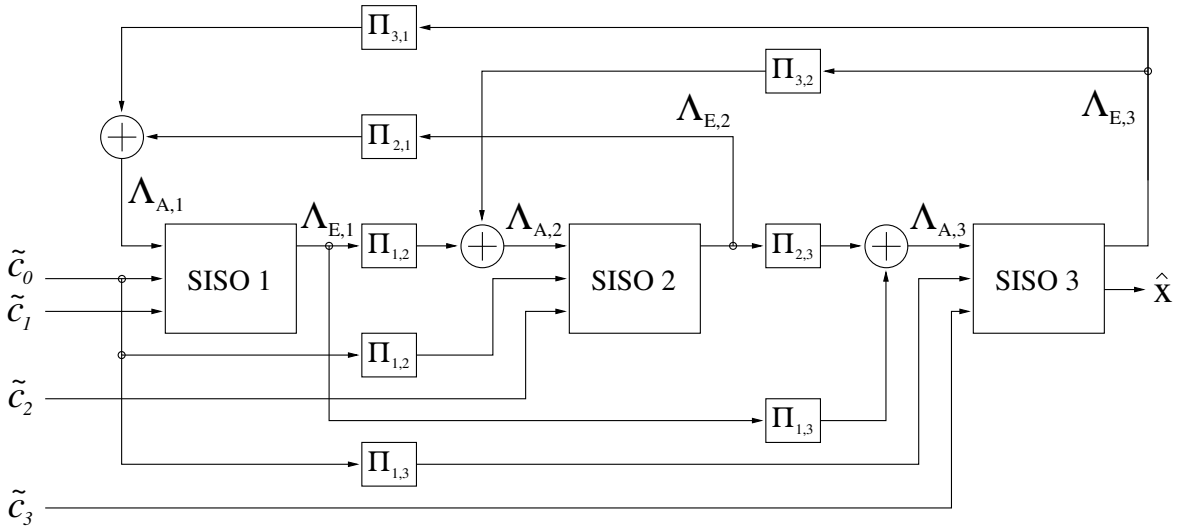

Figure 4.6: Three dimensional turbo decoder structure

with $\Lambda_{A,m}(x_k)$ the prior LLR of x_k and $\Lambda_c(\bar{r}_m|x_k)$ the channel LLR at the input of SISO module m. The received code block $\bar{r}_m$ consists of $\Pi_{1,m}\tilde{c}_0$ and $\tilde{c}_m$.

SISO module 3 provides the estimated soft output data cube $\hat{x}$, which is an approximation to the original transmitted data. Although each SISO module processes a single codeword at a time, an entire codeword cube must be processed before interleaving can be done. The interleavers shown in figure 4.6 are denoted as $\Pi_{i,j}$ with i and j the numbers corresponding to the input and output configurations of the interleaver respectively, e.g. interleaver $\Pi_{3,2}$ translates the configuration compatible with SISO 3 to the configuration compatible with SISO 2.

4.5 BLOCK CODE AS CONSTITUENT CODE OF TURBO CODE

The constituent codes used in turbo codes can either be convolutional or block codes, though in this study only linear block codes are considered. A soft-output bitwise decoding algorithm is required in a turbo coding scheme. Soft-output algorithms that can be applied on block codes include factor graph (e.g. the sum-product algorithm) and trellis decoding algorithms (e.g. the MAP and SOVA algorithms). In this study binary Reed-Muller codes are considered as constituent codes with a SOVA-based decoding algorithm. Binary Reed-Muller codes are chosen because of their relatively simple trellis

structures.

4.5.1 REED-MULLER CODES

4.5.1.1 REED-MULLER CODES AS BOOLEAN FUNCTIONS

Binary Reed-Muller (RM) codes can be described using binary polynomials or Boolean functions [12]. A Boolean function with m variables $f(x_1, x_2, \ldots, x_m)$ is a discrete function that has 2^m values corresponding to modulo-2 operations performed on all combinations of values of the function's variables $x_1, x_2, \ldots, x_m$. The Boolean function can be expressed in vector form as $\bar{f}(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m)$ where the argument vector $\bar{x}_i; i = 1, 2, \ldots, m$, is a vector of length 2^m such that all combinations of variable values can participate in the calculation of the Boolean vector $\bar{f}$.

To list all combinations of variable values of the Boolean function, a truth table such as table 4.1 can be used. The variable vectors $\bar{x}_1$, $\bar{x}_2$ and $\bar{x}_3$ are shown in the table with the Boolean vector $\bar{f}(\bar{x}_1, \bar{x}_2, \ldots, \bar{x}_m) = \bar{x}_1 + \bar{x}_2 + \bar{x}_3$, using modulo-2 addition.

Higher order Boolean functions are obtained by variable multiplication (dot product in modulo-2 math) resulting in 2^m elementary functions which consist of all possible combinations of variables multiplied together, including 1 (which represents degree 0), or in terms of vector expression, $\bar{1}$ which is the all-ones vector. For the $m = 3$ case, there are $2^m = 8$ elementary functions given as: $1, x_1, x_2, x_3, x_1x_2, x_1x_3, x_2x_3$ and $x_1x_2x_3$.

Table 4.1: Truth table of Boolean function with 3 variables

$\bar{x}_1$	0 0 0 0 1 1 1 1
$\bar{x}_2$	0 0 1 1 0 0 1 1
$\bar{x}_2$	0 1 0 1 0 1 0 1
$\bar{f}(x_1, x_2, x_3)$	0 1 1 0 1 0 0 1

The Boolean function can be expressed as the modulo-2 sum of the 2^m elementary functions in the general disjunctive normal form (DNF) as [12]:

$$\bar{f} = \bar{1} + a_1\bar{x}_1 + a_2\bar{x}_2 + \cdots + a_m\bar{x}_m + a_{12}\bar{x}_1\bar{x}_2 + \cdots + a_{12\cdots m}\bar{x}_1\bar{x}_2\cdots\bar{x}_m, \tag{4.17}$$

where the coefficients a_1 to $a_{12\cdots m}$ are used to control the degree of $\bar{f}$ by setting the coefficients to either 0 or 1.

4.5.1.2 DEFINITION OF REED-MULLER CODES

A binary RM $(2^m, K, 2^{m-r})$ code, denoted as $RM_{r,m}$ is the set of vectors associated with all Boolean functions with m variables and a maximum degree r. The dimension or bit length of the data input word of the $RM_{r,m}$ code is [12]:

$$K = \sum_{i=0}^{r} \binom{m}{i} \tag{4.18}$$

which is equal to the number of ways Boolean polynomials with m variables and a maximum degree r can be constructed.

4.5.1.3 REED-MULLER GENERATOR MATRIX

The generator matrix of an RM code is constructed by using the Boolean variable vectors as rows. The generator matrix $\mathbf{G}$ of the $RM_{1,3}$ or $RM(8,4,4)$ code is therefore given as [12]:

$$\mathbf{G} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \end{bmatrix} \tag{4.19}$$

by using the all-ones vector $\bar{1}$ and the vectors given in table 4.1. After performing row and column operations on $\mathbf{G}$, the systematic generator matrix is obtained and given as:

$$\mathbf{G_{sys}} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{4.20}$$

4.5.2 TRELLIS STRUCTURE OF BLOCK CODES

The construction of trellis structures for linear block codes as presented in [60] is considered in this section. The parity-check matrix of a linear (N, K) block code is given by:

$$\mathbf{H} = \left[\bar{h}_1\ \bar{h}_2 \cdots \bar{h}_N\right], \tag{4.21}$$

with $\bar{h}_i$ the i^{th} column vector of $\mathbf{H}$ with the index $i = 1, 2, \ldots, N$ also denoting the i^{th} trellis depth. Let $C_m = [c_{m,1},\ c_{m,2},\ \cdots,\ c_{m,N}]$ be the m^{th} valid codeword with $m = 1, 2, \ldots, 2^K$ and let the states in the trellis be denoted as $\bar{S}_{m,i}$ in column vector

format like $\bar{h}_i$. By including $S_{m,0}$, the trellis can be constructed using the following recursion formula [45, 60]:

$$\bar{S}_{m,0} = \bar{0} \tag{4.22}$$

$$\bar{S}_{m,i} = \bar{S}_{m,i-1} + c_{m,i}\bar{h}_i. \tag{4.23}$$

4.5.2.1 TRELLIS CONSTRUCTION PROCEDURE

Note that the trellis construction formula given in equation 4.23 requires knowledge of all valid codewords, though all information needed to construct the trellis is inherently available in **H**. A trellis construction procedure presented in [45] starts at $S_{m,0} = 0$ and considers both possible binary values for each code bit $c_{m,i} \in [0, 1]$ resulting in multiple paths, valid and invalid, in the trellis. The invalid paths are then removed by an expurgation procedure, i.e. removing all paths from the trellis that don't end in the zero state in the trellis at $i = N$. All valid codewords run from the zero state at depth $i = 0$ to the zero state at depth $i = N$ [13].

Using the generator matrix of the $RM(8, 4, 4)$ code given in equation 4.20, the parity check matrix can easily be shown to be:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.24}$$

from which the unexpurgated trellis structure from trellis depth $i = 0$ to $i = N$, shown in figure 4.7, is constructed. The maximum number of trellis states is 2^{N-K}, numbered as $S_{m,i} = 0, 1, \ldots, 2^{N-K} - 1$ which are the decimal values of the binary vector $\bar{S}_{m,i}$. The horizontal branches in the trellis correspond to the code bit 0 and diagonal branches to the code bit 1. Note that equations 4.22 and 4.23 can be rewritten to describe the trellis in reverse order as:

$$\bar{S}_{m,N} = \bar{0} \tag{4.25}$$

$$\bar{S}_{m,i} = \bar{S}_{m,i+1} + c_{m,i+1}\bar{h}_{i+1}. \tag{4.26}$$

Equation 4.26 follows from the fact that $c_{m,i}\bar{h}_i + c_{m,i}\bar{h}_i = \bar{0}$ in equation 4.23. The unexpurgated reverse trellis structure shown in figure 4.8 is constructed by moving from

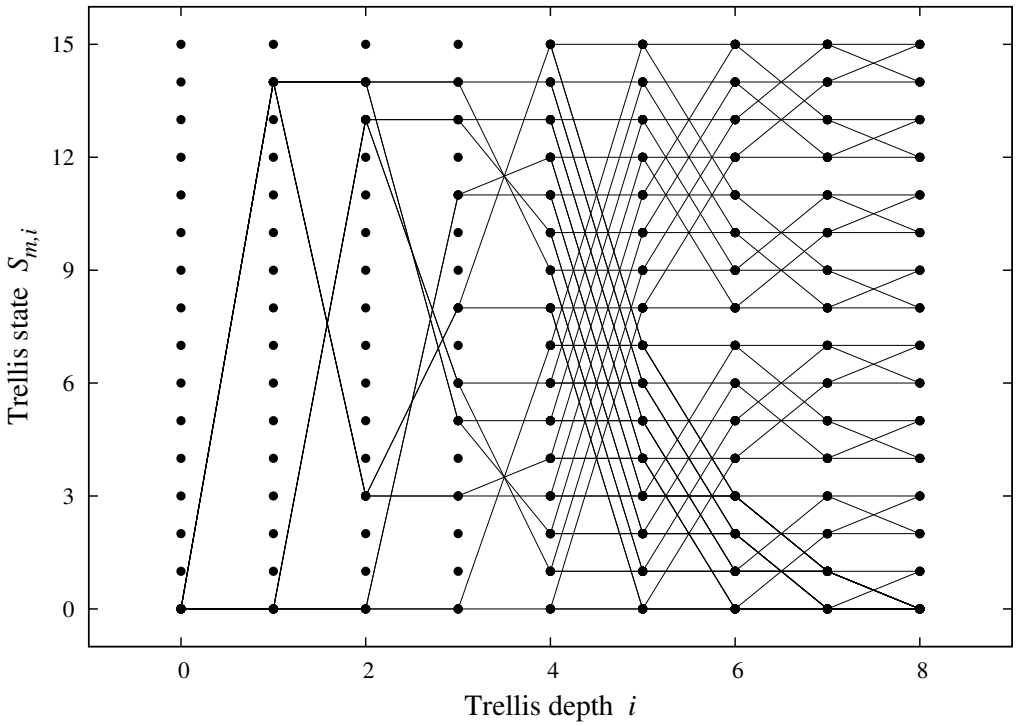

Figure 4.7: Unexpurgated trellis structure of the $RM(8,4,4)$ code in the forward direction

trellis depth $i = N$ to $i = 0$ using equations 4.25 and 4.26.

By combining the unexpurgated forward (figure 4.7) and reverse (figure 4.8) trellises, the expurgated trellis (figure 4.9) is obtained. The expurgated trellis structure is the union of branches of the forward and reverse trellises.

4.5.3 SOFT-OUTPUT TRELLIS DECODING

The aim of a soft-output decoding algorithm is to calculate the reliability or posterior LLR $\Lambda(x_k|\mathbf{r})$ of the transmitted information bits in order to provide the output as soft-input to a following decoder. In terms of trellis decoding, the posterior LLR (equation 4.9) can be expressed as [36, 63]:

$$\begin{aligned}\Lambda(x_k|\mathbf{r}) &= \ln\left[\frac{p(x_k=+1|\mathbf{r})}{p(x_k=-1|\mathbf{r})}\right] \\ &= \ln\left(\sum_{l=1}^{L_t} \mathrm{e}^{-PM_{l,k}^{x_k=+1}}\right) - \ln\left(\sum_{l=1}^{L_t} \mathrm{e}^{-PM_{l,k}^{x_k=-1}}\right) \end{aligned} \quad (4.27)$$

where L_t is the number of selected paths considered (depending on which decoding algo-

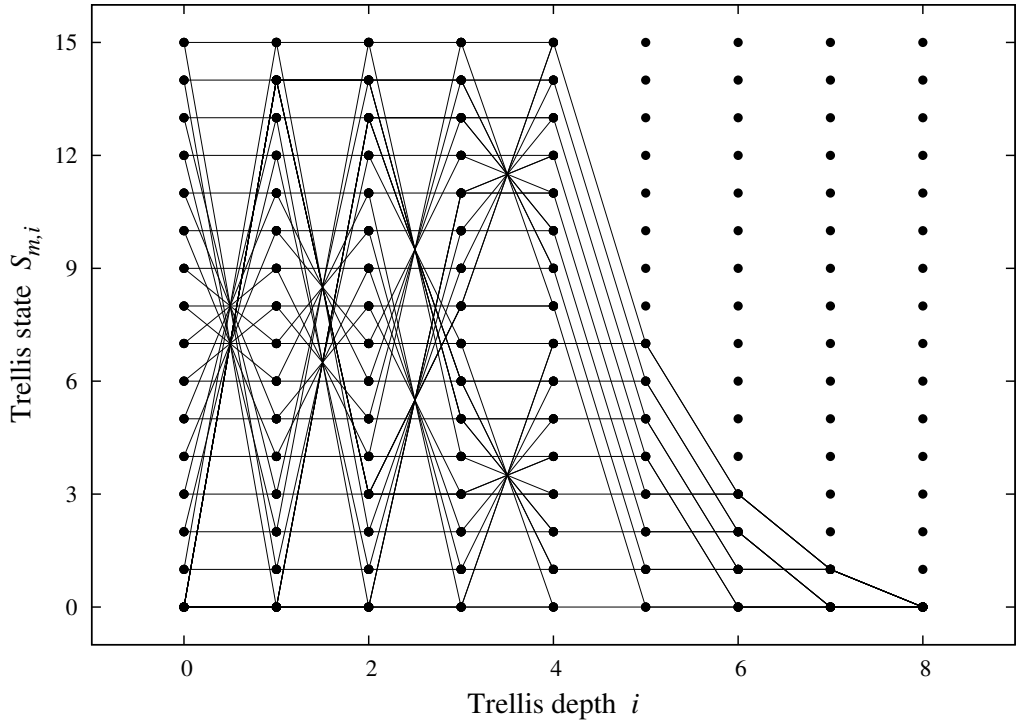

Figure 4.8: Unexpurgated trellis structure of the $RM(8,4,4)$ code in the reverse direction

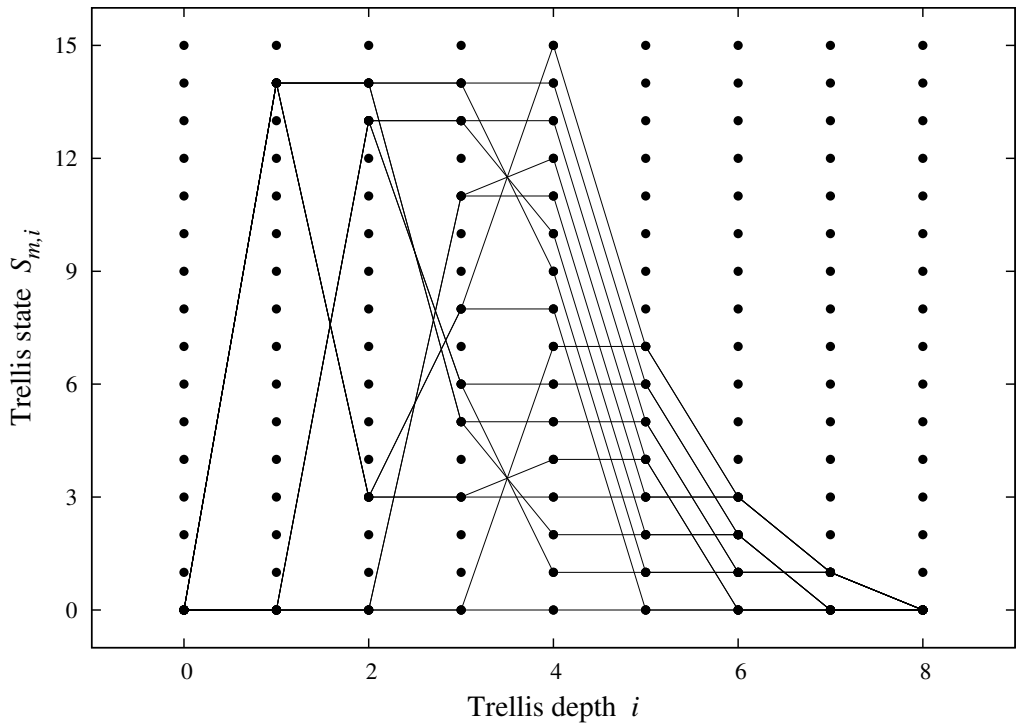

Figure 4.9: Expurgated trellis structure of the $RM(8,4,4)$ code

rithm is used), $PM_{l,k}^{x=+1}$ and $PM_{l,k}^{x=-1}$ are the l^{th} path metrics corresponding to $x_k = +1$ and $x_k = -1$ respectively. Equation 4.27 follows from the fact that $p(x_k = +1|\mathbf{r})$ is the sum of all probabilities of occurrences of $x_k = +1$ which correspond to the sum of probabilities of paths in the trellis with $x_k = +1$ (the same argument holds for $p(x_k = -1|\mathbf{r})$).

To calculate the probabilities $p(x_k = +1|\mathbf{r})$ and $p(x_k = -1|\mathbf{r})$ exactly, all possible paths in the trellis respectively corresponding to $x_k = +1$ and $x_k = -1$ must be considered and therefore $PM_{l,k}^{x=\pm 1}$ corresponds to the path metrics of all possible paths (not only ML paths) in the trellis.

4.5.3.1 DECODING SIMPLIFICATIONS

The most common soft-output trellis decoding algorithms used in turbo decoding are the MAP algorithm, the SOVA and simplifications of these two algorithms [63]. The MAP or BCJR [60] algorithm calculates the exact posterior LLR $\Lambda(x_k|\mathbf{r})$ by considering all possible paths in the trellis as discussed in section 4.5.2.1. In this study, a SOVA variation is used to decode RM codes in a turbo coding structure.

Equation 4.27 can be simplified using the approximation (also used in the suboptimal max-log-MAP algorithm) [12, 63]:

$$\ln\left(\sum_i \mathrm{e}^{x_i}\right) \approx \max_i(x_i). \tag{4.28}$$

Using equations 4.27 and 4.28, the posterior LLR can then be written as:

$$\begin{aligned}
\Lambda(x_k|\mathbf{r}) &\approx \left(\max_l\left\{-PM_{l,k}^{x=+1}\right\}\right) - \left(\max_l\left\{-PM_{l,k}^{x=-1}\right\}\right) && (4.29)\\
&= \left(-\min_l\left\{PM_{l,k}^{x=+1}\right\}\right) - \left(-\min_l\left\{PM_{l,k}^{x=-1}\right\}\right) && (4.30)\\
&= \min_l\left\{PM_{l,k}^{x=-1}\right\} - \min_l\left\{PM_{l,k}^{x=+1}\right\} && (4.31)\\
&= PM_{ML,k}^{x=-1} - PM_{ML,k}^{x=+1}, && (4.32)
\end{aligned}$$

where $PM_{ML,k}^{x=\pm 1}$ is the path metric of the best candidate (ML) path with $x_k = \pm 1$. Each trellis depth may have more than one branch corresponding to $x_k = \pm 1$, though the best candidate path is the path with minimum path metric of all paths with $x_k = \pm 1$.

Therefore, in stead of considering all the possible paths in the trellis to calculate the posterior LLR $\Lambda(x_k|\mathbf{r})$, only the paths with minimum path metrics corresponding to $x_k = +1$ and $x_k = -1$ are considered in each trellis depth. The posterior LLR $\Lambda(x_k|\mathbf{r})$ is

calculated using a forward-backward SOVA-based algorithm explained in the following section.

4.5.3.2 FORWARD-BACKWARD VITERBI ALGORITHM

The SOVA [64] is a variation of the classical Viterbi decoding algorithm used as constituent decoder in turbo coding. The modifications to the Viterbi algorithm include alternative path metric calculations to include prior information when selecting the ML path in the trellis and additional calculations to provide the posterior LLR $\Lambda(x_k|\mathbf{r})$ defined in equation 4.9 as output [63].

The soft-output decoding algorithm used in this study is essentially the SOVA-based algorithm or bidirectional soft output decoder presented in [12, 36]. The algorithm calculates an estimate of the posterior LLR $\Lambda(x_k|\mathbf{r})$ defined in equation 4.9 by performing a forward-backward Viterbi algorithm on an expurgated trellis structure such as the $RM(8,4,4)$ trellis shown in figure 4.9.

The algorithm basically computes and stores the path metrics in each trellis depth by running through the trellis independently in both forward and reverse directions. The posterior LLR $\Lambda(x_k|\mathbf{r})$ is then calculated using the forward and reverse calculated path metrics and the branch metrics corresponding to both $x_k = +1$ and $x_k = -1$ in each trellis depth.

The detailed forward-backward soft-output decoding algorithm used in this study is explained as follows.

- By running in the forward direction through the trellis according to equation 4.23 starting at state $\bar{S}_{m,0}$ (see equation 4.22), the forward path metrics are calculated as:

$$PM^f_{s,i} = \min_s \left(PM^f_{s',i-1} + BM_{s,s'} \right), \tag{4.33}$$

  where (s,i) indicates the trellis state $\bar{S}_{m,i}$ (as in equations 4.22 and 4.23) in the i^{th} trellis depth and $(s',i-1)$ indicates the trellis state in the previous trellis depth. $BM_{s,s'}$ is the branch metric of the branch connecting states s and s'. The path metric $PM^f_{s,i}$ is therefore the survivor path metric at state s up to trellis depth i in the forward direction. For each trellis coordinate (s,i) the survivor path metric should be stored.

- By running in the reverse direction through the trellis according to equation 4.26 starting at state $\bar{S}_{m,N}$ (see equation 4.25), the reverse path metrics are calculated as:

$$PM^r_{s,i} = \min_s \left(PM^r_{s',i+1} + BM_{s,s'} \right), \tag{4.34}$$

  where (s, i) indicates the trellis state $\bar{S}_{m,i}$ (as in equations 4.25 and 4.26) in the i^{th} trellis depth and $(s', i+1)$ indicates the trellis state in the next trellis depth. The path metric $PM^r_{s,i}$ is therefore the survivor path metric at state s up to trellis depth i in the reverse direction.

- For each trellis depth the best candidate path metric for $x_k = +1$ is calculated using the equation:

$$PM^{x=+1}_{ML,k} = \min_{x_k=+1} \left(PM^f_{s,i} + PM^r_{s',i+1} + BM^{x_k=+1}_{s,s'} \right), \tag{4.35}$$

  where $BM^{x_k=+1}_{s,s'}$ is the metric of the branch corresponding to $x_k = +1$. Note that the argument of the *min*-operator in equation 4.35 is the survivor path metric of a complete path in the trellis given that the branch with metric $BM^{x_k=+1}_{s,s'}$ is included in the path.

- For each trellis depth the best candidate path metric for $x_k = -1$ is calculated using the equation:

$$PM^{x=-1}_{ML,k} = \min_{x_k=-1} \left(PM^f_{s,i} + PM^r_{s',i+1} + BM^{x_k=-1}_{s,s'} \right), \tag{4.36}$$

  where $BM^{x_k=-1}_{s,s'}$ is the metric of the branch corresponding to $x_k = -1$.

- The posterior LLR $\Lambda(x_k|\mathbf{r})$ is finally calculated for each code bit using equations 4.32, 4.35 and 4.36.

The forward-backward decoding algorithm is used in each SISO decoding module (in figures 4.3 and 4.6) to calculate the posterior LLR $\Lambda(x_k|\mathbf{r})$ with $k = 1, \ldots, N$ for each code bit in the received codeword. The extrinsic LLR $\Lambda_E(x_k)$ with $k = 1, \ldots, K$ is calculated for each information bit and passed to other SISO modules in the decoder using the posterior LLR $\Lambda(x_k|\mathbf{r})$ and equation 4.15 or 4.16.

4.5.3.3 BRANCH METRIC CALCULATIONS

Each SISO module in a turbo decoder provides as output the extrinsic LLR $\Lambda_E(x_k)$ which is interleaved to provide the prior LLR $\Lambda_A(x_k)$ as input to a following SISO

module. The prior LLR is utilised in each SISO decoder to improve decoding decisions by incorporating $\Lambda_E(x_k)$ in the branch metric equation.

The branch metric is a branch transition probability, which can be expressed as a joint probability function [3]:

$$\mathrm{e}^{-BM_{s,s'}^{x_k}} = p(r_n, s|s'), \tag{4.37}$$

which is the probability that the n^{th} received code bit r_n causes a trellis transition to state s, given that the trellis was in state s'. Note that the branch metric $BM_{s,s'}^{x_k}$ in equation 4.37 is a LLR and is therefore written as an exponent. Using Bayes' rule, equation 4.37 can further be developed as:

$$\begin{aligned} \mathrm{e}^{-BM_{s,s'}^{x_k}} &= p(r_n|s, s')p(s|s') &(4.38)\\ &= p(r_n|t_n)p(x_k), &(4.39)\end{aligned}$$

with $p(r_n|t_n)$ the channel-dependent probability defined in equation 4.11. The prior probability $p(x_k)$ of the k^{th} data bit defined in equations 4.7 and 4.8 can be written in a single equation as [63]:

$$p(x_k) = \left(\frac{\mathrm{e}^{-\Lambda(x_k)/2}}{1+\mathrm{e}^{-\Lambda(x_k)}}\right)\mathrm{e}^{x_k\Lambda(x_k)/2} \tag{4.40}$$

with $\Lambda(x_k)$ the prior LLR of the data bit x_k, defined in equation 4.5. By substituting equations 4.11 and 4.40 in equation 4.39 and taking the logarithm on both sides, the branch metric can be written as [36]:

$$BM_{s,s'}^{x_k} = \kappa + (r_n - \alpha_n t_n)^2 - \frac{1}{2}x_k\Lambda(x_k), \tag{4.41}$$

with κ a constant term absorbing all terms that are constant for all branches in each trellis depth. The constant term therefore has no influence on the probability values and can be neglected. The second term is the branch metric as in the classical Viterbi algorithm, where the ML decoded sequence is the path in the trellis that differs the least from the received coded sequence in terms of the Euclidean distance. The Euclidean distance metric in equation 4.41 is written in terms of bits, though in a practical communication system the branch metric is calculated in terms of samples. Equation 4.41 can therefore be written as:

$$BM_{s,s'}^{x_k} = \sum_{l=1}^{N_s}(r_{n,l} - a_{n,l}t_{n,l})^2 - \frac{1}{2}x_k\Lambda(x_k), \tag{4.42}$$

where $r_{n,l}$ with $l = 1, \ldots, N_s$ is the n^{th} received code bit sequence with N_s samples per bit. The sequence corresponding to the sampled transmitted bit $t_{n,l}$ is used as branch weight vector in the decoder with instantaneous fading amplitude coefficient $a_{n,l}$ accompanying each received sample.

4.5.3.4 BRANCH METRICS IN 3D TURBO DECODER

An alternative, though closely related method as discussed in section 4.5.3.3 to calculate the branch metrics in the 3D turbo decoding structure, shown in figure 4.6, used in this study is considered in this section. Correlation type demodulation is performed on the received 4D signal (see figure 2.10) to extract the soft value representing the $k^{\text{th}}; k = 1, \ldots, K^3$, bit of the $n^{\text{th}}; n = 0, 1, 2, 3$, code cube shown in figure 4.5 and can be summarised by the equation:

$$\tilde{c}_{n,k} = \sum_{l=1}^{L.N_s} \alpha_{k,l} \cdot r_{k,l} \cdot \Psi_{n,l} \tag{4.43}$$

with $\alpha_{k,l}$ and $r_{k,l}$ respectively the l^{th} instantaneous fading amplitude and received samples of the composite 4D signal containing the k^{th} group of parallel code bits. $\Psi_{n,l}$ is the l^{th} sample of the orthonormal CSS base vector (see section 2.5.1), distinguishing the n^{th} dimension of the 4D signal. L is the CSS length and N_s the number of samples per bit.

The branch metric used in the 3D decoding structure at SISO module m can be described similar to equation 4.42 using equation 4.43 as:

$$BM_{s,s'}^{m,x_k} = (\tilde{c}_{n,k} - x_k)^2 - \frac{1}{2} x_k \Lambda_{A,m}(x_k). \tag{4.44}$$

4.6 LDPC CODES

Low-density parity-check (LDPC) codes were originally invented by Gallager [65] and rediscovered by MacKay and Neal [66, 67] and others [68] and have become a major research topic [19]. LDPC codes are powerful error correcting codes outperforming turbo codes if the code length is large and the parity check matrix is designed correctly [20, 21].

A regular LDPC code is a linear (N, K) block code with a sparse parity-check matrix $\mathbf{H}$ having a constant column weight w_c and constant row weight w_r, both much smaller than the code length N. LDPC codes are typically constructed by random generation of the parity-check matrix under row and column weight and cycle/girth [67] constraints. The sparse nature of $\mathbf{H}$ allows the implementation of efficient encoding and decoding algorithms making large code sizes ($N \simeq$ 1E5) practical. This section considers the encoding and decoding algorithms and the simulation platform to implement LDPC codes in the MD communication system presented in Chapter 2.

4.6.1 ENCODING LDPC CODES

4.6.1.1 ENCODING LINEAR BLOCK CODES

The easiest way to encode a (N, K) linear block code is with systematic encoding using the equation [12]:

$$\bar{\mathrm{c}} = \bar{\mathrm{x}} \cdot \mathbf{G_{sys}}, \tag{4.45}$$

where $\bar{\mathrm{c}}$ is the $1 \times N$ codeword, $\bar{\mathrm{x}}$ is the $1 \times K$ message word and $\mathbf{G_{sys}}$ is the $K \times N$ generator matrix in systematic form which can be expressed as:

$$\mathbf{G_{sys}} = [\ \mathbf{I_K} \mid \mathbf{P}\ ], \tag{4.46}$$

where $\mathbf{I_K}$ is the $K \times K$ identity matrix and $\mathbf{P}$ is a $K \times (N-K)$ parity sub-matrix. The generator matrix can either be created directly or obtained from the parity-check matrix (using the fact that $\mathbf{G} \cdot \mathbf{H}^T = \bar{\mathbf{0}}$) by performing elementary row and column operations on $\mathbf{H}$ to obtain the systematic form:

$$\mathbf{H_{sys}} = \left[\ \mathbf{P}^T \mid \mathbf{I_{N-K}}\ \right]. \tag{4.47}$$

An alternative encoding procedure which is computationally more efficient for high-rate codes ($K/N > 1/2$) using $\mathbf{H_{sys}}$ and the equation:

$$\mathbf{H} \cdot \bar{\mathrm{c}}^T = \bar{\mathbf{0}}, \tag{4.48}$$

is to calculate the values of the parity positions in the codeword using the equation [12]:

$$\mathrm{c}_i = \bar{\mathrm{x}} \cdot \mathbf{P}_{i-K}, \tag{4.49}$$

where $\mathrm{c}_i; i = K+1, \ldots, N$ is the i^{th} code bit (c_1 to c_K is the systematic data bits) and $\mathbf{P}_{i-K}$ is the $(i-K)^{\text{th}}$ column of $\mathbf{P}$.

Note that the parity-check matrix of an LDPC code is not necessarily in systematic form and by performing row and column operations on the randomly generated matrix, which is generated using special algorithms to ensure that row and column constraints are fulfilled, the special characteristics of $\mathbf{H}$ will be changed and may even result in a less sparse parity-check matrix $\mathbf{H}$. Alternative LDPC encoding algorithms have been developed and those used in this study are discussed next.

4.6.1.2 SPARSE MATRIX ENCODING OF LDPC CODES

In this section an encoding method for binary LDPC codes without using $\mathbf{H_{sys}}$, developed by Neal [38] is considered. By performing elementary matrix operations, a parity-check matrix can be written as (without substantially affecting the row or column weights):

$$\mathbf{H} = [\ \mathbf{A}\ \mathbf{B}\ ], \tag{4.50}$$

with a non-singular sub-matrix $\mathbf{A}$ $(M \times M)$ and sub-matrix $\mathbf{B}$ $(M \times K)$. The codeword can be written as $\bar{\mathrm{c}} = [\ \bar{\mathrm{p}}\ \bar{\mathrm{x}}\ ]$ with $\bar{\mathrm{p}}$ $(1 \times M)$ the parity vector and $\bar{\mathrm{x}}$ $(1 \times K)$ the systematic information vector. From equations 4.48 and 4.50 the following equation is obtained:

$$\mathbf{A}\bar{\mathrm{p}}^T + \mathbf{B}\bar{\mathrm{x}}^T = \mathbf{0}, \tag{4.51}$$

from which the parity vector can be calculated as:

$$\bar{\mathrm{p}}^T = \mathbf{A^{-1}B}\bar{\mathrm{x}}^T. \tag{4.52}$$

Systematic encoding can therefore be performed by pre-computing $\mathbf{A^{-1}B}$ and using equation 4.52 to calculate the parity bits in the codeword. To calculate the M parity bits requires $M \times K$ multiplications and $M \times (K-1)$ additions (multiplication is the primary coding complexity factor). If $\mathbf{H}$ is sparse, the encoding complexity can be reduced by calculating $\mathbf{A^{-1}B}\bar{\mathrm{x}}^T$ in two steps. In the first step

$$\mathbf{z} = \mathbf{B}\bar{\mathrm{x}}^T, \tag{4.53}$$

is calculated and by exploiting the sparseness of $\mathbf{B}$, the complexity can be reduced to $M \times w_r$ (assuming an LDPC code with average row weight $w_r < K$). Finally the parity vector is computed as:

$$\bar{\mathrm{p}}^T = \mathbf{A^{-1}z}, \tag{4.54}$$

which has a complexity of M^2 due to the fact that $\mathbf{A}^{-1}$ is not necessarily sparse. The resultant encoding complexity is therefore proportional to M^2 and more efficient than the

encoding method using equation 4.52 directly if $M < K$ (when the coderate $R_c > 1/2$).

The encoding complexity can be reduced further by also exploiting the sparseness of $\mathbf{A}$, by not using the inverse of $\mathbf{A}$ but an $\mathbf{LU}$-decomposition [69] ($\mathbf{L}$ is a lower-triangular and $\mathbf{U}$ is an upper-triangular matrix) of $\mathbf{A}$. Equation 4.54 can be written as:

$$\mathbf{A}\bar{\mathrm{p}}^T = \mathbf{z}, \tag{4.55}$$

which can then be written as:

$$\mathbf{LU}\bar{\mathrm{p}}^T = \mathbf{z}. \tag{4.56}$$

Finding the parity vector $\bar{\mathrm{p}}^T$ is done in two steps; in the first step the equation

$$\mathbf{LY} = \mathbf{z}, \tag{4.57}$$

is solved for $\mathbf{Y}$ $(M \times 1)$ using forward-substitution and secondly the equation

$$\mathbf{U}\bar{\mathrm{p}}^T = \mathbf{Y}, \tag{4.58}$$

is solved for $\bar{\mathrm{p}}^T$ using backward-substitution. Forward-substitution is accomplished by applying the equation:

$$Y_i = L_{i,i}\left[z_i + \sum_{j=1}^{M-1} L_{i,j}Y_i\right] \tag{4.59}$$

in modulo-2 math with Y_i the i^{th} column entry of $\mathbf{Y}$, $L_{i,j}$ the i^{th} row and j^{th} column entry of $\mathbf{L}$ and z_i the i^{th} column entry of $\mathbf{z}$. Similarly, backward-substitution is accomplished by applying the equation:

$$p_i = U_{i,i}\left[Y_i + \sum_{j=i+1}^{M} U_{i,j}p_i\right] \tag{4.60}$$

in modulo-2 math with p_i the i^{th} row entry of $\bar{\mathrm{p}}$ and $U_{i,j}$ the i^{th} row and j^{th} column entry of $\mathbf{U}$. Both forward- and backward-substitution algorithms have a complexity cost proportional to M^2, but if $\mathbf{L}$ and $\mathbf{U}$ are sparse matrices, the complexity is $k_{LU}M^2$ with k_{LU} proportional to w_r, which is typically a small fraction of N.

All three encoding methods (corresponding to equations 4.52, 4.54 and 4.58) are equivalent, though encoding complexity differs. A more efficient (or less complex) encoding algorithm typically requires more pre-processing effort and the encoding method used is chosen according to code size, code rate and allowed complexity.

4.6.1.3 APPROXIMATE TRIANGULAR ENCODING OF LDPC CODES

An efficient encoding algorithm for LDPC codes, presented by Richardson and Urbanke in [70], performs pre-processing on an LDPC matrix to change the matrix to an approximate lower triangular matrix allowing fast encoding, by exploiting the sparse nature of $\mathbf{H}$. The encoding complexity is minimised such that LDPC codes with length $N \simeq 1E5$ are still practical. In this section the encoding algorithm is explained.

A sparse matrix of an (N, K) LDPC code is altered by elementary matrix operations to obtain the approximate lower triangular form:

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{T} \\ \mathbf{C} & \mathbf{D} & \mathbf{E} \end{bmatrix} \tag{4.61}$$

with six sub-matrices $\mathbf{A}$ $((M-g)\times(N-M))$, $\mathbf{B}$ $((M-g)\times G)$, $\mathbf{T}$ $((M-G)\times(M-G))$, $\mathbf{C}$ $(G \times (N-M))$, $\mathbf{D}$ $(G \times G)$ and $\mathbf{E}$ $(G \times (M-G))$ where $M = N - K$ and g is the gap (the middle column of $\mathbf{H}$) dimension. $\mathbf{T}$ is a lower triangular matrix with ones along the diagonal. $\mathbf{H}$ is further processed to:

$$\mathbf{H} = \begin{bmatrix} \mathbf{A} & \mathbf{B} & \mathbf{T} \\ -\mathbf{E}\mathbf{T}^{-1}\mathbf{A} + \mathbf{C} & \phi & \mathbf{0} \end{bmatrix} \tag{4.62}$$

with non-singular sub-matrix $\phi = -\mathbf{E}\mathbf{T}^{-1}\mathbf{B} + \mathbf{D}$. The codeword can be expressed as $\bar{\mathrm{c}} = [\bar{\mathrm{x}}\ \bar{\mathrm{p}}_1\ \bar{\mathrm{p}}_2]$ and using equation 4.48 the parity vectors can be calculated from the following equations:

$$\mathbf{A}\bar{\mathrm{x}}^T + \mathbf{B}\bar{\mathrm{p}}_1^T + \mathbf{T}\bar{\mathrm{p}}_2^T = \bar{0}, \tag{4.63}$$

and

$$[-\mathbf{E}\mathbf{T}^{-1}\mathbf{A} + \mathbf{C}]\bar{\mathrm{x}}^T + \phi\bar{\mathrm{p}}_1^T = \bar{0}. \tag{4.64}$$

By rewriting equation 4.64, the first parity vector can be calculated most efficiently as:

$$\bar{\mathrm{p}}_1^T = -\phi[-\mathbf{E}\mathbf{T}^{-1}\mathbf{A}\bar{\mathrm{x}}^T + \mathbf{C}\bar{\mathrm{x}}^T]. \tag{4.65}$$

The second parity vector can then be calculated as:

$$\bar{\mathrm{p}}_2^T = -\mathbf{T}^{-1}[\mathbf{A}\bar{\mathrm{x}}^T + \mathbf{B}\bar{\mathrm{p}}_1^T]. \tag{4.66}$$

The encoding cost $(N + g^2)$ can be minimised by minimising g resulting in approximately linear complexity. Algorithms to design the parity-check matrix in equation 4.61 presented in [70] ensure a small value for g and a non-singular matrix ϕ. Alternatively, $\mathbf{H}$ can be generated randomly under certain constraints (such as row and column weight specifications, maximum allowed cycles in the matrix and maximum value for g) with a computer-search algorithm.

4.6.2 DECODING LDPC CODES

Regardless of the encoding method used, LDPC codes can be decoded by the same iterative algorithm using the code's factor graph constructed from the parity-check matrix. The decoding algorithm is a message-passing (MP) algorithm [14] known as Pearl's iterative belief propagation (IBP) algorithm [12]. In this section the construction of the factor graph and the MP algorithm are explained in terms of a non-systematic Hamming $(7, 4, 3)$ code. The algorithm presented here can be adapted to any binary error correcting code describable by a parity-check matrix $\mathbf{H}$.

4.6.2.1 FACTOR GRAPH CONSTRUCTION

A factor or Tanner graph [71] can be constructed for any parity check matrix $\mathbf{H}$ $(M \times N)$. The graphs used in decoding of linear block codes are bipartite graphs, with nodes connected with edges defined by the ones in $\mathbf{H}$. The graph contains two node types, including M check nodes $z_m; m = 1, \ldots, M$, and N code bit nodes $\zeta_n; n = 1, \ldots, N$ [14]. The parity-check matrix of the Hamming $(7, 4, 3)$ code considered is given as:

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{4.67}$$

Each row in $\mathbf{H}$ can be seen as a mask vector placed on the received codeword; the number of ones in the mask positions with value 1 should be even and with different parity-check rows with ones in different positions, the decoder can determine which bits in the codeword are erroneous. Each parity-check row is therefore represented by a check node in the factor graph with edges connecting the bit nodes corresponding to the ones in $\mathbf{H}$. The set of code bit positions from a codeword $\bar{\mathrm{c}} = [c_1, c_2, \ldots, c_N]$, participating in the $m^{\text{th}}; m = 1, \ldots, M$ check bit is given as [14]:

$$C_{bit}(m) = \{n : H_{m,n} = 1\}, \tag{4.68}$$

where the code position $n = 1, \ldots, N$, is only included in the set if the m^{th} row and n^{th} column entry of $\mathbf{H}$ is a 1. The code bit sets for the Hamming code given in equation 4.67 are therefore:

$$C_{bit}(1) = \{1, 2, 3, 5\} \tag{4.69}$$

$$C_{bit}(2) = \{2, 3, 4, 6\} \tag{4.70}$$

$$C_{bit}(3) = \{3, 4, 5, 7\}, \tag{4.71}$$

which completely describe the factor graph. However, the message passing algorithm also requires a description of the factor graph in terms of check bits associated with code bits. The set of check bit positions participating in the n^{th} code bit is given as:

$$C_{check}(n) = \{m : H_{m,n} = 1\}, \tag{4.72}$$

using the same notation as in equation 4.68. The check bit sets for the Hamming code are therefore:

$$\begin{aligned} C_{check}(1) &= \{1\} && (4.73)\\ C_{check}(2) &= \{1,2\} && (4.74)\\ C_{check}(3) &= \{1,2,3\} && (4.75)\\ C_{check}(4) &= \{2,3\} && (4.76)\\ C_{check}(5) &= \{1,3\} && (4.77)\\ C_{check}(6) &= \{2\} && (4.78)\\ C_{check}(7) &= \{3\}. && (4.79) \end{aligned}$$

The factor graph of the Hamming $(7,4,3)$ code is shown in figure 4.10. The graph is irregular because $\mathbf{H}$ does not have a constant row weight w_r (which is equal to the number of edges connected to a check node z_m) or a constant column weight w_c (which is equal to the number of edges connected to a code bit node ζ_n).

The factor graph of a regular LDPC $(36,18)$ code defined by a parity-check matrix with a form as in equation 4.61 with $w_r = 6$ and $w_c = 3$ is shown in figure 4.11. Each check node is connected with w_r edges to different code bit nodes and each code bit node is connected with w_c edges to different check nodes. The factor graph shown in figure 4.11 is sparse because the number of connected edges are far less than the maximum number of possible edges. In both figures 4.10 and 4.11 the white circle nodes represent transmitted bits and the square plus nodes represent parity constraints (the sum of all bits connected to a square node is forced to be even).

Each received codeword bit c_n is represented by a corresponding code bit node ζ_n on the factor graph and each check node is satisfied if the number of code bits with value 1, connected to each check node, is even. The decoding algorithm tries to satisfy all the check nodes by passing messages in the form of probabilities between nodes connected via edges. The algorithm iterates repeatedly over the graph to extract all

information from the received sequence in order to maximise the posterior probability $p(t_n|\mathbf{r})$ for each code bit. The detailed decoding algorithm is considered in the next section.

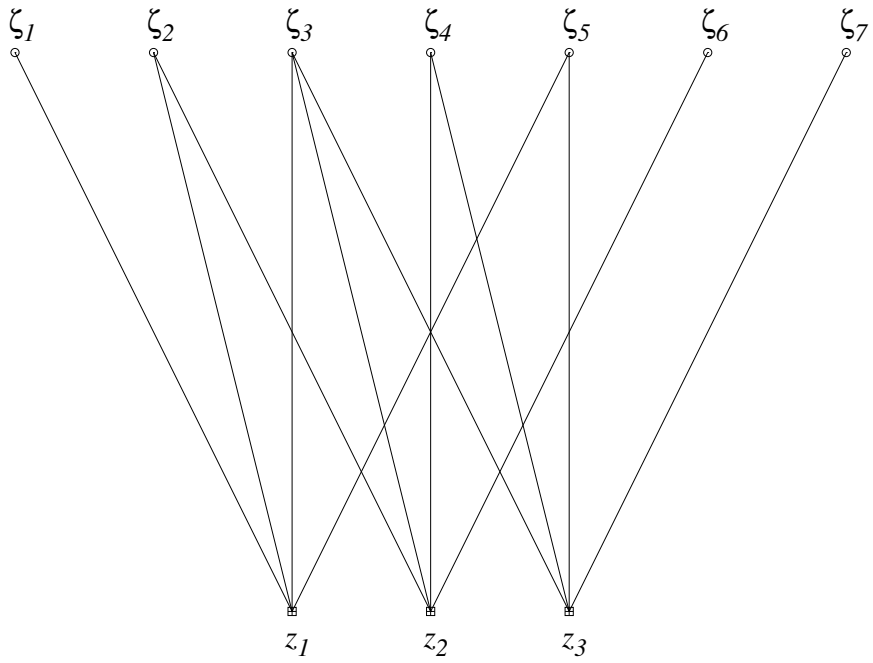

Figure 4.10: Irregular factor graph of the Hamming $(7, 4, 3)$ code with code bit nodes $\zeta_n; n = 1, \ldots, N$, and check nodes $z_m; m = 1, \ldots, M$

4.6.2.2 BELIEF PROPAGATION DECODING

The message-passing or sum-product algorithm used to decode LDPC codes is known as Pearl's belief propagation algorithm [14]. The algorithm takes as input the channel LLR $\Lambda_c(r_n|t_n)$ (see equation 4.12) and ideally provides the exact posterior probability $p(t_n|\mathbf{r})$ for every transmitted bit. In the ideal case the factor graph is cycle-free (a cycle is a path, starting and ending in the same node, across edges where an edge participates only once in the path) and the algorithm converges to the exact solution after a fixed number of iterations.

The algorithm decodes the received sequence in three phases of which the first is an initialisation phase and the second and third phases are repeated until the specified number of iterations is reached (or a stopping condition is satisfied).

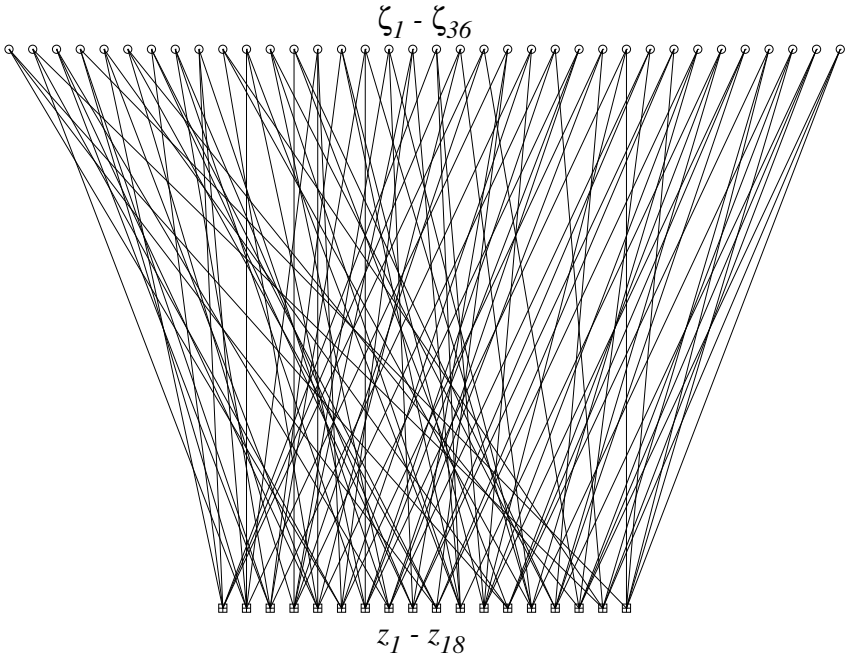

Figure 4.11: Regular factor graph of an LDPC $(36, 18)$ **code with** $w_r = 6$ **and** $w_c = 3$

The algorithm passes the following probabilities between nodes in order to perform decoding:

$q_{m,n}^{t=\pm 1}$ The probability that the transmitted bit takes on values $t_n = +1$ or $t_n = -1$, given the information obtained from all the check nodes, excluding check node z_m

$w_{m,n}^{t=\pm 1}$ The probability that check node z_m is satisfied if $t_n = +1$ or $t_n = -1$ and the other bits $t_{n'}$ have probabilities $q_{m,n'}^{t}$ with $n' \in \{C_{bit}(m) \backslash n\}$ (a set $\{S(i) \backslash j\}$ contains all elements s_i but not s_j)

The algorithm as presented in [12, 14] is discussed next and a detailed example is provided in Addendum B.

- In the initialisation phase, the code bit node probabilities are set equal to the likelihoods of transmitted bits (see equations 4.7, 4.8, 4.10 and 4.12). The probability that code bit node $\zeta_n; n = 1, \ldots, N$, is $+1$ (equivalent to the probability that the transmitted bit $t_n = +1$) is:

$$p_n^{t=+1} = p(r_n | t_n = +1) = \frac{1}{1 + e^{-\Lambda_c(r_n|t_n)}} \tag{4.80}$$

and the probability that code bit ζ_n is -1, is:

$$p_n^{t=-1} = p(r_n|t_n = -1) = 1 - p_n^{t_n=+1} \tag{4.81}$$

The q-values are initialised by setting $q_{m,n}^{t=+1} = p_n^{t=+1}$ and $q_{m,n}^{t=-1} = p_n^{t=-1}$ for every edge connecting a code bit node ζ_n to a check node z_m.

- In the second phase (the horizontal step in which the algorithm passes through the rows of $\mathbf{H}$) the algorithm runs through the check nodes z_m and computes for each $n \in C_{bit}(m)$ (all code bit nodes connected to z_m) the probabilities $w_{m,n}^{t=\pm1}$ which are calculated from products of the differences $\delta q_{m,n} = q_{m,n}^{t=-1} - q_{m,n}^{t=+1}$. The probability $w_{m,n}^t$ is obtained from the identity [12, 14]:

$$\delta w_{m,n} = \prod_{n' \in C_{bit}(m) \backslash n} \delta q_{m,n'} \tag{4.82}$$

by applying the following transformations:

$$w_{m,n}^{t=+1} = \frac{1}{2}(1 - \delta w_{m,n}), \tag{4.83}$$

and

$$w_{m,n}^{t=-1} = \frac{1}{2}(1 + \delta w_{m,n}). \tag{4.84}$$

- In the third phase (the vertical step in which the algorithm passes through the columns of $\mathbf{H}$) the algorithm runs through the code bit nodes ζ_n and updates the probabilities $q_{m,n}^{t=\pm1}$ using the probabilities $w_{m,n}^{t=\pm1}$ calculated in the second phase, by applying the equations:

$$\tilde{q}_{m,n}^{t=+1} = p_n^{t=+1} \prod_{m' \in C_{check}(n) \backslash m} w_{m',n}^{t=+1} \tag{4.85}$$

and

$$\tilde{q}_{m,n}^{t=-1} = p_n^{t=-1} \prod_{m' \in C_{check}(n) \backslash m} w_{m',n}^{t=-1} \tag{4.86}$$

Since it is required that $q_{m,n}^{t=+1} + q_{m,n}^{t=-1} = 1$, equations 4.85 and 4.86 are normalised by a scaling factor $\alpha_{m,n} = (\tilde{q}_{m,n}^{t=+1} + \tilde{q}_{m,n}^{t=-1})^{-1}$ to provide:

$$q_{m,n}^{t=+1} = \alpha_{m,n} \tilde{q}_{m,n}^{t=+1} \tag{4.87}$$

and

$$q_{m,n}^{t=-1} = \alpha_{m,n} \tilde{q}_{m,n}^{t=-1} \tag{4.88}$$

The estimated posterior probabilities $q_n^{t=\pm 1}$ are calculated after the maximum number of iterations are reached, similar to calculating $q_{m,n}^{t=\pm 1}$ but without excluding any check node, by applying the equations:

$$q_n^{t=+1} = \alpha_n p_n^{t=+1} \prod_{m \in C_{check}(n)} w_{m,n}^{t=+1} \tag{4.89}$$

and

$$q_n^{t=-1} = \alpha_n p_n^{t=-1} \prod_{m \in C_{check}(n)} w_{m,n}^{t=-1} \tag{4.90}$$

with α_n a constant such that $q_n^{t=+1} + q_n^{t=-1} = 1$. The estimated posterior LLR for each code bit can then be calculated as:

$$\Lambda(t_n|\mathbf{r}) = \ln(q_n^{t=+1}) - \ln(q_n^{t=-1}), \tag{4.91}$$

If a fixed number of iterations is specified, the decoder might converge to the correct codeword before the number of iterations is reached, resulting in unnecessary iterations. To prevent unnecessary iterations a stopping condition (e.g. stopping the decoder if $\mathbf{H} \cdot \tilde{c}^T = \bar{\mathbf{0}}$ with $\tilde{c}$ the decoded codeword) can be implemented, which will also require the calculation of equation 4.91 at the end of each iteration.

4.6.2.3 LLR BELIEF PROPAGATION DECODING

In stead of using probabilities, LLRs can be used as messages to perform BP decoding on a factor graph. Using LLRs offers implementation advantages in the BP algorithm, such as the replacement of multiplications by additions and the removal of the normalisation step [37]. LLR-BP decoding is employed in the decoding of RA codes, considered in the next section.

4.7 RA CODES

In a search for coding theorem proofs for turbo codes, Divsalar et al. [29] considered several Turbo-like codes, including Repeat-Accumulate (RA) codes. RA codes are extremely simple codes which allow linear-time encoding with performance results comparable to Turbo and LDPC codes. It is shown in [72] that RA codes are capacity-approaching codes. This study only considers the original non-systematic regular RA codes, though systematic [39] and irregular [73] RA codes have been proposed. RA codes can be decoded using the MP algorithm on the code's factor graph, similar to decoding LDPC codes.

4.7.1 ENCODING RA CODES

The encoder structure for a non-systematic RA (N, K) code is shown in figure 4.12.

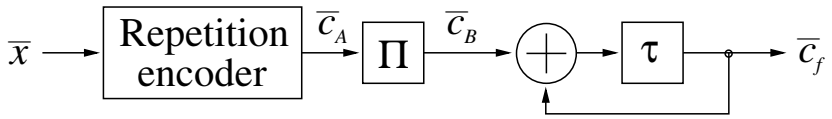

Figure 4.12: Non-systematic Repeat-Accumulate encoder

The encoding algorithm based on figure 4.12 is given below (see also [40]).

- The K-bit input data block $\bar{x}$ passes through a rate-$\frac{1}{\gamma}$ repetition encoder to provide the N-bit codeword $\bar{c}_A$. Every bit in $\bar{x}$ is repeated γ times ($N = \gamma K$) with $\gamma \geq 3$ (see [29]), e.g. if $\gamma = 3$ the codeword is:

$$\bar{c}_A = [x_1, x_1, x_1, x_2, x_2, x_2, \ldots, x_K, x_K, x_K], \tag{4.92}$$

  or alternatively:

$$\bar{c}_A = [x_1, x_2, \ldots, x_\gamma,\ x_1, x_2, \ldots, x_\gamma,\ \ \ldots,\ \ x_1, x_2, \ldots, x_\gamma], \tag{4.93}$$

  where the group $[x_1, x_2, \ldots, x_\gamma]$ repeats K times in $\bar{c}_A$.

- The codeword $\bar{c}_A$ is interleaved to provide $\bar{c}_B = \mathbf{\Pi} \cdot \bar{c}_A$. The interleaver $\mathbf{\Pi}$ is a fixed random interleaver permuting each codeword $\bar{c}_A$ in the same way.

- The codeword $\bar{c}_B$ is encoded by a rate-1 recursive convolutional encoder with transfer function $T(D) = (1+D)^{-1}$, providing as output the final N-bit codeword $\bar{c}_f$. The convolutional encoder performs an accumulating function described by the equation:

$$c_{f,n} = \begin{cases} c_{B,1} & for\ n = 1 \\ c_{f,n-1} + c_{B,n} & for\ 2 \leq n \leq N \end{cases} \tag{4.94}$$

  using modulo-2 addition with $c_{f,n}$ and $c_{B,n}$ the n^{th} bits of the codewords $\bar{c}_f$ and $\bar{c}_B$ respectively.

4.7.1.1 GENERATOR MATRIX DESCRIPTION

The function of the RA encoder in figure 4.12 can also be described as a generator matrix:

$$\mathbf{G_{RA}} = \mathbf{G_R} \cdot \mathbf{\Pi} \cdot \mathbf{G_A}, \tag{4.95}$$

with $\mathbf{G_R}$ and $\mathbf{G_A}$ the generator matrices of the repetition encoder and accumulator respectively. The generator matrix $\mathbf{G_R}$ for $\gamma = 3$ and $K = 3$ is:

$$\mathbf{G_R} = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 \end{bmatrix} \tag{4.96}$$

which will produce an encoded sequence $\bar{c}_A$ as given in equation 4.92. By rearranging the columns in equation 4.96, the generator matrix can be written as:

$$\mathbf{G_R} = [\ \mathbf{I_K}\ \mathbf{I_K}\ \mathbf{I_K}\ \cdots\ \mathbf{I_K}\ ], \tag{4.97}$$

with γ identity matrices $\mathbf{I_K}$ (the dimension of $\mathbf{G_R}$ is $K \times \gamma K = K \times N$). The generator matrix in equation 4.97 will produce the encoded sequence $\bar{c}_A$ given in equation 4.93.

The interleaver $\mathbf{\Pi}$ can be described as an $N \times N$ matrix obtained from the identity matrix $\mathbf{I_N}$ by exchanging the columns of $\mathbf{I_N}$. By exchanging the i^{th} and j^{th} column of $\mathbf{I_N}$, the i^{th} and j^{th} bits in the codeword $\bar{c}_A$ are interleaved to become the j^{th} and i^{th} bits in the codeword $\bar{c}_B$. The repeat generator matrices $\mathbf{G_R}$ given in equations 4.96 and 4.97 can be made equivalent by adapting $\mathbf{\Pi}$.

The accumulator (convolutional encoder) can be viewed as a block code with an $N \times N$ dimension generator matrix which is an upper triangular matrix filled with ones, given as:

$$\mathbf{G_A} = \begin{bmatrix} 1 & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ 0 & 1 & & & & & \cdot \\ \cdot & 0 & \cdot & & & & \cdot \\ \cdot & & \cdot & \cdot & & & \cdot \\ \cdot & & & \cdot & \cdot & & \cdot \\ \cdot & & & & 0 & 1 & \cdot \\ 0 & \cdot & \cdot & \cdot & \cdot & 0 & 1 \end{bmatrix} \tag{4.98}$$

4.7.2 DECODING RA CODES

RA codes are decoded using a message passing algorithm on the code's factor graph similar to the method used to decode LDPC codes (see section 4.6.2). In this section the construction of the factor graph of an RA code and the MP decoding algorithm are considered. A short RA $(6, 2)$ code (with repetition factor $\gamma = 3$) is used to explain the decoding algorithm, which can be extended to any length RA code.

4.7.2.1 FACTOR GRAPH CONSTRUCTION

A few factor graph constructions for RA codes are possible, including a graph which contains the trellis of the convolutional accumulator [14] and the graph constructed from the encoder using elementary factor graph nodes [40, 74]. The latter factor graph construction is considered in this study. The factor graph of the RA $(6, 2)$ code is shown in figure 4.13.

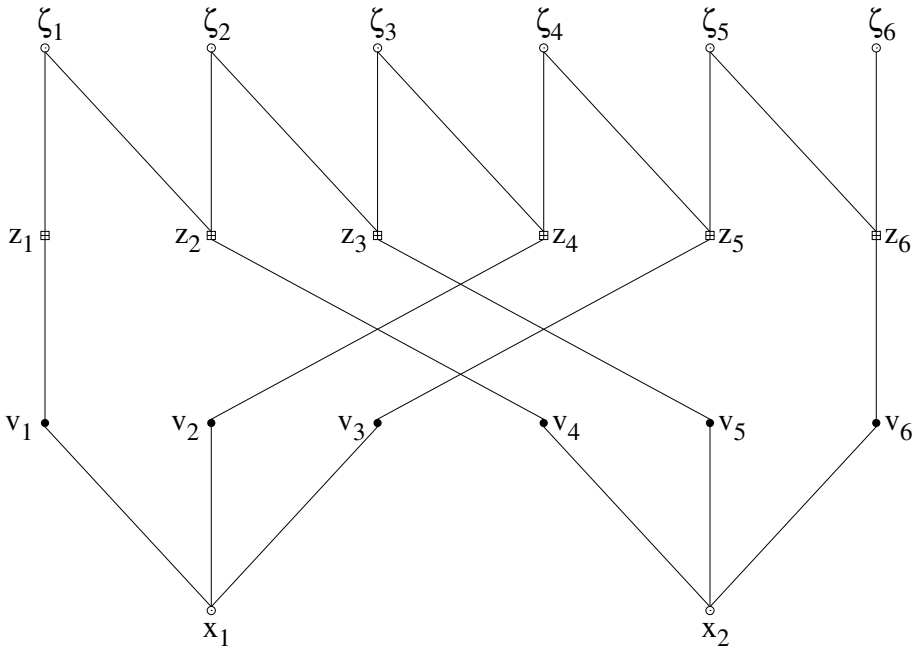

Figure 4.13: Factor graph of RA $(6, 2)$ code

The elementary nodes [14] used in figure 4.13 consist of variable (circles) and check (squares) nodes including:

- Equality constraint variable nodes shown as circles associated with the data bits x_k, forcing the bits connected to each constraint node to be equal

- Intermediate variable nodes v_n shown as black circles representing the code bits of the codeword $\bar{c}_A$ (see figure 4.12)
- Parity constraint nodes z_n shown as plus squares representing the code bits of the interleaved codeword $\bar{c}_B$ (see figure 4.12), forcing the sum of the bits connected to each constraint node to be even
- Variable nodes ζ_n shown as white circles representing the code bits of the codeword $\bar{c}_f$

The factor graph shown in figure 4.13 is divided into 4 layers corresponding to the encoder structure shown in figure 4.12. The layers are listed below.

- The x-layer which contains the nodes representing the original uncoded data bits
- The v-layer showing the repetition encoder function; each data bit in the x-layer is connected to γ variable nodes in the v-layer
- The z-layer showing the interleaver function; each node in the v-layer is connected to a check node in the z-layer according to $\mathbf{\Pi}$ on a one-to-one basis
- The ζ-layer which contains the nodes representing the transmitted code bits

4.7.2.2 LLR BASED MESSAGE PASSING

The factor graph messages are passed as LLRs (see equation 4.5) which can be written as:

$$\Lambda(m_{a,b}) = \ln\left[\frac{p(m_{a,b}=+1)}{p(m_{a,b}=-1)}\right] \tag{4.99}$$

where $m_{a,b}$ is the message (or variable) passed from the a-layer to the b-layer in the factor graph. Using equations 4.7 and 4.8, the probabilities can be expressed as:

$$p(m_{a,b}=+1) = \frac{e^{\Lambda(m_{a,b})}}{1+e^{\Lambda(m_{a,b})}} \tag{4.100}$$

and

$$p(m_{a,b}=-1) = \frac{1}{1+e^{\Lambda(m_{a,b})}} \tag{4.101}$$

The difference in probabilities can then be expressed as:

$$p(m_{a,b}=+1) - p(m_{a,b}=-1) = \frac{e^{\Lambda(m_{a,b})}-1}{e^{\Lambda(m_{a,b})}+1} \tag{4.102}$$

which is equal to the hyperbolic tangent function [37]:

$$p(m_{a,b} = +1) - p(m_{a,b} = -1) = \tanh\left[\frac{\Lambda(m_{a,b})}{2}\right] \tag{4.103}$$

The input probabilities at a variable node in a factor graph are multiplied together to produce the output probability (see section 4.6.2), though if LLRs are employed, message addition is used in stead of multiplication. The output LLR message at a variable node can be expressed as:

$$\Lambda(m) = \sum_i \Lambda(m_i), \tag{4.104}$$

with $\Lambda(m_i)$ the input LLR message. Figure 4.14 shows a variable node with three input LLR messages $\Lambda(m_i)$ with $i = 1, 2, 3$.

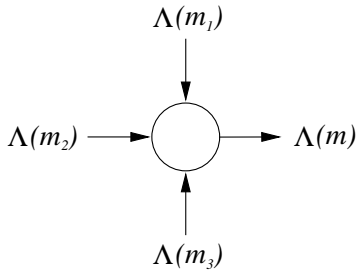

Figure 4.14: Illustration of LLR message updating at a variable node

To update messages at a check node, with statistically independent binary random variables as input messages, the *tanh*-rule given by [37, 75] is used as follows:

$$\tanh\left[\frac{\Lambda(m)}{2}\right] = \prod_i \tanh\left[\frac{\Lambda(m_i)}{2}\right] \tag{4.105}$$

Figure 4.15 shows a check node with three input LLR messages $\Lambda(m_i)$ with $i = 1, 2, 3$.

4.7.2.3 MESSAGE PASSING DECODING OF RA CODES

The iterative message passing decoding algorithm used to decode RA codes is discussed in this section in terms of the RA $(6, 2)$ code with factor graph shown in figure 4.13. The decoding algorithm based on [37, 40, 74] functions as follows.

- During the first iteration the input messages $\Lambda_n(m_\zeta)$ at the ζ-layer of the factor graph are set equal to the channel LLR given in equations 4.10 and 4.12:

$$\Lambda_n(m_\zeta) = \Lambda_c(r_n|t_n) = \frac{4}{N_0}\alpha_n r_n \tag{4.106}$$

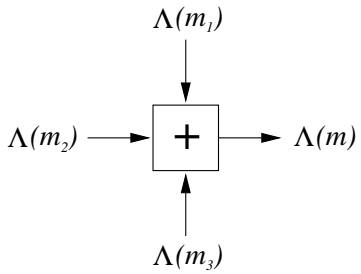

Figure 4.15: Illustration of LLR message updating at a check node

- The $\Lambda_{n,n'}(m_{\zeta,\mathrm{z}})$ messages (LLR messages passed from the n^{th} node in the ζ-layer to the $(n')^{\text{th}}$ node in the z-layer) are updated using equation 4.104 with the $\Lambda_n(m_\zeta)$ input messages and the $\Lambda(m_{\mathrm{z},\zeta})$ messages (which are zero during the first iteration) as shown in figure 4.16

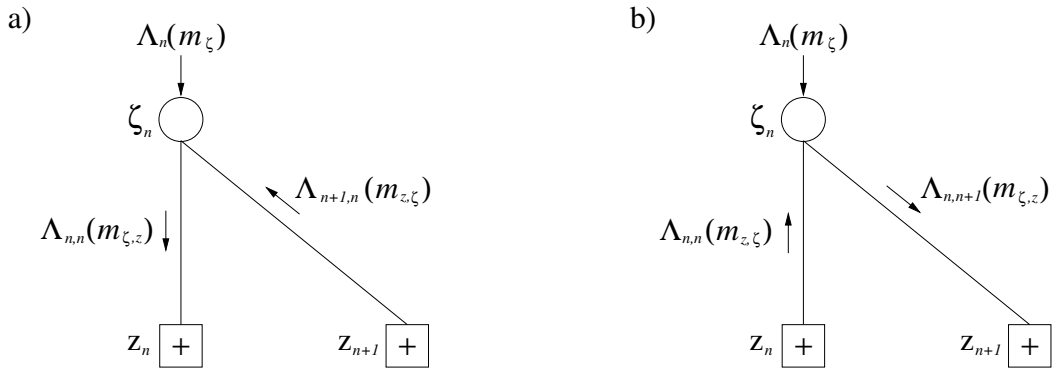

Figure 4.16: Illustration of message updating at the ζ-layer during decoding

In figure 4.16(a) the LLR message passed along the vertical edge is:

$$\Lambda_{n,n}(m_{\zeta,\mathrm{z}}) = \Lambda_n(m_\zeta) + \Lambda_{n+1,n}(m_{\mathrm{z},\zeta}), \tag{4.107}$$

with $n \in \{1, \ldots, N-1\}$. For the case where $n = N$ only the vertical edge is considered. In figure 4.16(b) the LLR message passed along the diagonal edge is:

$$\Lambda_{n,n+1}(m_{\zeta,\mathrm{z}}) = \Lambda_n(m_\zeta) + \Lambda_{n,n}(m_{\mathrm{z},\zeta}), \tag{4.108}$$

with $n \in \{1, \ldots, N-1\}$.

- The $\Lambda_{n,n'}(m_{\mathrm{z,v}})$ messages are updated using equation 4.105 with the $\Lambda(m_{\zeta,\mathrm{z}})$ messages as shown in figure 4.17. The message passed from the z-layer to the

v-layer is:

$$\Lambda_{n,n'}(\mathrm{z},\mathrm{v}) = 2\tanh^{-1}\left\{\tanh\left[\frac{\Lambda_{n-1,n}(m_{\zeta,\mathrm{z}})}{2}\right]\tanh\left[\frac{\Lambda_{n,n}(m_{\zeta,\mathrm{z}})}{2}\right]\right\} \tag{4.109}$$

with $n \in \{2,\ldots,N\}$. For the case $n = 1$, the message passes only through node z_1 and is updated with the equation $\Lambda_{1,1}(m_{\mathrm{z},\mathrm{v}}) = \Lambda_{1,1}(m_{\zeta,\mathrm{z}})$. The output message $\Lambda_{n,n'}(m_{\mathrm{z},\mathrm{v}})$ is passed from node z_n to node $\mathrm{v}_{n'}$ as determined by the interleaver.

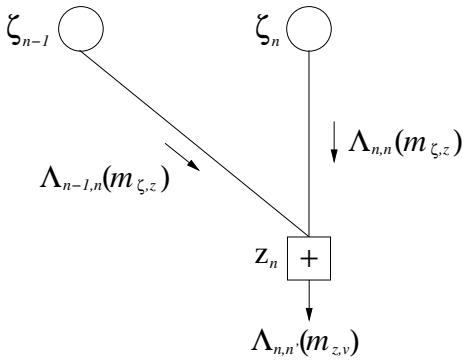

Figure 4.17: Illustration of message updating at the z-layer during decoding

- The $\Lambda_{n,k}(m_{\mathrm{v},\mathrm{x}})$ messages are updated by setting each of them equal to the message passed from the z-layer to the v-layer as dictated by the interleaver:

$$\Lambda_{n,k}(m_{\mathrm{v},\mathrm{x}}) = \Lambda_{n',n}(m_{\mathrm{z},\mathrm{v}}), \tag{4.110}$$

  where the (n,k) and (n',n) combinations are obtained from the factor graph shown in figure 4.13.

- The $\Lambda_{k,n}(m_{\mathrm{x},\mathrm{v}})$ messages are updated using equation 4.104 with the $\Lambda(m_{\mathrm{v},\mathrm{x}})$ messages as shown in figure 4.18. The messages passed from the x-layer to the v-layer are:

$$\Lambda_{k,3k}(m_{\mathrm{x},\mathrm{v}}) = \Lambda_{3k-1,k}(m_{\mathrm{v},\mathrm{x}}) + \Lambda_{3k-2,k}(m_{\mathrm{v},\mathrm{x}}) \tag{4.111}$$

$$\Lambda_{k,3k-1}(m_{\mathrm{x},\mathrm{v}}) = \Lambda_{3k,k}(m_{\mathrm{v},\mathrm{x}}) + \Lambda_{3k-2,k}(m_{\mathrm{v},\mathrm{x}}) \tag{4.112}$$

$$\Lambda_{k,3k-2}(m_{\mathrm{x},\mathrm{v}}) = \Lambda_{3k,k}(m_{\mathrm{v},\mathrm{x}}) + \Lambda_{3k-1,k}(m_{\mathrm{v},\mathrm{x}}), \tag{4.113}$$

  with $k \in \{1,\ldots,K\}$. Equations 4.111, 4.112 and 4.113 correspond respectively to figure 4.18(a), (b) and (c).

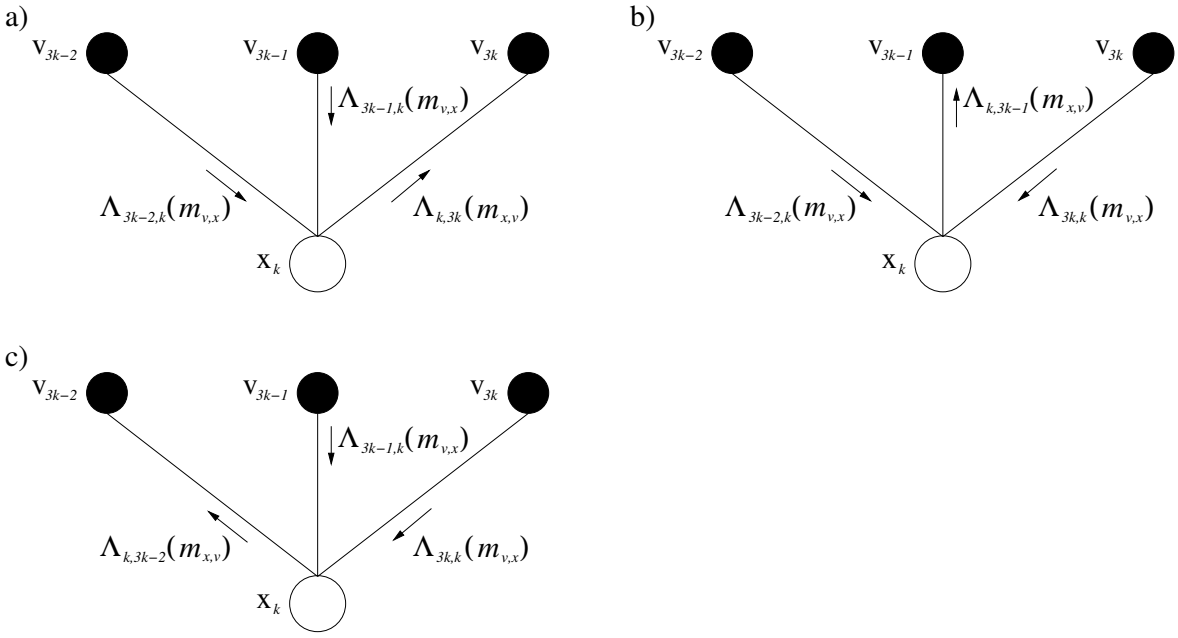

Figure 4.18: Illustration of message updating at the x-layer during decoding

- The $\Lambda_{n,n'}(m_{\mathrm{v,z}})$ messages are updated by setting each of them equal to the message passed from the x-layer to the v-layer as dictated by the interleaver:

$$\Lambda_{n,n'}(m_{\mathrm{v,z}}) = \Lambda_{k,n}(m_{\mathrm{x,v}}), \tag{4.114}$$

  where the (n, n') and (k, n) combinations are obtained from the factor graph shown in figure 4.13.

- The $\Lambda_{n,n'}(m_{\mathrm{z},\zeta})$ messages are updated using equation 4.105 with the $\Lambda(m_{\mathrm{v,z}})$ messages as shown in figure 4.19. In figure 4.19(a) the LLR message passed along the vertical edge is:

$$\Lambda_{n,n}(\mathrm{z},\zeta) = 2\tanh^{-1}\left\{\tanh\left[\frac{\Lambda_{n',n}(m_{\mathrm{v,z}})}{2}\right]\tanh\left[\frac{\Lambda_{n-1,n}(m_{\zeta,\mathrm{z}})}{2}\right]\right\} \tag{4.115}$$

  with $n \in \{1, \ldots, N\}$. For the case where $n = 1$ only the vertical edge is considered. In figure 4.19(b) the LLR message passed along the diagonal edge is:

$$\Lambda_{n,n-1}(\mathrm{z},\zeta) = 2\tanh^{-1}\left\{\tanh\left[\frac{\Lambda_{n',n}(m_{\mathrm{v,z}})}{2}\right]\tanh\left[\frac{\Lambda_{n,n}(m_{\zeta,\mathrm{z}})}{2}\right]\right\} \tag{4.116}$$

  with $n \in \{2, \ldots, N\}$.

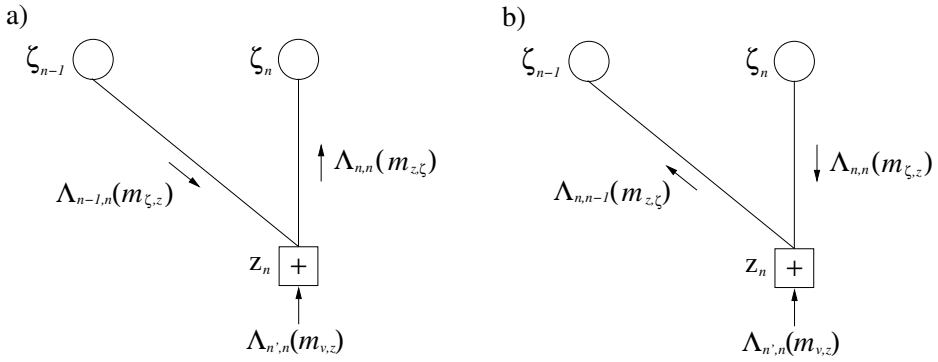

Figure 4.19: Illustration of message updating at the z-layer during decoding

The received sequence is decoded by repeatedly executing the above message passing algorithm until the specified number of iterations is reached. The output of the decoding algorithm is the estimated posterior LLR for each data bit, which is expressed as:

$$\Lambda(t_n|\mathbf{r}) = \Lambda_k(m_{\mathrm{x}}) = \sum_i \Lambda_{3k-i}(m_{\mathrm{v,x}}), \tag{4.117}$$

with $i \in \{0, 1, 2\}$ as shown in figure 4.20.

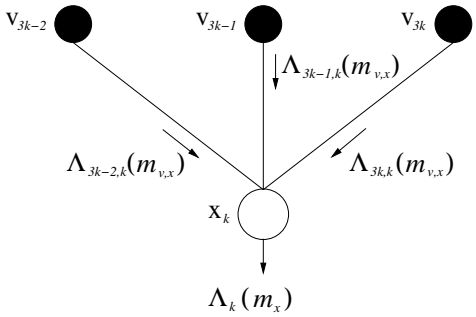

Figure 4.20: Calculation of MP algorithm output at the x-layer

4.8 CODING COST AND COMPLEXITY

The cost of encoding and decoding in terms of the number of mathematical operations required plays a significant role in the choice of which coding scheme to implement. The number of multiplications necessary to perform coding dominates the coding cost and is the parameter with which different schemes can be compared. Table 4.2 shows how the coding cost (number of multiplications required) scales with coding dimensions

Table 4.2: Dominant coding cost factor for a single coding module

Cost	RM code	LDPC code	RA code
Encoding	NK or MK	MK, M^2, $k_{LU}M^2$ or $N+g$	Additions only
Decoding	$3(2^{N/2+2})$	Mw_r^2	$12N$

(N, K) and parameters $M = N - K$, k_{LU} an LU-factor (see section 4.6.1.2), g the gap dimension (see section 4.6.1.3) and w_r the row weight of $\mathbf{H}$ (see section 4.6).

4.8.1 ENCODING COST OF A SINGLE CODER

The encoding cost of the 3 codes shown in table 4.2 is explained in this section. The encoding cost of the RM code is either NK (non-systematic) or MK (systematic) if the standard matrix multiplication method discussed in section 4.6.1.1 is used. Sparse matrix encoding of LDPC codes (section 4.6.1.2) has different encoding costs depending on which method is used. The two-step method (equations 4.53 and 4.54) has cost $M^2 + Mw_r$ and the **LU**-decomposition method has cost proportional to $k_{LU}M^2$. The approximate triangular LDPC encoding method discussed in section 4.6.1.3 has cost $N + g$. The RA encoder does not require any multiplications, because the encoder (see section 4.7.1) consists of a repetition encoder, interleaver and accumulator. The RA encoder only requires additions, which scales linearly with N, though the cost in terms of additions is much less than multiplication cost.

4.8.2 DECODING COST OF A SINGLE CODER

By counting the number of multiplications required for each decoding algorithm, the decoding cost factors in table 4.2 are obtained. The number of branches in the half-rate systematic trellises considered in this study (see section 4.5.2) is $2^{N/2+2} - 4$ and the cost of calculating a branch metric (see section 4.5.3.3) is 3 multiplications and 2 additions and therefore the dominant scaling factor is $3(2^{N/2+2})$. The decoding cost of LDPC codes (belief propagation decoding discussed in section 4.6.2.2) can be shown to be $Mw_r^2 + 4(w_c + 1) + 2$. The cost related to RA decoding (section 4.7.2) is $12N - 12$ multiplications (assuming the *tanh*-rule is implemented using a look-up table) and $K\gamma(\gamma - 2) + 2N - 2$ additions (γ is the RA repetition factor) per iteration (neglecting computations necessary for the initial and final decoding stages) and hence the dominant cost factor is $12N$.

Table 4.3: Effect of coding dimension n on coding parameters

Coding Parameter	Block Turbo Code	LDPC/RA code
Encoder complexity	Linear with n	Linear with n
Decoder complexity	Quadratically with n	Linear with n
Encoding time	Single constituent encoding time	Linear with n
Decoding time	Linear with n	Linear with n
Encoder components	Interleavers, encoders	S/P converter, encoder
Decoder components	Interleavers, decoders	P/S converter, decoder
Encoder structure	Parallel	Serial
Decoder structure	Serial	Serial

4.8.3 CODING COST OF AN MD CODER

The complexity of the coding and modulation structures depends on the dimension n of the MD platform. Table 4.3 indicates the relationship between coding dimension n, the overall complexity of the coding and modulation structures in terms of the number and size of components required and the coding delay (or processing) times. In general, complexity increases with increasing n, though the actual complexity depends on the constituent coding modules' complexity. The exact encoding and decoding delay times also depend on the coding complexity of the constituent coders, the structure of each coder and the implementation platform.

4.8.3.1 CODING COST OF BLOCK TURBO CODE

The n-dimensional block turbo code presented in this study is implemented on an $(n+1)$-dimensional modulation platform. The input data occupies one dimension and the coded information occupies n dimensions in the MD communication system.

The complexity of the turbo encoder structure increases linearly with the encoding dimension n in terms of the number of constituent encoders and interleavers required, assuming each constituent code's dimensions N and K remain constant. The 2D turbo encoder (figure 4.2) requires 2 constituent encoders (typically identical) and 1 matrix (rows-and-columns) interleaver. The 3D encoder (figure 4.4) requires 3 constituent encoders and 2 block interleavers. In general, an n-dimensional (with $n \geq 2$) turbo encoder requires n constituent encoders and $(n-1)$ n-dimensional interleavers.

The number of data bits processed simultaneously by each interleaver increases with a factor K with increasing dimension n. An n-dimensional interleaver therefore requires memory space for K^n bits, though all the interleavers can be implemented by temporarily storing the n-dimensional input data structure and performing a read-out interleaving rule for each constituent encoder.

Due to the parallel processing structure of turbo codes, the encoding delay depends only on the delay induced by the constituent encoders (and not on the encoding dimension n, assuming a parallel implementation is possible and no delay is incurred by loading the data bits into the interleaver), though the encoder complexity increases linearly with increasing n. Therefore, a fixed number of available data bits (enough data bits to fill an n-dimensional interleaver structure) can be encoded in equal time for any encoding dimension less than or equal to n.

The complexity of the turbo decoder structure increases quadratically with the encoding dimension n in terms of the number of constituent encoders and interleavers required, assuming each constituent code's dimensions N and K remain constant. The 2D turbo decoder (figure 4.3) requires 2 constituent decoders and 3 matrix interleavers. The 3D decoder (figure 4.6) requires 3 constituent decoders and 8 block interleavers. In general, an n-dimensional turbo decoder requires n constituent decoders and $(n^2 - 1)$ n-dimensional interleavers. Due to the serial structure of a turbo decoder, the decoding time delay of a single decoding iteration increases linearly with increasing encoding dimension n, assuming a parallel processing platform.

4.8.3.2 CODING COST OF LDPC/RA CODE

The coder structures of both the LDPC and RA codes consist of S/P and P/S converters and coding modules. A higher modulation platform therefore demands larger S/P and P/S converters, though the converter sizes and coding modules grow linearly with n. The LDPC encoder structure is serial, because only one encoder works on the input stream at the transmitter and a serially connected S/P converter adapts the coded sequence for transmission across the MD platform. The RA encoder (figure 4.12) functions similar to the LDPC encoder, with serially connected modules. The decoding structures consist of P/S converters connected in series with the actual decoders.

CHAPTER 5

ERROR PERFORMANCE RESULTS

5.1 CHAPTER SUMMARY

This chapter contains simulated error performance results of the uncoded and coded WCDMA communication system presented in Chapter 2. Both the 2D and 4D systems are evaluated, though the focus is on the 4D system employing ZCC CSSs.

The sparse graph channel coding schemes discussed in Chapter 4 are evaluated on the MD-WCDMA platform and include the following codes:

- 3D Turbo code with RM (8,4) and extended RM (12,6) constituent codes, forming respectively (256,64) and (864,216) codes
- LDPC (36,18), LDPC (256,128) and LDPC (256,64) codes constructed by random parity check matrix generation under constraints placed on cycle length (avoiding cycles of length 4 and shorter) and row and column weight (the column weight is mostly $w_c = 3$ and sometimes $w_c = 4$ and the row weight varies between $w_r = 2$ and $w_r = 10$ in order to satisfy cycle length constraints)
- RA (36,18), RA (256,128) and RA (256,64) codes

The channel conditions include AWGN and multipath frequency selective fading as discussed in Chapter 2. All results were obtained by baseband simulations with a transmission rate of 1000 symbols per second and 8 samples per chip. CSS lengths of 63 chips were used, resulting in a sampling rate of 504 kHz.

The performance evaluation setup (see Chapter 3) and theoretical performance curves and limits are given. Simulation studies are necessary to evaluate the error performance because of the vast number of parameters influencing the performance, making the derivation of theoretical expressions cumbersome. Performance bounds such as the

union bound for block codes [7, 12] and more accurate performance expressions [76] exist, though little work has been done on theoretical performance expressions of sparse graph codes [29].

5.2 EVALUATION SETUP

The channel simulation models and setup are discussed in detail in Chapter 3. This section is a short summary of the communication system setup to evaluate the system's error performance under different channel conditions.

The uncoded simulation platform shown in figure 3.6 is shown again in figure 5.1 (uncoded system) and figure 5.2 (coded system) indicating how the input $x(t)$ and output $y(t)$ data sequences are compared to obtain the BER. The simulation platform is configurable to either introduce multipath fading effects or AWGN channel effects only, by bypassing the fading channel simulator. Using the spreading sequences discussed in Chapter 2, the performance of 1 to 7 CDMA users can be evaluated by adding additional user data to the transmitted data of user 1 (which is the reference communication system). Due to ZCC CSS characteristics, only 1 to 5 users are considered in the coded system. The multipath channel setup is given in table 3.1 in Chapter 3.

In the coded communication system (figure 5.2) the data sequence $x(t)$ originating from a random data generator, is encoded and then passed through an interfacing module to adapt the signal for transmission by the 4D transmitter. The code interface is an S/P module at the transmitter and a P/S module at the receiver. The channel encoder encodes a serial data sequence (or block) into a serial coded sequence and the 4D transmitter takes as input 4 parallel bit sequences which are given as 4 parallel estimated sequences at the output of the 4D receiver. It is therefore necessary to interface the encoder output to the transmitter input and receiver output to the decoder input. In the turbo coded system, the channel coding and interfacing modules are integrated into a single module due to the turbo encoder and decoder structures.

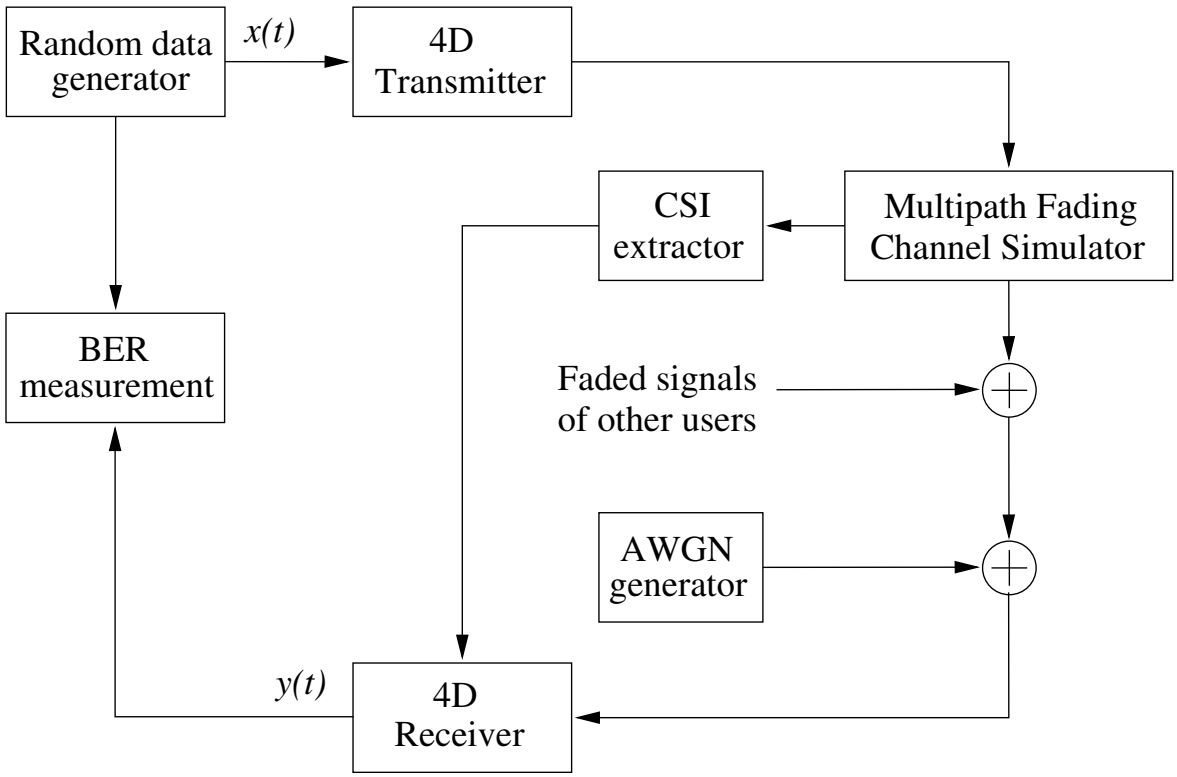

Figure 5.1: Uncoded simulation platform

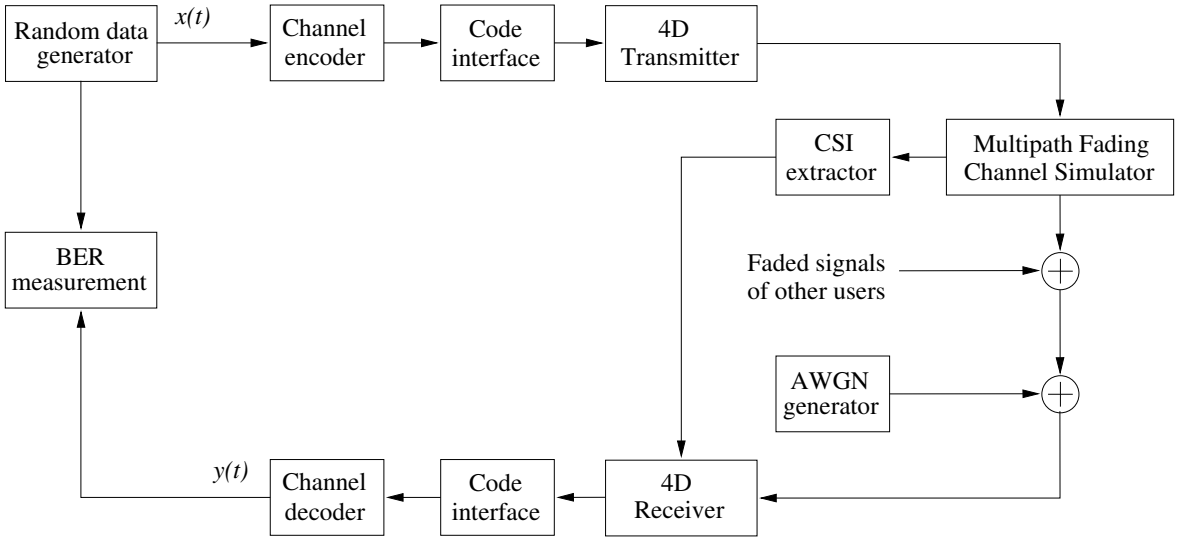

Figure 5.2: Coded simulation platform

5.3 THEORETICAL PERFORMANCE

The bit error probability of uncoded BPSK in an AWGN channel is given by [7]:

$$P_e(bit) = Q\left(\sqrt{\frac{2\varepsilon_b}{N_0}}\right) \tag{5.1}$$

if antipodal signalling is used. In a Rayleigh fading channel the uncoded bit error probability is [7]:

$$P_e(bit) = \frac{1}{2}\left(1 - \sqrt{\frac{(\varepsilon_b/N_0)}{1+(\varepsilon_b/N_0)}}\right) \tag{5.2}$$

The Rayleigh and AWGN BER curves shown in figure 5.3 are respectively the theoretical upper and lower limits of uncoded BPSK error performance. Simulated BER curves for Rician channels with $K = -100$ dB, $K = 3$ dB and $K = 9$ dB using the flat fading channel model presented in section 3.6.2 are shown in figure 5.4.

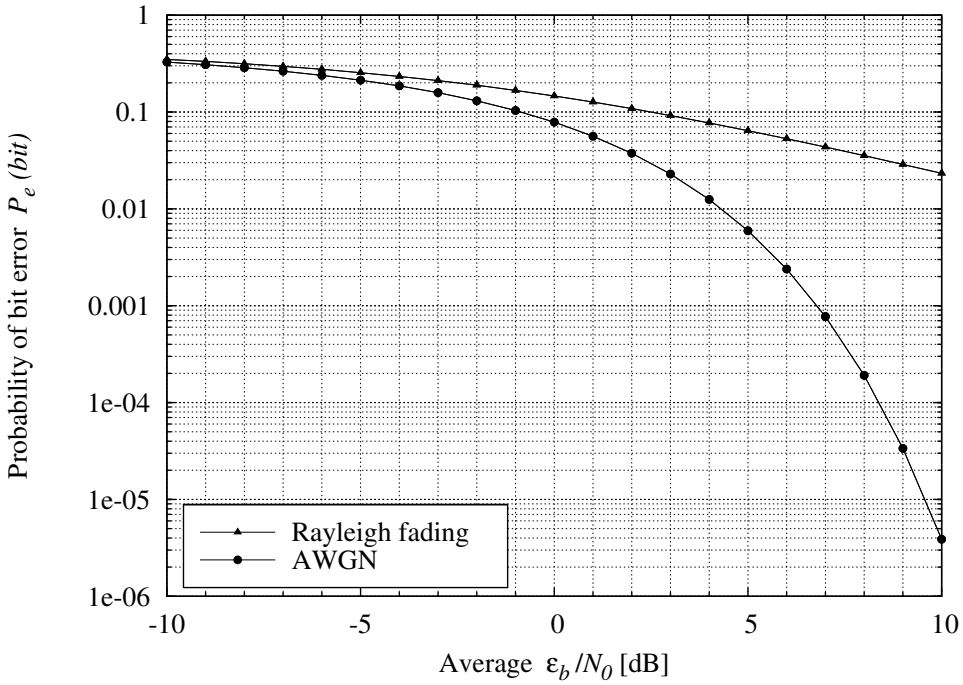

Figure 5.3: Theoretical BER curves

5.3.1 SHANNON CAPACITY LIMIT

Calculation of the theoretical Shannon limit, forming the lower error performance limit, is considered in this section.

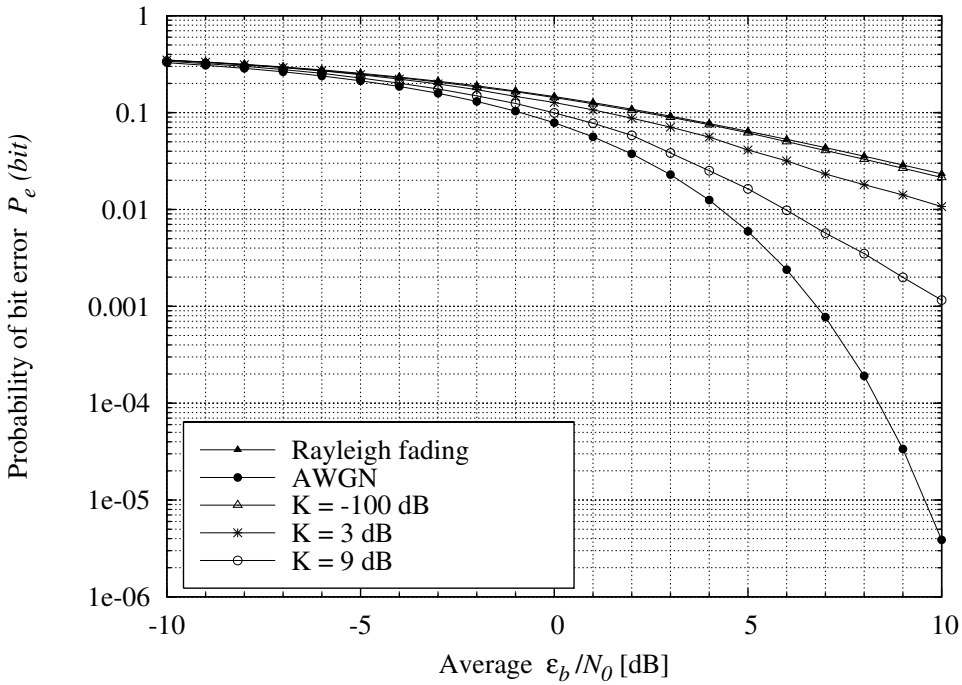

Figure 5.4: Simulated BER curves for different Rician factors with $f_D = 100$ Hz

5.3.1.1 CAPACITY AS A FUNCTION OF SNR PER BIT

The capacity of a binary-input AWGN memoryless channel with antipodal inputs $X = +A$ and $X = -A$ is given as [7]:

$$C = \frac{1}{2}\int_{-\infty}^{\infty} p(y|A)\log_2\frac{p(y|A)}{p(y)}dy + \frac{1}{2}\int_{-\infty}^{\infty} p(y|-A)\log_2\frac{p(y|-A)}{p(y)}dy, \tag{5.3}$$

with

$$p(y|A) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left\{\frac{-(y-A)^2}{2\sigma^2}\right\} \tag{5.4}$$

If antipodal inputs $X = \pm 1$ are used (unity variance $\sigma_x^2 = 1$) the noise variance and $\frac{\varepsilon_b}{N_0}$ are related as [14]:

$$\sigma^2 = \frac{1}{2R_c\,\frac{\varepsilon_b}{N_0}} \tag{5.5}$$

By combining equations 5.3, 5.4 and 5.5, the capacity can be described as a function

$$C = f\left(\frac{\varepsilon_b}{N_0}, R_c\right) \tag{5.6}$$

which can be evaluated numerically.

5.3.1.2 CAPACITY AS A FUNCTION OF ERROR PROBABILITY

Channel capacity can also be related to error probability as [14]:

$$C = R_c[1 - H_2\{P_e(bit)\}], \tag{5.7}$$

with $H_2\{P_e(bit)\}$ the binary entropy function, defined as [14]:

$$H_2(x) = x \ln\left[\frac{1}{x}\right] + (1-x)\ln\left[\frac{1}{(1-x)}\right]. \tag{5.8}$$

5.3.1.3 CAPACITY LIMIT AS BER CURVE

The capacity limit for different code rates can be calculated by matching $\left(P_e(bit), \frac{\varepsilon_b}{N_0}\right)$ pairs using the same capacity value in equations 5.6 and 5.7.

Figure 5.5 shows the Shannon capacity limit as an error probability curve against $\frac{\varepsilon_b}{N_0}$ for different code rates. The code limit curve is calculated in the limit $R_c \to 0$ which corresponds to the theoretical -1.6 dB limit.

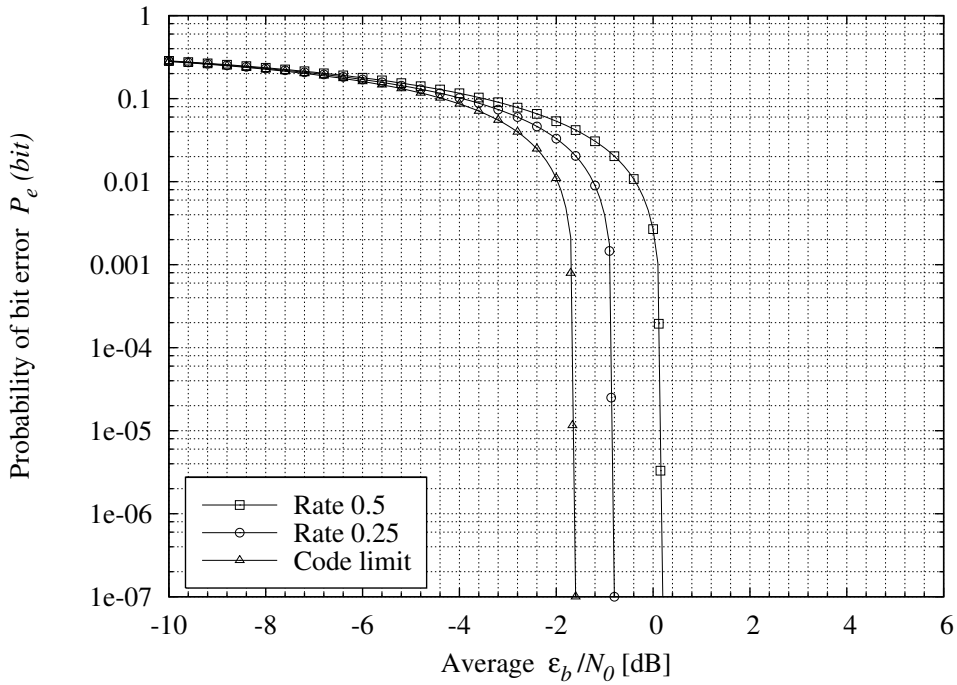

Figure 5.5: Shannon capacity limit for $R_c \to 0$, $R_c = 0.25$ and $R_c = 0.5$

5.4 UNCODED PERFORMANCE

The simulated error performance results of the MD-WCDMA communication system are shown in this section. Both the 2D and 4D systems using both GCL and ZCC CSSs are considered.

5.4.1 2D SYSTEM EMPLOYING CE-GCL-CSS

The AWGN and multipath fading error performance of the uncoded 2D communication system (shown in figures 2.4 and 2.5 in section 2.2.2.1) employing GCL CSSs is shown in figure 5.6. The performance of a single user under AWGN channel conditions is identical to the theoretical AWGN (BPSK) curve with worse performance for additional users as indicated by the 2 users AWGN curve. The performance of a single user under multipath effects is nearly identical to the $K = 9$ dB curve shown in figure 5.4, which is due to the fact that the main path in the multipath channel has a Rician factor $K = 9$ dB as shown in table 3.1 in Chapter 3. Additional CDMA users under multipath conditions cause worse performance approaching the Rayleigh fading curve for 7 concurrent users.

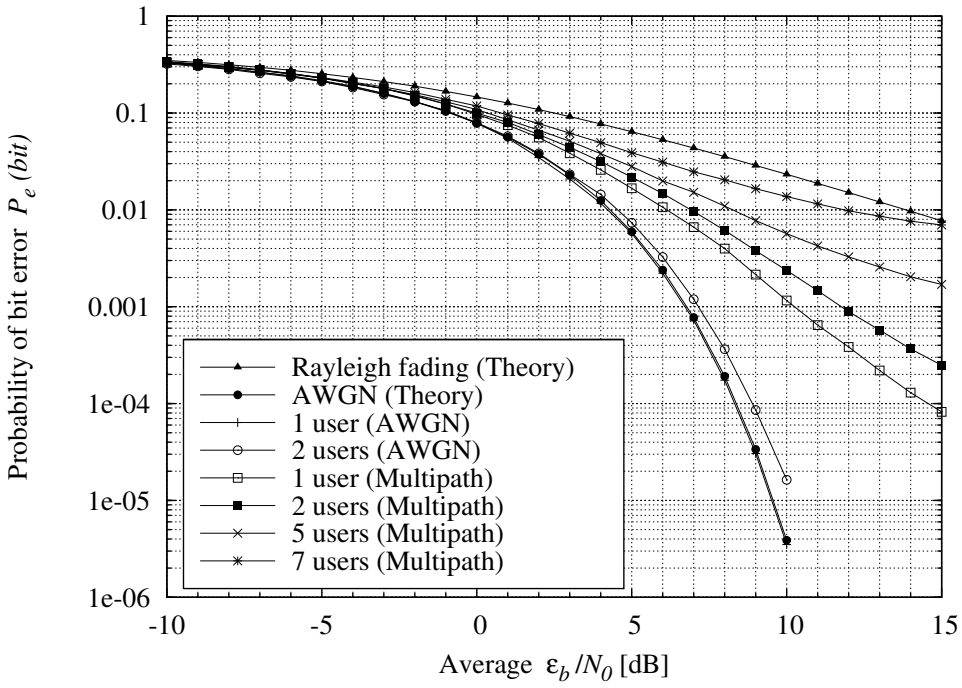

Figure 5.6: Uncoded BER performance for CE-GCL CSSs employed in a 2D communication system

5.4.2 2D SYSTEM EMPLOYING ZCC-CSS

5.4.2.1 AWGN PERFORMANCE

Figure 5.7 shows the uncoded BER performance for 1 and 7 CDMA users for different combinations of ZCC CSSs used in the 2D communication system shown in figures 2.4 and 2.5 in Chapter 2.

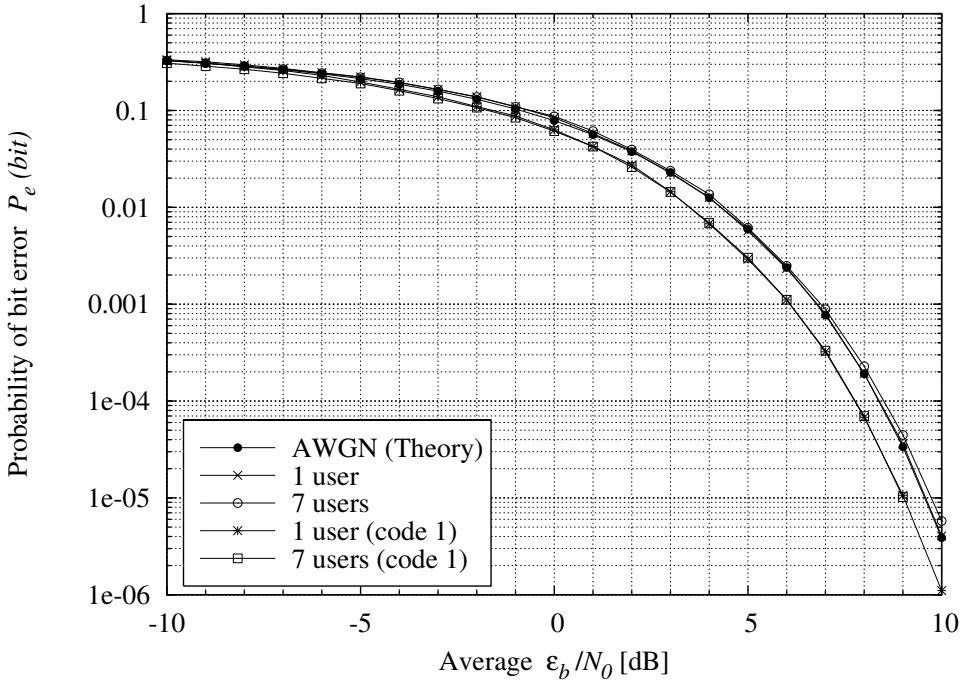

Figure 5.7: Uncoded AWGN BER performance for different combinations of ZCC CSSs employed in a 2D communication system

As shown in figure 5.7, the AWGN performance of all (1 to 7) CDMA users fall within 0.2 dB on the same curve, which is also the theoretical AWGN (BPSK) curve when any reference code other than code 1 (the first CSS of the length-63 ZCC family) is used. If code 1 is used as reference, the BER performance of all users improves slightly (about 0.5 to 1 dB) over the theoretical AWGN (BPSK) curve.

The performance of all users fall closely on the same curve due to the fact that each CSS's ACC with all the other CSSs is near-zero (see figure 2.33 in Chapter 2). The slight improvement over the theoretical AWGN (BPSK) curve, if code 1 is used as reference, can be attributed to code 1's special characteristics including the DC-offset (see figure 2.20) and superior correlation characteristics. By removing code 1's DC-offset

the correlation characteristics are lost and the BER performance is worse. All other ZCC CSSs of length-63 in the family are zero average codes and are therefore rather used than code 1 in the CDMA system. The 3GPP WCDMA standard [77] follows a similar method; only a subset of the orthogonal spreading codes are used as signature sequences to spread data, while those codes excluded due to their inferior spectral and correlation characteristics are used as control sequences.

5.4.2.2 MULTIPATH FADING PERFORMANCE

Figure 5.8 shows the uncoded BER performance for 1 to 7 CDMA users in the 2D communication system employing ZCC CSSs.

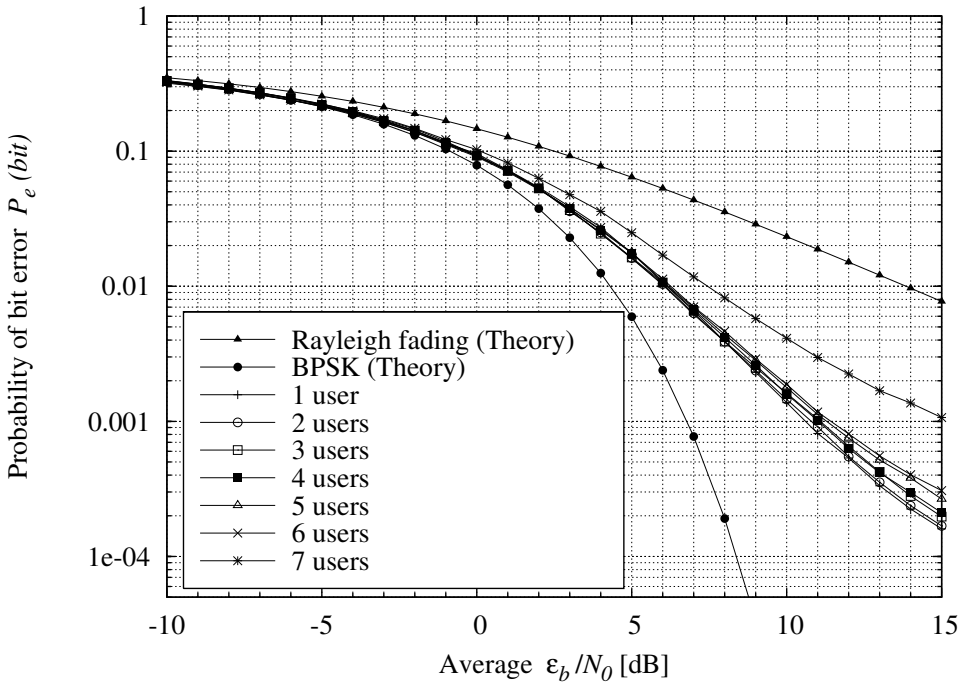

Figure 5.8: Uncoded BER performance for the 2D communication system employing ZCC CSSs

The multipath performance shown in figure 5.8 shows a performance degradation for each additional CDMA user. The multipath channel setup (table 3.1) dictates the performance curve for each additional user. The performance of a single user under multipath conditions is slightly worse for the ZCC CSS than for the GCL CSS (shown in figure 5.6), though for more CDMA users the performance of the ZCC CSS is greatly superior than the GCL CSS.

5.4.3 4D SYSTEM EMPLOYING CE-GCL-CSS

The AWGN and multipath fading performance of the 4D communication system (shown in figures 2.4 and 2.5 in section 2.2.2.1) is shown in figure 5.9.

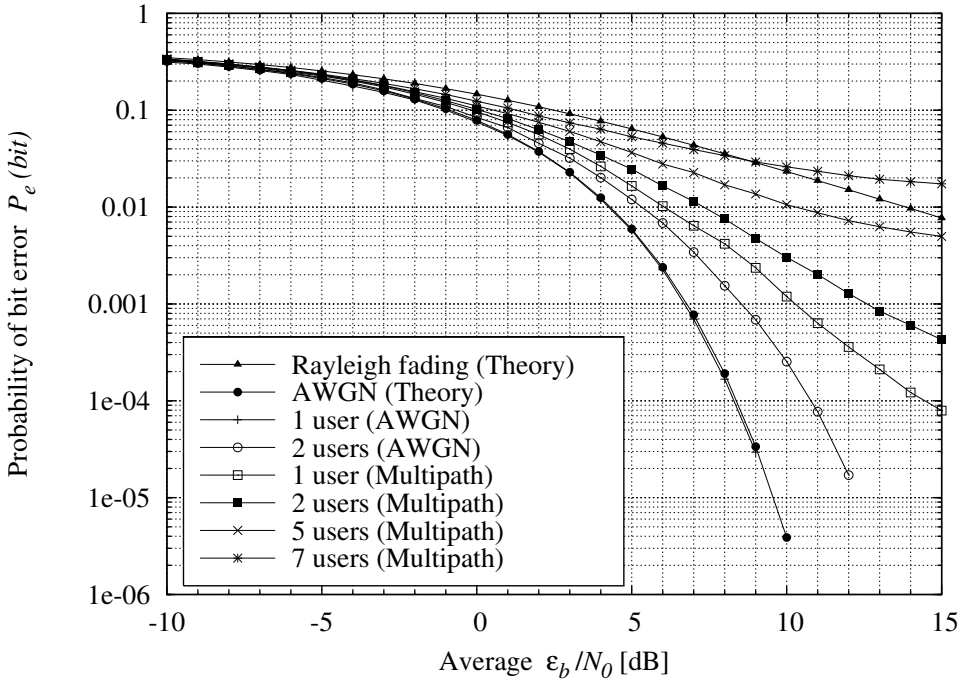

Figure 5.9: Uncoded BER performance for the 4D communication system employing CE-GCL CSSs

The performance of the 4D system shown in figure 5.9 is similar to the performance of the 2D system shown in figure 5.6. Again, the performance of a single user under AWGN channel conditions is identical to the theoretical AWGN (BPSK) curve with worse performance for additional users as indicated by the 2 users AWGN curve. The performance of a single user under multipath effects for the 4D system is nearly identical to the 2D system. Additional CDMA users under multipath conditions cause worse performance and performance even worse than the Rayleigh fading curve for 7 concurrent users.

5.4.4 4D SYSTEM EMPLOYING ZCC-CSS

5.4.4.1 AWGN PERFORMANCE

Figure 5.10 shows the uncoded BER performance for different combinations of ZCC CSSs used in the 4D communication system shown in figures 2.9 and 2.10 in Chapter 2.

The performance results can be understood by considering the correlation comparison graph shown in figure 2.34 in section 2.5.3.6.

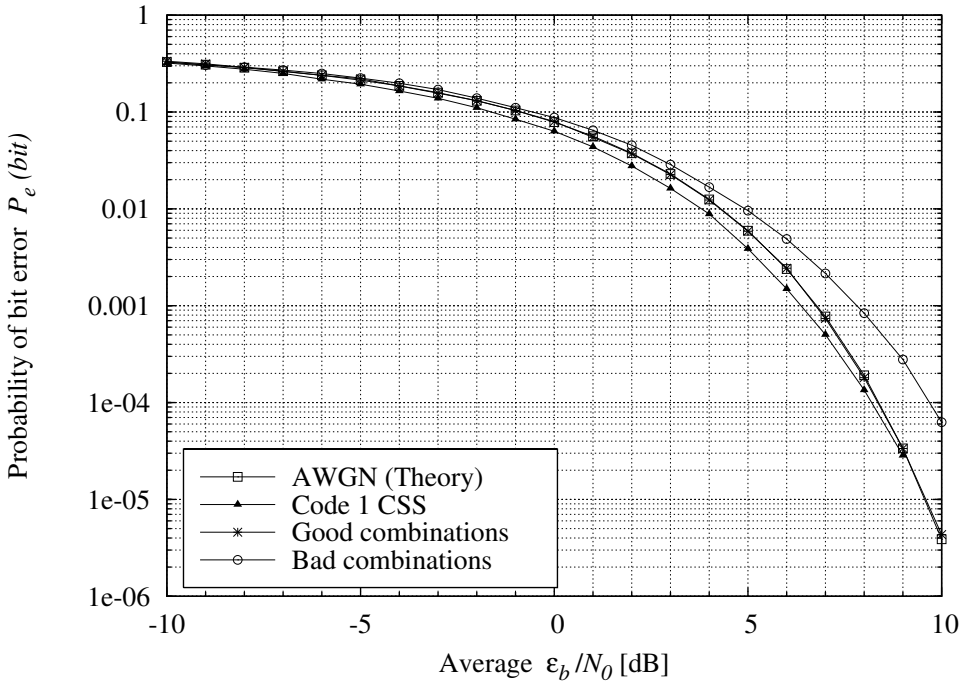

Figure 5.10: Uncoded BER performance for different combinations of ZCC CSSs employed in a 4D communication system

Good combination CSSs are code combinations of which the ACC functions are nearly zero (excluding code 1). Using the length-63 ZCC CSS family, each reference CSS has 5 potential good combinations of CSS codes and the performance of $1-5$ users (each using one of the good combination CSSs) fall on the same curve (on the theoretical AWGN (BPSK) curve) as shown in figure 5.10.

Bad combination CSSs are code combinations of which the ACC functions are significantly greater than zero (with a correlation factor of about 0.17 - see figure 2.34) resulting in worse BER performance. By using any 2 CSSs with a non-zero ACC the BER performance are about 1.5 dB worse than when using good combination CSSs.

Bad combination CSSs include the sets $(2,7)$, $(3,6)$ and $(4,5)$. By avoiding these sets, theoretical BPSK performance is guaranteed for all users. Alternatively, a reference CSSs must be used which will not be used in conjunction with a CSSs forming a bad combination, e.g. code 2 can be used as reference with codes 3-6 assigned to 4 other

CDMA users and user 1 (with code 2) will then experience BPSK performance for 1 - 5 concurrent CDMA users.

5.4.4.2 MULTIPATH FADING PERFORMANCE

Figure 5.11 shows the uncoded BER performance for 1 - 7 CDMA users using ZCC CSSs in a multipath fading environment in the 4D communication system shown in figures 2.9 and 2.10 in Chapter 2.

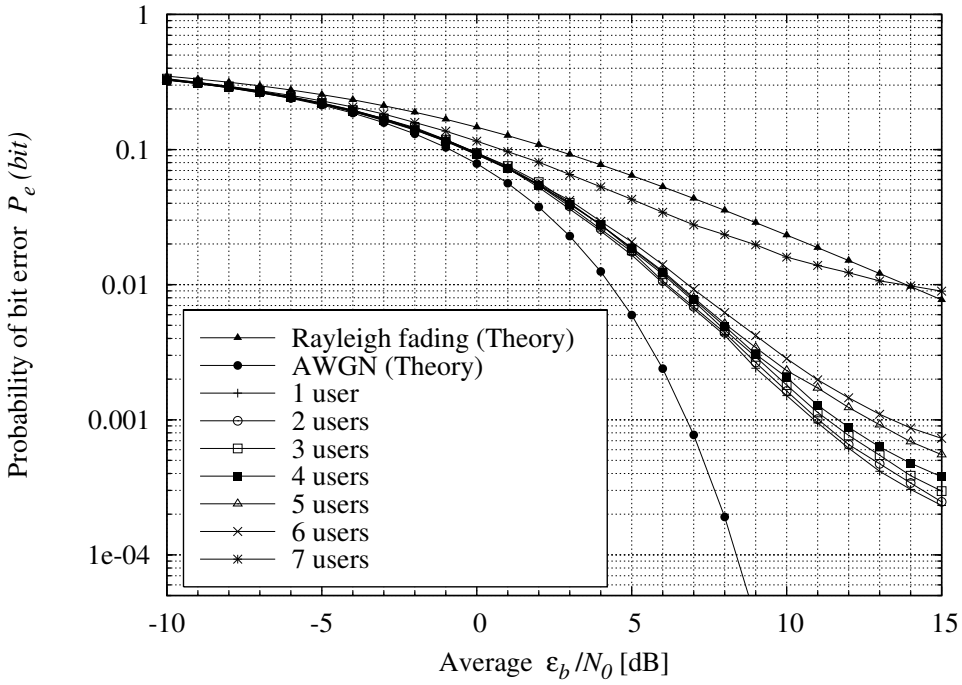

Figure 5.11: Uncoded BER performance for 1-7 CDMA users in the 4D system using ZCC CSSs on a multipath fading channel

The 4D performance results shown in figure 5.11 are similar to the 2D results shown in figure 5.8 and were obtained by using ZCC CSS code 2 as the reference code. The performance degrades slightly for each additional CDMA user for the first 6 users. The performance of 7 users is significantly worse due to the bad CSS combination (code 2 and code 7).

5.5 CODED PERFORMANCE

The performance of sparse graph codes, including Turbo, LDPC and RA codes, evaluated on the 4D communication platform employing ZCC CSSs is the main focus of

this chapter and is considered in this section.

5.5.1 TURBO CODES

This section contains the performance results of mainly the 3D Turbo code construction presented in section 4.4 in Chapter 4. Constituent codes used in the Turbo coding scheme include the RM (8,4) and (12,6) codes. The (12,6) code is not an RM code according to the definition of RM codes (see section 4.5.1) but is an extension of the RM (8,4,4) code by adding two rows and four columns to $\mathbf{G_{sys}}$ in equation 4.20 such that the matrix consist of an identity matrix and an all-ones matrix with zeros along the diagonal from lower left to upper right corners.

5.5.1.1 TURBO CODE PERFORMANCE IN AN AWGN CHANNEL

Figure 5.12 shows the simulated error performance of the RM (8,4,4) turbo code in a narrowband AWGN channel. Two and three dimensional code constructions are used and the code is decoded iteratively (the performance of 1 and 10 decoding iterations are shown) with the SOVA decoding algorithm explained in section 4.5.3. The performance of a 1D Viterbi soft decision decoded RM (8,4) code is also shown. Performance gains due to iterative decoding (considering the improvement from 1 to 10 decoding iterations) vary between 0.5 and 1 dB for both 2D and 3D turbo coding schemes. The 3D coding scheme has a performance gain of about 0.4 dB above the 2D scheme and the overall performance improvement from the 1D to 3D scheme is 1.5 dB.

Figure 5.13 shows the performance of the RM(8,4,4) 3D turbo coding scheme employing ZCC CSSs for 1 and 5 users and 1 to 10 decoding iterations. The performance of the coding schemes on the wideband system is identical to the performance on the narrowband system shown in figure 5.12. After 2 decoding iterations no significant additional performance gains are achieved. At lower ε_b/N_0 values, the performance gains due to increasing decoding iterations are larger and for higher ε_b/N_0 values the performance curves merge towards the single iteration case. As for the uncoded error performance curves, there is no difference between the performance for a single or 5 CDMA users.

Figure 5.14 shows the performance of the (12,6) 3D turbo coding scheme employing ZCC CSSs for 1 to 5 users and 1, 2, 5 and 10 decoding iterations. The performance of all 5 CDMA users fall on the same curve and each bundle of curves is identified only

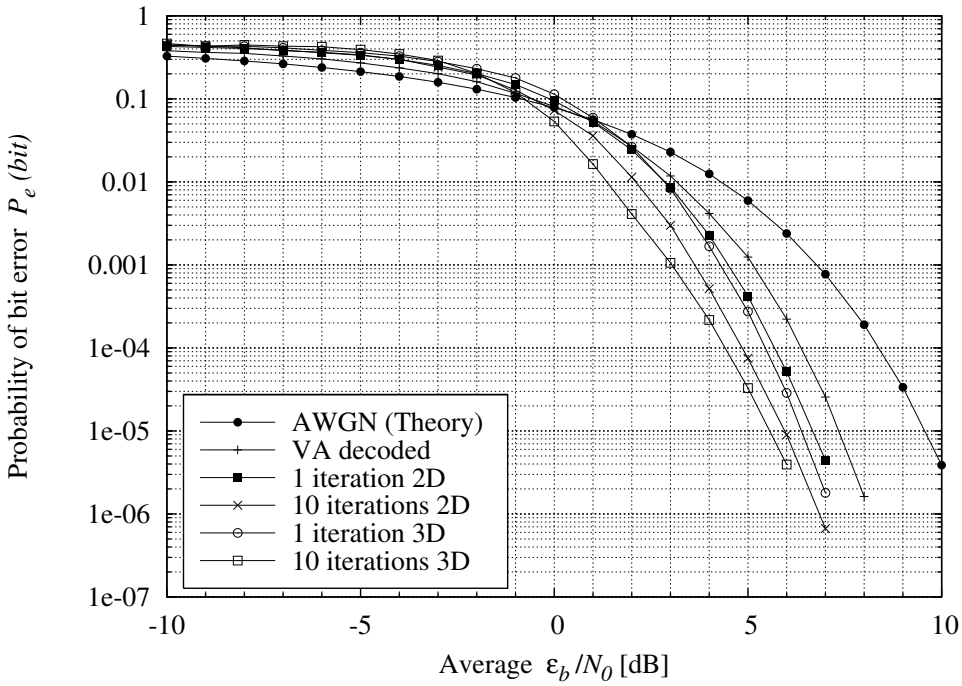

Figure 5.12: AWGN performance of the 2D and 3D RM(8,4,4) TC (Narrowband)

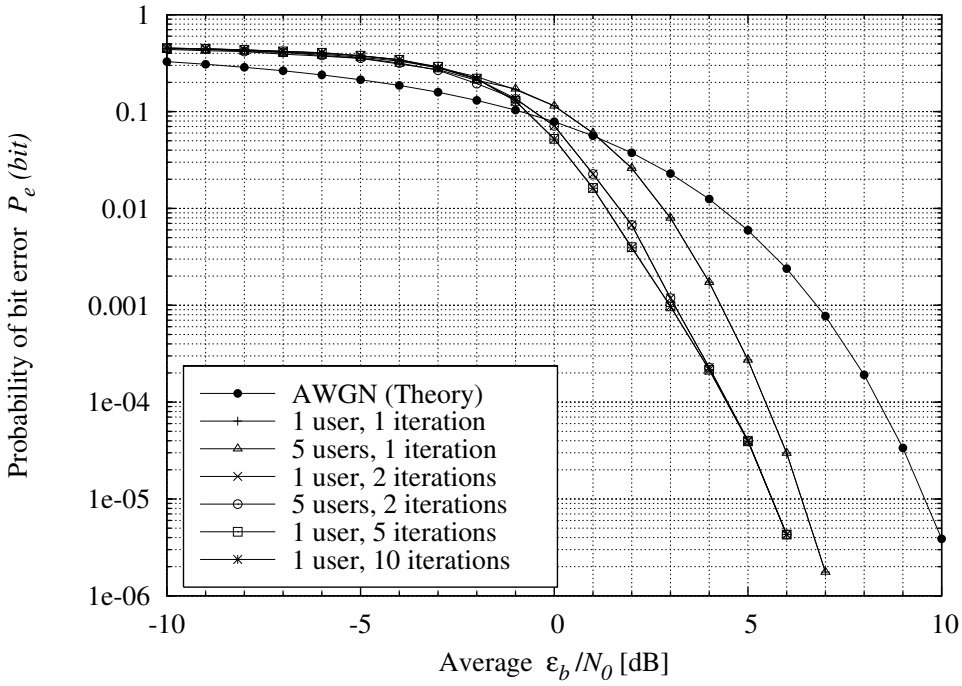

Figure 5.13: AWGN performance of the 3D RM(8,4,4) TC employing ZCC CSSs

by the number of decoding iterations. The Shannon limit for a rate-0.25 code is also shown in the figure. The performance improvement from 1 to 10 decoding iterations is about 2 dB, though after 5 iterations no significant additional gains are achieved. The same phenomenon as in figure 5.13 occurs in figure 5.14; for low ε_b/N_0 values the increasing decoding iteration curves have larger performance differences and for high ε_b/N_0 values the differences decrease and finally merge into the same curve.

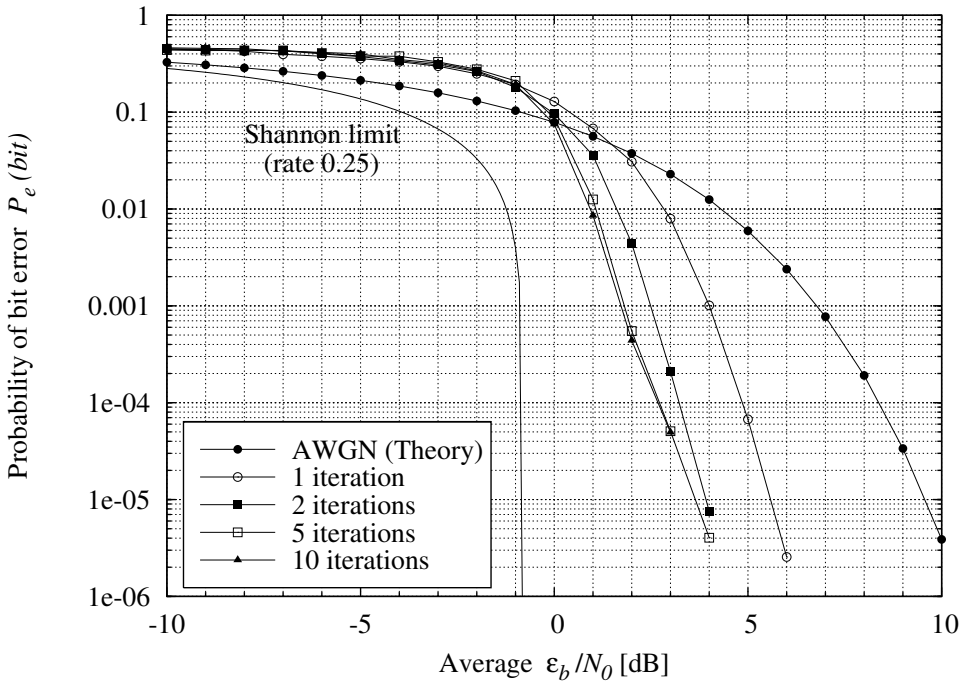

Figure 5.14: AWGN performance of the 3D RM(12,6) TC employing ZCC CSSs

5.5.1.2 TURBO CODE PERFORMANCE IN A MULTIPATH CHANNEL

Figure 5.15 shows the simulated error performance results of the RM(8,4,4) 3D turbo coding scheme employing ZCC CSSs for 1 and 5 users in a multipath frequency selective fading channel with setup given in table 3.1 in Chapter 3. The performance curves for 1 and 10 decoding iterations are given.

Similarly figure 5.16 shows the performance of the (12,6) 3D turbo coding scheme employing ZCC CSSs for 1 and 5 users in a multipath fading channel with performance curves for 1, 2, 5 and 10 decoding iterations.

The multipath performance results in figures 5.15 and 5.16 are respectively similar to the AWGN performance results given in figures 5.13 and 5.14 but slightly worse due to channel condition differences. The relative path strengths of the multipath channel play a significant role in the error performance of the communication system creating a Rician multipath channel. In this case the Rician factor of the main path bundle is $K = 9$ dB (see table 3.1 in Chapter 3) and the error performance can therefore be expected to be slightly worse than the AWGN curve (see figure 5.4).

From figures 5.11, 5.15 and 5.16 it can be observed that the multipath error performance difference from 1 to 5 CDMA users in the uncoded system is reduced (practically removed) in the coded system.

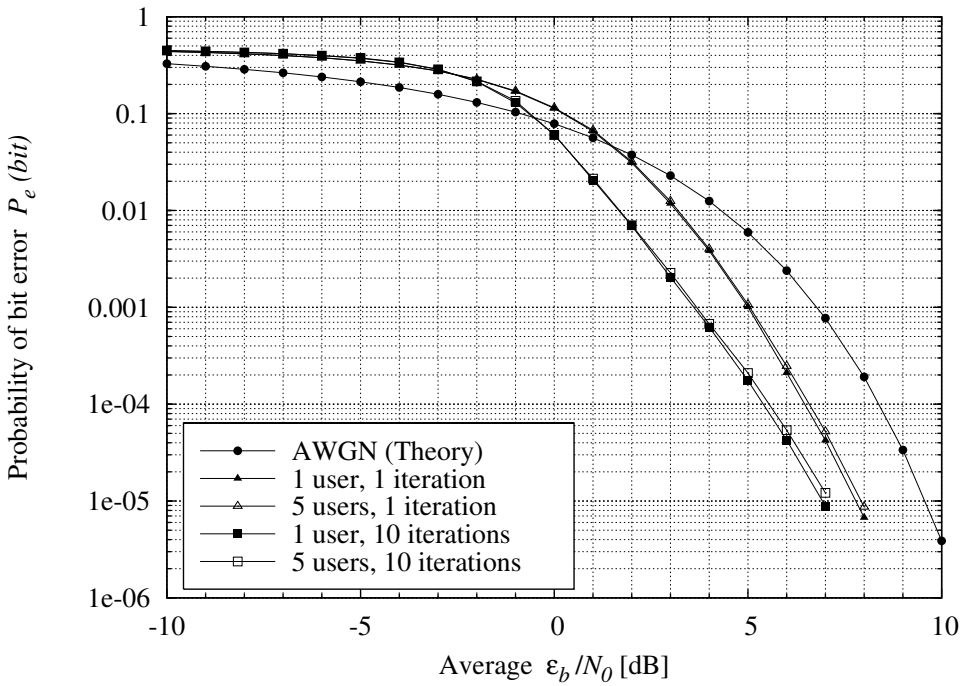

Figure 5.15: Multipath performance of the 3D RM(8,4,4) TC employing ZCC CSSs

5.5.2 LDPC CODES

5.5.2.1 LDPC CODE PERFORMANCE IN AN AWGN CHANNEL

Figure 5.17 shows the performance of the regular $(36, 18)$ LDPC coding scheme (with factor graph shown in figure 4.11 in Chapter 4) employing ZCC CSSs for 1 to 5 users

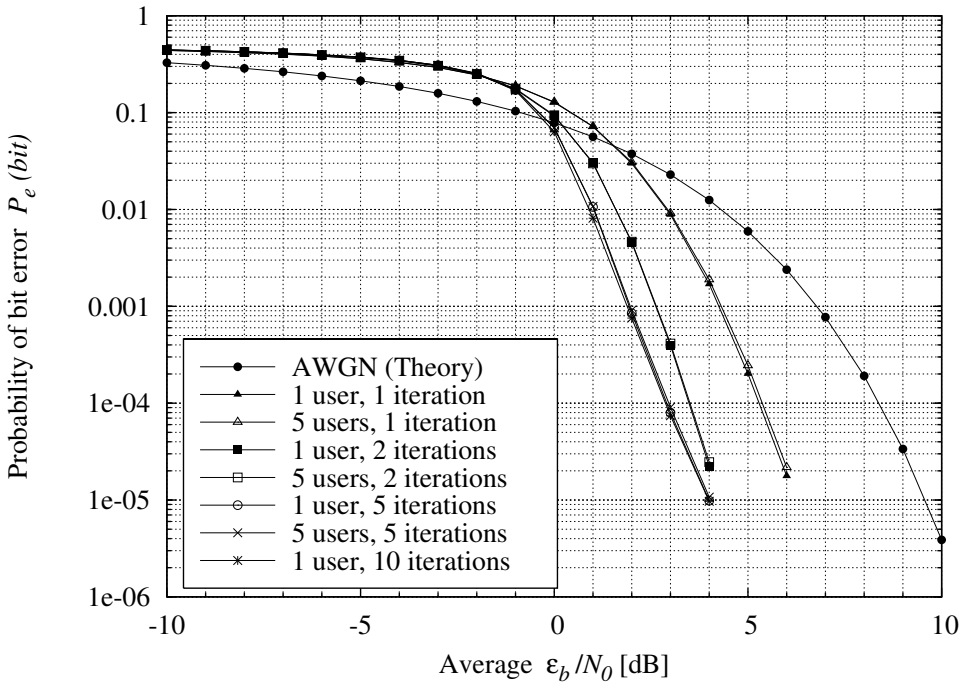

Figure 5.16: Multipath performance of the 3D (12,6) TC employing ZCC CSSs

in an AWGN channel for 1, 2, 10 and 100 decoding iterations. For each iteration curve shown, the performance for 1 to 5 CDMA users is identical. The performance of a single iteration is only slightly better than the uncoded system, though at $P_e(bit) = 10^{-6}$ a performance gain of about 3 dB is evident for 100 decoding iterations.

Figures 5.18 and 5.19 respectively show the performance of the approximately regular (256,128) and (256,64) LDPC coding schemes employing ZCC CSSs for 1 and 5 users in an AWGN channel for 1, 10 and 100 decoding iterations. The half rate code shows an improvement of nearly 5.5 dB at $P_e(bit) = 10^{-6}$, whereas the quarter rate code only shows about a 5 dB gain above the uncoded system. After a single iteration the quarter rate code performs much worse than the half rate code and even worse than the uncoded system, though as the decoding iterations increase the two codes show nearly the same performance results.

5.5.2.2 LDPC CODE PERFORMANCE IN A MULTIPATH CHANNEL

Figures 5.20, 5.21 and 5.22 respectively show the simulated error performance results of the (36,18), (256,128) and (256,64) LDPC coding schemes employing ZCC CSSs for 1

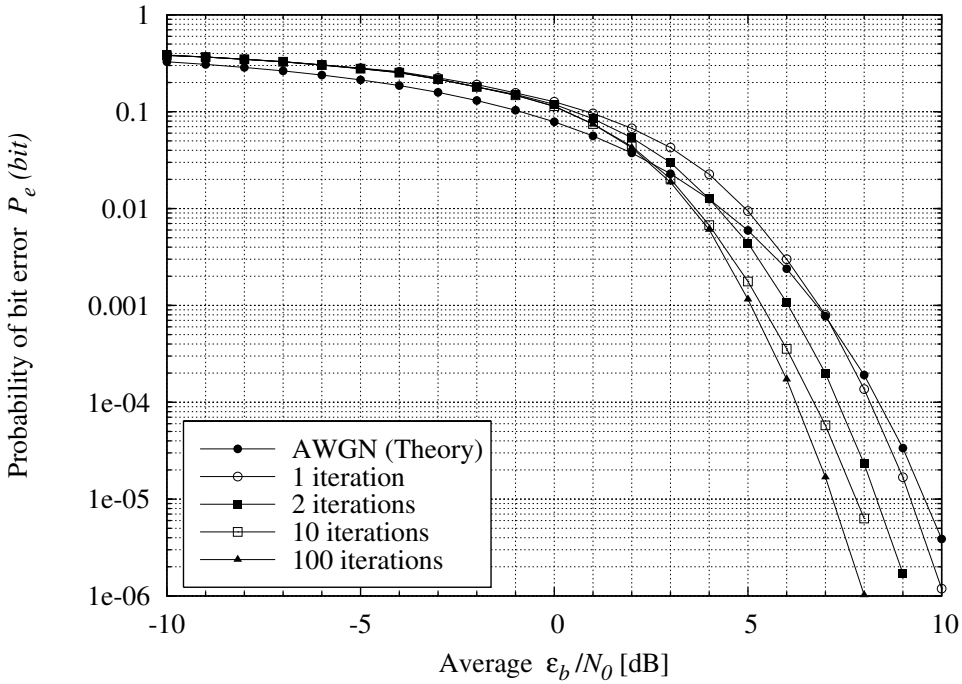

Figure 5.17: AWGN performance of the (36,18) LDPC code employing ZCC CSSs

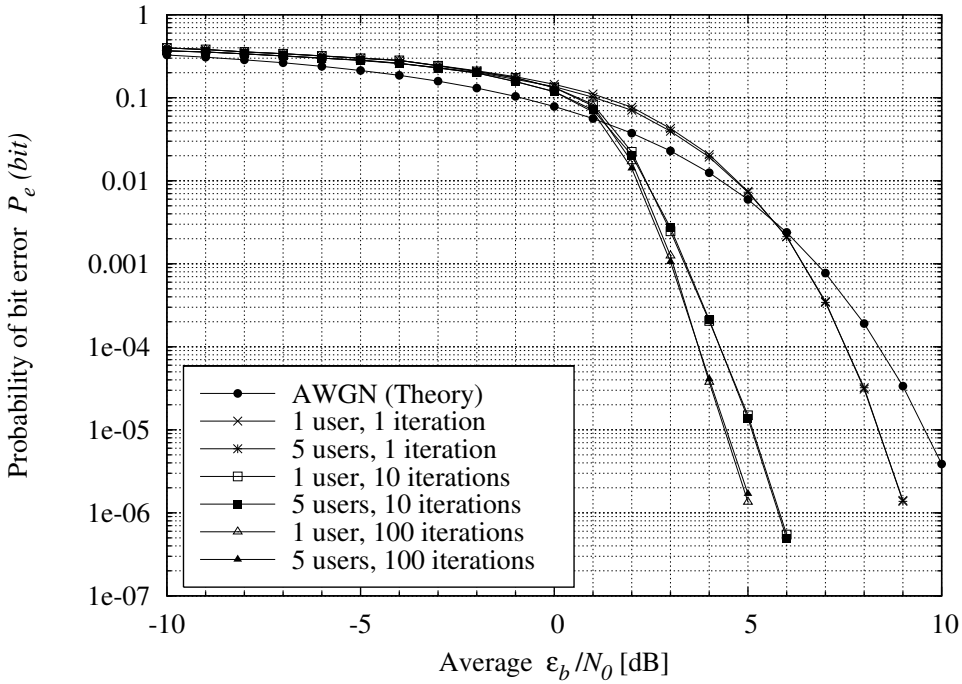

Figure 5.18: AWGN performance of the (256,128) LDPC code employing ZCC CSSs

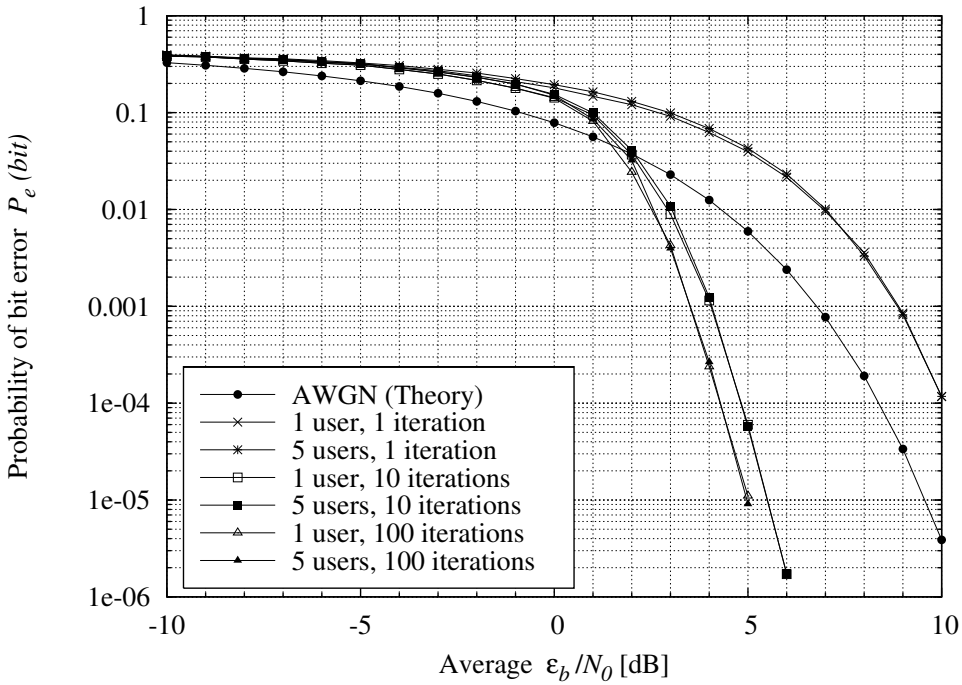

Figure 5.19: AWGN performance of the (256,64) LDPC code employing ZCC CSSs

and 5 users in a frequency selective fading channel for 1, 10 and 100 decoding iterations. The performance results are similar but slightly worse than the AWGN performance results shown in figures 5.17, 5.18 and 5.19.

5.5.2.3 THE EFFECT OF FADING AMPLITUDE IN LDPC CODES

Figures 5.23 and 5.24 show the performance of the regular (36,18) and (256,64) LDPC coding schemes employing ZCC CSSs for 1 user in a frequency selective fading channel for 1 and 100 decoding iterations to illustrate the effect on performance when the CSI fading amplitude is not used in the decoder. If knowledge of the fading amplitude is not available (though phase information is still available) or not used in the decoder the performance degradation can be as much as 0.5 dB.

5.5.3 RA CODES

5.5.3.1 RA CODE PERFORMANCE IN AN AWGN CHANNEL

Figure 5.25 shows the performance of the (36,18) RA coding scheme employing ZCC CSSs for 1 to 5 users in an AWGN channel for 1, 10 and 100 decoding iterations. For

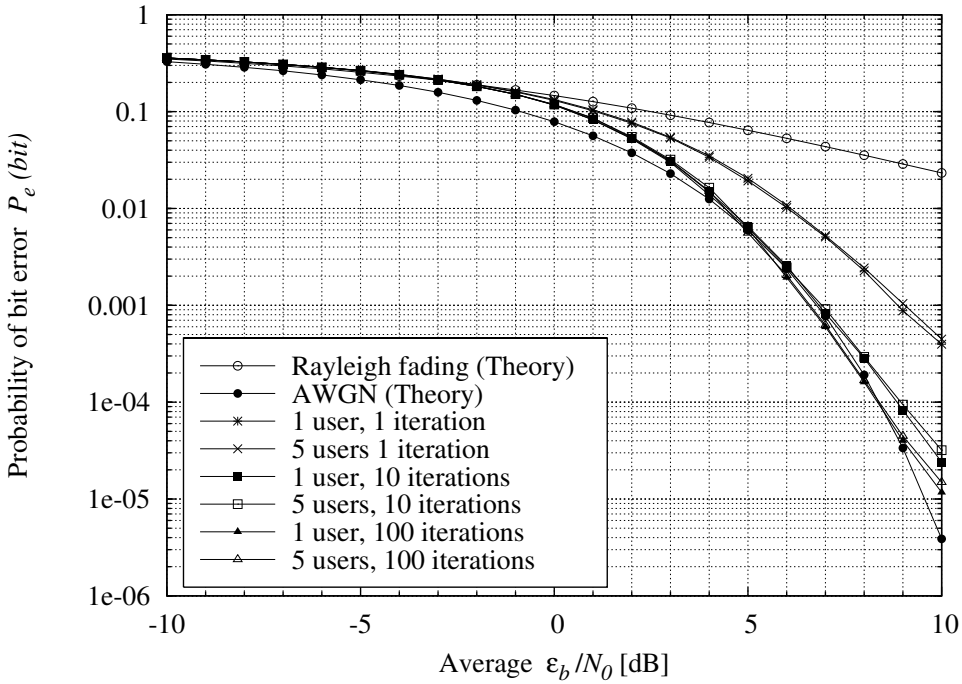

Figure 5.20: Multipath fading performance of the (36,18) LDPC code employing ZCC CSSs

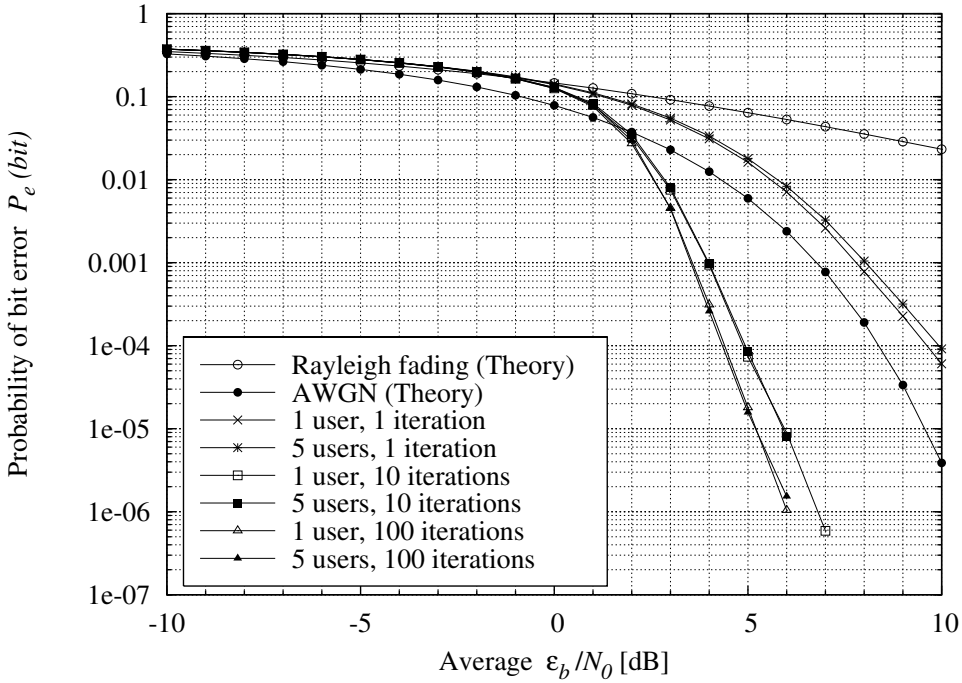

Figure 5.21: Multipath fading performance of the (256,128) LDPC code employing ZCC CSSs

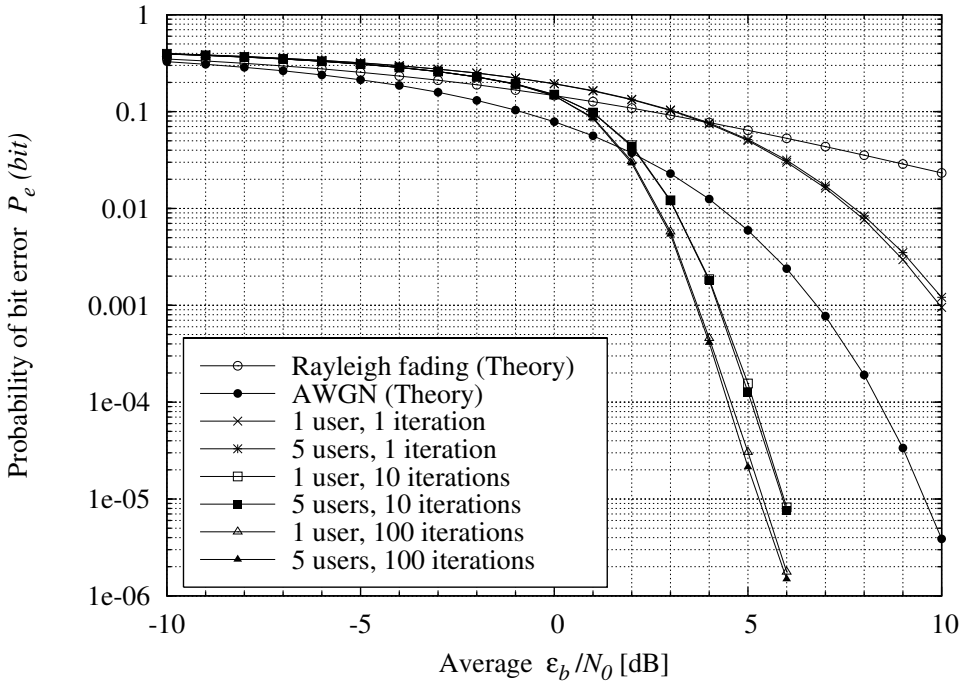

Figure 5.22: Multipath fading performance of the (256,64) LDPC code employing ZCC CSSs

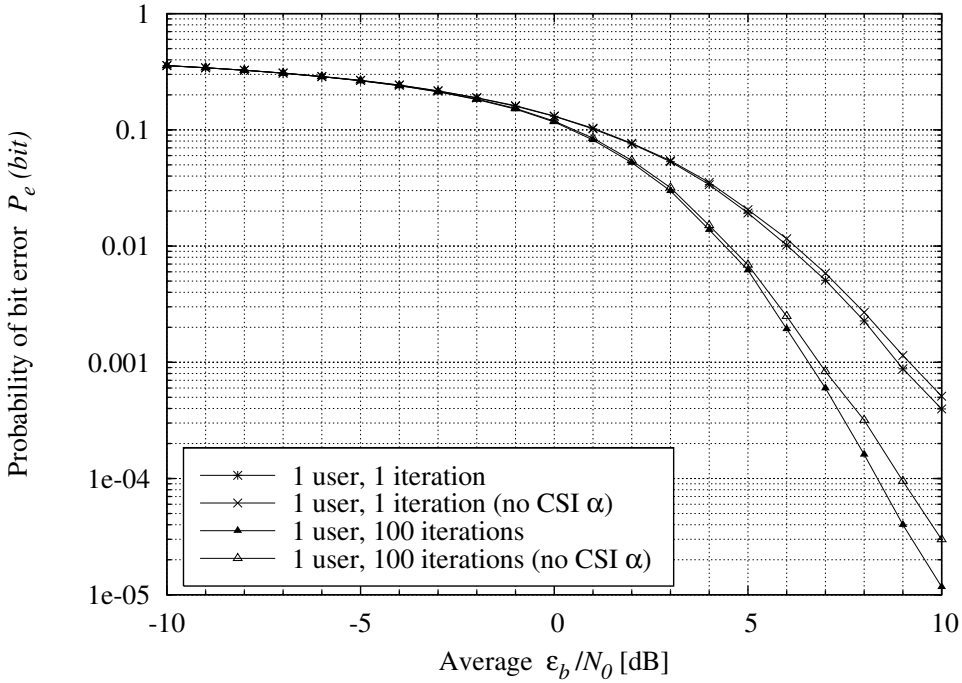

Figure 5.23: Multipath fading performance of the (36,18) LDPC code illustrating the importance of CSI fading amplitude

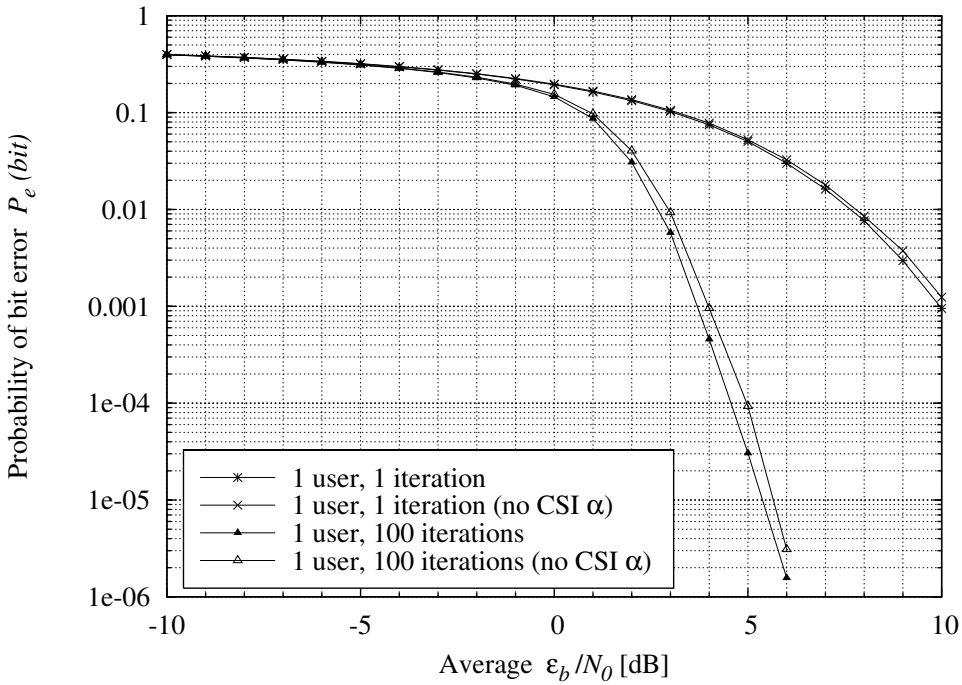

Figure 5.24: Multipath fading performance of the (256,64) LDPC code illustrating the importance of CSI fading amplitude

each iteration curve shown, the performance for 1 to 5 CDMA users is identical. The performance of a single iteration is slightly worse than the uncoded system and after 10 iterations no additional performance gains are achieved. At $P_e(bit) = 10^{-6}$ a performance gain of approximately 1.1 dB is evident after 10 decoding iterations.

Figures 5.26 and 5.27 respectively show the performance of the regular (256,128) and (256,64) RA coding schemes employing ZCC CSSs for 1 and 5 users in an AWGN channel for 1, 10 and 100 decoding iterations. For a single iteration, both the half rate and quarter rate codes show worse performance than the uncoded system. The half rate code exhibits error floor behaviour and only shows improved performance over the uncoded system from 1.2 to 9.5 dB for 10 decoding iterations. The quarter rate code shows superior performance over the half rate code with a performance gain of approximately 4.7 dB above the uncoded system at $P_e(bit) = 10^{-6}$ for 100 decoding iterations. No error floor is visible in the error probability range shown for the quarter rate code.

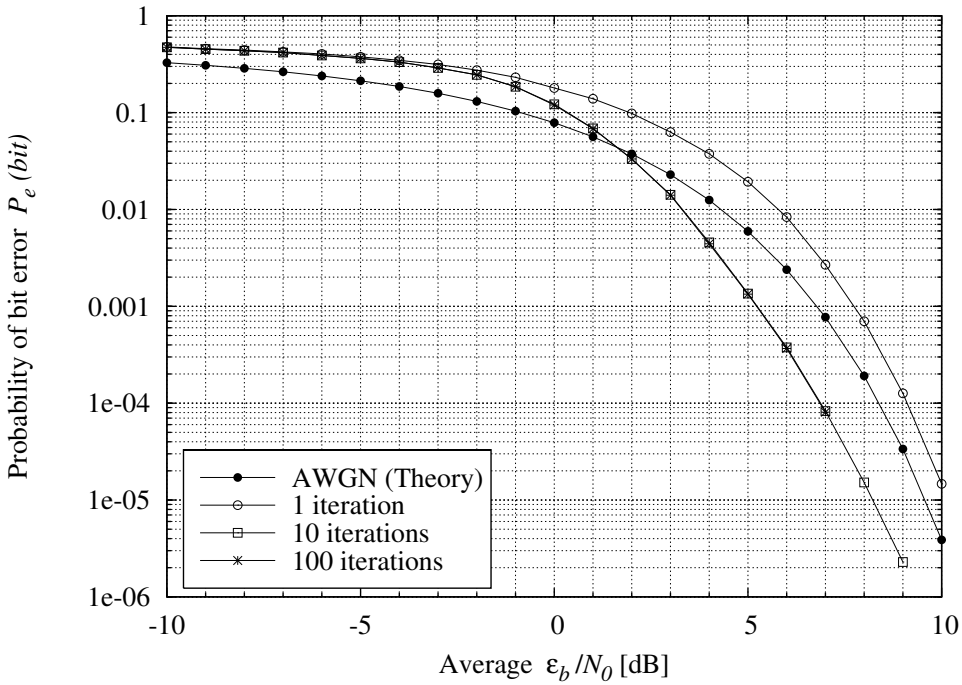

Figure 5.25: AWGN performance of the (36,18) RA code employing ZCC CSSs

5.5.3.2 RA CODE PERFORMANCE IN A MULTIPATH CHANNEL

Figures 5.28, 5.29 and 5.30 respectively show the simulated error performance results of the (36,18), (256,128) and (256,64) RA coding schemes employing ZCC CSSs for 1 and 5 users in a frequency selective fading channel for 1, 10 and 100 decoding iterations. The performance results are similar but worse than the AWGN performance results shown in figures 5.25, 5.26 and 5.27.

5.5.3.3 THE EFFECT OF FADING AMPLITUDE IN RA CODES

Figure 5.31 shows the performance of the (256,64) RA coding scheme employing ZCC CSSs for 1 user in a frequency selective fading channel for 1 and 100 decoding iterations to illustrate the effect on performance when the CSI fading amplitude is not used in the decoder. For 100 decoding iterations the performance degradation is between 0.4 and 0.5 dB.

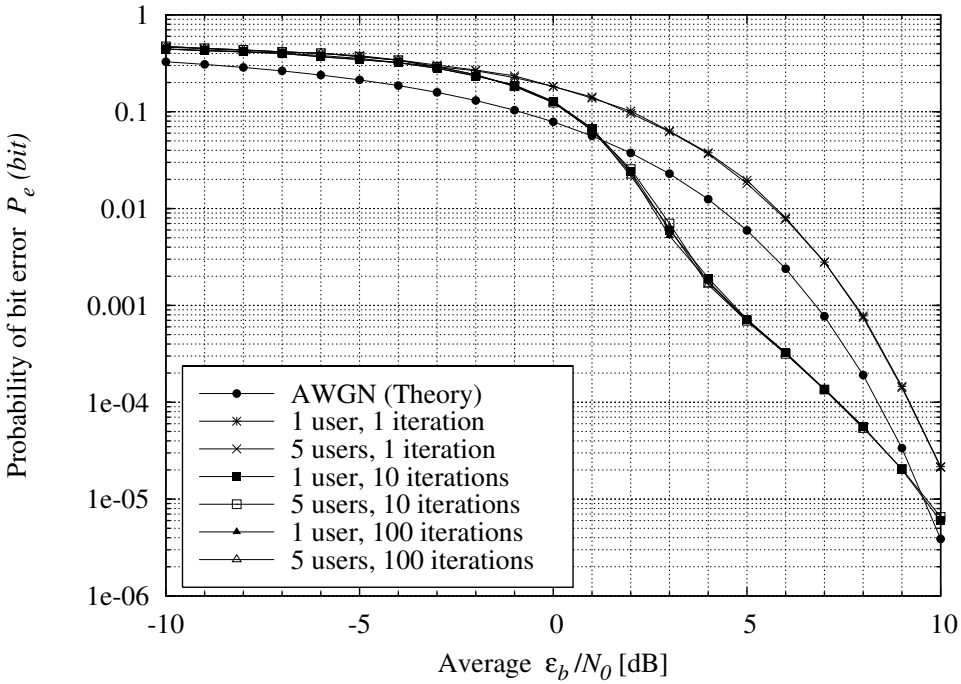

Figure 5.26: AWGN performance of the (256,128) RA code employing ZCC CSSs

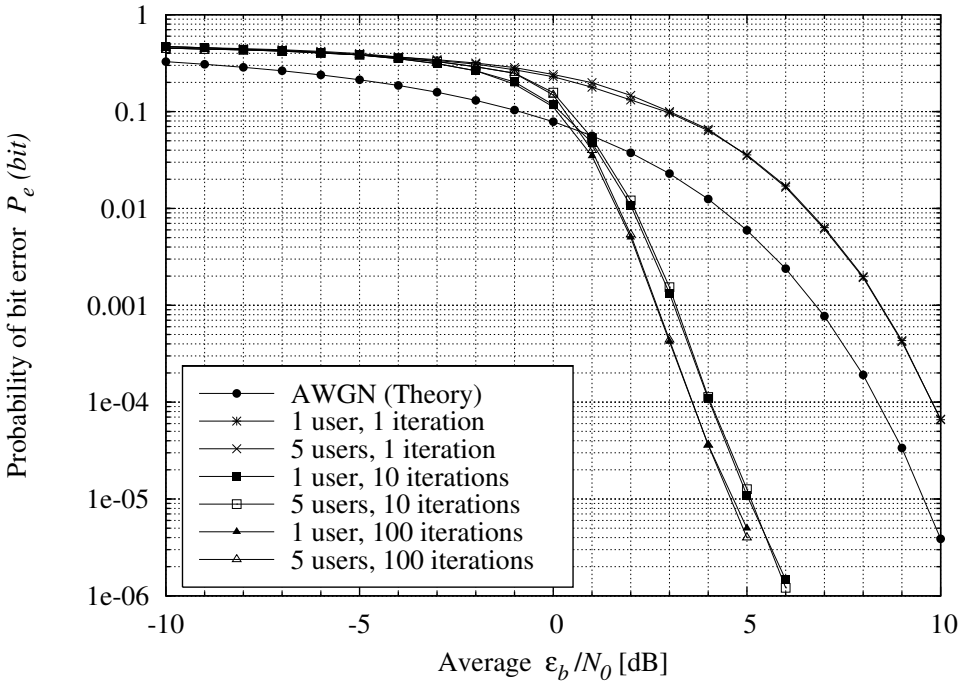

Figure 5.27: AWGN performance of the (256,64) RA code employing ZCC CSSs

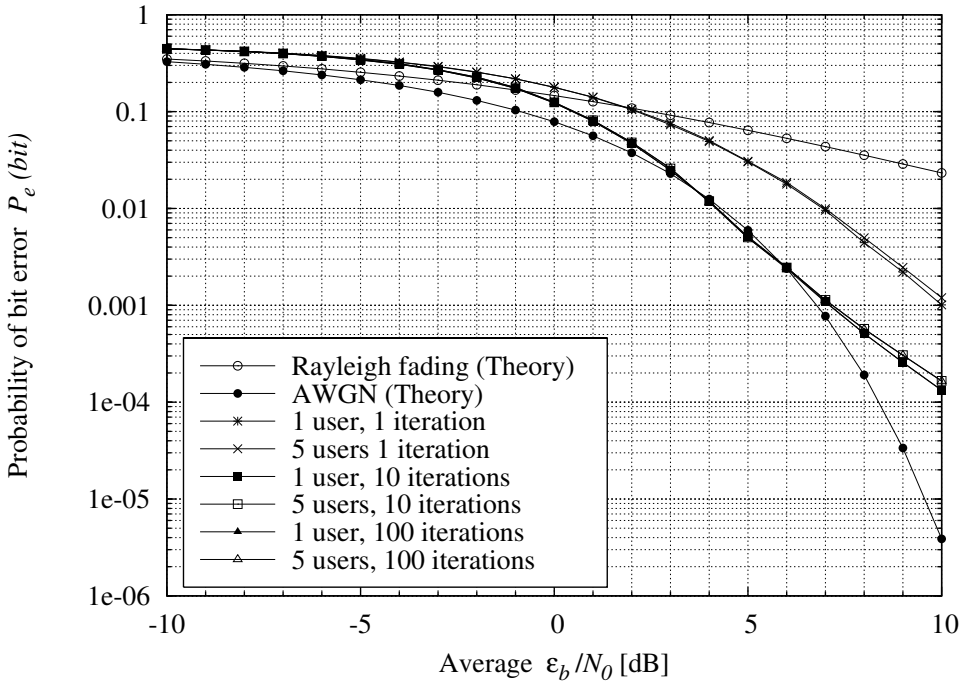

Figure 5.28: Multipath fading performance of the (36,18) RA code employing ZCC CSSs

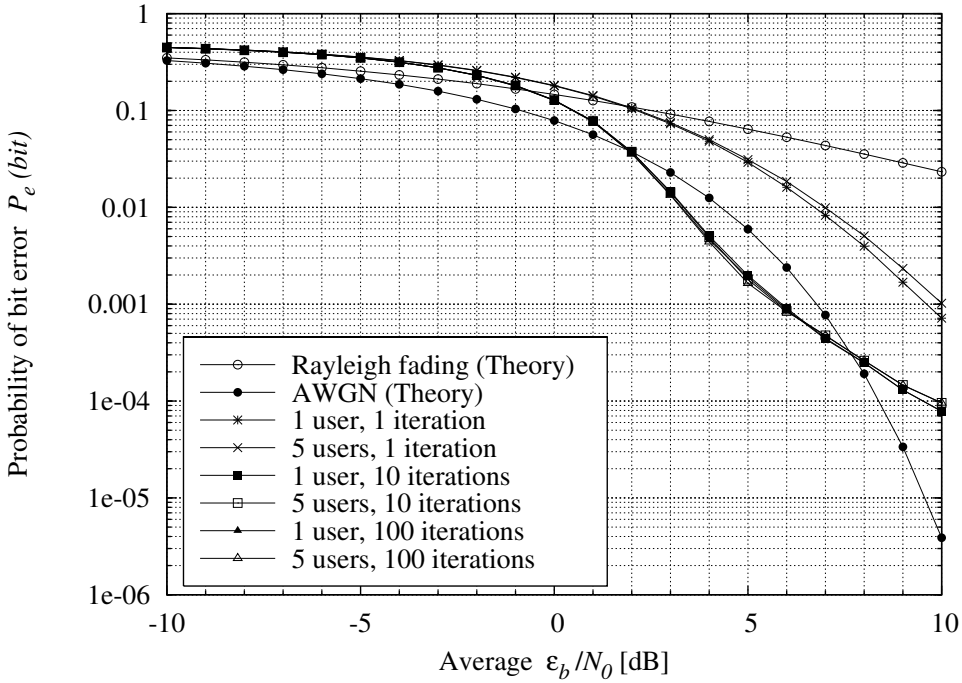

Figure 5.29: Multipath fading performance of the (256,128) RA code employing ZCC CSSs

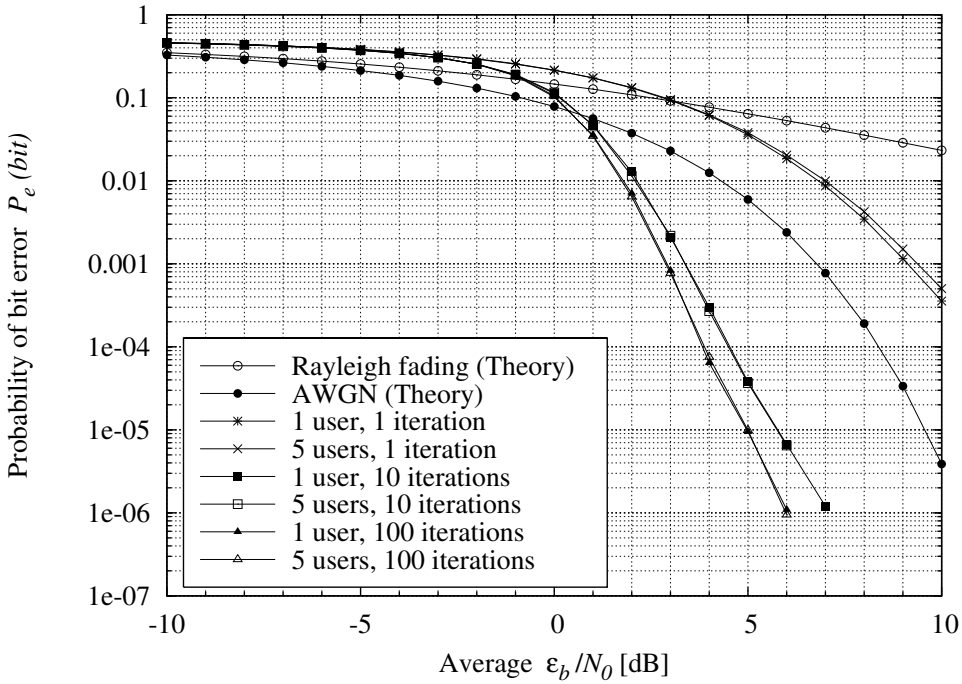

Figure 5.30: Multipath fading performance of the (256,64) RA code employing ZCC CSSs

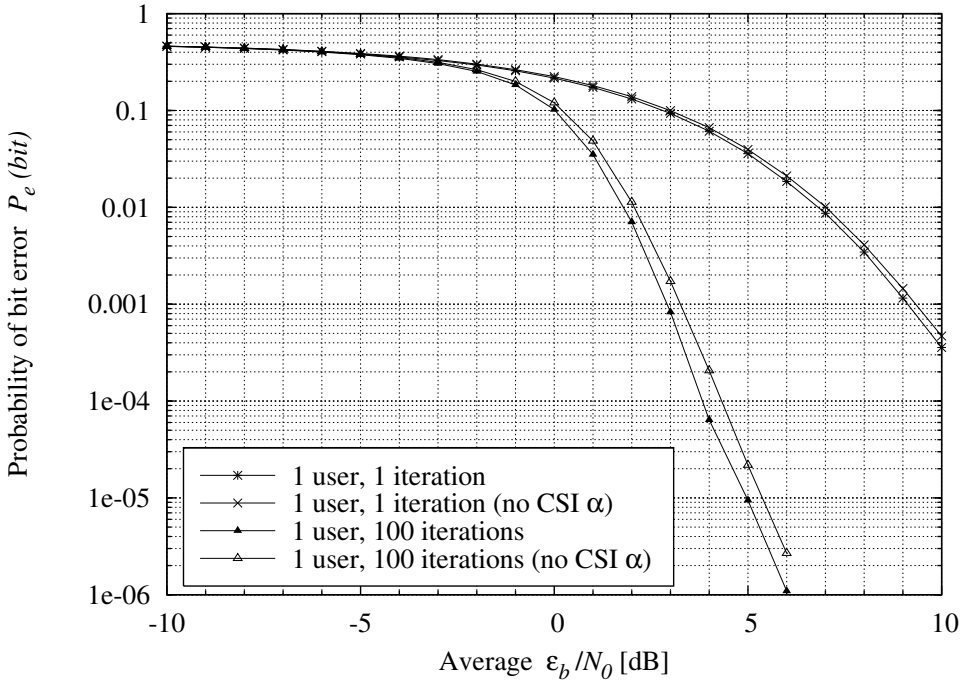

Figure 5.31: Multipath fading performance of the (256,64) RA code illustrating the importance of CSI fading amplitude

5.5.4 COMPARING CODING SCHEMES

5.5.4.1 AWGN PERFORMANCE

Figure 5.32 shows the simulated AWGN error performance results of the uncoded 4D system (from figure 5.10), the 3D turbo code with the RM (8,4,4) constituent code (which is effectively a (256,64) code) for 10 decoding iterations (from figure 5.13), the LDPC (256,64) code for 100 decoding iterations (from figure 5.19) and the RA (256,64) code for 100 decoding iterations (from figure 5.27). All curves shown are the single user error performance curves (which are identical to the performance curves for up to 5 CDMA users) using ZCC code 2 as reference code.

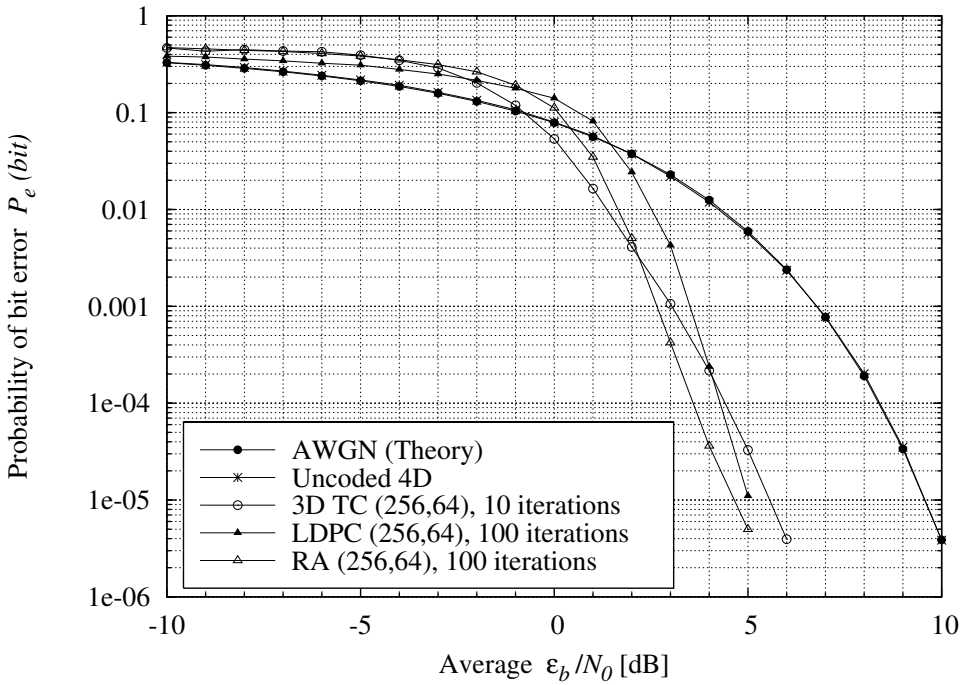

Figure 5.32: AWGN performance comparison of (256,64) coding schemes

5.5.4.2 MULTIPATH FADING PERFORMANCE

Figure 5.33 shows the simulated multipath fading error performance results of the uncoded 4D system for 1 and 5 users (from figure 5.11), the 3D turbo code with the RM (8,4,4) constituent code for 1 and 5 users with 10 decoding iterations (from figure 5.15), the LDPC (256,64) code (from figure 5.22) and the RA (256,64) code (from figure 5.30). The curves for the LDPC and RA codes shown in figure 5.33 represent the performance curves for 1 to 5 users with 100 decoding iterations each.

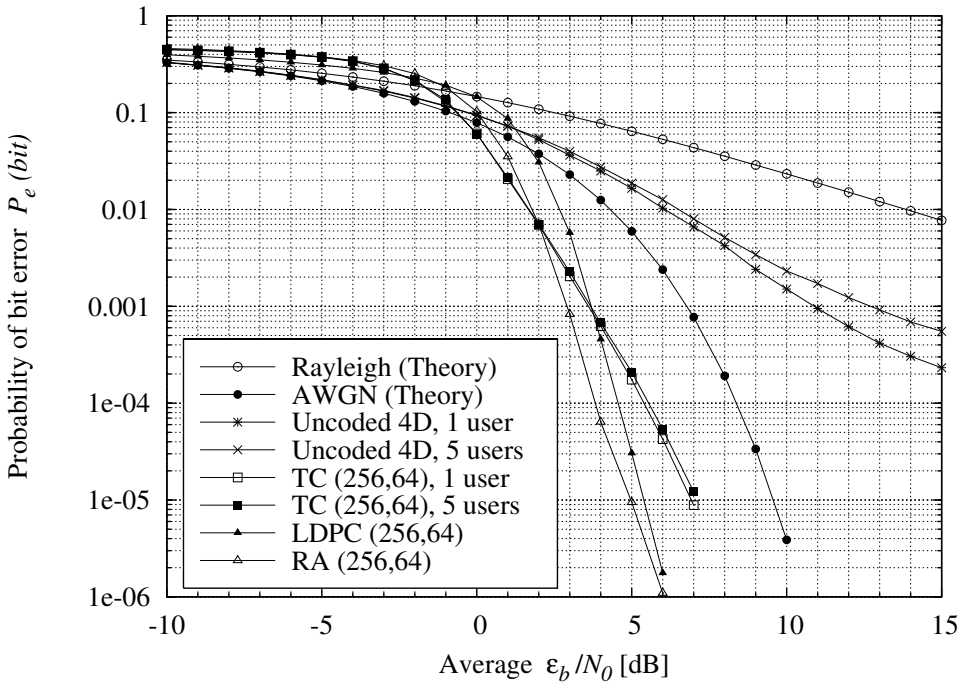

Figure 5.33: Multipath performance comparison of (256,64) coding schemes

5.5.4.3 ERROR PERFORMANCE COMPARISON

The 3 coding schemes show comparable error performance results (within 2 dB), though at low ε_b/N_0 values (around 0 dB) the turbo code performs a little better than the other two coding schemes and at 3 dB and higher the LDPC and RA codes outperform the turbo code. The RA code shows the best overall error performance. It is also clear that the turbo code requires less decoding iterations than the other coding schemes, though different decoding structures and algorithms (with differing levels of complexity) are employed for each scheme (see Chapter 4).

CHAPTER 6

CONCLUSIONS

6.1 CHAPTER SUMMARY

This chapter contains a discussion of all the major conclusions drawn from studying the WCDMA platform and CSSs, the channel simulator, sparse graph codes and the simulated error performance results. The research performed during the course of this study is critically evaluated and possible areas for future investigation are briefly explored.

Probably the most important conclusion summarising this chapter and the dissertation is as follows. MLM (multi-layered-modulation) can be used to transmit parallel data streams with improved spectral efficiency compared with single-layer modulation, providing data throughput rates proportional to the number of modulation layers at performances equivalent to single-layer modulation. Alternatively, multiple modulation layers can be used to transmit coded information to achieve improved error performance at throughput rates equivalent to a single layer system.

6.2 WCDMA PLATFORM

The main conclusions drawn from studying and developing the WCDMA platform and CSSs presented in Chapter 2 are considered in this section.

6.2.1 TRANSMITTER AND RECEIVER STRUCTURES

The WCDMA platform consists of dual-channel QPSK-like MD transmitter and receiver structures. Both 2D (section 2.2) and 4D systems (section 2.3) were evaluated mathematically and the actual implementable structures were presented. The main conclusions are given below.

- A dual-channel QPSK-like system can transmit a CSS, with the real and imaginary sequence components (or combinations thereof) respectively modulating the

inphase and quadrature carriers, creating a 2D SS system.

- Two 2D systems in parallel form a 4D system where the four dimensional space is created by all possible combinations of CSS components and quadrature carriers.
- The basic 4D (or 2D) system can be expanded to create an MDM system with $4m$ (or $2m$) dimensions, by placing m 4D (or 2D) systems in parallel. The maximum value of m is limited by the length and type of CSS and the number of sequences in the family employed.
- Simple correlation-type receivers can be used as MD demodulators, with performance depending on basis vector orthogonality and carrier phase synchronisation.

6.2.2 COMPLEX SPREADING SEQUENCES

Two classes of CSSs, including SO-GCL CSSs (derived from Chu sequences) and ZCC CSSs (derived from PS sequences), were evaluated in section 2.4 in terms of correlation and spectral characteristics for implementation on the MDM system. Important findings are listed below.

- RU-CSSs can be filtered using complex phase interpolation to obtain a CE signal with minimum Nyquist bandwidth (using linear interpolation creating GCL from Chu CSSs) or near-Nyquist bandwidth (using non-linear interpolation creating ZCC from PS CSSs) with the complex rotation direction determining upper or lower sideband spectrum elements (both CSS classes considered in this study rotate unrestrained in both directions on the complex unity circle and are hence DSB signals).
- Chu CSSs exhibit near-ideal AAC (a single correlation peak at $\tau = 0$ with small side lobes around zero) and perfect PAC characteristics (zero correlation for all time shifts except at $\tau = 0$). When complex components are considered separately, the AAC and PAC functions are slightly worse (the single correlation peak at $\tau = 0$ is about 50 % smaller and the sidelobes are slightly larger when compared with the complex AC characteristics), though the ACC and PCC between complex components of the same CSS are zero at $\tau = 0$. The ACC and PCC between different CSSs are non-ideal and have non-zero values (approximately 13 % of the complex main peak) at $\tau = 0$.

- RUF-GCL CSSs show AAC and PAC with the same main characteristics as Chu CSSs, though GCL CSSs have larger sidelobes and the complex PAC is non-ideal, though near-perfect. The ACC and PCC of the GCL CSSs show nearly identical characteristics.

- ZCC CSSs exhibit the most severe non-ideal AAC and PAC characteristics of all the CSSs considered in this study, making ZCC CSSs the most timing-sensitive sequences (very accurate synchronisation is required to avoid ambiguities in the demodulator). The AAC shows the maximum correlation peak at $\tau = 0$ with several large peaks periodically spaced on either side of $\tau = 0$. The same correlation graph structure is observed at the component AAC functions with a 50 % reduction in correlation amplitude (as is the case with Chu and GCL CSSs). The PAC is similar to the AAC, though all the large correlation sub-peaks are equal in amplitude to the main peak at $\tau = 0$.

- The multiple AAC and PAC peaks are expected when observing the periodic nature of ZCC CSSs in time (see figure 2.20) and these peaks are a trade-off providing zero cross correlation characteristics, hence the name ZCC CSSs. The ACC exhibits zero-values at $\tau = 0$ (see figure A.6) and the PCC shows zero-values for all time shifts for all combinations of CSS components considered (see figure A.8), making ZCC CSSs excellent candidates for employment in synchronous CDMA systems.

- The AAC and ACC values are important (especially at time shift $\tau = 0$) and affect error performance of the communication system, while PAC determines timing recovery and symbol synchronisation (though not considered in this study as perfect synchronisation was assumed).

6.2.3 MD BASIS FUNCTIONS

In order to predict performance, the behaviour of the uncoded communication system was evaluated by studying the synchronous correlation, dot product functions and output power of the basis functions in multidimensional space. The MD space is created by combining orthogonal components of CSSs and quadrature carriers. The orthogonality and power efficiency of the MD basis functions were considered in section 2.5 and some major conclusions are provided below.

- Since quadrature carriers are perfectly orthogonal, the orthogonality between basis functions is dictated by the ACC characteristics between the complex components of a given CSS. In a multi-user environment, the orthogonality between all complex components (real and imaginary sequences considered separately) of all the CSS family members employed determines the performance of the CDMA system.
- The average and instantaneous powers of the MD transmitted signal depend on the orthogonality between basis vectors (see equations 2.51 and 2.54). Both GCL and ZCC CSSs display near perfect (constant) instantaneous output power due to the point-by-point orthogonality between basis vectors. ZCC CSSs show perfect orthogonality over every symbol period for all family members (except code 1 which displays a small ACC value between the real and imaginary components) and therefore have constant average output powers (for all codes in the family). GCL CSSs exhibit significant ACC values between components of every CSS and therefore show varying average output powers for different CSSs (see table A.2).
- The MD-SO-WCDMA transmitted signal can be described as a vector traversing the surface of an M-dimensional (with $M = 4$ for the 4D system) hypersphere, by noting that the complex envelope of the basis functions are constant (due to the point-by-point orthogonality of the basis functions) and that the instantaneous and average output powers are constant.
- ZCC CSSs exhibit zero ACC at $\tau = 0$ between all complex sequences in the family (see figure 2.33), indicating that ZCC CSSs are perfectly complementary pairs (sidelobes cancel perfectly) and making them ideal for use in the 4D system. However, only certain combinations of ZCC component codes show non-zero ACC values (see figure 2.34). By choosing the correct subset of ZCC CSSs, the same performance can therefore be obtained using components of CSSs as with complete CSSs. GCL CSSs show much worse ACC values between both complete and component CSSs.

6.3 CHANNEL SIMULATION MODELS

This section includes conclusions drawn from studying and developing the wideband channel model presented in Chapter 2 to simulate AWGN and multipath effects, including frequency selective fading and Doppler spreads.

6.3.1 CHANNEL MODELLING

The channel simulator presented in sections 3.6 and 3.7 is based on mathematical channel models and equations given in sections 3.2 to 3.5. Conclusions are provided below.

- The presented channel simulator is a generic expandable multipath channel model with each path a flat fading channel with a unique Rician factor and Doppler spread.
- The power delay profile of the channel is adjusted by setting each of the parallel flat fading channels' relative time delay and power scaling factor. By adjusting the power delay profile, the channel can be set up as frequency selective or non-selective.
- The channel used in this study was set up to model a 3-path frequency selective (wideband) fading channel with statistically independent paths as shown in table 3.1, assuming a static power delay profile (i.e. each path's relative time delay and power scaling factor is constant for the duration of performance measurement though the instantaneous phase and fading amplitude of the channel are dynamic due to changes in the instantaneous Doppler frequency).
- In a CDMA environment, several users may allocate the same channel and bandwidth in relative close proximity concurrently, though each user may experience unique channel effects depending on exact geographical location, surroundings and relative motion between the transmitter and receiver. The channel conditions were therefore set up individually for each user (see table 3.1), though the overall system performance was based on one user's performance only (see figure 3.6).

6.3.2 CHANNEL MODEL VALIDATION

The channel model is validated in section 3.8 by comparing each module of the channel with theoretical descriptions. The following statements and the uncoded error performance results given in Chapter 5 (see section 6.5 below) indicate that the channel model functions according to theory and is therefore a valid model fit for use in error performance analysis.

- It is shown in figures 3.7 and 3.8 that the AWGN model in figure 3.2 functions correctly according to the theoretical PDF given in equation 3.1.
- The theoretical amplitude PDF graphs given in figure 3.10 and described in equation 3.10 correspond closely with the flat fading channel model's measured amplitude PDF graphs given in figure 3.11 for different Rician factors.
- The theoretical phase PDF graphs given in figure 3.12 and described in equation 3.11 correspond closely with the flat fading channel model's measured phase PDF graphs given in figure 3.13 for different Rician factors.
- The channel model is capable of simulating Doppler spreads accurately as shown in figure 3.14 (in comparison with the theoretical Doppler spectrum given in figure 3.1).

6.4 SPARSE GRAPH CODING

Sparse graph channel codes were considered in Chapter 4, with basic channel coding concepts discussed in section 4.2. Conclusions related to each of the main coding classes studied in this dissertation are considered next.

6.4.1 BLOCK TURBO CODES

General turbo encoding and decoding structures were considered in sections 4.3 and 4.4, while section 4.5 focused on block codes and decoding algorithms to implement block codes in a turbo coding structure. Main conclusions and supporting definitions are given below.

- Turbo codes are parallel concatenated convolutional or block codes which are iteratively decoded by passing LLRs between interconnected SISO decoders to estimate the posterior probability of each transmitted information bit, with increasing accuracy as the number of decoding iterations increase (though the accuracy converges after a number of iterations). Several SISO algorithms are available which can be simplified by sacrificing error performance.
- Turbo codes, usually implemented in conjunction with S/P and P/S converters, can easily be adjusted to match the MDM platform by expanding the code's dimension in order to transmit systematic and code information in parallel.

- Block codes, having highly structured and predictable trellises (starting and ending in the zero state for each codeword), are ideal to use in turbo coding structures. SISO decoding algorithms developed for convolutional codes can be efficiently employed in the turbo coding structure.
- The trellis of any linear block code can be constructed using the bi-directional trellis expurgation technique presented in section 4.5.2.

6.4.2 LDPC CODES

Conclusions related to LDPC codes presented in section 4.6 are provided below.

- LDPC codes can be encoded using different encoding algorithms with differing complexity, though the same factor graph is used to perform decoding, yielding identical error performance results.
- The sparse nature of an LDPC code's parity check matrix can be exploited by using encoding algorithms that avoid multiple zero multiplications, resulting in more efficient encoding algorithms without sacrificing error performance.
- LDPC encoding algorithms typically require pre-processing to be performed on the parity check matrix, though pre-processing is a once-off process and therefore does not contribute to encoding complexity.

6.4.3 RA CODES

RA codes (presented in section 4.7) are low-complexity (the complexity increases linearly with code length N) codes, encoded with a serial concatenation of a repetition encoder and rate-1 recursive convolutional encoder connected via an interleaver. Decoding is performed with a LLR based message passing algorithm on the factor graph of the code, similar to the decoding of LDPC codes.

6.5 ERROR PERFORMANCE RESULTS

The simulation platform and error performance results were given in Chapter 5. Main conclusions for the uncoded and coded CDMA systems employing GCL and ZCC CSSs are given in this section.

6.5.1 UNCODED PERFORMANCE

The evaluation setup and theoretical error performance probabilities were considered in sections 5.2 and 5.3. The uncoded error performance results were given in section 5.4 and related findings are discussed below.

- The uncoded simulation platform functions correctly. This is verified by noting that a single path in the multipath channel simulator provides the correct error probability for different Rician factors within theoretical limits (see figure 5.4).
- The error performance of the CDMA communication system depends on the correlation characteristics of the spreading codes used and the simulation channel's configuration settings.
- The ZCC family ensures improved performance over the GCL family due to the ZCC CSS family's superior correlation characteristics as can be seen in figures 2.27 and 2.28 (GCL CSS) and figures 2.33 and 2.34 (ZCC CSS) in Chapter 2.
- The use of GCL CSSs results in worse performance if the number of CDMA users increases to such an extent that the system setup becomes unfeasible (for both 2D and 4D systems in both AWGN and multipath channel conditions) long before the system's user capacity is reached (for length 63 GCL CSSs, 36 BPSK users can be accommodated, though 5 concurrent users are already too much in terms of the resultant error performance). The use of GCL CSSs must therefore be combined with MUI cancellation techniques to warrant their use in CDMA applications.
- By choosing the correct combination of ZCC CSSs, no performance degradation is experienced with an increase in the number of concurrent CDMA users in AWGN channel conditions. A slight performance degradation is however experienced in multipath fading conditions as the number of CDMA users increases, though the degradation is much less than experienced with GCL CSSs.
- In general the 2D system shows better error performance (at a lower data throughput rate) than the 4D system due to the fact that the 2D system employs complete CSSs per user while the 4D system employs only components of CSSs per user. However, by choosing the correct subset of spreading codes from the ZCC family (by avoiding bad CSS combinations), the same performance on the 2D and 4D

systems can be obtained, which means the same performance can be obtained while doubling the data throughput rate.

6.5.2 CODED PERFORMANCE

The simulated error performance results of the 3 coding schemes (3D turbo code with RM constituent codes, LDPC and RA codes) on the MD-WCDMA platform were given in section 5.5 and related findings are discussed below.

- The use of CSSs does not improve the system's AWGN error performance, but has advantages in terms of CDMA applications and power efficiency.
- The code size N is the main factor determining the code's AWGN and multipath fading error performance, together with other performance factors including the type of coding scheme used, the number of decoding iterations, code rate and application of CSI knowledge in the receiver. The 3D Turbo code with $(12, 6)$ constituent code shows performance closest to the Shannon limit (see figure 5.14), because of its code size ($N = 864$).
- Due to the use of SO-ZCC CSSs, the performance of all CDMA users are equivalent and the multi-user interference effects (visible as performance degradation with increasing users) are also reduced when using a coding scheme.
- Each iterative decoding scheme has a point of performance saturation where additional decoding iterations do not improve error performance. For the schemes evaluated in this study, the turbo codes reach the saturation point faster (after 10 iterations) than the LDPC and RA codes (after 100 iterations). The first few decoding iterations are responsible for the largest coding gains achieved (e.g. in figure 5.16, 2 dB is gained from 1 to 2 decoding iterations, though after 5 iterations less than an additional 0.2 dB is gained).
- Codes with lower code rates do not necessarily show improved performance over higher rate codes under all circumstances. The half rate and quarter rate LDPC codes evaluated in this study show approximately the same performance results, though initially (after 1 iteration) the quarter rate code show much worse performance than the half rate code. If the code size increases, the performance of lower rate codes improves over higher rate codes and ultimately the performance

Table 6.1: Coding complexity and performance

Criteria	BTC	LDPC	RA
Encoding complexity	Low	Moderate	Very low
Decoding complexity	Low-Moderate	Moderate	Moderate
Low ε_b/N_0 performance	Best	Worst	Good
High ε_b/N_0 performance	Worst	Good	Best

of lower rate codes are superior to higher rate codes. This can be ascribed to the fact that a lower rate coding scheme has a larger bandwidth than a higher rate coding scheme (assuming equal transmission periods of complete codewords in order to compare the two coding schemes) and therefore the lower rate coding scheme introduces more noise in the transmission bandwidth. Equation 3.13 in section 3.6.1.1 can also be used to explain this phenomenon; the noise constant for lower rate coding systems is higher than for high rate coding systems.

- The three coding schemes show comparable error performance results (by choosing the same code parameters for all schemes) as shown in figures 5.32 and 5.33, though at low ε_b/N_0 values (around 0 dB) the turbo code performs a little better than the other two coding schemes and at 3 dB and higher the LDPC and RA codes outperform the turbo code. The RA code is the best choice of the three coding schemes presented in this dissertation, in terms of error performance and coding complexity (see table 6.1).

- The cost of implementation of a coding system depends on coding module complexity and though a trade-off between complexity and performance exist, higher complexity (and thus cost) does not necessarily guarantee improved performance. Table 6.1 shows the relation between coding complexity (measured in terms of the overall number of computations required in a 4D system as considered in this study) and error performance. The BTC has low encoding complexity due to the simplicity of the parallel encoder structure (see figure 4.4) and the BTC decoder has low to moderate decoding complexity for a 3D coding structure, though for higher dimensions the BTC decoder will become more complex than the LDPC and RA decoders (see table 4.3).

6.6 CRITICAL EVALUATION OF RESEARCH

This dissertation presented the design details and simulated error performance results of an uncoded and sparse-graph-coded MLM MD-SO-WCDMA modulation platform in four dimensions. The MLM platform and a selection of CSSs were investigated assuming perfect synchronisation over the air interface. AWGN and multipath channel simulators were developed, evaluated and validated against theoretic statistical models to analyse the error performance of the communication system under realistic channel conditions.

This section provides a short critical evaluation of the research performed in terms of the aim and assumptions as stated in section 1.5.2.

6.6.1 COMMUNICATION PLATFORM

The basic 4D modulator provides a generic upwards-expandable power and spectrally efficient WCDMA modem as an alternative for deployment in 4G wireless cellular and WLAN standards. The primary advantage of this modulator compared with existing 3G WCDMA and MC-CDMA WLAN standards is the ability to operate close to the 1 dB saturation point of transceiver power amplifiers with negligible performance loss. Consequently, no power back-offs are required (a 6 dB back-off is typical in current standards) and improved communication range and/or handset battery life are made possible. The use of SO-CSSs reduces MUI and a simplified correlation-type receiver structure can be employed in stead of complex MUI-suppressing receivers.

This study only focused on the WCDMA platform and CSSs employed on the platform, assuming perfect synchronisation, code lock and power control over the air interface for all concurrent users. In a real system, subsystems must be implemented and integrated to ensure such conditions. Pilot channels and signalling schemes are therefore necessary, introducing additional overheads into the system and effectively limiting the overall system capacity and/or data throughput rates.

The basic 4D modulator was implemented successfully and the correlation characteristics of two classes of CSSs and the effect on the output power and error performance of the communication system (both 2D and 4D) were evaluated and explained.

Another aspect of the communication platform of concern is the data throughput rate. Although modern wideband communication systems have much higher data throughput rates and wider bandwidths, the data speed in the simulated communication system considered in this study is scaled down to 1000 symbols per second to simplify and speed up simulations. Simulated error performance results obtained from scaled-down data throughput rate models are however still valid.

For AWGN simulations the average ε_b/N_0 error performance is independent of the data speed, as long as the bandwidth of the system does not exceed the bandwidth for which AWGN has a flat PSD which is about 1 THz. Equations 3.12 and 3.13 show that the AWGN scaling factor depends on the number of samples per bit, the energy per bit and the noise power spectral density and therefore the actual data rate does not influence error performance results.

The multipath fading error performance results depend on the sampling and Doppler frequencies as discussed in section 3.6.2. It is important to note however that everything inside the simulation space is relative to the sampling time and actual data speeds and Doppler frequencies should be interpreted in terms of the sampling frequency. In this dissertation Doppler frequencies of 33, 67 and 100 Hz were assumed on a signal bandwidth of 63 kHz (assuming a sampling frequency of 504 kHz). The channel model was set up as a frequency selective fading channel and although the signal bandwidth is only 63 kHz, the results are an indication of wideband behaviour.

6.6.2 CHANNEL MODELLING

The multipath channel model presented in this study is an accurate theoretical model and the delay profile was chosen to approximate a frequency selective fading or wideband wireless channel. To simplify the channel model, the delay profile was configured to be static (the average path power of each channel path was chosen to be constant) to simulate an instance of typical urban channel conditions. It can be expected that delay profiles of a real channels will be more dynamic, though a static channel is sufficient to evaluate and compare different communication systems and channel coding schemes.

The multipath error performance results depend on the configuration of the channel simulator as given in table 3.1, resembling an exponentially decaying power delay profile with a strong main channel path component ($K_{Rice,1} = 9$ dB) and two much weaker

delayed multipath components. The resultant error performance is therefore close to AWGN performance (see Chapter 5). By adjusting the channel simulator, error performance curves anywhere between the Rayleigh and AWGN curves can be obtained. To compare the multipath results presented in this study with possible future simulation results, the exact same channel configuration should be used. The AWGN error performance curves (both coded and uncoded) therefore provide the best benchmark for comparison to other communication systems.

During the simulation study CSI was extracted directly from the channel model, making exact instantaneous CSI knowledge available at the receiver. In a practical system additional subsystems will be needed to ensure accurate CSI knowledge at the receiver. The simulation results presented in this dissertation are therefore optimistic in terms of CSI availability.

6.6.3 CHANNEL CODING

The channel codes presented in this study are existing codes or combinations of existing coding schemes, with the uncoded platform showing error performance results as predicted by theory. The best coding scheme in terms of error performance considered in this study (the 3D RM(12,6) TC with results shown in figure 5.14) is about 6 dB from the Shannon limit at $P_e = 10^{-6}$.

To approach the Shannon limit, larger code sizes can be used (see for example [21]), though larger code sizes might not be practical. The choice of coding scheme and code size depends on the specific implementation and generally a trade-off between required quality of service and acceptable level of complexity exists.

Three classes of sparse graph codes with relative short code sizes were presented in this dissertation and though the Shannon limit is still at least 6 dB's away, these codes have relatively low implementation complexities. The RA code presented in this dissertation provides the best trade-off between implementation complexity and error performance (considering the quarter rate codes), though there are still many coding and modulation scheme combinations not yet considered and the search for optimal coding performance and data throughput will therefore continue.

6.7 FUTURE STUDY

This study involved a variety of aspects in digital communication theory, including multi-user access communication systems, channel modelling and channel coding. Possible future research ventures related to this study are listed below.

- In order to increase user capacity in the 4D communication system, techniques to extend CSS family sizes need to be investigated. The length-63 GCL CSSs considered in this study have 36 sequences in the family, though the ZCC CSSs (also length-63) only have 7 family members. Difficulties to overcome include creating larger family sizes without compromising the CE and ZCC characteristics of the ZCC CSSs.

- Although the ZCC CSSs have perfectly ZCC characteristics when employed as complex sequences (see figure 2.33), the correlation between complex components (components of CSSs are used in the 4D system) show that certain sequence combinations are not perfectly orthogonal (see figure 2.34). Techniques to improve the correlation between these combinations should be investigated.

- The 4D transmitter and receiver presented in this study form only the basic building block of a larger communication system. Other important aspects that need to be addressed before the 4D system can be considered as a communication standard (or incorporated into existing standards) include synchronisation, code-lock and multi-user coordination strategies (how CSSs and 4D channels will be assigned to different users according to user bandwidth demands).

- The MD system's capacity and spectral efficiency can be improved further by extending the MD system to use multiple carrier frequencies in stead of using a single carrier, to create a MD-MC-CDMA or MD-OFDM system. The MD-MC-CDMA system and rules to assign MD data streams to different carrier frequencies are possible areas of further research.

- The incorporation of antenna diversity techniques, such as space time coding, into the 4D system can be considered.

- The possibilities of channel coding schemes and combinations thereof, implemented on the MD platform are virtually endless and can be investigated further in a search of the optimal MD coding scheme. Hybrid turbo and product codes

with different constituent code combinations are a few of the myriad codes that can be considered. The MD platform can also be combined with mapping-by-set-partitioning [31] codes and compared to the results presented in this dissertation.

- Combinations of coding modules, such as MD turbo codes with sparse graph constituent codes, that will require novel message passing decoding algorithms in a turbo decoding structure, can be investigated. Only linear block codes are considered as constituent codes of the turbo code structure in this dissertation, necessitating implementation of SISO block decoders. By considering sparse graph codes as constituent turbo codes, new code combinations and possible performance gains can be realised.

- The development of a multi-user MD coded modulation system can be undertaken by extending CSS family sizes and using coded DSSS techniques.

- Though theoretical bounds for turbo-like codes exist, more research into the theoretical behaviour of sparse graph codes is required.

REFERENCES

[1] C. E. Shannon, "A mathematical theory of communication," *Bell System Technical Journal*, vol. 27, pp. 379–423 and 623–656, July and October 1948.

[2] C. E. Shannon, "Communication in the presence of noise," *Proceedings of the IEEE*, vol. 86, no. 2, pp. 447–457, February 1998.

[3] J. G. Proakis and M. Salehi, *Communications Systems Engineering*, 2nd ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 2002.

[4] W. Stallings, *Wireless Communications and Networks*, 1st ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 2002.

[5] T. S. Rappaport, *Wireless Communications Principles and Practice*, 2nd ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 2002.

[6] M. A. Rosengrant, *Introduction to Telecommunications*, 1st ed. Upper Saddle River, New Jersey, USA: Prentice Hall, 2002.

[7] J. G. Proakis, *Digital Communications*, 4th ed. Boston, Massachusetts, USA: McGraw-Hill, 2001.

[8] M. P. Lötter and L. P. Linde, "A class of bandlimited complex spreading sequences with analytic properties," *Proceedings of the 4th IEEE International Symposium on Spread Spectrum Techniques and Applications, Mainz, Germany*, vol. 2, pp. 662–666, September 1996.

[9] I. Pryra, L. P. Linde, and S. A. Swanepoel, "New family of constant-envelope root-of-unity filtered complex spreading sequences with zero cross-correlation properties," *Proceedings of 6th AFRICON IEEE Conference*, vol. 1, pp. 299–304, October 2002.

[10] D. C. Chu, "Polyphase codes with good periodic correlation properties," *IEEE Transactions on Information Theory*, pp. 531–532, July 1972.

[11] R. D. Austin and S. P. Bradley, *The Broadband Explosion Leading Thinkers on the Promise of a Truly Interactive World*, 1st ed. Boston, Massachusetts, USA: Harvard Business School Press, 2005.

[12] R. H. Morelos-Zaragoza, *The Art of Error Correcting Coding*, 1st ed. Chichester, UK: John Wiley & Sons Limited, 2004.

[13] J. K. Wolf, "Efficient maximum likelihood decoding of linear block codes using a trellis," *IEEE Transactions on Information Theory*, vol. 24, pp. 76–80, January 1978.

[14] D. J. C. MacKay, *Information Theory, Inference, and Learning Algorithms*, 1st ed. Cambridge, UK: Cambridge University Press, 2005.

[15] R. W. Hamming, "Error detecting and error correcting codes," *Bell System Technical Journal*, vol. 26, no. 2, pp. 147–160, April 1950.

[16] I. S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society of Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, June 1960.

[17] E. R. Berlekamp, "Nonbinary BCH decoding," *IEEE Transactions on Information Theory*, vol. 14, no. 2, pp. 242–242, March 1968.

[18] R. G. Gallager, "Low-density parity-check codes," *IRE Transactions on Information Theory*, vol. 8, no. 1, pp. 21–28, January 1962.

[19] T. J. Richardson and R. L. Urbanke, "The Renaissance of Gallager's Low-Density Parity-Check codes," *IEEE Communications Magazine*, vol. 41, no. 8, pp. 126–131, August 2003.

[20] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, "Design of capacity-approaching irregular low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 619–637, February 2001.

[21] S. Y. Chung, G. D. Forney, T. J. Richardson, and R. L. Urbanke, "On the Design of Low-Density Parity-Check Codes within 0.0045 dB of the Shannon limit," *IEEE Communications Letters*, vol. 5, no. 2, pp. 58–60, February 2001.

[22] P. Elias, "Coding of noisy channels," *IRE Convention Record*, vol. 4, pp. 37–47, March 1955.

[23] J. M. Wozencraft, "Sequential decoding for reliable communication," *IRE Convention Record*, vol. 5, no. 2, pp. 11–23, August 1957.

[24] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Transactions on Information Theory*, vol. 13, no. 2, pp. 260–269, April 1967.

[25] P. Elias, "Error-free coding," *IEEE Transactions on Information Theory*, vol. 4, no. 4, pp. 29–37, September 1954.

[26] G. D. Forney, *Concatenated Codes*, 1st ed. Cambridge, Massachusets, USA: MIT Press, 1966.

[27] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," *Proceedings of 1993 IEEE International Conference on Communications*, vol. 2, pp. 1064–1070, May 1993.

[28] B. Yin and M. R. Soleymani, "Design and implementation of three-dimensional block turbo codes," *Proceedings of 2003 IEEE Canadian Conference on Electrical and Computer Engineering - CCECE2003, Montreal, Canada*, pp. 1625–1628, May 2003.

[29] D. Divsalar, H. Jin, and R. J. McEliece, "Coding theorems for Turbo-Like codes," *Proceedings of 36th Allerton Conference on Communication, Control, and Computing*, pp. 201–210, September 1998.

[30] J. L. Massey, "Coding and modulation in digital communications," *Proceedings of International Zurich Seminar on Digital Communications*, pp. E2(1)–E2(4), March 1974.

[31] G. Ungerboeck, "Channel coding with multilevel/phase signals," *IEEE Transactions on Information Theory*, vol. 28, no. 1, pp. 55–67, January 1982.

[32] V. Tarokh, N. Seshadri, and A. R. 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, March 1998.

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

[34] L. P. Linde and M. Lötter, "Spread spectrum modulator and method," South African Complete Patent 96/0355, January 17, 1996, earliest priority claimed: ZA 94/9014, 14 November 1994.

[35] F. E. Marx and L. P. Linde, "Four dimensional modem employing complex spreading sequences," *Proceedings of 5th AFRICON IEEE Conference*, vol. 1, pp. 221–226, September 1999.

[36] W. Feng and B. Vucetic, "A list bidirectional soft output decoder of turbo codes," *Proceedings of the International Symposium on Turbo Codes and Related Topics*, pp. 288–292, September 1997.

[37] J. Chen, A. Dholakia, E. Eleftheriou, M. P. C. Fossorier, and X. Y. Hu, "Reduced-complexity decoding of LDPC codes," *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1288–1299, August 2005.

[38] R. M. Neal, "Sparse matrix methods and probabilistic inference algorithms - Part I: Faster encoding for low density parity check codes using sparse matrix methods," IMA Programs on Codes, Systems and Graphical Models, Tech. Rep., August 1999.

[39] S. Johnson and S. Weller, "Interleaver and accumulator design for systematic Repeat-Accumulate codes," *Proceedings of 6th Australian Communication Theory Workshop*, pp. 1–7, February 2005.

[40] H. Jin, "Analysis and design of Turbo-like codes," Ph.D. thesis, California Institute of Technology, Pasadena, Calafornia, 2001.

[41] L. P. Linde and J. D. Vlok, "Power and spectrally efficient four-dimensional super-orthogonal WCDMA building block for next generation wireless applications," *IEEE Communications Letters*, vol. 10, no. 7, pp. 519–521, July 2006.

[42] J. D. Vlok and L. P. Linde, "Performance of a Block-Turbo-Coded Four-Dimensional WCDMA Communication System in AWGN and Multipath Fading," *Submitted to IEE Electronics Letters*, August 2006.

[43] L. P. Linde, L. Staphorst, and J. D. Vlok, "Performance of a Quasi-Synchronous Four-Dimensional Super-Orthogonal WCDMA Modulator for Next Generation Wireless Applications," *To appear in the South African Journal of Science*, March 2007.

[44] R. H. Clarke, "A statistical theory of mobile-radio reception," *Bell Systems Technical Journal*, vol. 47, pp. 957–1000, July 1968.

[45] L. Staphorst, "Viterbi decoded linear block codes for narrowband and wideband communication over mobile fading channels," Master's dissertation, University of Pretoria, South Africa, 2005.

[46] C. Komninakis, "A fast and accurate Rayleigh fading simulator," *Proceedings of GLOBECOM Global Telecommunications IEEE Conference*, vol. 6, pp. 3306–3310, December 2003.

[47] M. P. Lötter and L. P. Linde, "Constant envelope filtering of complex spreading sequences," *Electronics Letters*, vol. 31, no. 17, pp. 1406–1407, August 1995.

[48] L. P. Linde, F. E. Marx, and W. R. Malan, "Power and spectral efficiency of a family of constant-envelope root-of-unity filtered complex spreading sequences in WCDMA non-linear power amplification," *Proceedings of 6th AFRICON IEEE Conference*, vol. 1, pp. 395–400, October 2002.

[49] L. Staphorst, M. Jamil, and L. P. Linde, "Performance of a synchronous balanced QPSK CDMA system using complex spreading sequences in AWGN," *Proceedings of 5th AFRICON IEEE Conference*, vol. 1, pp. 215–220, September 1999.

[50] F. E. Marx, "DSSS communication link employing complex spreading sequences," Master's dissertation, University of Pretoria, South Africa, 2005.

[51] R. L. Frank and S. A. Zadoff, "Phase shift pulse codes with good periodic correlation properties," *IRE Transactions on Information Theory*, pp. 381–382, October 1962.

[52] R. L. Frank, "Comments on Polyphase codes with good periodic correlation properties by Chu, David C," *IEEE Transactions on Information Theory*, vol. 19, no. 2, pp. 244–244, March 1973.

[53] N. Suehiro, "A signal design without co-channel interference for approximately synchronized CDMA systems," *IEEE JSAC*, vol. 12, pp. 837–841, June 1994.

[54] L. Staphorst and L. P. Linde, "Optimal ABC sequence selection for MUI minimization," *Submitted to IEE Communications letters*, 2006.

[55] M. J. E. Golay, "Complementary series," *IEEE Transactions on Information Theory*, vol. 7, no. 2, pp. 82–87, April 1961.

[56] L. Staphorst and L. P. Linde, "Evaluating Viterbi decoded Reed-Solomon block codes on a complex spreaded DS/SSMA CDMA system: Part II - Channel model, evaluation and results," *Proceedings of AFRICON IEEE Conference*, pp. 335–340, 2004.

[57] B. Wichmann and D. Hill, "Building a random-number generator," *Byte Magazine*, pp. 127–128, March 1987.

[58] G. Marsaglia and T. A. Bray, "A convenient method for generating normal variables," *SIAM Rev.*, vol. 6, pp. 260–264, 1964.

[59] S. O. Rice, "Statistical properties of a sine wave plus random noise," *Bell Systems Technical Journal*, vol. 27, pp. 109–157, January 1948.

[60] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal decoding of linear codes for minimizing symbol error rate," *IEEE Transactions on Information Theory*, vol. 20, no. 2, pp. 284–287, March 1974.

[61] N. Wiberg, "Codes and decoding on general graphs," Ph.D. thesis, Linköping University, Sweden, 1996.

[62] R. J. McEliece, D. J. C. MacKay, and J. F. Cheng, "Turbo Decoding as an Instance of Pearl's Belief Propagation Algorithm," *IEEE Journal on Selected Areas of Communications*, vol. 16, no. 2, pp. 140–152, February 1998.

[63] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo-Coding, Turbo Equalisation and Space-Time Coding for transmission over Fading Channels*, 1st ed. Chichester, UK: John Wiley & Sons Limited, 2002.

[64] J. Hagenhauer and P. Hoeher, "A Viterbi algorithm with soft-decision outputs and its applications," *Proceedings of GLOBECOM Global Telecommunications IEEE Conference*, vol. 3, pp. 1680–1686, November 1989.

[65] R. G. Gallager, "Low-density parity-check codes," Ph.D. thesis, M.I.T., Cambridge, Massachusetts, 1963.

[66] D. J. C. MacKay and R. M. Neal, "Near Shannon limit performance of low density parity check codes," *Electronics Letters*, vol. 32, no. 18, pp. 1645–1646, August 1996.

[67] D. J. C. MacKay, "Good error-correcting codes based on very sparse matrices," *IEEE Transactions on Information Theory*, vol. 45, no. 2, pp. 399–431, March 1999.

[68] M. Sipser and D. A. Spielman, "Expander codes," *IEEE Transactions on Information Theory*, vol. 42, no. 6, pp. 1710–1722, November 1996.

[69] R. L. Burden and J. D. Faires, *Numerical Analysis*, 7th ed. Pacific Grove, California, USA: Brooks/Cole Thomson Learning, 2001.

[70] T. J. Richardson and R. L. Urbanke, "Efficient encoding of low-density parity-check codes," *IEEE Transactions on Information Theory*, vol. 47, no. 2, pp. 638–656, February 2001.

[71] R. M. Tanner, "A recursive approach to low complexity codes," *IEEE Transactions on Information Theory*, vol. 27, no. 5, pp. 533–547, September 1981.

[72] H. Jin and R. J. McEliece, "RA codes achieve AWGN channel capacity," *Proceedings of 13th International Symposium on Applied Algebra, Algebraic Algorithms and Error Correcting Codes*, pp. 10–18, November 1999.

[73] A. Roumy, S. Guemghar, G. Caire, and S. Verdú, "Design methods for irregular Repeat-Accumulate codes," *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1711–1727, August 2004.

[74] R. J. McEliece, "Achieving the Shannon limit: A progress report," *Plenary talk presented at 38th Allerton Conference on Communication, Control, and Computing*, October 2000.

[75] J. Hagenhauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 3, pp. 429–445, March 1996.

[76] C. Desset, B. Macq, and L. Vandendorpe, "Computing the word-, symbol-, and bit-error rates for block error-correcting codes," *IEEE Transactions on Communications*, vol. 52, no. 6, pp. 910–921, June 2004.

[77] *3GPP Technical Specification 25.213 v.7.0.0 (2006-03); Spreading and Modulation (FDD), Release 7*, 3rd Generation Partnership Project Radio Access Network Group, http://www.3gpp.org/ftp/Specs/html-info/25213.htm, Last accessed on 28 September 2006.

ADDENDUM A

COMPLEX SPREADING SEQUENCES

A.1 ADDENDUM SUMMARY

This addendum provides additional figures and tables related to the CSSs employed on the WCDMA platform presented in Chapter 2. The figures contain additional correlation graphs and the tables contain output power and correlation values of the CE-GCL and ZCC CSSs employed on the MDM platform.

A.2 CORRELATION GRAPHS

The figures provided in this addendum contain correlation graphs of the CSSs presented in section 2.4, including Chu (section 2.4.1), CE-GCL (section 2.4.2) and ZCC (section 2.4.3) CSSs. All CSSs considered have length $L = 63$, all AC graphs show the correlation of the first code in the relevant family and all CC graphs show the correlation between the first and second codes in the relevant family. Each figure shows four correlation graphs, including the correlation between 2 complex sequences (in the AC graphs only 1 complex sequence is used), between the 2 complex sequences' real components separately, between the 2 complex sequences' imaginary components separately and the correlation between one sequence's real component and the other sequence's imaginary component. Table A.1 shows the figure numbers of the correlation graphs given in Chapter 2 and in this addendum.

Table A.1: Figure numbers of correlation graphs

CSS	AAC	ACC	PAC	PCC
Chu	2.13	A.1	2.14	A.2
CE-GCL	2.18	A.3	A.4	A.5
ZCC	2.21	A.6	A.7	A.8

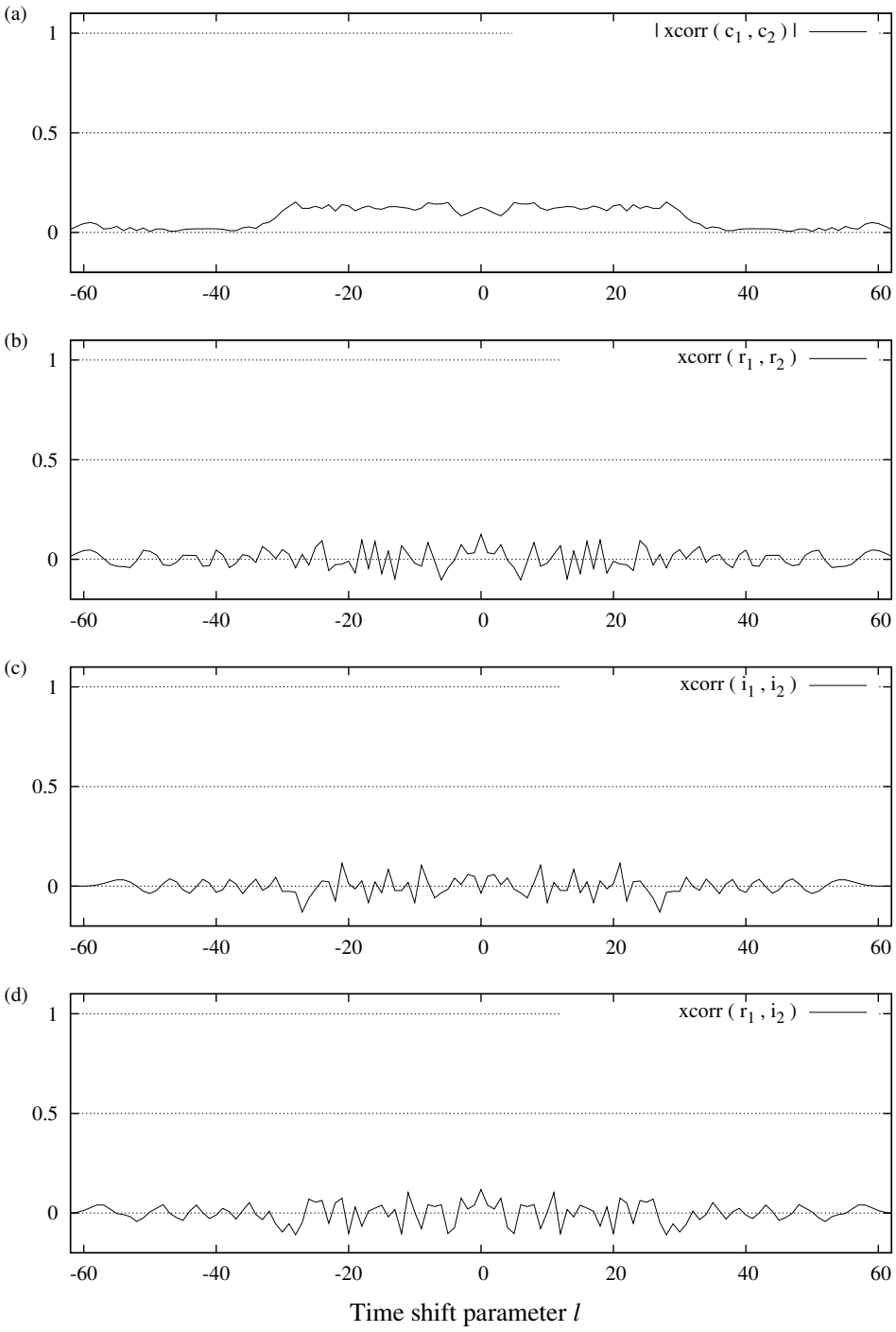

Figure A.1: The figure shows the ACC between two length $L = 63$ Chu $(r = 1, 2)$ (a) complex sequences (b) real sequences (c) imaginary sequences and (d) the ACC between real and imaginary sequences using only the primary sequence values.

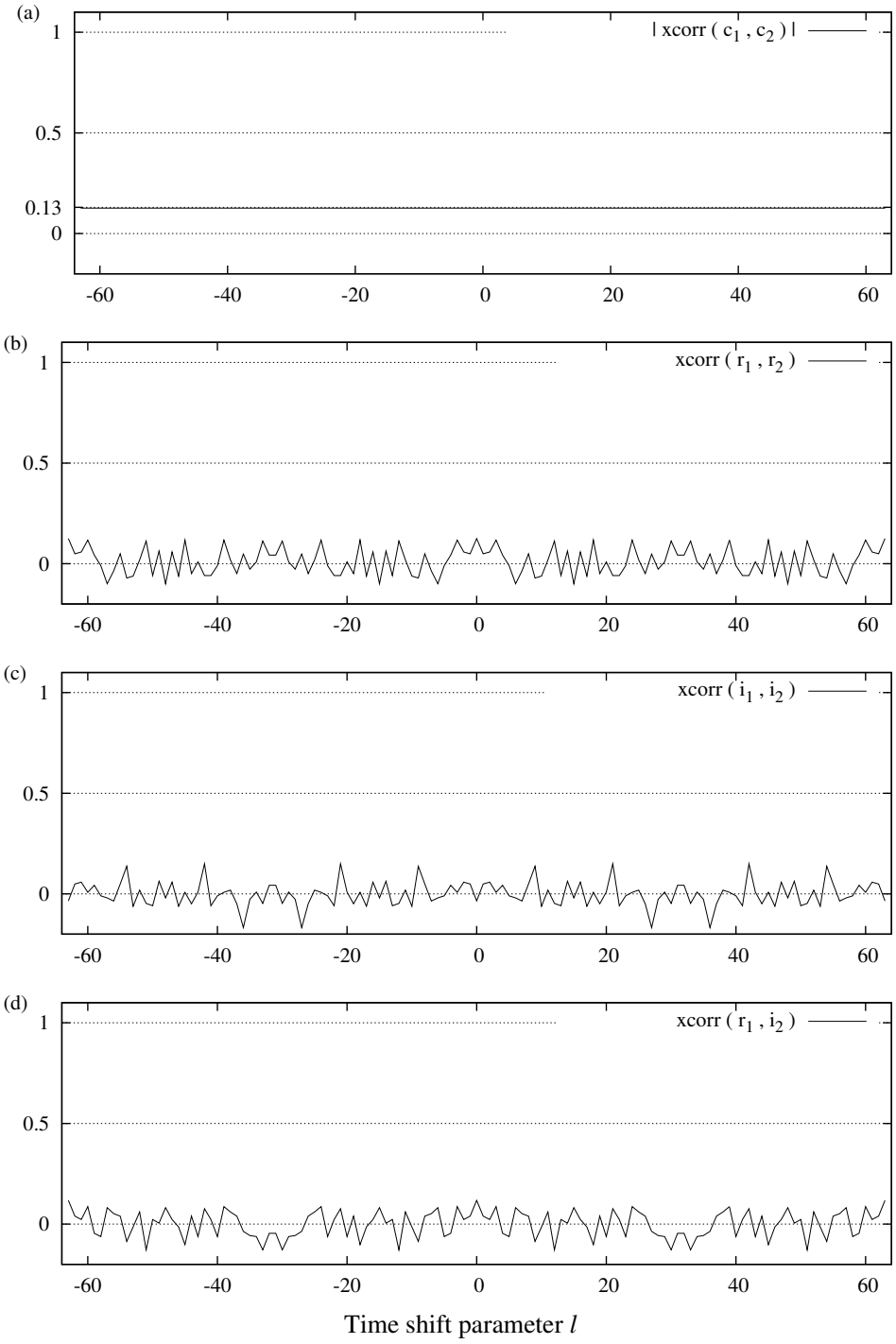

Figure A.2: The figure shows the PCC between two length $L = 63$ Chu (a) complex sequences (b) real sequences (c) imaginary sequences and (d) the PCC between real and imaginary sequences using only the primary sequence values.

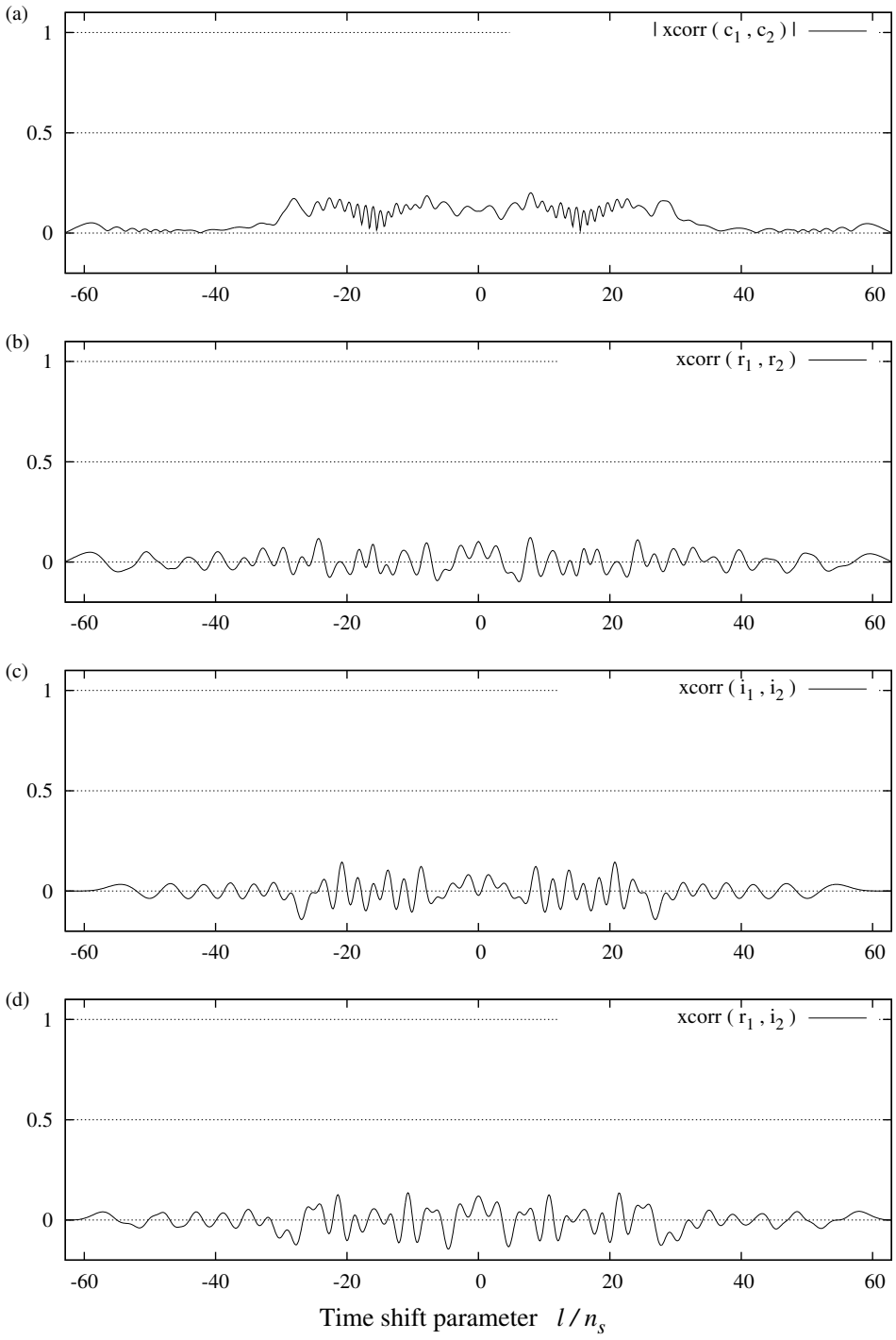

Figure A.3: The figure shows the ACC between two length $L = 63$ CE-RUF-GCL (a) complex sequences (b) real sequences (c) imaginary sequences and (d) the ACC between real and imaginary sequences.

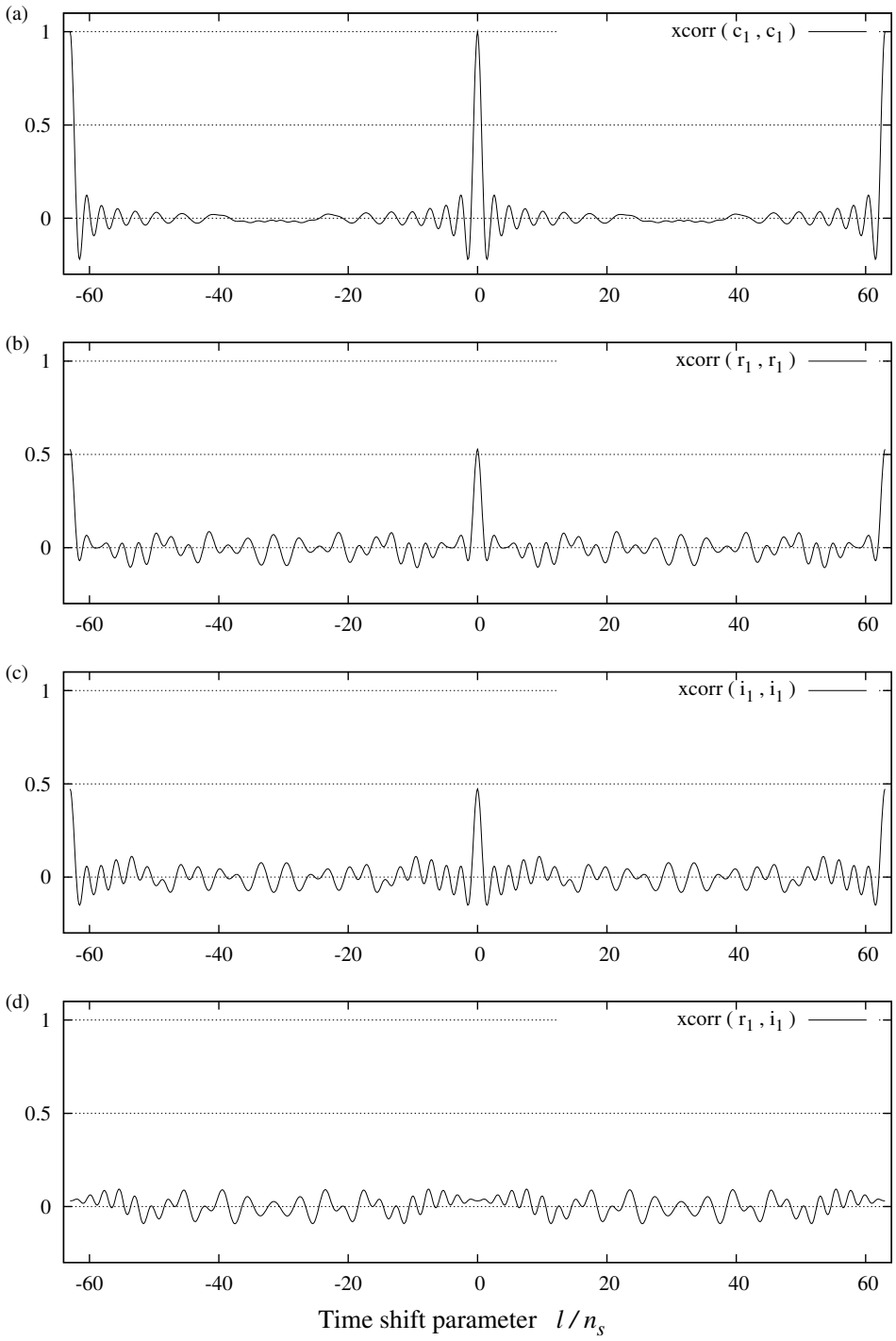

Figure A.4: The figure shows the PAC of a length $L = 63$ CE-RUF-GCL (a) complex sequence (b) real sequence only (c) imaginary sequence only and (d) the PCC between real and imaginary sequences.

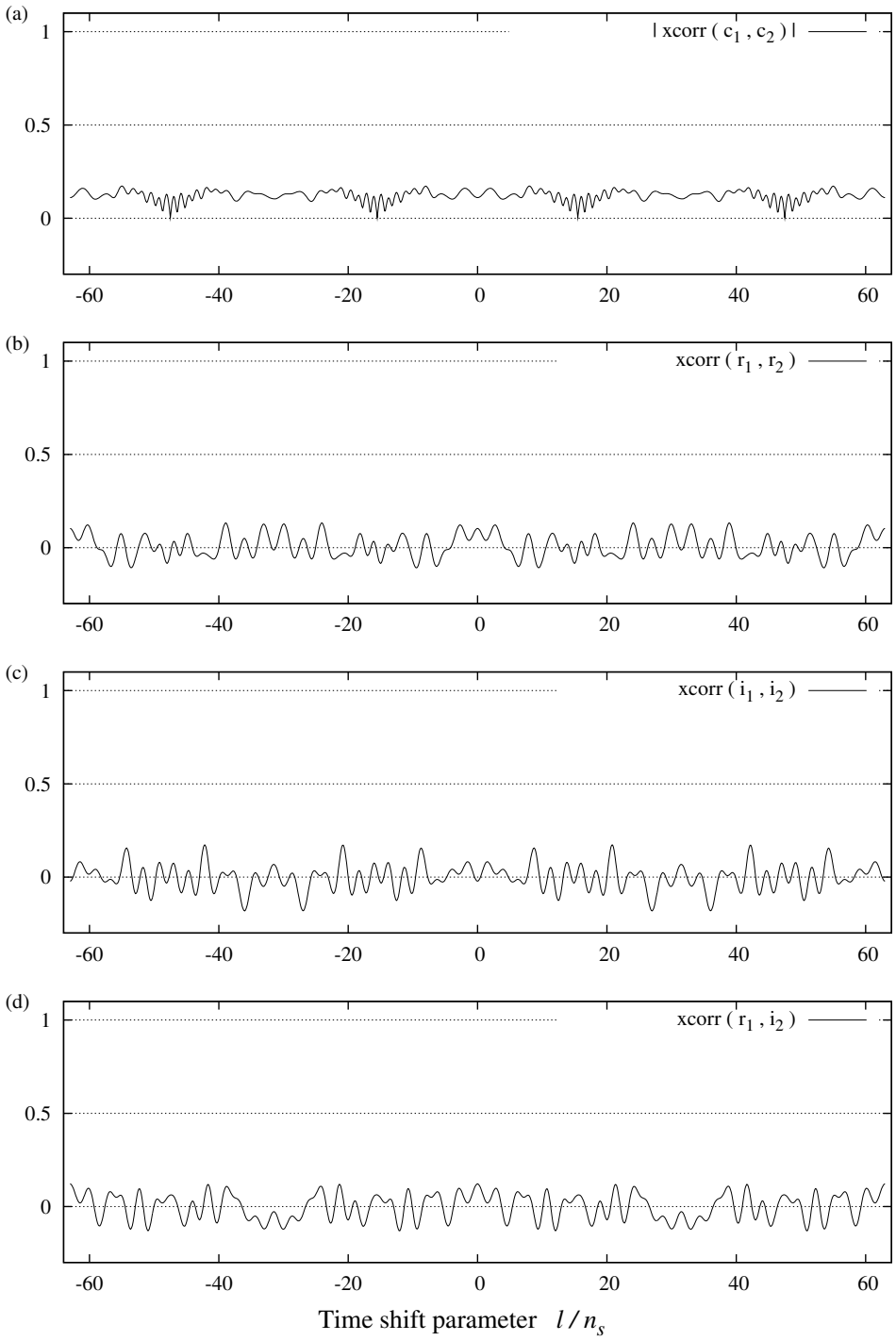

Figure A.5: The figure shows the PCC between two length $L = 63$ CE-RUF-GCL (a) complex sequences (b) real sequences (c) imaginary sequences and (d) the PCC between real and imaginary sequences.

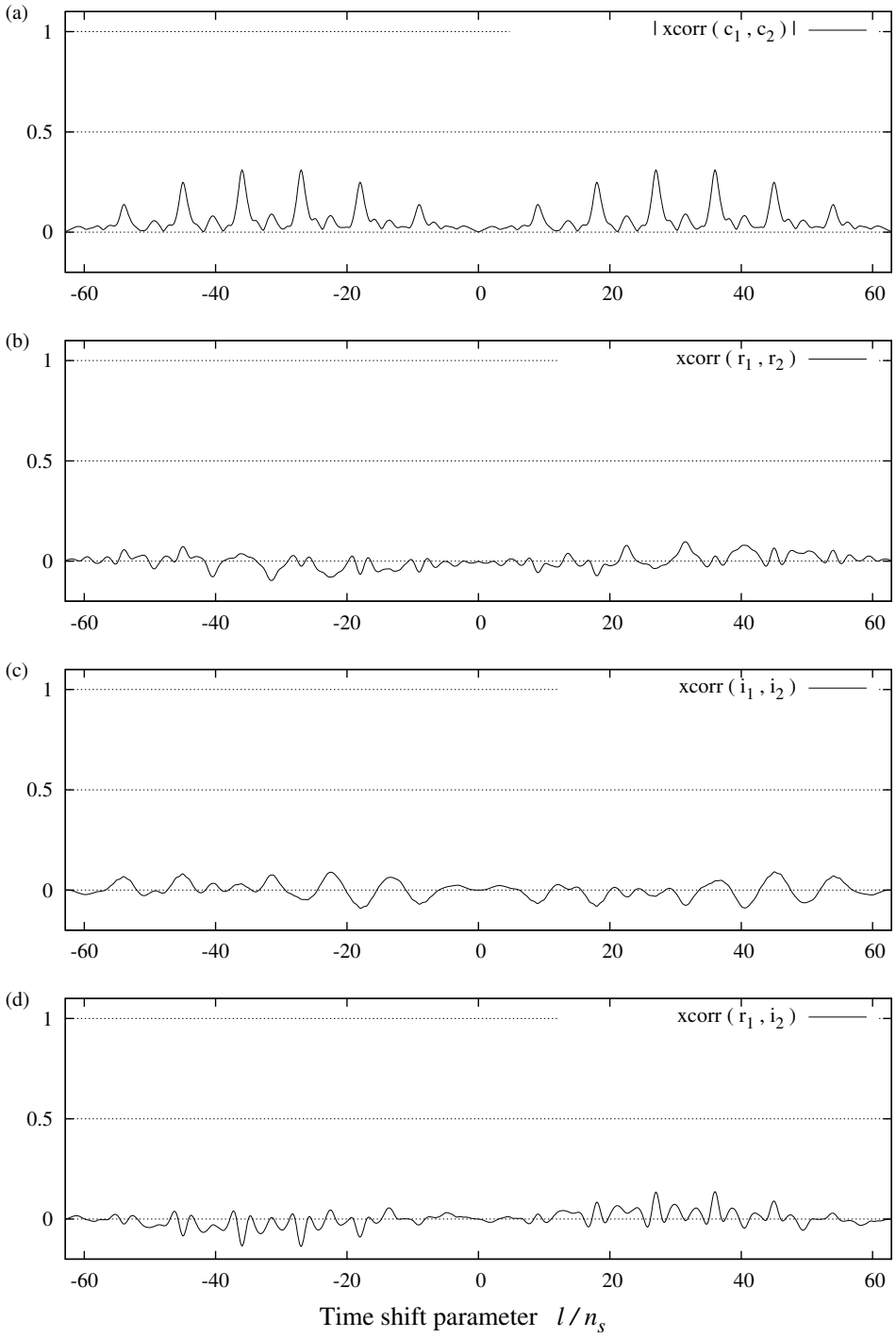

Figure A.6: The figure shows the ACC between two length $L = 63$ ZCC (a) complex sequences (b) real sequences (c) imaginary sequences and (d) the ACC between real and imaginary sequences.

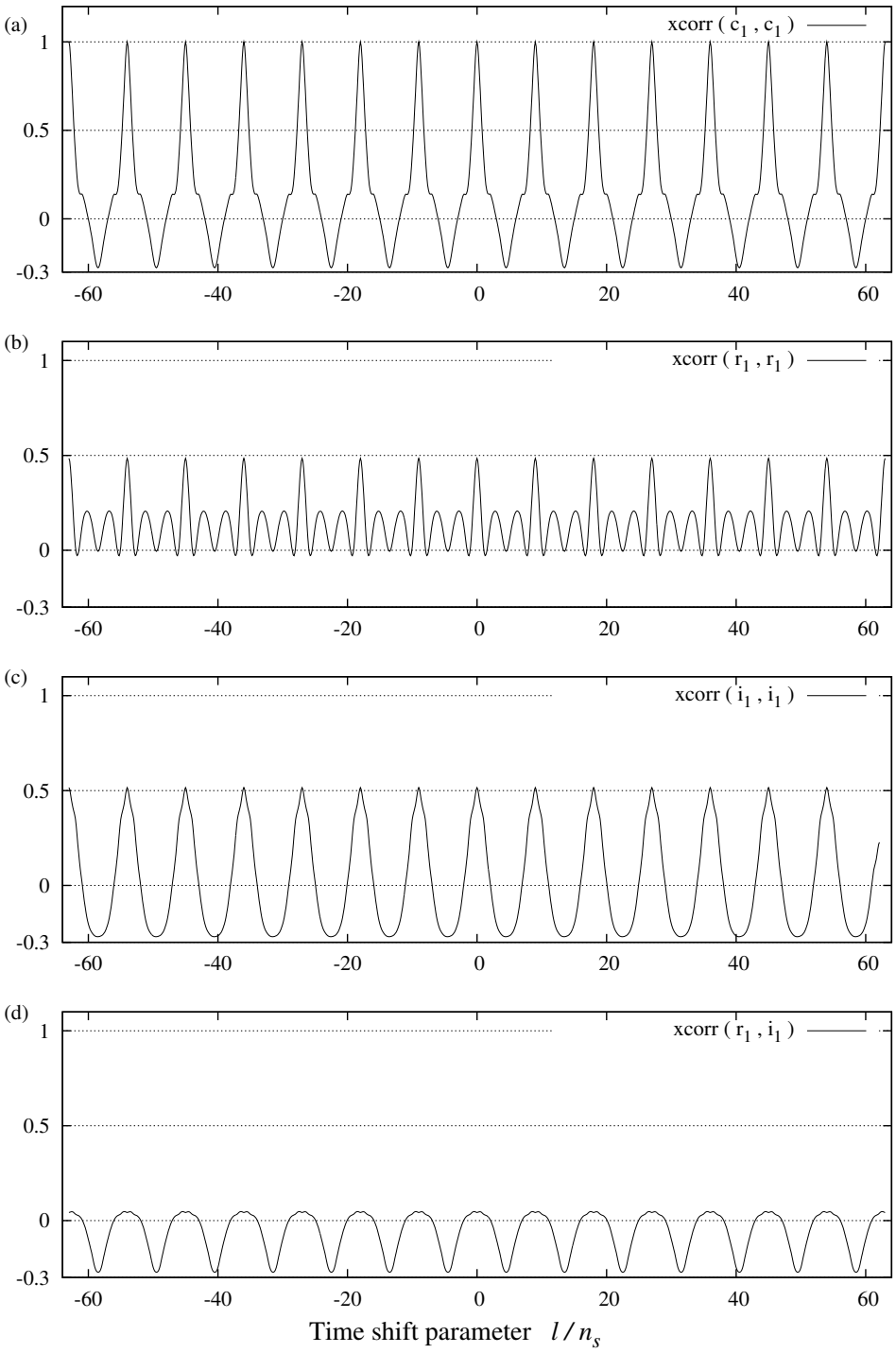

Figure A.7: The figure shows the PAC of code $r = 1$, length $L = 63$ ZCC (a) complex sequence (b) real sequence only (c) imaginary sequence only and (d) the PCC between real and imaginary sequences.

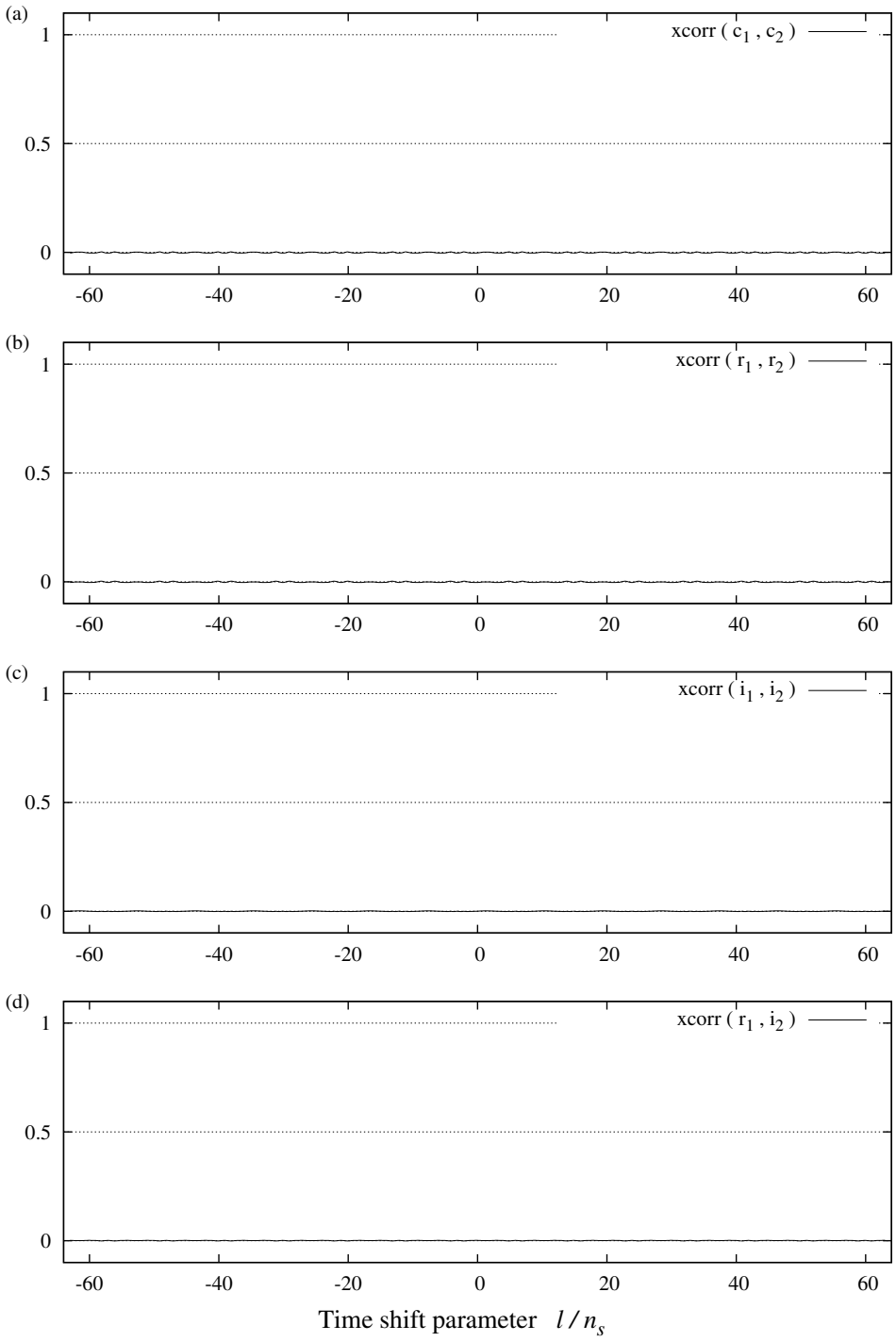

Figure A.8: The figure shows the PCC between two length $L = 63$ ZCC (a) complex sequences (b) real sequences (c) imaginary sequences and (d) the PCC between real and imaginary sequences, showing zero cross correlation values for all time shifts.

A.3 OUTPUT POWER

The average power of the 4D modulator (figure 2.9) output signal $s(t)$ defined in equations 2.31 and 2.50, taken over a single symbol period for each CSS in both CE-GCL and ZCC CSS classes employed are given in table A.2. The average power of $s(t)$ is calculated using equation 2.51, repeated here:

$$\begin{aligned} P_{s,ave} &= \frac{1}{T_s}\int_{t_0}^{t_0+T_s} |s(t)|^2\, dt \qquad (A.1)\\ &= \frac{1}{T_s}\int_{t_0}^{t_0+T_s} \left\{\Psi_1^2(t)+\Psi_2^2(t)+\Psi_3^2(t)+\Psi_4^2(t)\right\} dt + \zeta \end{aligned}$$

with the cross term (from equation 2.52):

$$\zeta = \frac{2}{T_s}\int_{t_0}^{t_0+T_s} \{\Psi_1(t)\Psi_2(t)+\Psi_3(t)\Psi_4(t)\}\, dt \qquad (A.2)$$

Table A.2: Average output power and cross term values of the 4D modulator for different CSSs employed

r_c	CE-GCL		ZCC	
	$P_{s,ave}$	ζ	$P_{s,ave}$	ζ
1	0.93778	-0.062224	1.0829	8.2861e-02
2	0.99273	-0.0072682	1.0000	4.9978e-11
3	0.87513	-0.12487	1.0000	-1.6176e-11
4	1.09855	0.098553	1.0000	-2.3681e-11
5	0.96979	-0.030213	1.0000	1.1784e-10
6	0.95673	-0.043271	1.0000	-5.6810e-12
7	0.87813	-0.12187	1.0000	6.2994e-11

The first term in equation A.1 which can be expressed as $(P_{s,ave} - \zeta)$ is exactly 1.0000 for all CSSs considered in table A.2. The values in the table were obtained by assuming data transmission at $f_b = 1000$ symbols per second on a $f_c = 8$ kHz carrier sampled at $f_s = 32$ kHz.

A.4 TABLES WITH CORRELATION VALUES

A.4.1 CORRELATION VALUES FOR CE-GCL-CSSs

The correlation values between complete CE-GCL-CSSs (see figure 2.27) are shown in table A.3 and the correlation values between the real and imaginary components separately (see figure 2.28) are shown in tables A.4 and A.5.

Table A.3: Correlation values of 7 best CE-GCL-CSSs of length 63

r_c	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	504.00	55.215	153.71	170.95	266.21	132.75	28.449
2	55.215	504.00	85.304	81.336	79.444	155.01	70.226
3	153.71	85.304	504.00	37.993	160.39	266.21	84.803
4	170.95	81.336	37.993	504.00	104.89	32.502	90.961
5	266.21	79.444	160.39	104.89	504.00	171.84	82.596
6	132.75	155.01	266.21	32.502	171.84	504.00	92.387
7	28.449	70.226	84.803	90.961	82.596	92.387	504.00

Table A.4: Correlation values of 7 best CE-GCL-CSSs (complex components separated) of length 63 - Part I

r_{cc}	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	266.72	15.671	51.909	60.525	127.06	100.01	111.66
2	15.671	237.28	23.337	11.096	2.8771	7.9312	66.789
3	51.909	23.337	249.98	1.7415	14.595	28.178	91.220
4	60.525	11.096	1.7415	254.02	54.909	33.915	90.109
5	127.06	2.8771	14.595	54.909	253.82	31.463	14.668
6	100.01	7.9312	28.178	33.915	31.463	250.18	77.025
7	111.66	66.789	91.220	90.109	14.668	77.025	266.48
8	11.131	49.965	30.215	146.25	63.730	50.259	24.733
9	100.54	22.301	54.673	48.509	38.701	33.052	35.436
10	98.023	136.93	24.325	22.946	117.89	15.543	36.741
11	50.266	18.313	5.0432	93.821	77.586	43.828	45.166
12	66.523	73.424	46.934	59.884	76.497	159.88	5.1393
13	22.281	41.055	8.2953	18.350	10.148	4.2512	91.051
14	68.756	15.797	49.127	27.752	64.510	39.485	9.3576

Table A.5: Correlation values of 7 best CE-GCL-CSSs (complex components separated) of length 63 - Part II

r_{cc}	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	11.131	100.54	98.023	50.266	66.523	22.281	68.756
2	49.965	22.301	136.93	18.313	73.424	41.055	15.797
3	30.215	54.673	24.325	5.0432	46.934	8.2953	49.127
4	146.25	48.509	22.946	93.821	59.884	18.350	27.752
5	63.730	38.701	117.89	77.586	76.497	10.148	64.510
6	50.259	33.052	15.543	43.828	159.88	4.2512	39.485
7	24.733	35.436	36.741	45.166	5.1393	91.051	9.3576
8	237.52	47.120	27.569	37.624	44.104	7.1116	1.5934
9	47.120	243.18	7.3260	105.30	64.951	39.429	30.450
10	27.569	7.3259	260.82	37.780	32.445	69.269	112.33
11	37.624	105.30	37.780	263.31	10.600	39.971	63.949
12	44.104	64.951	32.445	10.599	240.69	19.284	0.12552
13	7.1116	39.429	69.269	39.971	19.284	263.34	30.495
14	1.5934	30.450	112.33	63.949	0.12552	30.495	240.66

A.4.2 CORRELATION VALUES FOR ZCC CSSs

The correlation values between complete ZCC CSSs (see figure 2.33) are shown in table A.6 and the correlation values between the real and imaginary components separately (see figure 2.34) are shown in tables A.7 and A.8.

Table A.6: Correlation values of the 7 ZCC CSSs of length 63

r_c	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	5.04e+02	9.50e-04	8.26e-04	1.94e-03	8.47e-04	3.92e-04	1.80e-03
2	9.50e-04	5.04e+02	8.97e-04	5.36e-04	9.37e-04	1.03e-03	2.90e-04
3	8.26e-04	8.97e-04	5.04e+02	9.07e-04	3.65e-04	2.90e-03	1.09e-03
4	1.94e-03	5.36e-04	9.07e-04	5.04e+02	2.08e-03	1.25e-03	2.02e-03
5	8.47e-04	9.37e-04	3.65e-04	2.08e-03	5.04e+02	7.88e-04	3.67e-04
6	3.92e-04	1.03e-03	2.90e-03	1.25e-03	7.88e-04	5.04e+02	5.92e-04
7	1.80e-03	2.90e-04	1.09e-03	2.02e-03	3.67e-04	5.92e-04	5.04e+02

Table A.7: Correlation values of the 7 ZCC CSSs (complex components separated) of length 63 - Part I

r_{cc}	**1**	**2**	**3**	**4**	**5**	**6**	**7**
1	2.44e+02	2.05e+01	6.33e-04	3.43e-05	3.35e-04	1.12e-04	5.02e-04
2	2.05e+01	2.60e+02	3.51e-04	2.63e-04	3.21e-04	3.69e-04	1.10e-03
3	6.33e-04	3.51e-04	2.52e+02	4.11e-05	1.16e-04	3.07e-04	8.48e-05
4	3.43e-05	2.63e-04	4.11e-05	2.52e+02	3.18e-04	7.59e-04	1.32e-04
5	3.35e-04	3.21e-04	1.16e-04	3.18e-04	2.52e+02	1.74e-04	6.32e-04
6	1.12e-04	3.69e-04	3.07e-04	7.59e-04	1.74e-04	2.52e+02	9.10e-04
7	5.02e-04	1.10e-03	8.48e-05	1.32e-04	6.32e-04	9.10e-04	2.52e+02
8	3.28e-04	8.09e-04	1.59e-04	4.50e-04	1.01e-03	2.69e-04	1.15e-03
9	1.04e-04	2.37e-05	7.76e-04	4.63e-05	2.32e-04	5.10e-04	1.16e+01
10	5.96e-05	7.42e-04	4.71e-04	5.05e-06	1.96e-04	4.16e-04	4.16e+01
11	2.50e-05	3.37e-04	1.47e-04	1.54e-03	1.16e+01	4.16e+01	5.84e-04
12	4.14e-04	4.10e-04	5.83e-04	2.32e-04	4.16e+01	1.16e+01	3.11e-04
13	1.21e-05	8.65e-04	1.16e+01	4.16e+01	8.40e-04	5.55e-04	8.82e-04
14	8.95e-04	3.69e-04	4.16e+01	1.16e+01	5.24e-04	9.88e-04	8.08e-04

Table A.8: Correlation values of the 7 ZCC CSSs (complex components separated) of length 63 - Part II

r_{cc}	**8**	**9**	**10**	**11**	**12**	**13**	**14**
1	3.27e-04	1.04e-04	5.96e-05	2.50e-05	4.14e-04	1.21e-05	8.95e-04
2	8.09e-04	2.37e-05	7.42e-04	3.37e-04	4.10e-04	8.65e-04	3.69e-04
3	1.59e-04	7.76e-04	4.71e-04	1.47e-04	5.83e-04	1.16e+01	4.16e+01
4	4.50e-04	4.63e-05	5.05e-06	1.54e-03	2.32e-04	4.16e+01	1.16e+01
5	1.01e-03	2.32e-04	1.96e-04	1.16e+01	4.16e+01	8.40e-04	5.24e-04
6	2.69e-04	5.10e-04	4.16e-04	4.16e+01	1.16e+01	5.55e-04	9.88e-04
7	1.15e-03	1.16e+01	4.16e+01	5.84e-04	3.11e-04	8.82e-04	8.08e-04
8	2.52e+02	4.16e+01	1.16e+01	4.22e-04	6.61e-04	6.42e-04	5.23e-04
9	4.16e+01	2.52e+02	4.44e-04	5.83e-04	9.07e-04	3.78e-04	3.54e-04
10	1.16e+01	4.44e-04	2.52e+02	1.20e-04	6.17e-04	7.14e-04	4.47e-04
11	4.22e-04	5.83e-04	1.20e-04	2.52e+02	1.04e-03	4.99e-04	2.98e-04
12	6.61e-04	9.07e-04	6.17e-04	1.04e-03	2.52e+02	2.40e-04	7.45e-04
13	6.42e-04	3.78e-04	7.14e-04	4.99e-04	2.40e-04	2.52e+02	6.99e-05
14	5.23e-04	3.54e-04	4.47e-04	2.98e-04	7.45e-04	6.99e-05	2.52e+02

ADDENDUM B

BELIEF PROPAGATION DECODING

B.1 ADDENDUM SUMMARY

This addendum contains an example of the belief propagation decoding algorithm considered in section 4.6.2.2 for a linear (N, K) block code. The example shows the decoding of the Hamming $(7, 4, 3)$ code, defined by the parity-check matrix (equation 4.67):

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix} \tag{B.1}$$

of which the factor graph (shown in figure 4.10) is shown again in figure B.1.

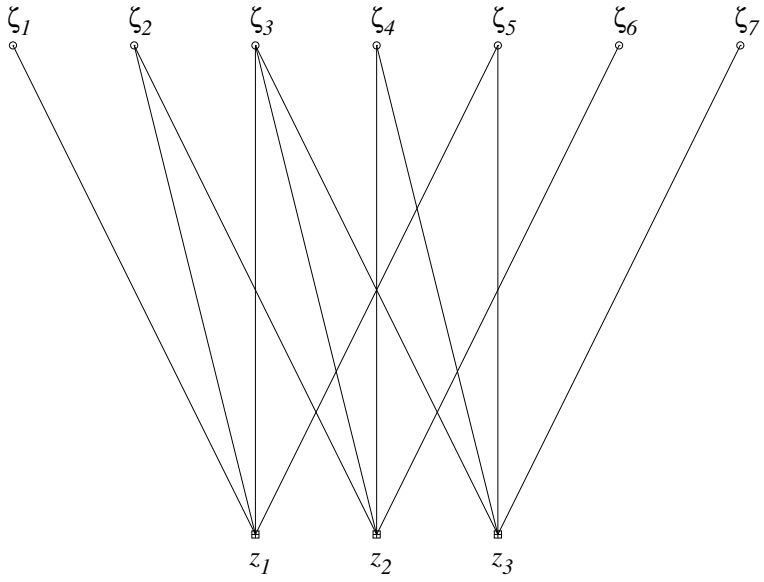

Figure B.1: Irregular factor graph of the Hamming $(7, 4, 3)$ code with code bit nodes $\zeta_n; n = 1, \ldots, N$ and check nodes $z_m; m = 1, \ldots, M$

B.2 INITIALISATION PHASE

After setting the code bit node probabilities $p_n^{t=\pm 1}$ using equations 4.80 and 4.81, the q-values are initialised by setting $q_{m,n}^{t=+1} = p_n^{t=+1}$ and $q_{m,n}^{t=-1} = p_n^{t=-1}$ for every edge connecting a code bit node ζ_n to a check node z_m as shown in figure B.2 below.

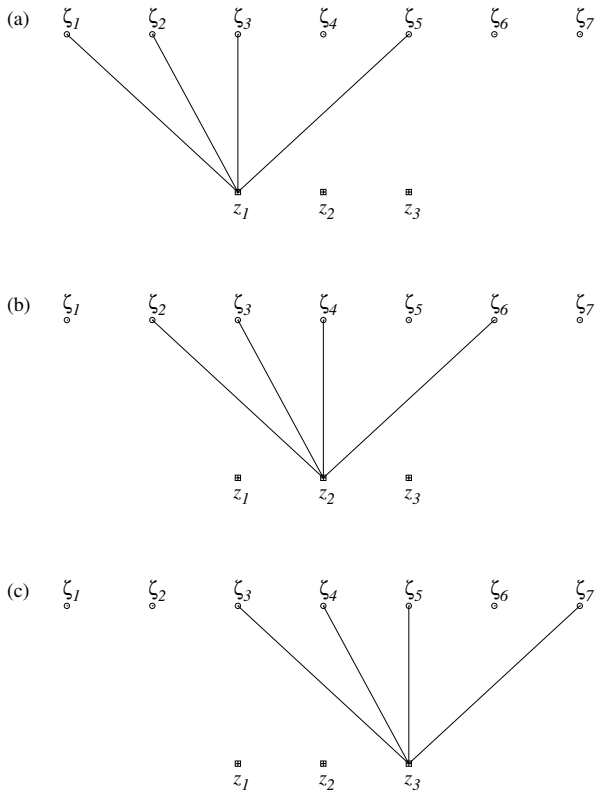

Figure B.2: Initialisation phase of belief propagation decoding

The q-values are given below according to figure B.2. The q-values associated with check node z_1 (figure B.2(a)) are:

$$q_{1,1}^{t=\pm 1} = p_1^{t=\pm 1} \tag{B.2}$$
$$q_{1,2}^{t=\pm 1} = p_2^{t=\pm 1} \tag{B.3}$$
$$q_{1,3}^{t=\pm 1} = p_3^{t=\pm 1} \tag{B.4}$$
$$q_{1,5}^{t=\pm 1} = p_5^{t=\pm 1} \tag{B.5}$$

The q-values associated with check node z_2 (figure B.2(b)) are:

$$q_{2,2}^{t=\pm 1} = p_2^{t=\pm 1} \tag{B.6}$$
$$q_{2,3}^{t=\pm 1} = p_3^{t=\pm 1} \tag{B.7}$$

$$q_{2,4}^{t=\pm 1} = p_4^{t=\pm 1} \tag{B.8}$$

$$q_{2,6}^{t=\pm 1} = p_6^{t=\pm 1} \tag{B.9}$$

The q-values associated with check node z_3 (figure B.2(c)) are:

$$q_{3,3}^{t=\pm 1} = p_3^{t=\pm 1} \tag{B.10}$$

$$q_{3,4}^{t=\pm 1} = p_4^{t=\pm 1} \tag{B.11}$$

$$q_{3,5}^{t=\pm 1} = p_5^{t=\pm 1} \tag{B.12}$$

$$q_{3,7}^{t=\pm 1} = p_7^{t=\pm 1} \tag{B.13}$$

B.3 HORIZONTAL DECODING STEP

In the second phase, the algorithm runs through the check nodes z_m and computes for each $n \in C_{bit}(m)$ (see equations 4.68 to 4.71) the probabilities $w_{m,n}^{t=\pm 1}$ which are calculated from products of the differences $\delta q_{m,n} = q_{m,n}^{t=-1} - q_{m,n}^{t=+1}$ (see equation 4.82).

Figure B.3 shows the horizontal decoding step associated with check node z_1 where the dotted edge illustrates which code bit node n does not participate in the calculation of $\delta q_{m,n}$.

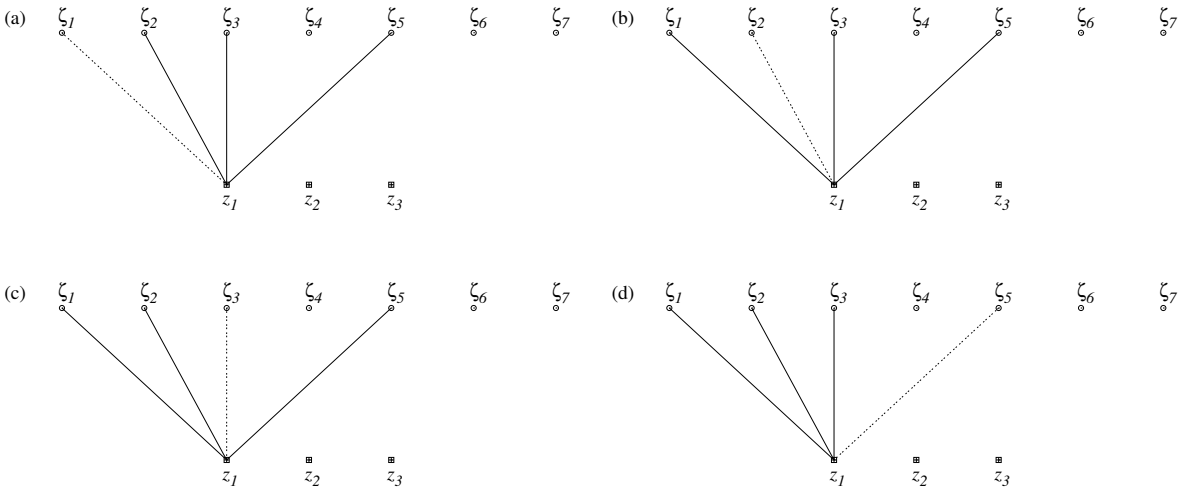

Figure B.3: Horizontal decoding step for check node z_1 where the values for $\delta w_{m,n}$ are calculated using the factor graph of the code

The δr-values associated with check node z_1 are:

$$\delta w_{1,1} = (\delta q_{1,2})(\delta q_{1,3})(\delta q_{1,5}) \tag{B.14}$$

$$\delta w_{1,2} = (\delta q_{1,1})(\delta q_{1,3})(\delta q_{1,5}) \quad \text{(B.15)}$$

$$\delta w_{1,3} = (\delta q_{1,1})(\delta q_{1,2})(\delta q_{1,5}) \quad \text{(B.16)}$$

$$\delta w_{1,5} = (\delta q_{1,1})(\delta q_{1,2})(\delta q_{1,3}) \quad \text{(B.17)}$$

where equations B.14 to B.17 apply respectively to figures B.3(a) to (d). The values $w_{m,n}^{t=\pm 1}$ are calculated by using the transformations given in equations 4.83 and 4.84.

The horizontal decoding step continues by calculating the values $w_{m,n}^{t=\pm 1}$ associated with check nodes z_2 and z_3, which is done exactly as for z_1.

B.4 VERTICAL DECODING STEP

In the third phase, the algorithm runs through the code bit nodes ζ_n and updates the probabilities $q_{m,n}^{t=\pm 1}$ using the probabilities $w_{m,n}^{t=\pm 1}$ calculated in the second phase, by applying equations 4.85 and 4.86 and performing probability normalisation using equations 4.87 and 4.88. In this section equations 4.85 to 4.88 are all combined into a single equation:

$$q_{m,n}^{t=\pm 1} = \alpha_{m,n} p_n^{t=\pm 1} \prod_{m' \in C_{check}(n) \setminus m} w_{m',n}^{t=\pm 1} \quad \text{(B.18)}$$

Figure B.4 shows the vertical decoding step associated with code bit nodes ζ_1, ζ_2 and ζ_3 where the dotted edge illustrates which check node m does not participate in the calculation of $q_{m,n}^{t=\pm 1}$.

The $q_{m,n}^{t=\pm 1}$-values associated with code bit node ζ_1 (figure B.4(a)) are:

$$q_{1,1}^{t=\pm 1} = p_1^{t=\pm 1} \quad \text{(B.19)}$$

which are the same values assigned in the original initialisation phase. Check node ζ_1 will therefore never be updated, because there is only a single connection between ζ_1 and the rest of the factor graph (in an LDPC code this will typically never happen).

The $q_{m,n}^{t=\pm 1}$-values associated with code bit node ζ_2 (figure B.4(b) and (c) respectively) are:

$$q_{1,2}^{t=\pm 1} = \alpha_{1,2} p_2^{t=\pm 1}(w_{2,2}^{t=\pm 1}) \quad \text{(B.20)}$$

$$q_{2,2}^{t=\pm 1} = \alpha_{2,2} p_2^{t=\pm 1}(w_{1,2}^{t=\pm 1}) \quad \text{(B.21)}$$

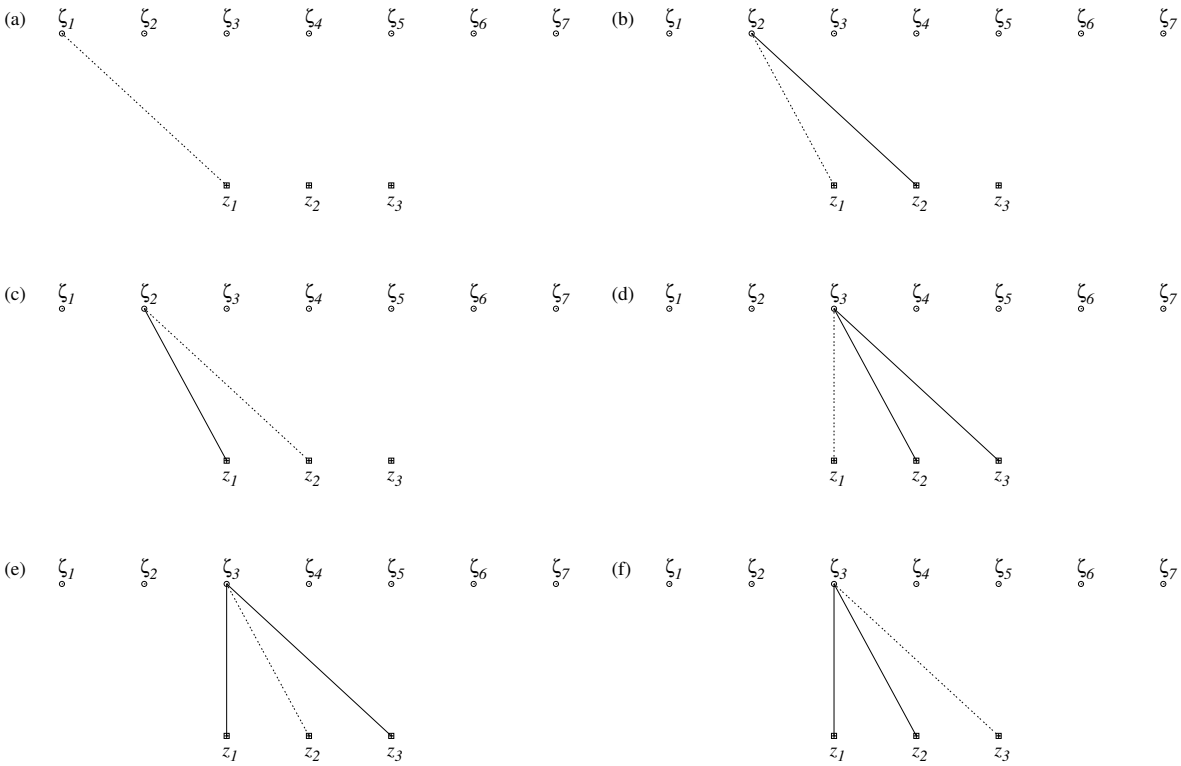

Figure B.4: Vertical decoding step for code bit nodes ζ_1, ζ_2 and ζ_3 where the values for $q_{m,n}^{t=\pm 1}$ are calculated using the factor graph of the code

The $q_{m,n}^{t=\pm1}$-values associated with code bit node ζ_3 (figure B.4(d),(e) and (f) respectively) are:

$$q_{1,3}^{t=\pm1} = \alpha_{1,3} p_3^{t=\pm1}(w_{2,3}^{t=\pm1})(w_{3,3}^{t=\pm1}) \quad \text{(B.22)}$$

$$q_{2,3}^{t=\pm1} = \alpha_{2,3} p_3^{t=\pm1}(w_{1,3}^{t=\pm1})(w_{3,3}^{t=\pm1}) \quad \text{(B.23)}$$

$$q_{3,3}^{t=\pm1} = \alpha_{3,3} p_3^{t=\pm1}(w_{1,3}^{t=\pm1})(w_{2,3}^{t=\pm1}) \quad \text{(B.24)}$$

The algorithm then continues to calculate the $q_{m,n}^{t=\pm1}$-values associated with code bit nodes ζ_4 to ζ_7.

B.5 FINAL DECODING STEP

After the maximum number of iterations are reached, the algorithm calculates the estimated posterior probabilities $q_n^{t=\pm1}$, similar to calculating $q_{m,n}^{t=\pm1}$ in the vertical decoding step, but without excluding any check node by applying equations 4.89 and 4.90.

The estimated posterior probabilities are:

$$q_1^{t=\pm1} = \alpha_1^{t=\pm1}(w_{1,1}^{t=\pm1}) \quad \text{(B.25)}$$

$$q_2^{t=\pm1} = \alpha_2^{t=\pm1}(w_{1,2}^{t=\pm1})(w_{2,2}^{t=\pm1}) \quad \text{(B.26)}$$

$$q_3^{t=\pm1} = \alpha_3^{t=\pm1}(w_{1,3}^{t=\pm1})(w_{2,3}^{t=\pm1})(w_{3,3}^{t=\pm1}) \quad \text{(B.27)}$$

$$q_4^{t=\pm1} = \alpha_4^{t=\pm1}(w_{2,4}^{t=\pm1})(w_{3,4}^{t=\pm1}) \quad \text{(B.28)}$$

$$q_5^{t=\pm1} = \alpha_5^{t=\pm1}(w_{1,5}^{t=\pm1})(w_{3,5}^{t=\pm1}) \quad \text{(B.29)}$$

$$q_6^{t=\pm1} = \alpha_6^{t=\pm1}(w_{2,6}^{t=\pm1}) \quad \text{(B.30)}$$

$$q_7^{t=\pm1} = \alpha_7^{t=\pm1}(w_{3,7}^{t=\pm1}) \quad \text{(B.31)}$$

The estimated posterior LLR for each code bit can then be calculated from equations B.25 to B.31, using equation 4.91 repeated here:

$$\Lambda(t_n|\mathbf{r}) = \ln(q_n^{t=+1}) - \ln(q_n^{t=-1}) \quad \text{(B.32)}$$

ADDENDUM C

SIMULATION SOFTWARE

C.1 ADDENDUM SUMMARY

The error performance results presented in Chapter 5 were obtained by performing extensive simulation studies which required the development of simulation software. All software was developed in *C++* and compiled with either the *GNU g++* or *Intel icpc* compilers on a *Linux* platform. All error performance results and related graphs presented in this dissertation were generated by *GNU Octave*. This addendum contains a brief overview of the design of the simulation software including a description of the *C++* classes and important functions developed during the course of this study.

C.2 SOFTWARE DESIGN

The object orientation programming philosophy was followed by developing several source files and classes separately and fusing them together using *GNU make* with the *Makefile*-file defining source files, objects, executables, the compiler and related parameters.

Each error performance simulation program takes inputs from the command line and header files containing configuration settings and variables. The command line inputs are listed below.

- The error limit which is the number of errors the program must count before completion of a single ε_b/N_0 simulation
- The energy per bit per noise power spectral density in dB (ε_b/N_0)
- The number of simultaneous CDMA users
- The number of decoding iterations
- The names and directories of output files

The output includes the simulated bit error probability, which is written periodically to a progress output file as the simulation continues. The final simulated error probability is written to another output file after completion of the simulation. The output files are used by *GNU Octave* to generate BER curves.

To coordinate the simulation of a range of ε_b/N_0 points to create a BER curve, *Perl* scripts were used. These scripts were loaded and executed on multi-node computing clusters (the *Ipercube* and *Collective* found in the *Department of Electrical, Electronic and Computer Engineering, University of Pretoria*) together with the accompanying simulation program packages.

C.2.1 PRIMARY HEADER FILES

Several header files were used to define general configuration settings, programming variables and constants. Most header files were generated by supporting configuration programs, e.g. large LDPC matrices were generated and stored by programs employing random computer search algorithms. The header files used in all the simulation programs developed during this study are listed in table C.1.

Table C.1: Simulation program primary header files

Header file	Contents
codeX.h	CSSs; X is the sequence number
constvar.h	Coderate, **G**, **H**, interleaver index array
delayprofile.h	Power delay profile settings
genset.h	General configuration settings
NKMonly.h	Channel code dimensions

C.2.2 C++ CLASSES

The *C++* classes, host files and member functions developed during the course of this study are given in table C.2 to C.4. Public and private functions are given respectively as *italic* and roman text.

Table C.2: Simulation program primary classes - Part I

Files and class	Functions	Description
enc3D.h enc3D.cc encode3DTurbo	*cube_encode*	Generates random data, performs interleaving and populates 3D code cubes
	encode	Performs systematic LBC encoding
	antipode	Generates random antipodal numbers using the Wichmann-Hill algorithm
linearblockSOVA.h linearblockSOVA.cc LBSOVA	*calculate_trellis*	Constructs LBC trellis from **H**
	nodeXOR	Calculates next node address by performing XOR operation
	dec2bin	Converts decimal numbers (trellis depth) to binary words
	cube_decode	Manages 3D cube decoding by performing interleaving and calculation of extrinsic posterior probabilities
	decode	Calculates metrics of a codeword in the forward trellis direction
	Backtrace_decode	Calculates metrics of a codeword in the reverse trellis direction
	SOVA_decode	Calculates soft output values of a codeword using the forward and reverse trellis metrics
	min_or_max_val	Returns minimum or maximum value of 2 parameters given, depending on simulation configuration. Used to calculate survivor branches.
	validNode	Determines if a given trellis address exists in the trellis defined by **H**
	BranchMetric	Calculates branch metric value for redundant bits
	BranchMetricX	Calculates branch metric value for information bits

Table C.3: Simulation program primary classes - Part II

Files and class	Functions	Description
mpxdecode.h mpxdecode.cc messagepassing	*analyzeH*	Analyse **H** to construct factor graph
	decode1	Performs generic message passing decoding using Pearl's belief propagation algorithm
qpskawgn.h qpskawgn.cc AWGNchannel	*qpskmodawgnchan*	Performs QPSK modulation and introduces AWGN channel effects into the modulated signal
	AWGN_sample	Generates a zero-mean, unity-variance sample with Gaussian PDF
	WichmanHill	Generates a uniform random number between 0 and 1
RAdecode.h RAMP	*decodeRA*	Performs LLR MP decoding algorithm on RA code's factor graph
main.cc channelcorruption	*AWGN*	Generates a zero-mean, unity-variance sample with Gaussian PDF
	init_filters	Initialise Doppler filters before start of simulation
	chAWGN	Adds multiple user data together in the AWGN channel
	multipath	Simulates a single path in the multipath channel
	WichmannHill	Generates statistically independent uniform random numbers
	elipIIR	Performs elliptic filtering on uniform numbers
	AWGN_Doppler	Generates uniform numbers for elliptic filter
	Doppler	Performs interpolation to produce Doppler filtered samples
main.cc Q4DMod	*spreadcss*	Performs 4D modulation
	demodulate	Performs carrier phase synchronisation and integrate-and-dump demodulation

Table C.4: Simulation program primary classes - Part III

Files and class	Functions	Description
ZCC4D.cc ZCC2D.cc DSB4D.cc CSS	*ZadoffChu*	Generates a family of ZC CSSs
	DSB	Generates CE-GCL CSSs by performing RU filtering on ZC CSSs
	eqenergy	Equalise power of each CSS
	gcd	Calculates greatest common divider of 2 integers

C.2.3 SPECIAL FUNCTIONS

Apart from the classes given in section C.2.2, a number of special functions were developed separately. Table C.5 lists these functions, their header files and a description of each.

Table C.5: Simulation program special functions

Files	Functions	Description
LDPCencNeal.h	encodeNealDense	Performs Neal's sparse matrix encoding method
LDPCXenc.h	encode	Performs Richardon and Urbanke's approximate triangular efficient encoding method
loadcodes.h	loadandequalise	Loads CSSs from header files and performs power scaling
RAenc.h	encode	Performs RA encoding using **G**
RMenc.h	encode	Performs systematic LBC encoding

C.3 ADDITIONAL SOFTWARE

Several software modules, including mostly *Octave* and some *C++* programs were developed to perform essential support and test functions. Table C.6 lists functions related to Chapter 2 including modules that generate CSSs, evaluate correlation, generate and plot spectra and functions to study the power output of the MD modulator. Table C.7 shows functions related to Chapter 3 and include software to create, evaluate and verify the channel simulator. Table C.8 shows functions related to Chapter 4 and include

software to generate **G** and **H** matrices, trellises, factor graphs and modules to test encoding and decoding algorithms. Table C.9 shows functions related to Chapter 5 and include software to calculate channel capacity and to plot measured error performance results.

Table C.6: Development software related to the WCDMA platform

Files	**Description**
Chu.m DSB.m	Generates Chu or CE-GCL CSS family of specified length and number of samples per chip
processDSB.cc	Reads CSSs, performs processing and writes output as header files
aperiodic.m	Calculates aperiodic correlation between 2 CSSs
periodic.m	Calculates periodic correlation between 2 CSSs
specABS.m	Calculates and plots absolute PSD of a given CSS
specRe.m	Calculates and plots real PSD of a given CSS
specIm.m	Calculates and plots imaginary PSD of a given CSS
avgpow.m	Calculates average power content of a given CSS
pins.m	Calculates instantaneous output power of a given CSS
pshort.m	Calculates short-time average output power of a given CSS
plotDSBmanifold.m	Calculates and plots CSS basis vectors and the complex manifold
calculateXM.m	Calculates and plots synchronous aperiodic correlation between 2 complete CSSs
calcXMplain.m	Calculates and plots synchronous aperiodic correlation between components of 2 CSSs
plotCOR.m	Calculates and plots cross dot products between basis vectors

Table C.7: Development software related to channel models

Files	Description
Channel.cc	Simulates Clarke's channel model and evaluates I/O of channel
ChanF_2D.m	Performs tests on channel model using complex 2D modulation
ChanF_4D.m	Performs tests on channel model using complex 4D modulation
design2.m	Designs Doppler filter and plots frequency response
testD.m	Evaluates Doppler filter characteristics
tryInt.m	Interpolates Doppler sequence and plots frequency response
designchannel.m	Designs a multipath frequency selective fading channel
expprofile.m	Plots power delay profile of multipath channel given the path powers and relative delay times
profile.m	Calculates mean excess delay, RMS delay spread, coherence bandwidth of a given power delay profile
main_awgntry.cc	Generates AWGN samples and corrupted bit sequence
main_flatfading.cc	Generates fading amplitude and phase and flat faded bit sequence
plotpdf.m	Calculates and plots amplitude PDF of AWGN samples
plotRice.m	Calculates and plots PDF of fading amplitude
plotPhase.m	Calculates and plots PDF of fading phase angle
plotPx.m	Plot the time signal of path x in the multipath channel
plotPathxSpec.m	Calculates and plots PSD of path x in the multipath channel

Table C.8: Development software related to channel coding

Files	Description
calcG.m	Calculates **G** of systematic LBC
LDPCenc.cc	Richardson and Urbanke's approximate lower triangular check matrix test and evaluation program
Konqueror.cc	Program to count number of codewords satisfying nullspace of Gallager codes
genHapLT.cc	Creates approximate lower triangular LDPC matrix (Richardson and Urbanke) using random search method under row, column and cycle length constraints
gen_H.m	Creates LDPC matrix using a deterministic approach
importH.cc makeH.m	Creates header file containing LDPC and sub-matrices
RMtrellis.m	Calculates and plots forward direction LBC trellis structure using **H**
RMtrellisReverse.m	Calculates and plots reverse direction LBC trellis structure using **H**
RMtrellisExpurgate.m	Calculates and plots expurgated LBC trellis structure using **H**
Htest.m	Creates LDPC code's factor graph from **H**
RAG.m	Creates RA code's factor graph
HammBP.cc	Belief propagation test program on Hamming codes
BeliefP.cc	Generic belief propagation test program
genRA.m	Generates RA code generator matrix **G**
RA_MP.cc	Tests RA message passing decoder
script1.sh readtxt.m	Generates LDPC **G** and **H** matrices, using certain modules of Neal's software
Hamming_on_AWGN.cc	Evaluates Viterbi decoding (on Hamming codes)

Table C.9: Development software related to error performance calculation

Files	Description
BIAWGN.m	Matches capacity and binary entropy functions to plot capacity curve
BIAWGN2.m	Calculates channel capacity (see equation 5.3) of BI-AWGN curve using numerical integration
C_BIAWGN.m	Evaluates integral argument at a single function value
simpson.m	Evaluates capacity integral using C_BIAWGN.m
ShannonBI2.m	Calculates Shannon limit BER curve
biboawgn.m	Calculates capacity limit of a BIBO AWGN channel
bsc.m	Calculates capacity limit of a BSC channel
invH2.m	Calculates inverse of binary entropy function using Newton-Raphson method
QAMBER.m	Calculates BER of a QAM communication system
plotHC.m	Plots binary entropy function and capacity versus error probability for a given code rate
compare.m	Plots theoretic and measured error probability curves

LIST OF FIGURES

LIST OF TABLES