

University of Windsor Scholarship at UWindsor

Electronic Theses and Dissertations

2009

Performance Enhancement of the OFDM-Based DSRC System Using Frequency-Domain MAP Equalization and Soft-Output Demappers

Nabih Jaber
University of Windsor

Follow this and additional works at: <http://scholar.uwindsor.ca/etd>

Recommended Citation

Jaber, Nabih, "Performance Enhancement of the OFDM-Based DSRC System Using Frequency-Domain MAP Equalization and Soft-Output Demappers" (2009). *Electronic Theses and Dissertations*. Paper 355.

This online database contains the full-text of PhD dissertations and Masters' theses of University of Windsor students from 1954 forward. These documents are made available for personal study and research purposes only, in accordance with the Canadian Copyright Act and the Creative Commons license—CC BY-NC-ND (Attribution, Non-Commercial, No Derivative Works). Under this license, works must always be attributed to the copyright holder (original author), cannot be used for any commercial purposes, and may not be altered. Any other use would require the permission of the copyright holder. Students may inquire about withdrawing their dissertation and/or thesis from this database. For additional inquiries, please contact the repository administrator via email (scholarship@uwindsor.ca) or by telephone at 519-253-3000ext. 3208.

Performance Enhancement of the OFDM-Based DSRC System
Using Frequency-Domain MAP Equalization and Soft-Output
Demappers

By

Nabih Jaber

A Thesis

Submitted to the Faculty of Graduate Studies
through Electrical Engineering
in Partial Fulfillment of the Requirements for
the Degree of Master of Applied Science at the
University of Windsor

Windsor, Ontario, Canada

2009

© 2009 Nabih Jaber

All Rights Reserved. No Part of this document may be reproduced, stored or otherwise retained in a retrieval system or transmitted in any form, on any medium by any means without prior written permission of the author.

Author's Declaration of Originality

I hereby certify that I am the sole author of this thesis and that no part of this thesis has been published or submitted for publication.

I certify that, to the best of my knowledge, my thesis does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my thesis, including any final revisions, as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

Abstract

5.9 GHz Dedicated Short-Range Communication (DSRC) systems are based on the orthogonal frequency division multiplexing (OFDM) systems. OFDM systems are well known for their abilities to combat inter symbol interference (ISI) in time-invariant, frequency-selective channels.

We propose a receiver design to enhance the overall performance of the DSRC system to combat ICI instead of ISI caused by the time-varying channel. Most researchers focus on the time-domain channel model and MAP equalization to combat ISI. It is shown that the proposed receiver design outperforms the conventional DSRC system by up to and around 15 dB for the same bit error rate (BER). This improvement was achieved through a worst-case scenario study, i.e. time-varying Rayleigh faded channel.

Furthermore, soft demapping schemes were compared with one another at varying SNRs and velocities. When tested at worst-case scenario environments, very little improvements in the DSRC performance were achieved when it came to the soft demappers.

In the name of God, most gracious, most merciful.

For my father Riad Ali Jaber, who although is no longer with us to see the conclusion of this work, he gave support as no other could, even when he was far. For my mother Hannah Jaber, my wife Mahbegum Azimy, her parents and her family, and our two beloved children Riad and Iman. Also, for my supportive siblings Fouad and Ghinwa Jaber.

Acknowledgements

There are several people who deserve my sincere gratitude and appreciation for their invaluable contribution for this project.

I would first like to express my sincere appreciation to my supervisors, Dr. Kemal E. Tepe and Dr. Esam Abdel-Raheem for providing me with constant support, generosity and guidance throughout the entire project. They had a tremendous impact on me both academically and personally. I am honoured to have worked with them. I would also like to thank my committee members Dr. Mohammed A.S. Khalid and Dr. Nader Zamani for their support, guidance and time spent observing my project, and providing me with invaluable comments and suggestions. Also, I would like to thank my forever friend Bill Cassidy for his support throughout both my undergrad and graduate studies. He has always been there for me and my family whenever we needed a helping hand. Truly, a friend in need is a friend indeed. Also, I would like to thank Ishaq and Izhar for their beneficial suggestions.

Many thanks go to my parents Riad Ali Jaber, and Hannah Jaber, whom have sacrificed so much for our well being and protection. Although my father is no longer with us to see the conclusion of this work, he gave support as no other person could, even when he was far and ill. Thank you dad, and you will always be on our minds. God bless your soul. My mom's ongoing support and advice will never grow old. I would like to also thank my very supportive and caring siblings Fouad and Ghinwa Jaber. I am very grateful and thankful to God for giving me such wonderful parents and siblings.

Additionally, I would like to thank my beautiful wife Mahbegum Azimy, and my two most beloved children, Riad and Iman. My children's funny and relaxing gestures with their cute expressions keep me going in life. Thank you for all your patience and your confidence in my ability. I truly am blessed to have them in my life. My wife's talent and amazing character will always be an inspiration to me. Last but not least, I would like to thank my in-laws for their warm support and faith.

Table of Contents

Author’s Declaration of Originality	iii
Abstract.....	iv
Dedication	v
Acknowledgements	vi
List of Tables	x
List of Figures.....	xi
List of Abbreviations	xiv
Chapter 1 Introduction.....	1
1.1 Orthogonal Frequency Division Multiplexing	2
1.2 WAVE Channel Model.....	5
1.3 Organization of the thesis	10
Chapter 2 FEC coded and OFDM DSRC systems	12
2.1 Channel Coding Theory.....	12
2.2 The DSRC System Model.....	12
2.3 The DSRC Transmitter	14
2.3.1 Convolution Encoder.....	15
2.3.2 Interleaving.....	18
2.3.3 Modulation	19
2.3.4 IFFT.....	20
2.3.5 Guard Interval.....	20
2.4 DSRC Receiver.....	21
2.4.1 The time domain.....	21
2.4.2 The frequency domain.....	22
2.4.3 Channel Estimation	22

2.4.4	Signal Compensation.....	23
2.4.5	Demodulation and Deinterleaving.....	23
2.4.6	Decoding	23
Chapter 3	Hard vs. Soft Decoding Algorithms and Proposed Soft Demapping Scheme.	24
3.1	Optimal detection in the estimation of the states of a Markov process.....	25
3.2	The Viterbi Algorithm (VA/SOVA).....	25
3.2.1	Hard-decision Viterbi (VA).....	26
3.2.2	Soft-decision Viterbi (SOVA).....	30
3.3	BCJR/ MAP (Maximum A Posteriori) Algorithm	31
3.3.1	Forward Recursion	32
3.3.2	Backward Recursion.....	33
3.3.3	State Transition Matrix.....	33
3.3.4	APPs of the Symbols	34
3.4	Performance of DSRC Systems using Different Demapping and Decoding Schemes	35
3.4.1	Proposed Linear-based Demapper.....	37
3.4.2	Proposed LLR/Probabilistic Demapper	39
3.4.3	S and Π -Decision Demapper.....	40
3.4.4	Simulation Results.....	41
Chapter 4	Performance Enhancement of DSRC Systems	54
4.1	Tools for (Iterative) Decoding of Binary Codes	54
4.1.1	From LLR to Probabilistic Values.....	55
4.2	Proposed System Enhancement Using Frequency-Domain MAP Equalization	57
4.2.1	Simplified Channel Model in Frequency-Domain.....	58
4.2.2	State Diagrams of the F-D Channel Model	61
4.2.3	Frequency Domain MAP Equalization Model	62
4.2.4	Decoder Model.....	64
4.3	Simulation Results	66
Chapter 5	Conclusions and Future Works	74
References		76
Appendix A	Graphical User Interface (GUI) and Simulator for OFDM Systems	79
A.1	Introduction.....	79
A.2	Advantages	80
A.2.1	ease of use	80
A.3	Simulation Environment.....	80
A.3.1	Environment overview	80

A.3.2	Simulation and System Parameters	81
A.3.3	Model-Based/Object-Oriented Engineering	81
A.3.4	Simulation Configuration.....	82
A.3.5	Iterative Case.....	82
A.3.6	System Configuration.....	83
A.3.7	Simulation Settings in the GUI	83
A.4	Output Comparisons and Plotting	89
A.4.1	Plotting.....	89
A.5	Other Consideration.....	91
A.5.1	Motivation.....	91
A.6	Design.....	91
A.6.1	Language and Objects	91
A.6.2	Graphical Design Around a Simulation	91
A.6.3	Flexibility.....	92
A.6.4	Adding new components for simulating	92
A.6.5	Wrapper Object.....	92
A.6.6	Application to other Platforms	92
A.6.7	Simulation to Implementation.....	92
A.7	Performance.....	92
A.7.1	Parallelism.....	92
A.7.2	Alternative Improvements.....	93
A.8	Further Reading	93
Vita Auctoris	94	

List of Tables

Table 2-1: State transition of convolution example.....	17
Table 3-1: Example of SOVA Decoder.....	30
Table 3-2: Simulation Values.....	42
Table 4-1: Numerical Example of LLR.....	57
Table 4-2: State Diagram for Simplified F-D Channel Model with BPSK Modulated Input.....	61
Table 4-3: Simulation Values.....	66

List of Figures

Figure 1.1 (a) Multi-carrier Modulation and (b) Demodulation Using Correlators.....	2
Figure 1.2 Spectrum of the OFDM signal with 4 subcarriers.....	3
Figure 1.3 OFDM (a) Transmitter and (b) Receiver Systems	4
Figure 1.4 Jakes Fading Simulator	6
Figure 1.5 Channel Model with multipath scattering, specular (LOS), and Rayleigh (no-LOS) portions and AWGN[14].....	6
Figure 1.6 Scatter plots showing QPSK constellation distortion under various frequency shifts	7
Figure 1.7 view of scatter plots showing QPSK constellation distortion under various frequency shifts (vertical represents data samples).....	8
Figure 1.8 Scatter plots showing QPSK with low fade and noise of $10 E_b/N_0$	8
Figure 1.9 dispersion of amplitude and phase in the time domain for QPSK with no AWGN	9
Figure 1.10 side view of dispersion of amplitude and phase in the time domain for QPSK with no AWGN.	9
Figure 1.11 Samples vs. magnitude of 1 packet, 80 OFDM symbols under varying doppler frequencies....	10
Figure 2.1 DSRC System Model	12
Figure 2.2 DSRC transmission sequence	13
Figure 2.3 DSRC transmission packet format (time vs. frequency domain grid).....	14
Figure 2.4 Transmitter model of the DSRC system	15
Figure 2.5 Convolution Encoder with Rate of 1/2 and a constraint length of $K=7$ and a generator matrix based on 133 171(8)	15
Figure 2.6 Convolution Encoder with Rate of 1/2 and a constraint length of $K=3$ and a generator matrix of [58 78]	16
Figure 2.7 FSM chart for basic convolutional code	16
Figure 2.8 Transitions for RSC codes[23].....	17
Figure 2.9 Transition for NRC codes[23].....	17
Figure 2.10 Block Interleaver / De-Interleaver	18
Figure 2.11 Example QPSK (QAM4) Modulated Signal Constellations	19
Figure 2.12 Normalized signal constellations	19

Figure 2.13 OFDM frame with cyclic extension	20
Figure 2.14 DSRC Receiver	21
Figure 3.1 Decoding Algorithms	24
Figure 3.2 Viterbi Algorithm diagram labels	26
Figure 3.3 Viterbi Algorithm for (7, 5) Convolution Encoder	28
Figure 3.4 Example of Trellis Diagram for SOVA Decoder	30
Figure 3.5 System Diagram of the MAP decoding Algorithm	35
Figure 3.6 Gray-Coded of Constellation Points for QPSK, 16 QAM and 64 QAM Mappers.....	36
Figure 3.7 The demapping regions of all message bits sequence	38
Figure 3.8 Mapping of 16 QAM In-Phase Symbols into Binary Elements	39
Figure 3.9 S and II-Decision Rules for 16 QAM	41
Figure 3.10 16QAM: Conventional vs. Soft demappers using soft-input Viterbi under Rayleigh fading at 238 km/h.....	43
Figure 3.11 16QAM: Conventional vs. Soft demappers using BCJR under Rayleigh fading at 238 km/h under Rayleigh fading at 238 km/h	44
Figure 3.12 16QAM: Conventional vs. soft demappers using both soft-input Viterbi and BCJR under Rayleigh fading at 238 km/h under Rayleigh fading at 238 km/h.....	45
Figure 3.13 16QAM: Conventional vs. Soft demappers using soft-input Viterbi under Ricean fading channel $K=1$	46
Figure 3.14 QPSK: Conventional vs. Soft demappers using soft-input Viterbi under Ricean fading channel $K=1$	47
Figure 3.15 QPSK: Conventional vs. Soft demappers using soft-input Viterbi under Rayleigh fading at 238 km/h	48
Figure 3.16 16QAM velocity: Conventional vs. Soft demappers using soft-input Viterbi under Rayleigh fading varying velocities at 10dB.....	49
Figure 3.17 QPSK velocity: Conventional vs. Soft demappers using soft-input Viterbi under Rayleigh fading varying velocities at 30dB.....	50
Figure 3.18 16QAM velocity: Conventional vs. Soft demappers using BCJR under Rayleigh fading varying velocities at 30dB	51
Figure 3.19 QPSK velocity: Conventional vs. Soft demappers using BCJR under Rayleigh fading varying velocities at 30dB	52
Figure 3.20 QPSK: Conventional vs. Soft demappers using soft-input Viterbi under Rayleigh fading at varying high SNR.....	53
Figure 4.1 SISO Decoder	54
Figure 4.2 Proposed Receiver Design	58
Figure 4.3 Simplified Frequency Domain Channel Model.....	59
Figure 4.4 Subcarrier to Subcarrier ICI Contributions	60

Figure 4.5 Proposed vs. Conventional BPSK, QPSK modulation, with and without perfectly known coefficients at $f_d=1300\text{Hz}$	67
Figure 4.6 Proposed vs. Conventional BPSK modulation at $f_d=1300\text{Hz}$	68
Figure 4.7 Proposed vs. Conventional BPSK modulation at $f_d=550\text{Hz}$	69
Figure 4.8 Proposed vs. Conventional QPSK modulation at $f_d=1300\text{Hz}$	70
Figure 4.9 Proposed vs. Conventional QPSK modulation at $f_d=550\text{Hz}$	71
Figure 4.10 Soft Demapping BCJR vs. Proposed.....	72
Figure A.1 Illustration of Information exchange between GUI and Simulator	81
Figure A.2 Tiered Simulation with each path resulting in dataset for output.....	82
Figure A.3 Modified Tiered Simulation to include Iteration.....	83
Figure A.4 Main GUI page.....	84
Figure A.5 GUI Layout Sections.....	84
Figure A.6 Encoder Settings	85
Figure A.7 Decoder Selection process	85
Figure A.8 Transmitter and Receiver constants for interoperability between components	86
Figure A.9 varying E_b/N_0	86
Figure A.10 varying Doppler shift	87
Figure A.11 output settings	87
Figure A.12 configurable channel properties	88
Figure A.13 default action is to create a new simulation message	88
Figure A.14 msg load section with new message disabled, showing name of the message to load message.	88
Figure A.15 Choose the number of paths	89
Figure A.16 choose the channel noise method	89
Figure A.17 Different Receivers with different demodulations schemes.....	90
Figure A.18 QAM16 scatter plot at $E_b/N_0 = 20\text{dB}$	90

List of Abbreviations

The notation used in this thesis is as follows. $LLR(.)$ and $L(.)$ will be used interchangeably to mean the log-likelihood ratio operation on $(.)$. In general, the non-bold face upper case letters denote matrices in time-domain, while bold face upper case letters are used to denote frequency-domain matrices. Non-bold face lower case letters represent the time-domain of one data symbol (modulated symbol), while bold face lower case letters represent the frequency-domain of one data symbol. $[.]$ denotes a one-dimension array or a signal vector. $\langle.\rangle_{sc}$ represents the modular sc operation. $(.)^H$ denotes the complex conjugate transpose (Hermitian). $.$ denote a componentwise matrix multiplication, while \cdot operation is used when denoting a regular matrix multiplication, for example $A \cdot B$ is a componentwise matrix multiplication between A and B . $X_{a,b}$ indicates the a – th row and the b – th column of X . Some commonly used operators, abbreviations and symbols are listed below.

<u>Abbreviation</u>	<u>Definition</u>
APP	A Posteriori Probabilities
AWGN	Additive White Gaussian Noise
BCJR	Bahl, Coke, Jelinek, Raviv
BCJRA	Bahl, Coke, Jelinek, Raviv Algorithm
BER	Bit Error Rate
BPSK	Binary Phase Shift Keying
CIR	Channel Impulse Response
dB	Decibel
DFT	Discrete Fourier Transform
DMC	Discrete Memoryless Channel
DSRC	Dedicated Short Range Communication
FD or fd	Doppler Frequency
FEC	Forward Error Correction
FER	Frame Error Rate
FFT	Fast Fourier Transform
FIR	Finite Impulse Response

<u>Abbreviation</u>	<u>Definition</u>
FSM	Finite State Machine
FSSM	Finite State Sequential Machine
GF(g)	Galois or finite field
ICI	Inter Carrier Interference
IDFT	Inverse Discrete Fourier Transform
IEEE	Institute of Electrical and Electronics Engineers
IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response
ISI	Inter Symbol Interference
LLR	Log Likelihood Rate
MAP	Maximum A Posteriori
MAPdec	MAP Decoders
MAPEq	MAP Equalizer
ML	Maximum Likelihood
ML	Maximum Likelihood
MMLSE	Maximum Likelihood Sequence Estimation
NRC	Non-Recursive Codes
OFDM	Orthogonal Frequency Division Multiplexing
PDF	Probability Distribution Function
PER	Packet Error Rate
PHY	Physical Layer
QPSK	Quadrature Amplitude Modulation
RSC	Recursive systematic codes
SER	Symbol Error Rate
SISO	Soft Input Soft Output
SNR	Signal-to-Noise Ratio
SOVA	Soft-Output Viterbi Algorithm
VA	Viterbi Algorithm (Hard-Decision)
WAVE	Wireless Access Vehicular Environment

List of Symbols

T_c	Coherence Time
B_c	Coherence Bandwidth
N	Number of OFDM frequency Subcarriers
N_{sc}	sc^{th} Subcarrier of OFDM frequency Subcarrier

R_c	Code Rate
T_s	OFDM symbol duration
v	Velocity
f_c	Carrier Frequency
f_d	Doppler frequency
$(.)_8$	Octal representation
m	number of constellation points in a modulation scheme
O_{sc}	modulation waveform at sc^{th} subcarrier
X_{sc}	data symbol at sc^{th} subcarrier
R_d	data rate
t_r	time resolution
sc	Subcarrier or an integer used when used as a subscript to denote the location of the subcarrier

Chapter 1

Introduction

In 1999, the U.S Federal Communication Commission (FCC) allocated a 75 MHz spectrum at 5.9 GHz for Dedicated Short-Range Communication (DSRC) system for services that involve vehicle-to-vehicle and vehicle-to-roadside communications [1]. Although the DSRC band is a licensed spectrum, the FCC does not charge a fee for its usage [1]. The main purpose of its establishment is for improving road safety. The program, which is referred to as ‘Vehicular Infrastructure Integration’ (VII), is being considered by the United States Department of Transportation (USoT) [2]. The DSRC system is a short to medium range communication system, i.e. the distance range between the transmitter and the receiver should be around a 1000 meters. That is, unlike some other radio communication systems such as cellular or FM radio, where their communication range are in kilometres and hundreds of kilometres, respectively [3].

DSRC is also known as the IEEE 802.11p Wireless Access in Vehicular Environment (WAVE) [1], [2], and was based on the IEEE 802.11a standard [4]. The main difference between these two standards is the increase of the symbol duration in DSRC that resulted from the 10 MHz reduction in the channel spacing compared to the IEEE 802.11a standard, with 20 MHz channel spacing. IEEE 802.11a is one of the standards used in wireless local area network (WLAN), and was designed for time-invariant channels suitable for stationary indoor environments with low delay spread; hence it makes sense to extend the symbol duration in DSRC, which was justified in [5]. The DSRC physical layer, which is the main focus of this thesis, employs orthogonal frequency division multiplexing (OFDM) physical layer technique [6]. OFDM is popular for its ability to mitigate inter symbol interference (ISI) through the use of sufficient guard interval.

Due to the fact that DSRC is an OFDM-based system, the first section (section 1.1) of this chapter will be dedicated for it. An understanding of the characteristics of the radio channel plays a big role in trying to equalize the message signal to mitigate the effect of the channel on the DSRC system; hence section 1.2 will present the channel model of the DSRC system. Finally, section 1.3 will contain the objectives of this thesis.

1.1 Orthogonal Frequency Division Multiplexing

The basis for the DSRC Physical Layer is that of the 802.11a OFDM transmitter/receiver combination.

Orthogonal Frequency Division Multiplexing (OFDM) uses a single serial transmission that contains multiple subcarriers. OFDM was first introduced by Chang and Gibby in [6], for the purpose of transmitting information across harsh channel environments. This technique works by joining equally spaced subcarriers into one serial transmission using IFFT transformation.

The IFFT/FFT (Inverse Fourier Transform/ Fast Fourier Transform) are not the only method for modulation/demodulation in OFDM. Actually, in [7] the IFFT/FFT methods were shown to produce similar results to a different method used to generate the OFDM system and its orthogonality. The modulation uses oscillators and the demodulation uses matched filters (correlator implementation) as shown in Figure 1.1(a) and (b), respectively.

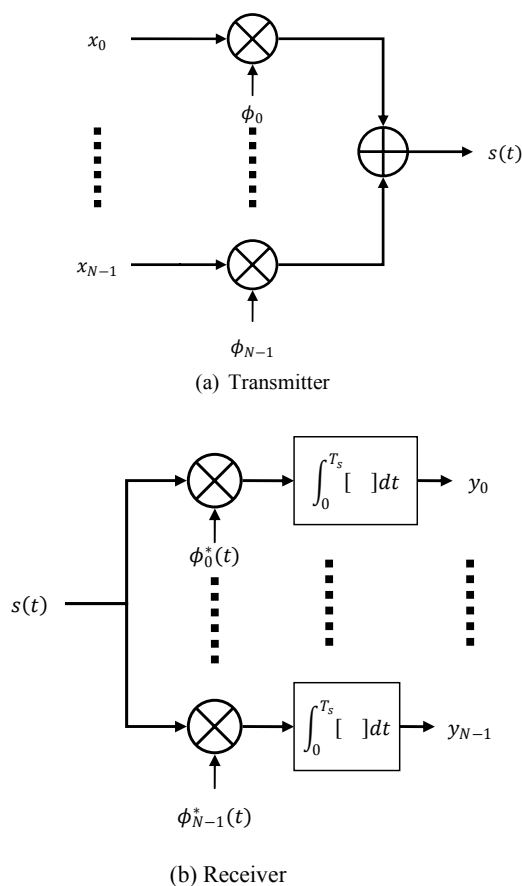


Figure 1.1 (a) Multi-carrier Modulation and (b) Demodulation Using Correlators

The OFDM signal can be written as follows [8],

$$s(t) = \sum_{sc=0}^{N-1} x_{sc} \theta_{sc}(t) \quad (1-1)$$

where:

$s(t)$ is the modulated multi-carrier signal and at time t .

x_{sc} is the data symbol at sc^{th} subcarrier, where $sc^{th} = 0, 1, 2, \dots, N$.

$\theta_{sc}(t)$ is the modulation waveform at sc^{th} subcarrier and at time t .

This relationship is also shown in Figure 1.1(a). For orthogonality the modulation waveform $\theta_{sc}(t)$ is chosen as a set of orthogonal waveforms [9][8],

$$\theta_{sc}(t) = \begin{cases} \frac{1}{\sqrt{T_s}} e^{j\pi f_{sc} t}, & t \in [0, T_s] \\ 0, & \text{otherwise} \end{cases} \quad (1-2)$$

f_{sc} is the frequency of the sc^{th} subcarrier. We can note that the window restriction of $[0, T_s]$ gives us the sinc functions that we see in frequency domain, as shown in Figure 1.2. The different colorings and/or symbols on the sinc function plots represent different subcarriers that are orthogonal to one another (sc1, sc2, sc3, sc4).

For the demodulation equation of Figure 1.1(b), we get [8],

$$y_{sc} = \int_0^{T_s} s(t) \theta_{sc}^*(t) dt \quad (1-3)$$

where

$s(t)$ is the modulated multi-carrier signal and at time t .

y_{sc} is the demodulated data symbol of sc^{th} subcarrier, where $sc^{th} = 0, 1, 2, \dots, N$.

$\theta_{sc}^*(t)$ is the complex conjugate of $\theta_{sc}(t)$.

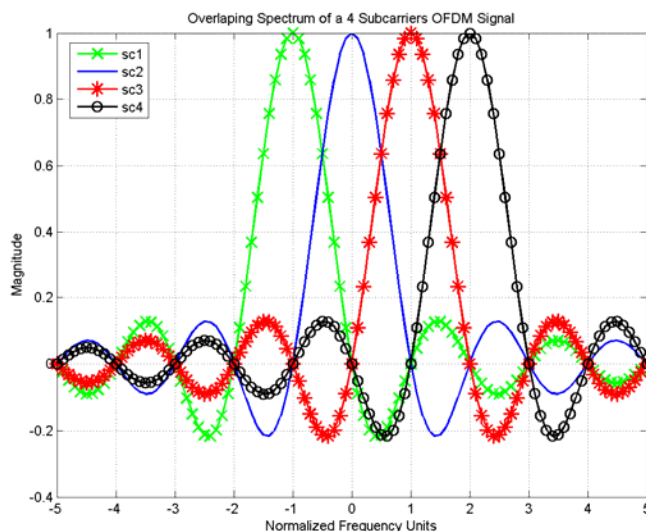


Figure 1.2 Spectrum of the OFDM signal with 4 subcarriers

In DSRC system, the baseband OFDM symbol is generated via serial binary data which has been modulated and then mapped onto parallel subcarriers. Furthermore, modulating in the discrete domain

using the Inverse Discrete Fourier Transform (IDFT) or the Inverse Fast Fourier Transform (IFFT) to get the symbol ready for transmission in the time domain, and conversely in the receiver, the DFT or FFT algorithms are used during the demapping, as shown in Figure 1.3.

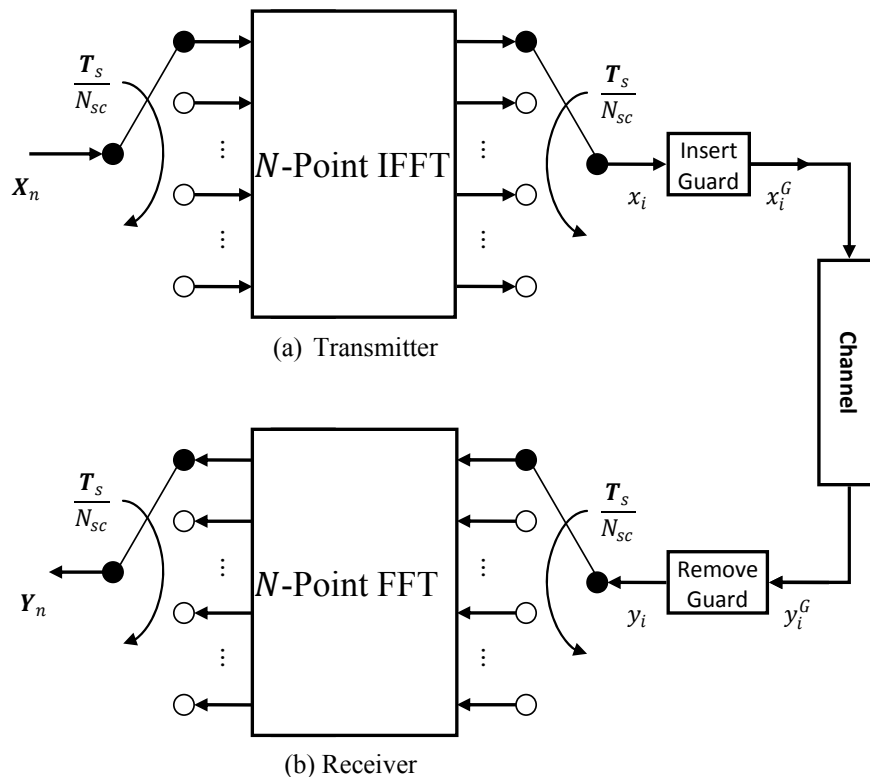


Figure 1.3 OFDM (a) Transmitter and (b) Receiver Systems

A cyclic prefix guard interval can be inserted before transmission and removed after reception (as seen in Figure 1.3) to reduce Inter Symbol Interference (ISI), which becomes the main benefit for OFDM systems when combating harsh channel environments. This is because the low symbol rate makes the use of a guard interval between symbols affordable, giving rise to the possibility of handling time spreading and eliminating ISI. The guard interval will be described in more details in Chapter 2 when the DSRC system model is defined.

OFDM has high spectral efficiency which permits high throughput data transmission in a small frequency band. As already expressed, the orthogonality of the subcarriers allows for overlapping subcarriers to still be separated, i.e. no interference at the carrier locations. Figure 1.2 illustrates how the subcarriers are overlapping. Although the guard interval helps combat ISI, increasing it reduces the spectral efficiency because of the extra redundancy.

For DSRC, the Fourier Transforms are performed using 64 point IFFT and FFT algorithms as in the standard for 802.11a [4].

Additionally, 802.11a/g/n, HYPERLAN/2, Digital audio and video broadcasting, i.e. DAB and DVB, WLAN, 802.20 Mobile Broadband Wireless Access and 802.16e WiMax standards all use OFDM as physical transmission modes.

1.2 WAVE Channel Model

Multipath fading and Doppler shift are two physical phenomena that are encountered in wireless communication systems between two Omni-directional antennas with nonzero relative velocity. Multipath refers to both line-of-sight (LOS) and non-line-of-sight (non-LOS) components, due to the reflections of the transmitted signals by the surrounding objects. Multipath may cause what is known as “frequency selective” fading, which induce inter symbol interference (ISI) when the symbol duration or symbol period (T_s) is less than 10 times the multipath RMS delay spread (σ_τ), ($T_s \leq 10\sigma_\tau$). The channel may be considered flat if the range of the channel frequencies or coherence bandwidth (B_c) satisfies [10],

$$B_c \approx \frac{1}{50\sigma_\tau} \quad (1-4)$$

In time-varying channels, a nonzero relative velocity is produced, because both antennas may be moving towards each other or in the same direction but at different speeds. The resulting maximum Doppler shift (f_D) is related to the relative velocity via the equation ($f_D = V \cdot f_c / c$), where V is the relative velocity in meters per second (m/s), f_c is the carrier frequency in Hertz (Hz), and c is the speed of light in (m/s) [10]. Also, the channel may be characterized as a “fast fading” channel if $T_s \cdot N_s \gg T_c$, where T_c is the coherence time and N_s is the number of OFDM symbols per packet,

$$T_c = \frac{0.423}{f_D} = \frac{0.423}{V \cdot f_c / c} = \frac{0.423 \cdot c}{V \cdot f_c} \mu sec \quad (1-5)$$

In other words, the channel impulse response is changing within the duration of a symbol or packet if $T_s \ll T_c$ or $T_s \cdot N_s \ll T_c$ is not satisfied, respectively. Since C and f_c remain constant throughout a particular system, the rate of fading is proportional to one of the parameters T_s , N_s or V while the others remain constant. *Rician fading* channel could be used if a strong LOS component exists, while Rayleigh fading could be considered when time-varying channels exist with No-LOS.

We tested our proposed system design in Chapter 4 in the worst case scenario environment, *Rayleigh fading* channel was considered. In Chapter 3, both no-LOS and LOS were considered in our simulations. For LOS, *Ricean fading* channel model was used. The time-varying Rayleigh fading channel in our study is produced using the Jakes’ fading channel simulator method with no-LOS [11]. Figure 1.4 below shows a diagram representation of a Jakes’ fading simulator.

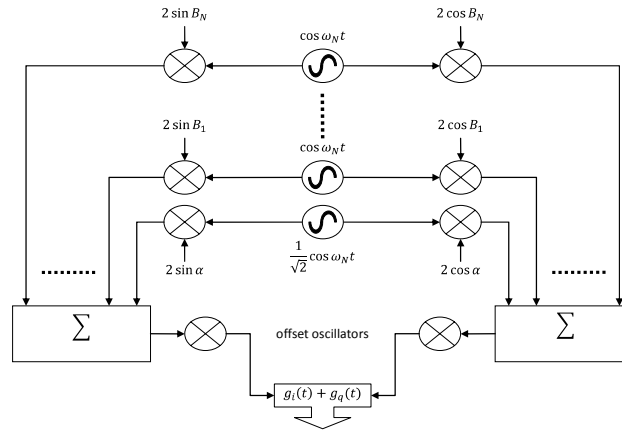


Figure 1.4 Jakes Fading Simulator

Jake’s method works by simulating the physical model of 2D isotropic scattering with no-LOS. It assumes that there are N equi-spaced or uniformly distributed scatters around the vehicle,

$$\theta_i = \frac{2\pi i}{N}, i = 1, 2, \dots, N \tag{1-6}$$

The phase angle associated with each scatter is chosen at random. The received complex envelope is treated as wide-sense stationary Gaussian random process with zero mean, when Rayleigh channel is concerned. The WAVE (wireless access vehicular environment) channel model can be modeled using statistical methods presented in [12] [13].

Figure 1.5 illustrates the WAVE channel model. It has both specular and scatter components. It is a model for simulating the effects of an L number of paths Rayleigh/Rician multipath scattering channel with AWGN and K factor affecting specular to scatter component fading path ratio.

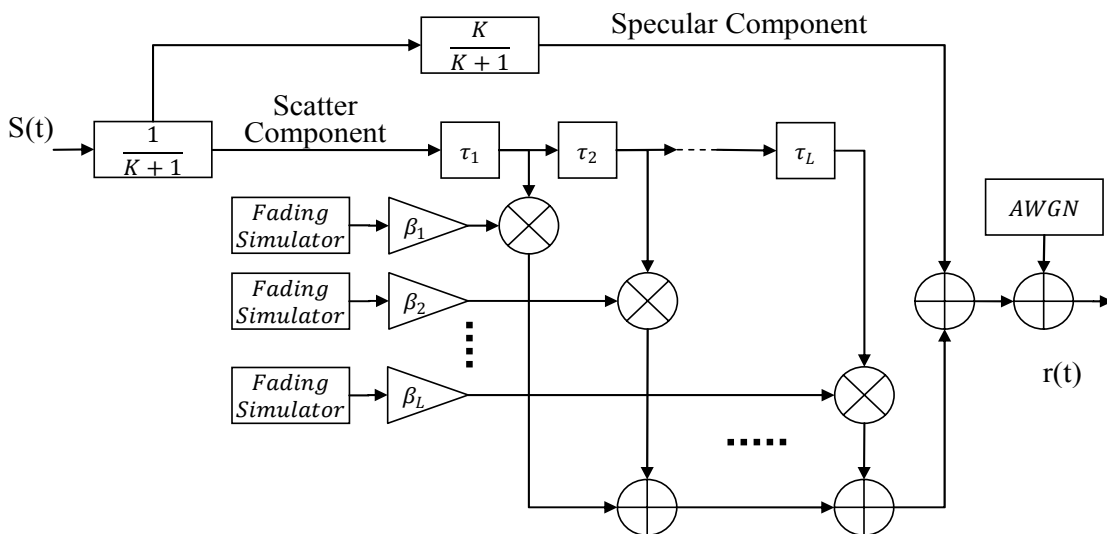


Figure 1.5 Channel Model with multipath scattering, specular (LOS), and Rayleigh (no-LOS) portions and AWGN[14]

Referring back to Figure 1.4, and in order to produce a zero mean Gaussian envelope, the following relation is used [12][13],

$$g(t) = g_I(t) + j g_Q(t) \quad (1-7)$$

$$g(t) = \sqrt{2} \left[2 \sum_{n=1}^M \cos \beta_n \cos 2\pi f_n t + \sqrt{2} \cos \alpha \cos 2\pi f_m t \right] + j \left[2 \sum_{n=1}^M \sin \beta_n \cos 2\pi f_n t + \sqrt{2} \sin \alpha \cos 2\pi f_m t \right] \quad (1-8)$$

where $\beta_n = \frac{\pi n}{M}$. We need M number of oscillators with frequencies,

$$f_n = f_m \cos \frac{2\pi n}{N}, n = 1, 2, \dots, M \quad (1-9)$$

According to [15] M greater than 6 should suffice.

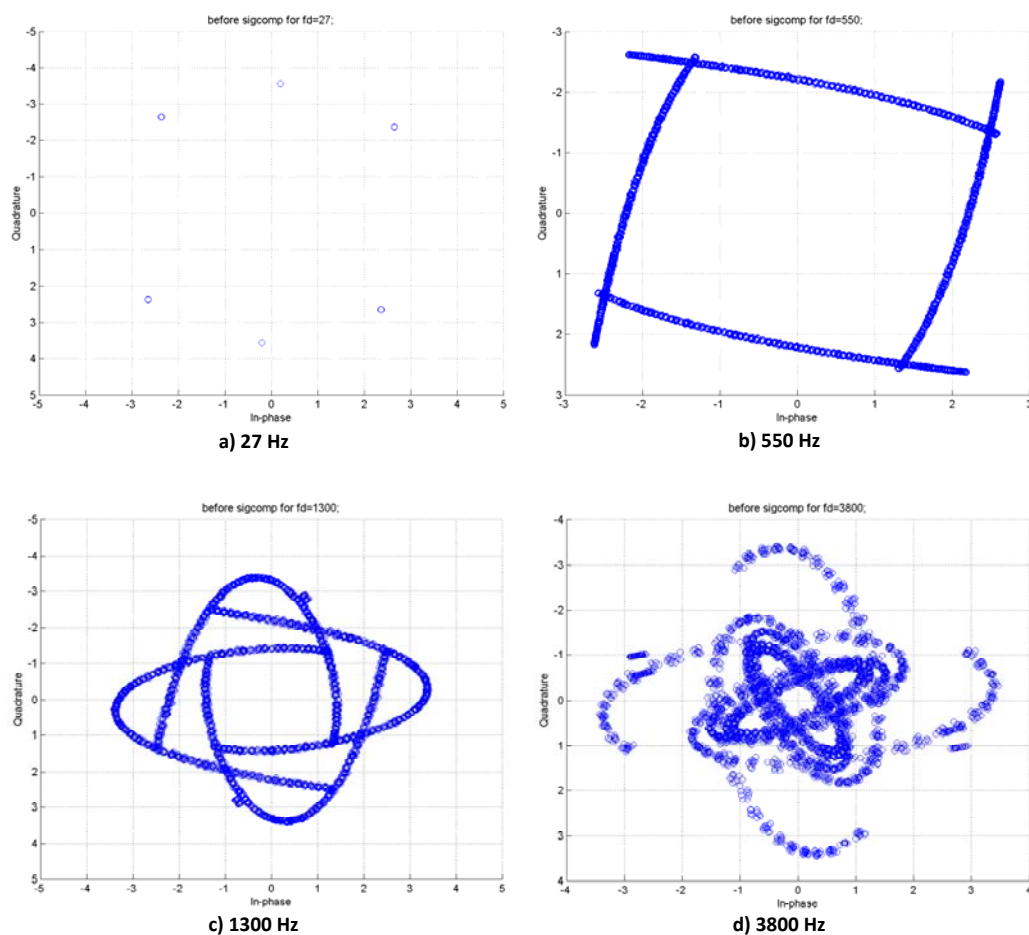


Figure 1.6 Scatter plots showing QPSK constellation distortion under various frequency shifts

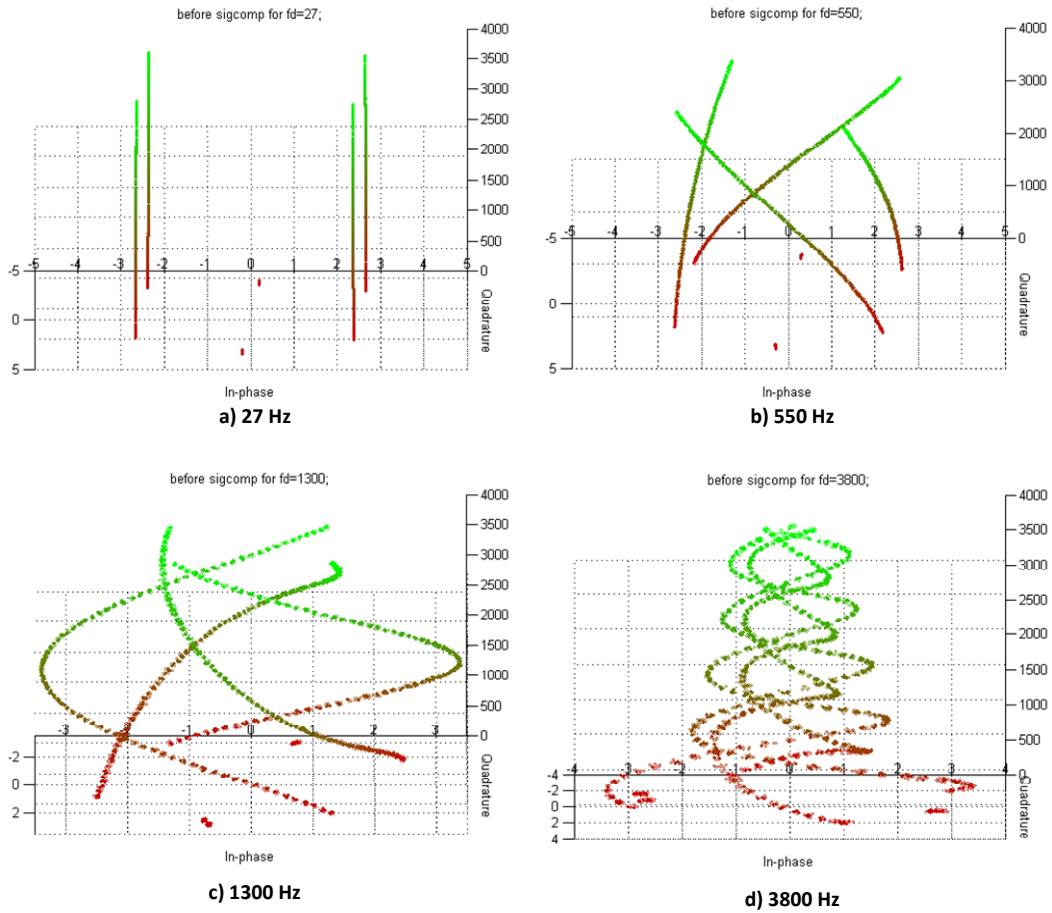


Figure 1.7 view of scatter plots showing QPSK constellation distortion under various frequency shifts (vertical represents data samples)

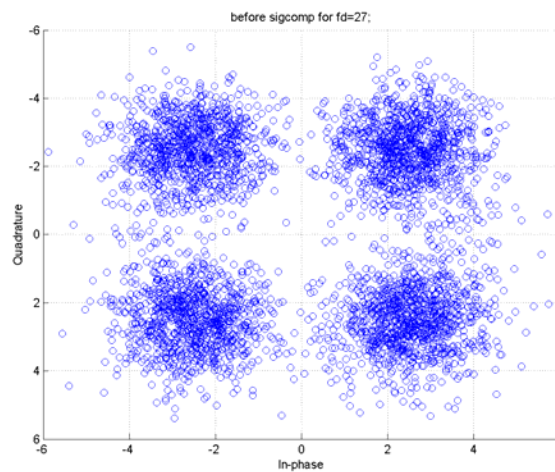


Figure 1.8 Scatter plots showing QPSK with low fade and noise of $10 \frac{E_b}{N_0}$

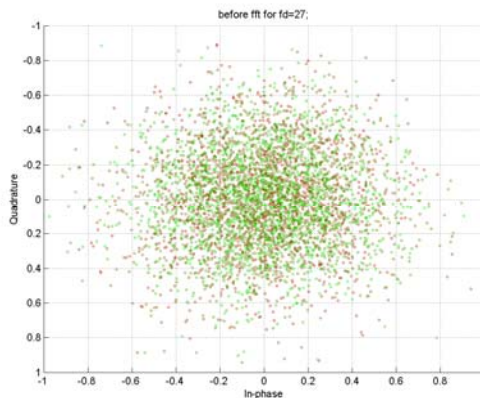


Figure 1.9 dispersion of amplitude and phase in the time domain for QPSK with no AWGN

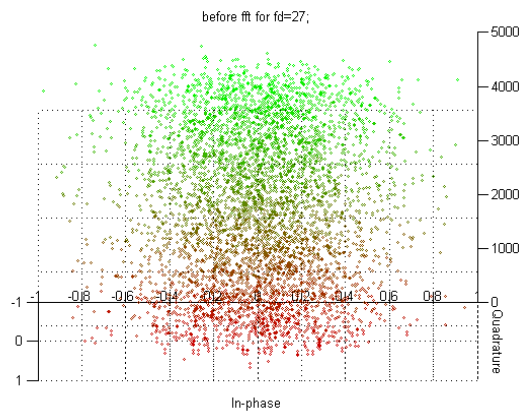


Figure 1.10 side view of dispersion of amplitude and phase in the time domain for QPSK with no AWGN

In order to visually demonstrate the effect of the channel on the message bits, scatter plots in Figure 1.6 are produced for 5.9 GHz DSRC system over a 1 packet of duration of 0.64 ms, using QPSK modulation scheme. Different Rayleigh fading envelopes are simulated using different velocities. Additive white Gaussian noise (AWGN) is also considered using different signal-to-noise ratio (SNR) values. As illustrated from the figure, as the velocity is increased there exist a greater shift in both amplitude and phase, which result in the deviation of the received symbols from their constellation points. As for AWGN, the increased velocity produces a more scattered version of the symbols, hence increases the chance of erroneously interpreting symbols. Figure 1.7 represent the 3-D version of Figure 1.6.

Figure 1.9 and Figure 1.10 show the scatter plots of the same variables and system simulated in Figure 1.6 and Figure 1.7, except that these simulations show the dispersion of amplitude and phase in time-domain rather than the frequency-domain. It may seem like there is AWGN, due to the scatter nature of signals in time-domain, but no AWGN was included in the simulations.

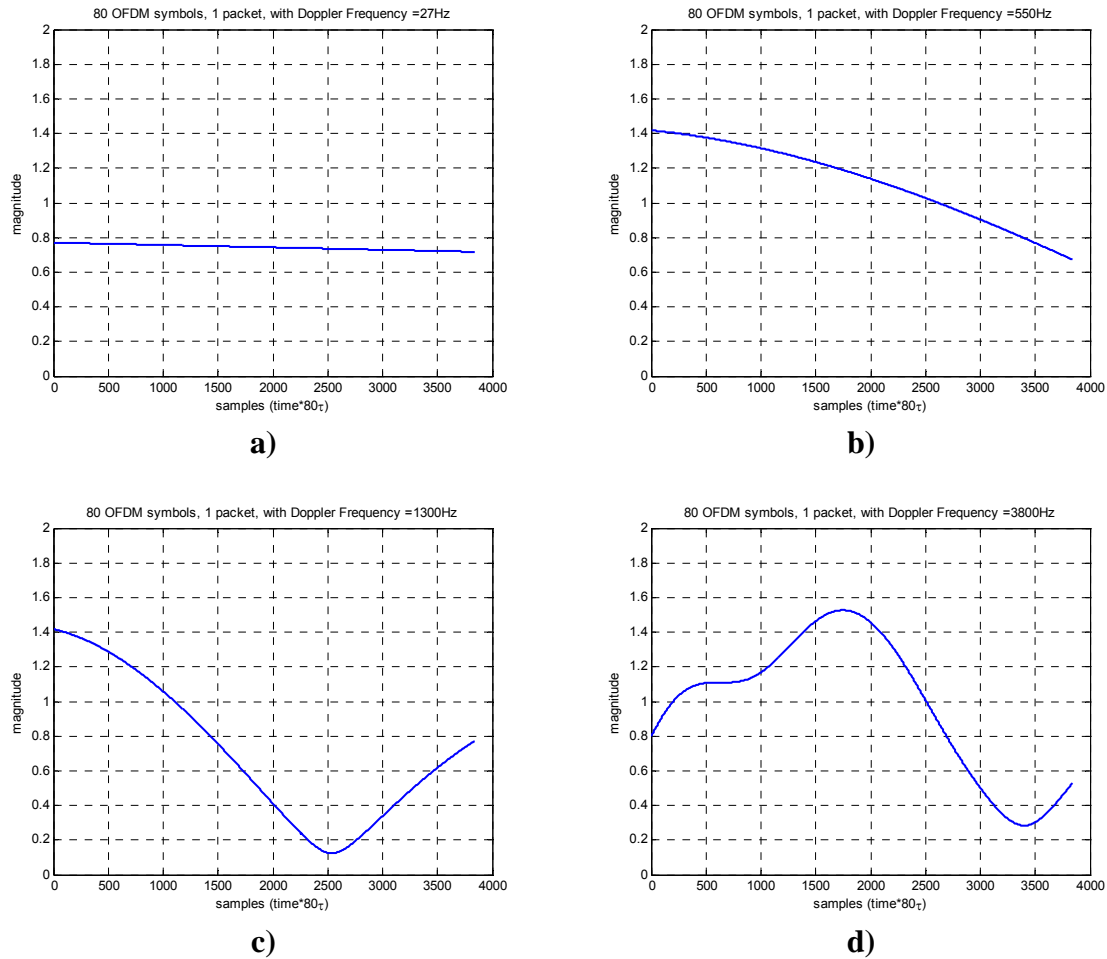


Figure 1.11 Samples vs. magnitude of 1 packet, 80 OFDM symbols under varying doppler frequencies

Figure 1.11 above, provide us with the information about the effect that the doppler frequencies have upon the data symbols. It can be seen here, that as the doppler frequencies increase the transmitted data experience the so-called deep fades, which represent the amplitude attenuation on the particular samples at the time that these deep fades occur.

1.3 Organization of the thesis

Previous sections provided a description of the OFDM system and the channel model, which are two critical components or factors that are needed to be understood in order to ensure successful state of the art design. What follows is the organization of this thesis:

Chapter 2 contains the description of the conventional DSRC system (or 802.11p) similar to that of the 802.11a standard [4], [16].

Chapter 3 discusses different types of demapping and decoding schemes that will be compared with one another through BER (bit error rate) and PER (packet error rate) performances with varying SNR (signal to noise ratio) and velocities. Two of the three demappers are the proposed demapping schemes.

Chapter 4 presents the detailed description of the proposed receiver design for enhancing the performance of the DSRC system using frequency domain MAP equalization algorithm.

Chapter 5 summarizes major accomplishments and suggestions for future research studies are given in this chapter.

Appendix A has been added to this thesis to focus on our proposed design of a simulation environment as a first stage of a design implementation, and as a learning/research tool. This contribution was not directly related to the major trend of the thesis, but is there to demonstrate the usefulness of the simulator's flexibility when it comes to simulating systems such as OFDM systems that contain a lot of variables that depend on one another.

Chapter 2

FEC coded and OFDM DSRC systems

2.1 Channel Coding Theory

Claude Shannon [17], "the father of information theory" [18], introduced the main concepts of information theory. Shannon went beyond the examinations of signals' frequencies, bandwidths and noise added to them during transmission [19]. He characterized the source that produces these messages and proved that there exists a theoretical limit at which error-free communication could take place using error-correcting codes. Several coding schemes in hopes of achieving this Shannon limit performance were developed. This family of codes are called Forward Error Correcting (FEC) codes. They are mainly composed of two parts: the convolutional and the block codes. FEC code went from a single error correcting code technique called Hamming block code, to maximum likelihood sequence algorithm Viterbi [20] for decoding convolutional code. Due to the famous Viterbi algorithm, convolution coding was more exploited and hence became more popular. The Viterbi decoder will be discussed in Chapter 3 along with maximum a-posteriori (MAP) and maximum likelihood (ML) decoders, which will be used in our DSRC receiver design. The description of the conventional DSRC system is presented in this chapter.

2.2 The DSRC System Model

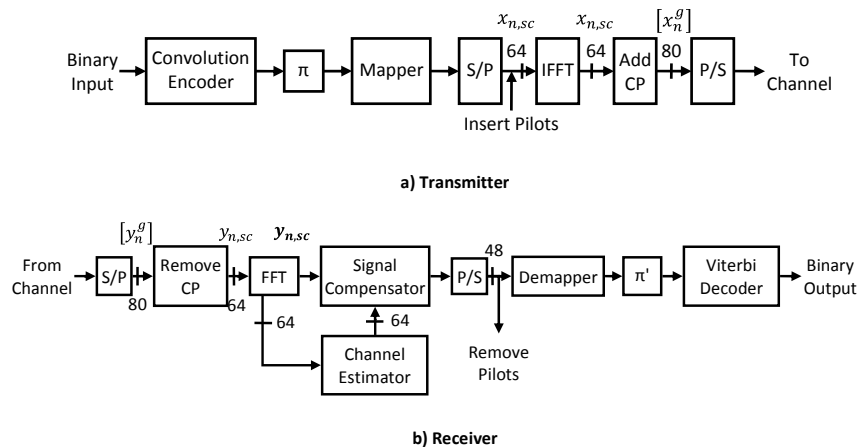


Figure 2.1 DSRC System Model

In this section we describe the DSRC system model in both the time and the frequency domain baseband model. Figure 2.1 shows the block diagram of the DSRC system model. DSRC physical layer (PHY) uses OFDM to mitigate the effect of ISI by using *guard interval* that exceeds the maximum excessive delay. The guard interval is added at the transmitter and later removed at the receiver. It is a copy of the symbol tail and placed in front of the symbol, known as a cyclic prefix. Its duration is usually computed as around 20% of the OFDM symbol duration. DSRC is very similar to the 802.11a standard [4] with the difference being in doubling the symbol duration in the case of the DSRC system. In DSRC, the guard interval is $1.6 \mu\text{s}$ with symbol duration of $8.0 \mu\text{s}$ and signal bandwidth of 10MHz. Figure 2.2 shows the packet format of the DSRC system.

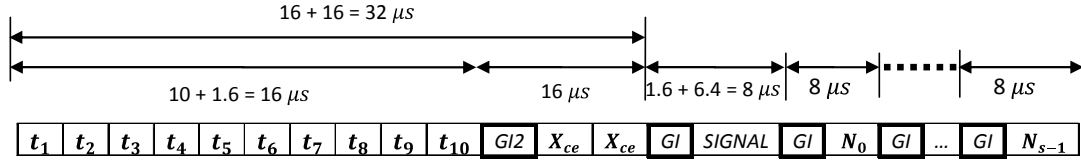


Figure 2.2 DSRC transmission sequence

It consists of a short and a long preamble. The short preamble is used for frequency offset estimation and symbol timing, while the long preamble is used for channel estimation through the use of two identical training symbols X_{ce} [4]. The guard intervals can also be seen in Figure 2.2. The DSRC symbol length is composed of the OFDM symbol length {64} and the guard interval length {16}. The minimum time resolution $tr = \frac{T_s}{64+16} = 0.1 \mu\text{s}$ is (resulting in 10MHz channel spacing). The number of subcarriers is ($N = 48$), and the symbol period is ($T_s = 8 \mu\text{s}$), the data rate (R_d) of a DSRC system depends on both the modulation scheme and the code rate (R_c) as follows,

$$R_d = \frac{R_c \cdot \log_2(m) \cdot N}{T_s} \quad (2-1)$$

The term $\log_2(m)$ gives the number of bits per modulated data symbol.

From Figure 2.1, the binary message goes through a series of components before transmission. First it goes through a convolution encoder with a generator of $(133_8, 177_8)$ and constraint length of 7 for forward error correction coding (FEC) reasons. The encoded message is then interleaved to avoid long burst errors due to fading and AWGN. The interleaved message is then digitally modulated using one of the Gray-Coded constellations BPSK, QPSK, 16 QAM, and 64 QAM [4]. The resulting modulated message is then divided into 64 shorter and parallel data symbols $X_{n,sc}$ (includes 4 pilot symbols, and 12 zero-padding), for the n^{th} OFDM symbol in N_s size packet ($0, 1, \dots, N_s - 1$), and the sc^{th} sample point (subcarrier: $0, 1, \dots, (N - 1 = 63)$) of the n^{th} OFDM symbol. The pilot symbols are inserted in the 6th, 20th, 34th, and 48th subcarriers for each OFDM symbol, in order to estimate the frequency and track the phase of the received message [4]. These symbols are then multiplexed into a 64-point (IFFT) using the OFDM modulation scheme, denoted by $x_{n,sc}$. $x_{n,sc}$ is transformed into an 80-point in length vector $[x_n^g]$ ($0, 1, \dots, N + g -$

1; $g = 16$) after the addition of the guard interval, and then serially transmitted over the channel. In order to satisfy equation (1-4) and with high delay spread involved in vehicular environments, the spacing of these pilots' tones needs to be at least 200KHz[21]. Hence, with the pilot spacing set as 1.85 MHz for the DSRC system, channel estimation using the current pilots is not feasible.

2.3 The DSRC Transmitter

As mentioned earlier, the DSRC 802.11p (WAVE) is an amendment to the IEEE 802.11a standard. Some of the revisions will be highlighted here. After the description of the DSRC transmission packet format, the blocks that make up the transmitter will be described.

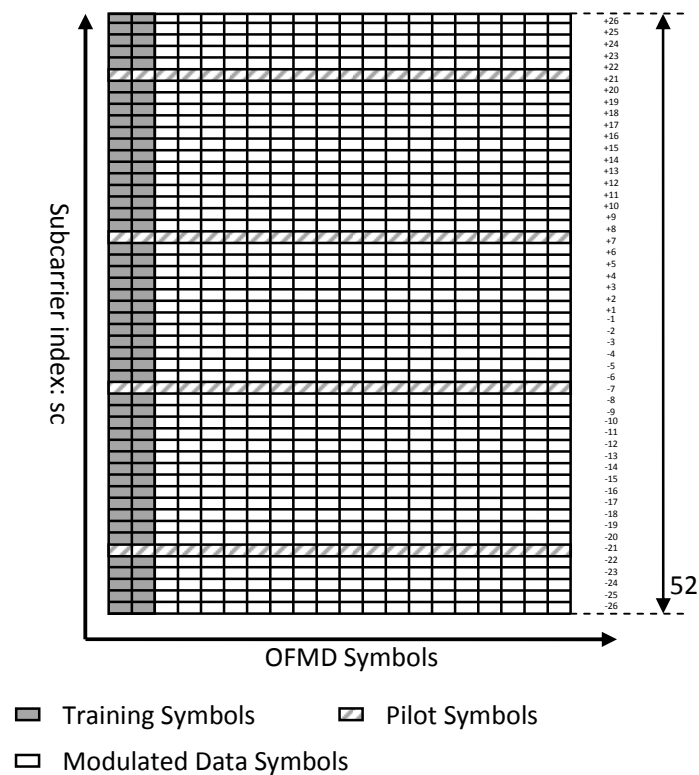


Figure 2.3 DSRC transmission packet format (time vs. frequency domain grid)

In Figure 2.3, the time grid represents the OFDM symbols in the 1 packet and the frequency grid represent the subcarriers in 1 packet. In this thesis, we will be using OFDM symbols to mean the DSRC symbols, while data symbols to mean modulated symbols. Each block in that grid represents the data symbols. The first two OFDM symbols represent the training data that is used by a 1 tap signal compensator to correct the fading of the channel. Subcarriers +21, +7, -7, and -21 are the pilot subcarriers.

2.3.1 Convolution Encoder

Figure 2.1(a) shows the transmitter model of the DSRC system. For convenience, it is also shown below.

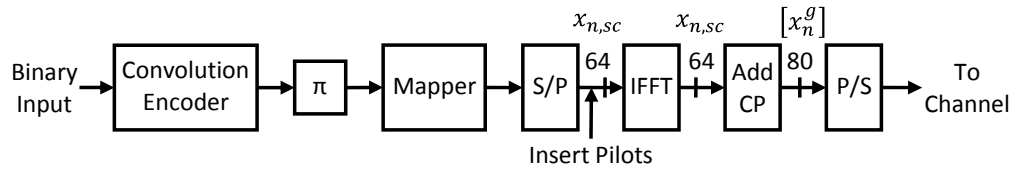


Figure 2.4 Transmitter model of the DSRC system

Convolution Encoding is frequently used for Forward Error Correcting (FEC) coding in communication systems. A convolution encoder can be described as a markov finite state machine which takes input as information bits and provides outputs as coded bits. In other words, the convolution encoder takes in statistically independent values from a discrete alphabet and outputs a sequence of values that are also taken from a statistically independent discrete alphabet [22]. The convolutional encoder can be represented by a generator sequence which is a set of impulse responses, whereby each information bit must pass through the generator constraint Length number of steps before the encoding is complete.

$$\mathbf{g}^{(i)} = (g_0^{(i)}, g_1^{(i)}, \dots, g_{K-1}^{(i)}) \quad (2-2)$$

where K is the constraint length, and i is from 1 to the number of outputs.

The number of inputs x_{conv} over the number of outputs y_{conv} of a convolutional code defines the code rate $R_c = x_{conv}/y_{conv}$ (i.e. for a 1 input, 2 output convolutional code, the code rate is said to be 1/2).

The encoding process is dependent on a trellis structure that defines the rate, and input/output relationship. A rate $1/y_{conv}$ convolutional encoder can be represented as a finite state machine (FSM) with the state of the encoder defined by the contents of $K - 1$ shift registers. The 802.11 Standard [4] defines the OFDM PHYS for which DSRC is based, and uses the industry standard $[\mathbf{g}^{(1)} = 133_8; \mathbf{g}^{(2)} = 171_8]$ generator matrix as seen in Figure 2.5.

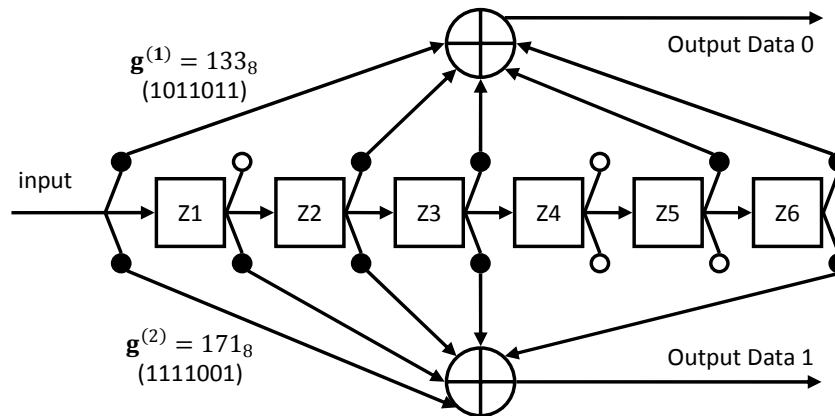


Figure 2.5 Convolution Encoder with Rate of 1/2 and a constraint length of $K=7$ and a generator matrix based on $[133 \ 171]_{(8)}$

The FSM describing information forms a trellis that shows the paths that the encoding take from one state to the other over time.

The output of the convolution encoder can be obtained by performing the discrete convolution,

$$\mathbf{output}^{(i)} = (\mathbf{input} * \mathbf{g}^{(i)})[n] \tag{2-3}$$

where * denotes the modulus-2 convolution.

Figure 2.7 shows an example of a FSM of a basic convolutional encoder example shown in Figure 2.6 , i.e. a convolution encoder with Rate of 1/2 and a constraint length of $K=3$ and a generator matrix of $[5_8 \ 7_8]$, and Table 2-1 shows its state transition.

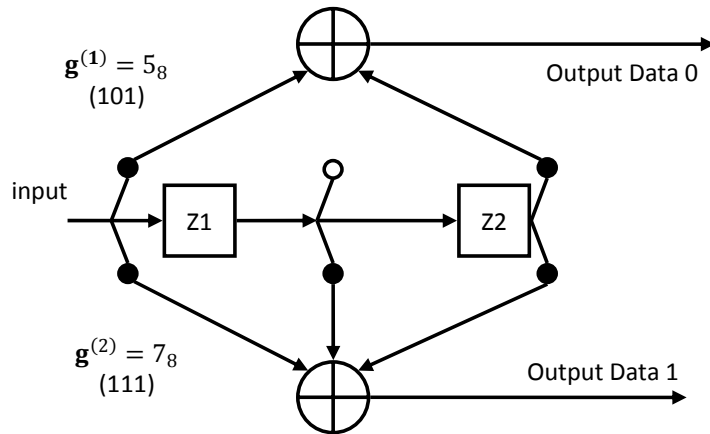


Figure 2.6 Convolution Encoder with Rate of 1/2 and a constraint length of $K=3$ and a generator matrix of $[5_8 \ 7_8]$

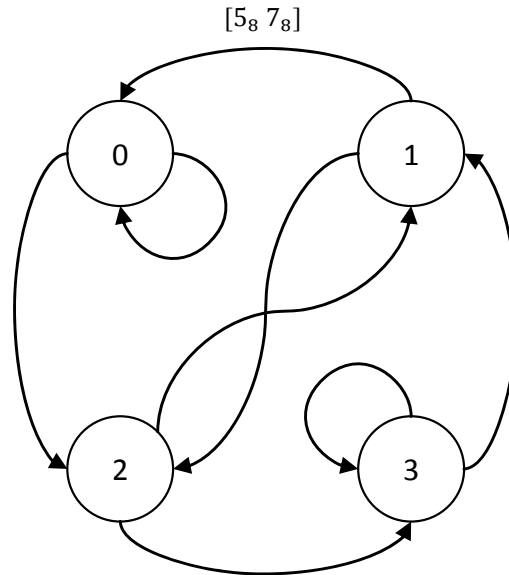


Figure 2.7 FSM chart for basic convolutional code

For the FSM chart in Figure 2.7 above, each state can be reached by exactly two paths, and each state has two paths leaving them as well.

Table 2-1: State transition of convolution example

In	Current State		S	Next State		S	O1	O2
0	0	0	0	0	0	0	0	0
1	0	0		1	0	2	1	1
0	0	1	1	0	0	0	1	1
1	0	1		1	0	2	0	0
0	1	0	2	0	1	1	1	0
1	1	0		1	1	3	0	1
0	1	1	3	0	1	1	0	1
1	1	1		1	1	3	1	0

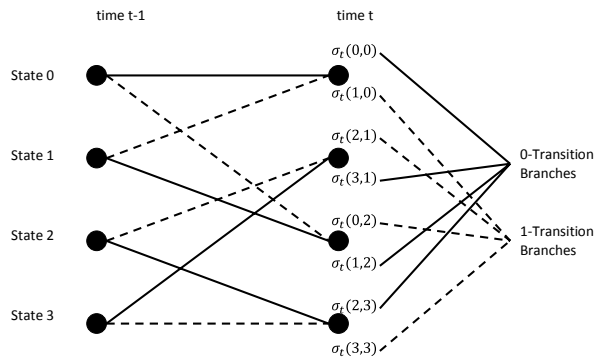


Figure 2.8 Transitions for RSC codes[23]

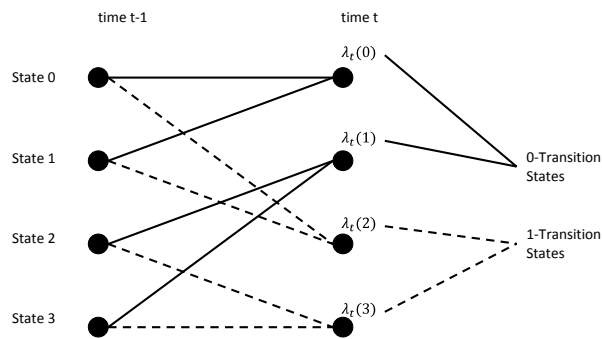


Figure 2.9 Transition for NRC codes[23]

Figure 2.8 and Figure 2.9 show the transition diagrams for recursive systematic (RSC) and non-recursive systematic codes (NRC) respectively. Recursive codes have direct feed-forward during the decoding stage,

meaning a portion of recursive decoder output depends directly on the input. While, the non-recursive codes do not.

Decoding of the convolutional code can be done by *Viterbi algorithm* [20] for Maximum Likelihood, or the BCJR for *Maximum a posteriori* [24] and are discussed in details in the next chapter.

2.3.2 Interleaving

Because of the fact that fading channels may have deep fades that can cause a long sequence of errors, a method used in combating burst errors is interleaving. Interleaving has the effect of reducing adjacent correlation in the bit stream. This can be done either pre-symbol mapping known as *bit interleaving* or post-symbol mapping, which is *symbol interleaving*.

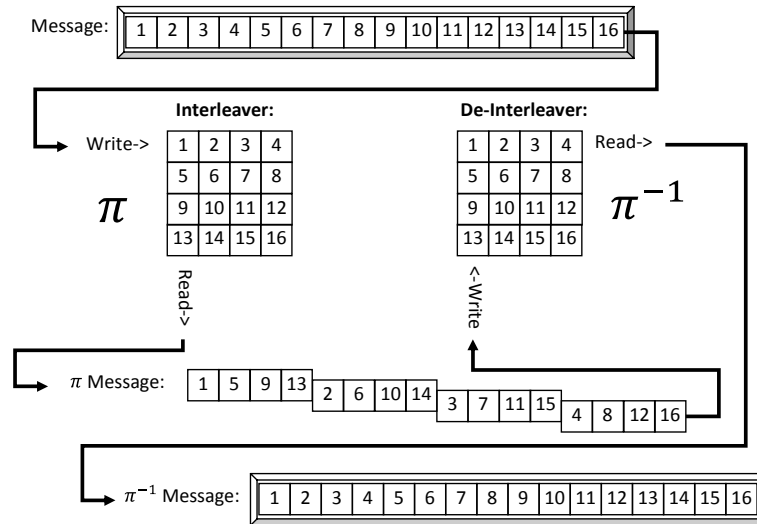


Figure 2.10 Block Interleaver / De-Interleaver

Both the interleaving and de-interleaving process can be done with different size blocks. Figure 2.10 shows the interleaver for a 4 row, 4 column block interleaver and de-interleaver showing the dispersed correlation of the interleaved (π) message from adjacent bits. In the interleaving example, each data symbol (numbered 1 through 16 in this case) is written to the rows of the interleaver block of size $r=4$, $c=4$. The interleaving is actually just a rotated read, which reads each data symbol column wise and concatenates the result to achieve the π message. The deinterleaving process is the reverse in that columns are written to first and then the reading is done row wise. Giving the final result in the receiver of data that was in the original order as it originally was in the transmitter.

Random interleavers can also be used which use a pseudo randomly permuted bit stream. Pseudo random is used because the inverse operation must be performed on the receiver side (i.e. the pattern must be predictable).

2.3.3 Modulation

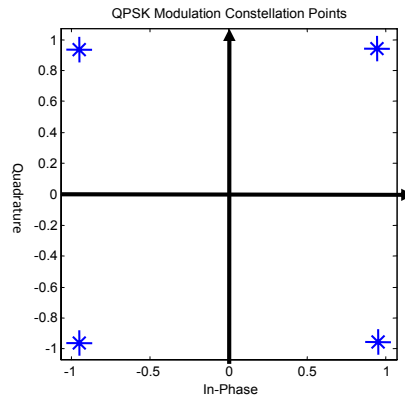


Figure 2.11 Example QPSK (QAM4) Modulated Signal Constellations

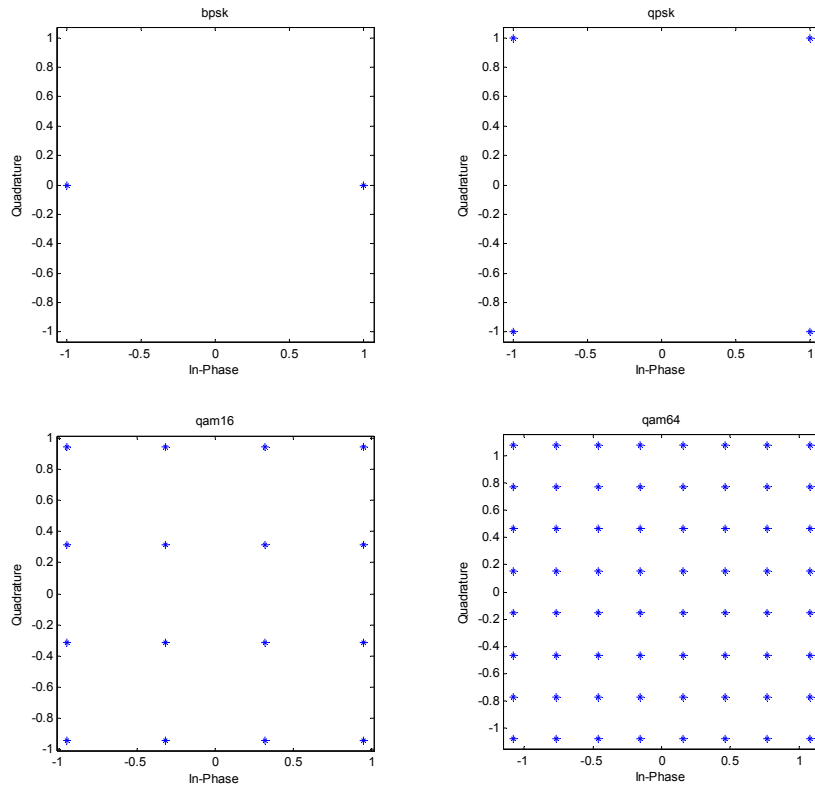


Figure 2.12 Normalized signal constellations

Many modulation techniques exist that attempt to compress data in a more efficient form that can use the same band width of the available channel. The adding of more variations to a given signal lead to increased diversity of the transmission. The higher the modulation level (i.e. the more information packed into 1 modulated symbol), the more susceptible to error causing channel conditions. Figure 2.11 shows an example modulated constellations with the in-phase representing the real component of the signal and the quadrature component representing the imaginary or phase shift portion of the signal.

As mentioned earlier, the modulations chosen can be BPSK, QPSK, QAM16, or QAM64, with modulation levels of $\log_2 2$, $\log_2 4$, $\log_2 16$, and $\log_2 64$, respectively.

After modulation, the complex data symbols should be grouped into 48 complex numbers for being used in the IFFT. The complex numbers will be numbered from 0 to 47 and mapped onto spaced data coordinated with 4 pilot symbols in between as shown in Figure 2.3.

Figure 2.12, below shows the normalized signal constellations for BPSK, QPSK, 16 QAM and 64 QAM.

2.3.4 IFFT

The Inverse Fast Fourier Transform is used to transform the modulated signal to the time domain for transmission as a serial transmission.

$$x_{n,sc} = F^{-1}(x_{n,sc}) \quad (2-4)$$

Where, $F^{-1}(\cdot)$ is the Inverse Fast Fourier Transform (IFFT). $x_{n,sc}$ and $x_{n,sc}$ are the time domain output and the frequency domain input to the IFFT block at the n^{th} OFDM symbol, and the sc^{th} subcarrier, respectively.

2.3.5 Guard Interval

The guard interval is a redundant portion of the transmission, a cyclic extension of the tail bits added to the beginning of the message to absorb inter-symbol interference in the time domain.

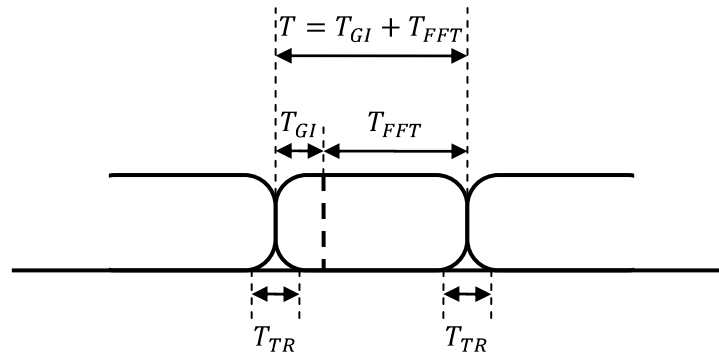


Figure 2.13 OFDM frame with cyclic extension

The guard interval is particularly useful in helping to correct multipath fading because the repetition can be detected and its effects can be cancelled through the use of techniques such as Hermitian transpose signal compensation/equalization methods in the receiver as discussed later. The guard interval length chosen for DSRC is $3.2 \mu\text{s}$ consisting of 16 symbols representing 20% of the FFT length (i.e. FFT length of 64) for a total transmission size of 80 data symbols per OFDM symbol. Hence the time of transmission for 1 OFDM symbol, T_i , where $T_i = T_{GI} + T_{FFT} = 16t_{ds} + 64t_{ds}$. t_{ds} is the inverse of the symbol rate recommended to

be $sr = 125000$ for DSRC, making $t_{ds} = 8\mu s$, hence $T_i = 80t_{ds} = 640\mu s$. For our analysis we are ignoring the overlap length T_{TR} shown in Figure 2.13, because we are not addressing synchronization in this thesis.

2.4 DSRC Receiver

This section will present a description of the DSRC receiver of Figure 2.14.

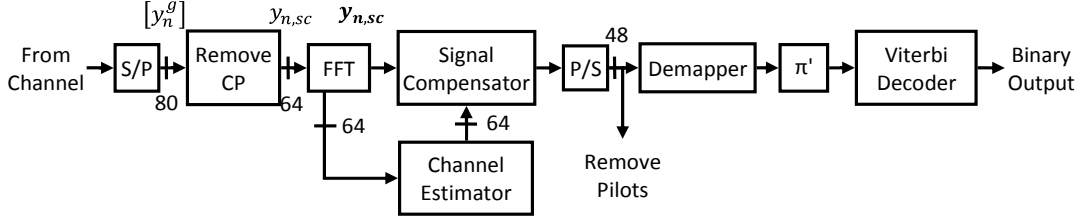


Figure 2.14 DSRC Receiver

The DSRC Receiver must cope with potential fading due to the high velocities and multipath fading in urban environments. The goal of reconstructing and obtaining the original transmission message can be challenging (if not impossible) if too much of the signal has been distorted or lost.

The receiver receives the signal and will attempt to correct errors that may have occurred during transmission through a Rayleigh faded and AWGN channel. The components that assist with error correction and/or prevention are the Guard Interval, Channel Estimator and Signal Compensator, FFT, Interleaver, and Decoder.

2.4.1 The time domain

At the receiver, the reverse of the transmission process starts to unfold. The received signal vector $[y_n^g]$ of one OFDM symbol is,

$$[y_n^g] = [x_n^g] \otimes [h_n] + [\omega_n] \quad (2-5)$$

Where $[h_n]$ is the channel impulse response (CIR) vector and can be expressed as

$$h_{n,sc} = \sum_{l=0}^{L-1} h_{n,l} e^{(j\frac{2\pi}{N_{sc}} f_{Dl} T_{sc})} \delta(t - \tau_l), \quad 0 \leq sc \leq N - 1 \quad (2-6)$$

where N is for number of OFDM subcarriers, $h_{n,l}$ is the complex impulse response for the l^{th} path of the CIR at time n (n^{th} OFDM symbol). f_{Dl} is the Doppler frequency in the l^{th} path as well; t is the delay spread index; τ_l is the time delay of the l^{th} path; while T_{sc} is the sample period. $[\omega_n]$ is the (AWGN) or additive white Gaussian noise vector at time n (for the n^{th} OFDM symbol). The symbol \otimes denotes the linear convolution operator. Equation (2-5) can also be expressed as

$$[y_n^g] = \sum_{l=0}^{L-1} h_{n,l} \cdot x_{n-l} + [\omega_n], \quad 0 \leq n \leq N + g - 1 \quad (2-7)$$

2.4.2 The frequency domain

After the removal of the guard intervals from the received data (or channelword), the parallel data symbols $y_{n,sc}$ can be expressed in terms of a matrix

$$Y = H \cdot (\mathbf{F})^H \cdot \mathbf{X} + W \quad (2-8)$$

The variables Y, X and W are $N \times 1$ vectors (N is for number of OFDM subcarriers). \mathbf{F} is the discrete Fourier transform (DFT) of size $N_s \times N_s$ (N_s is the number of OFDM symbols in a packet). $(\cdot)^H$ denotes the complex conjugate transpose (Hermitian). H is the time-domain channel matrix (circulant if the channel is time-invariant).

After that, the signal is demultiplexed and fast Fourier transformed (FFT) into the frequency domain output as follows,

$$\mathbf{y}_{n,sc} = \mathbf{h}_{n,sc} \cdot \mathbf{x}_{n,sc} + \mathbf{w}_{n,sc} \quad (2-9)$$

where $\mathbf{y}_{n,sc}, \mathbf{h}_{n,sc}, \mathbf{x}_{n,sc}, \mathbf{w}_{n,sc}$ denote the FFT of the received OFDM symbols, the channel response, the transmitted OFDM symbols, and the AWGN of the n^{th} OFDM symbol at the sc^{th} subcarrier, respectively.

2.4.3 Channel Estimation

The first two OFDM symbols called the training symbols, as shown in Figure 2.3, are then used in the channel estimation process. The channel estimation block estimates channel distortion and relays that information to the signal compensator to attempt to reconstruct the signal back to its undistorted form.

Assuming that $\mathbf{x}_{n,sc}$ is known, then the channel frequency response can be obtained as follows,

$$\mathbf{h}_{n,sc} = \frac{\mathbf{y}_{n,sc} - \mathbf{w}_{n,sc}}{\mathbf{x}_{n,sc}} = \frac{\mathbf{y}_{n,sc}}{\mathbf{x}_{n,sc}} - \frac{\mathbf{w}_{n,sc}}{\mathbf{x}_{n,sc}} = \tilde{\mathbf{h}}_{n,sc} + \mathfrak{N}_{n,sc} \quad (2-10)$$

Using two training symbols instead of 1 minimizes the effect of \mathfrak{N} , reducing noise enhancement by the signal compensator. Therefore, the two training symbols are used in obtaining the estimated channel response over the first two received symbols as follows,

$$\tilde{\mathbf{h}}_{0,sc} = \frac{\mathbf{h}_{ce,sc} + \mathbf{h}_{ce,sc}}{2 \cdot \mathbf{X}_{ce,sc}}, \quad \text{where } \tilde{\mathbf{h}}_{0,sc} \cong \tilde{\mathbf{h}}_{n,sc} \quad (2-11)$$

The subscript 'ce' represents the known or training data and not the actual message data. The conventional IEEE 802.11a system uses the same channel response estimation throughout the packet, in other words the channels were assumed time-invariant and not changing throughout the packet.

2.4.4 Signal Compensation

As shown in Figure 2.14, 1-tap signal compensator is used in the 802.11a standard to estimate the transmission distortion based on the channel estimation derived by the channel estimator using two training symbols. At the output of the signal compensator, the received OFDM symbols are compensated as follows,

$$\tilde{\mathbf{y}}_{n,sc} = \frac{\mathbf{y}_{n,sc}}{\tilde{\mathbf{h}}_{0,sc}} \quad (2-12)$$

The compensated symbols are then digitally demodulated, deinterleaved and finally decoded using a Viterbi decoder [20] , [4].

2.4.5 Demodulation and Deinterleaving

Demodulation, deinterleaving as inverse of their transmitter counter parts. Special care must be taken for soft demodulation if using a soft input decoder for preserving probability information. Soft demodulation is covered in the next chapter.

2.4.6 Decoding

Maximum likelihood (ML), Viterbi decoder [20] , [4] is used to decode the convolutional codes. Next chapter will have a detailed description of the Viterbi algorithm.

Maximum a posteriori (MAP) decoders would also be an excellent option for decoding convolutional codes. ML decoders minimize sequence error, while MAP decoders minimize symbol error. More will be discussed about these decoders in the next chapter.

Chapter 3

Hard vs. Soft Decoding Algorithms and Proposed Soft Demapping Scheme

In the previous chapter, the DSRC system was introduced. This chapter will be dedicated for deriving the algorithms behind the Viterbi decoder used in the conventional DSRC system and the BCJR algorithm in order to use them along side with our proposed demapping scheme that is introduced in this chapter. Figure 3.1 below shows the overall trellis-based algorithms used in decoding. Our proposed system in Chapter 4 will be developed using the trellis-based MAP BCJR algorithms [24].

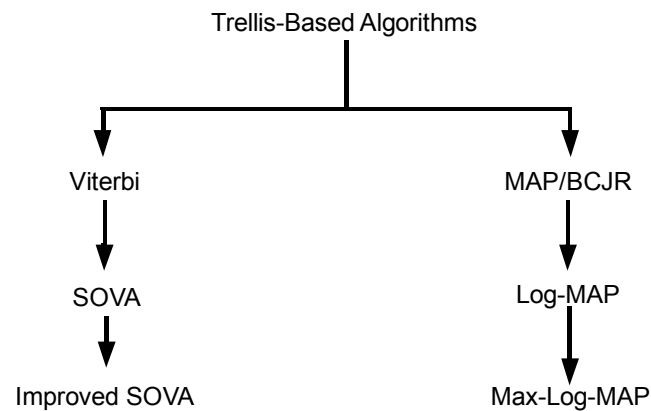


Figure 3.1 Decoding Algorithms

Information about the improved SOVA can be found in [25]. This chapter will present three different demapping schemes and compared with one another using both Viterbi [20] and BCJR [24] decoding schemes, under varying signal to noise ratio (SNR) and velocities. They will also be compared under different environments, i.e. *Rayleigh* and *Rician* faded channels. The next chapter (4) will present a receiver design for DSRC system that utilizes the concept behind the MAP algorithm (BCJR), which greatly outperforms the conventional system in both bit-error-rate (BER) and packet-error-rate (PER).

3.1 Optimal detection in the estimation of the states of a Markov process

The goal of an optimal receiver is to reproduce the message (m) that was sent at the transmitter as accurately as possible. In other words, the receiver must be able to minimize the probability of error $P(m_i \neq \hat{m}_i)$ for each message bit (m_i). The way to achieve that is to maximize the a posteriori probability (APP) $P(m_i = m|[\mathbf{y}])$ [26], where $[\mathbf{y}]$ is the observed sequence array. Therefore, for optimal detection, we have:

$$\hat{m}_i = \arg \left\{ \max_x P(m_i = m|[\mathbf{y}]) \right\} \quad (3-1)$$

Where the symbol ‘ $\hat{}$ ’ is used to indicate that the value or bit is an estimated one, that is, a value or bit after it undergoes some sort of a process in the receiver of a system, like demapping, deinterleaving and decoding. The equation above turns out to be the main inspiration of the symbol by symbol (MAP) or Maximum a posteriori BCJR algorithm [24]. The Map algorithm tries to find the most likely transmitted symbol (m_i), given the received sequence ($[\mathbf{y}]$). The BCJR algorithm will be explained in more details in the next few sections.

The Viterbi [20] algorithm is another well known algorithm that is used to estimate the states of a Markov process, but the difference between it and the BCJR algorithm is in the fact that the Viterbi algorithm tries to find the most probable transmitted sequence ($[\hat{m}]$) given the received sequence ($[\mathbf{y}]$), instead of what was mentioned about the BCJR algorithm in the previous paragraph. Therefore, it follows for the Viterbi algorithm that:

$$\hat{m} = \arg \left\{ \max_{m_i} P(m|[\mathbf{y}]) \right\} \quad (3-2)$$

From this brief introduction, it can be seen that the BCJRA (BCJR algorithm) minimizes the symbol error rate (SER), while VA (Viterbi algorithm) minimizes the frame error rate (FER).

3.2 The Viterbi Algorithm (VA/SOVA)

The previous section presented the general definition of VA. In this section VA will be described in more details with an example to demonstrate how this algorithm can decode an encoded message. The Viterbi decoder, also known as the Maximum Likelihood Sequence Estimator (MLSE), tries to follow the most likely path that the channel-word has taken through the trellis diagram. Hence, as mentioned earlier, it minimizes the error of a wrong path being chosen. Unlike the BCJR that minimizes the BER. There are two kinds of Viterbi decoders; the Hard-decision Viterbi (denoted by VA) uses Hamming distance and the soft-input Viterbi uses Euclidean distance for branch and path metric computations. The soft Viterbi decoders can also be classified as soft-input soft-output Viterbi algorithm (SOVA). As its name suggest, SOVA takes in soft values and outputs soft values. SOVA are useful in turbo decoding, where messages are transferred between two decoders in a receiver. More on SOVA will follow later on this chapter.

3.2.1 Hard-decision Viterbi (VA)

Hard-decision Viterbi [20] receives hard channel-bits that were given by the demodulator/demapper. Demodulator and demapper will be used interchangeably throughout this thesis. The demapper makes a decision on the channel-bits that it receives from the signal compensator and maps them into binary values ('0' or '1') according to the constellation points that is used in a modulator scheme at the transmitter. These binary values are then fed to the Viterbi decoder as input and the Viterbi decoder produces its own decoded hard decision of the message.

The Viterbi algorithm (VA) is best understood with an example. If we use the NRC encoder introduced in Chapter 2 as our example, we will see that the encoder encoded data-words, by constructing specific paths through a trellis diagram. Having the trellis as a tool, the Viterbi decoder knows that there exist specific transitions from one state to another. For example and for simplicity, let us assume a data-word (original message) of [0 0 0 0]. This message is first encoded to become the code-word of [00 00 00 00]. This codeword is then corrupted when sent through a noisy channel to become the channel-word [10 00 10 00]. Now, we will attempt to decode this channel-word. Figure 3.3 below shows the process of decoding the received channel-word by determining the most likely path using the trellis information generated by the convolution encoder of Figure 2.6.

Figure 3.2 defines the important terms used in defining the VA. The branch metric or Hamming distance in our case is the number of bits which differ between two binary strings. The branch labels are the code-words corresponding to the path of one state to the next defined by the convolution encoder's trellis. For example, the branch label or code-bits from 'state 0' to 'state 0' is always '00' for the convolution encoder of Figure 2.6

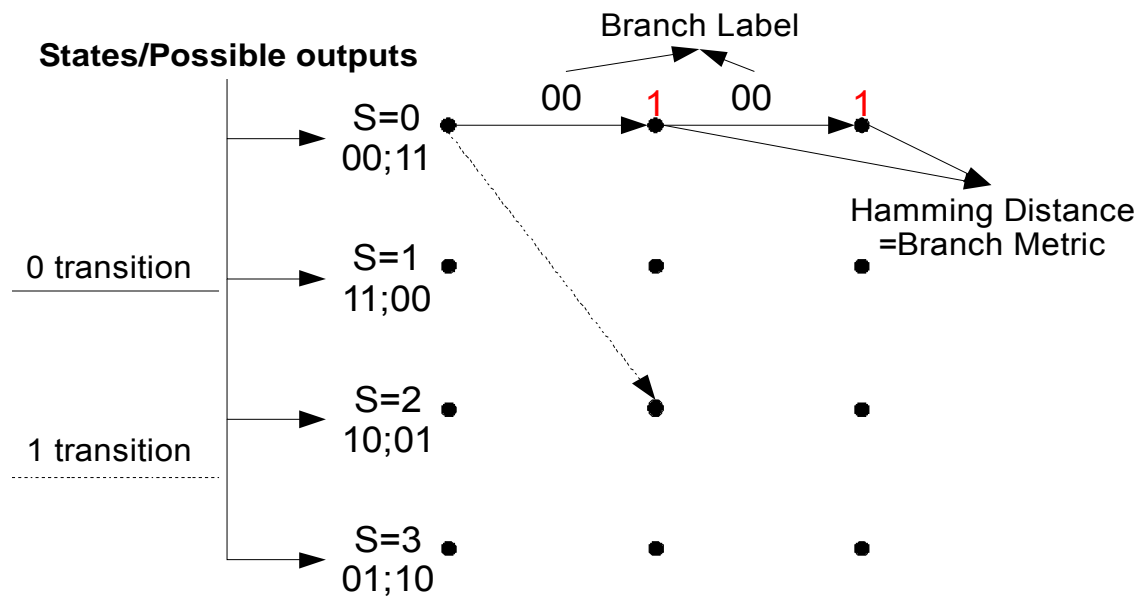
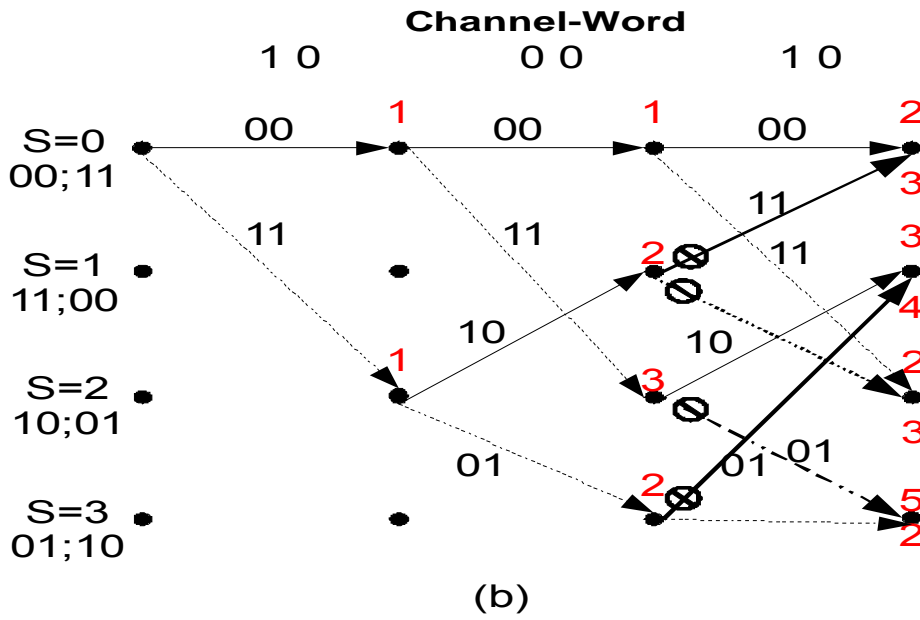
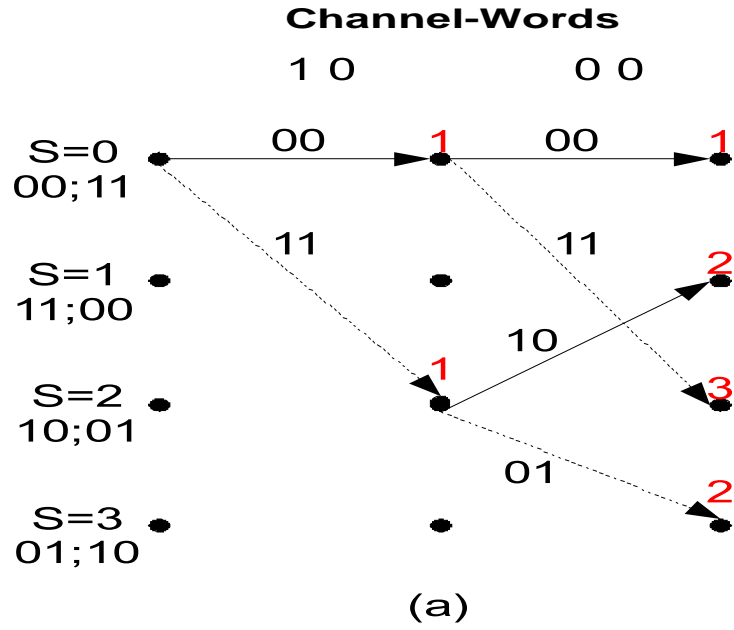


Figure 3.2 Viterbi Algorithm diagram labels



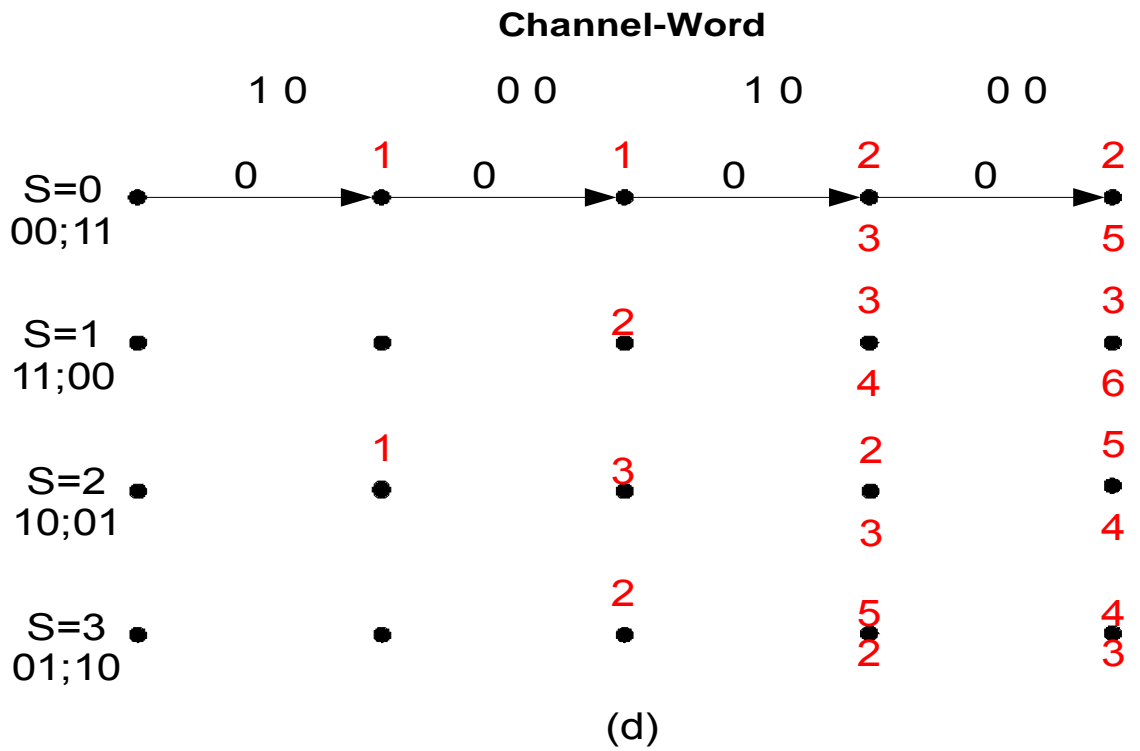
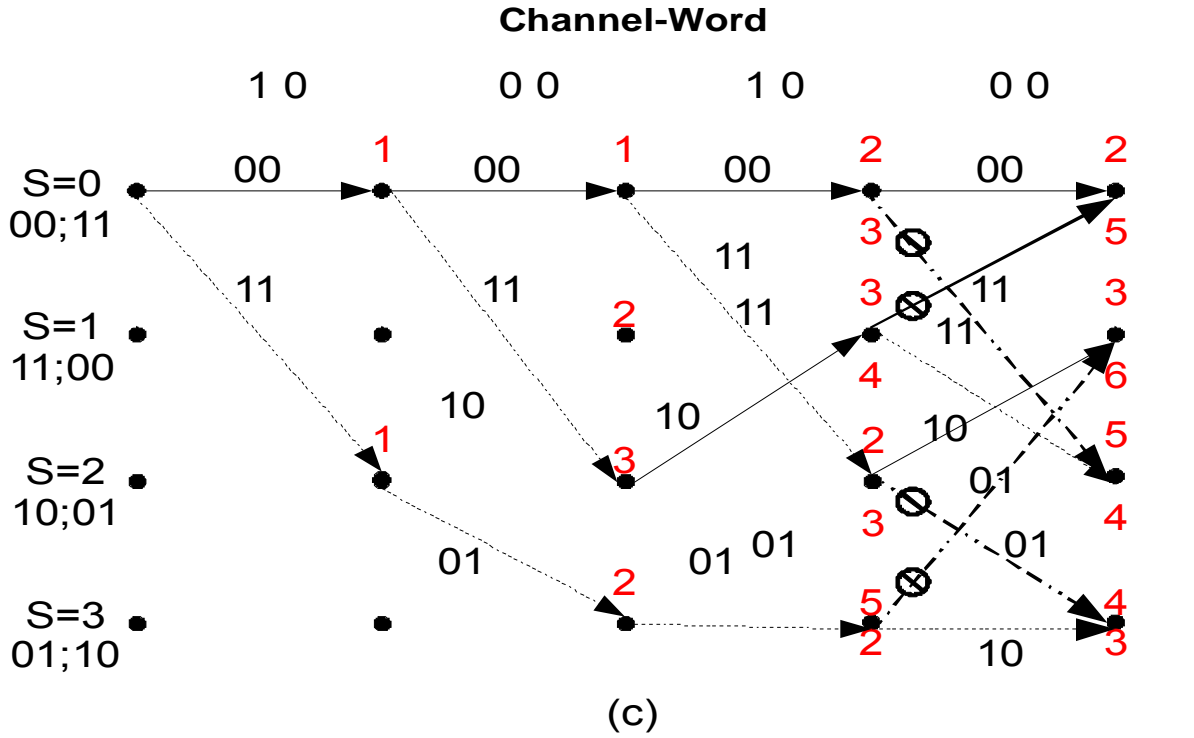


Figure 3.3 Viterbi Algorithm for (7, 5) Convolution Encoder

The encoder is composed of two memory blocks that is there are four possible states. Since the encoder is assumed to have started from state zero, then the VA starts at the initial state zero. As shown in Figure 3.3, the code-bits from state '0' to state '0' is expected to be '00', and '11' for the state '0' to state '2'. The channel-bits at $t=1$ are corrupted and were received as '10'. At that time, that is at that beginning state '0', the encoder have either gone from state '0' to state '0' (itself) or to state '2'. This means that there are two possible paths that the trellis would have taken. The branch metrics or Hamming distances between these initial received channel-bits (10) and the possible code-bits (00 and 11) for both paths ($S=0$ to $S=0$ and $S=0$ to $S=2$) respectively would be '1' for both paths. Next, the channel-bits at time $t=2$ will be considered, that is 00. At time $t=2$, we now have to "active" states that will be considered in the trellis to arrive their next states. In other words, the present states at this time are $S=0$ and $S=2$. Again for $S=0$, the only possible next states are $S=0$ and $S=2$ and as for $S=2$, the only possible next states are $S=1$ and $S=3$, each with their own branch labels or possible code-bits (10 and 01 respectively). The received channel-bits here are 00, which correspond to Hamming distances of (0, 2, 1, 1) for ($S=0$ to $S=0$, $S=0$ to $S=2$, $S=2$ to $S=1$, $S=2$ to $S=3$) respectively. Note that these Hamming distances were just for that time slot $t=2$. Therefore, before going to time $t=3$, we add all the Hamming distances obtained in the prior time slot or states with paths connecting to the next states, specifically (1, 3, 2, 2) for ($S=0$ to $S=0$, $S=0$ to $S=2$, $S=2$ to $S=1$, $S=2$ to $S=3$) respectively as seen in Figure 3.3(a). $S=1$ and $S=3$ are now been added to the equation. $S=1$ can only go to the next states $S=0$ (with only possible bits of 11) and $S=2$ (with only possible bits of 10), while $S=3$ can only go to the next states $S=1$ (with only possible bits of 01) and $S=3$ (with only possible bits of 10). Using the same procedure, we obtain the total Hamming distances for $t=3$ Figure 3.3(b). Before proceeding to time $t=4$ in Figure 3.3(c), we see that the paths start to merge together (Figure 3.3(b)), which cause "conflict", hence a decision has to been as to which paths need to "stay" and which paths need to be "eliminated". The paths marked with the symbol "⊗" are the eliminated paths. Elimination occurs on the paths with the higher accumulated metric at the "conflict" points. In Figure 3.3(b), we can see the paths from ($S=1$ to $S=0$, $S=1$ to $S=2$, $S=2$ to $S=3$, $S=3$ to $S=1$) are eliminated as they have higher accumulated metric values against ($S=0$ to $S=0$, $S=0$ to $S=2$, $S=3$ to $S=3$, $S=2$ to $S=1$) respectively. This process is continued until the end of the received channel-bits, where the path with the lowest accumulated metric is chosen as seen in Figure 3.3(d). That "winner" or "survivor" path is called the survivor path. Therefore, the final decoded message is [0 0 0 0], due to the fact that these bits correspond to the inputs that follow these survivor path created by the trellis encoder.

Sometimes, there are cases (not in our example) where the "conflicting" paths have equal accumulated metric values. In these cases, an arbitrary choice must be made as to which path must be dropped. This is due to the fact that the Viterbi makes its decision based on the maximum likelihood, and hence in this case it cannot decide between these conflicting paths as to which is more likely to be the most probable path.

The complexity of VA can be seen from this example that it increases with the increase of memory blocks of a convolution encoder. An encoder with "m" number of memory blocks will have 2^m number of states. For example, in DSRC systems we will have 2^6 states. Also, it can be seen in Figure 3.3, that there are

always 4 survivor paths for $t \geq 2$. Therefore, for convolution encoder, we will have 2^{n-1} survivor paths after $t \geq n - 1$ for constraint length 'n' (Recall $n = m - 1$). It can therefore be concluded that the Viterbi algorithm is not an ideal decoding scheme when a large constraint length is used, due the exponential increase of the number of survivor paths with increasing constraint length.

3.2.2 Soft-decision Viterbi (SOVA)

Although the demappers of this chapter will not include Soft-input soft-output Viterbi algorithm (SOVA), but it is worth defining it here as it is an important decoding scheme when it comes to turbo decoding. In other words, when the VA is to be used in turbo decoding, then SOVA replaces the soft-input VA. As it will be seen in the next chapter, it is necessary to have the decoders be able to accept soft information as their inputs and produce soft output information when it comes to turbo decoding. However, SOVA is found to be impractical in some scenarios where minimal delay is required, due to the fact that SOVA is found to work best when large packet length are present, hence creating more delays. This section will present the Viterbi algorithm with soft-input and soft-output capabilities.

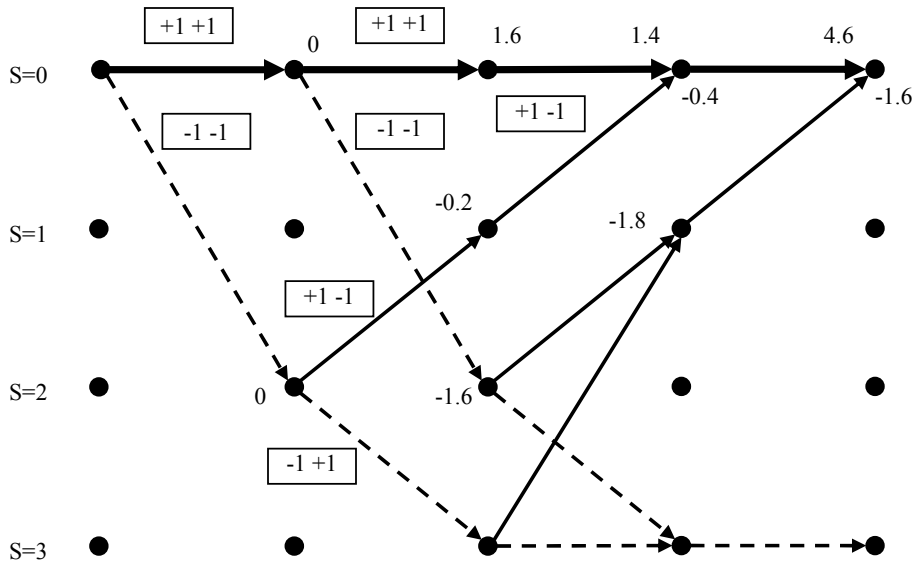


Figure 3.4 Example of Trellis Diagram for SOVA Decoder

Table 3-1: Example of SOVA Decoder

Modulated Message:	+1	+1	+1	+1
Codeword:	+1+1	+1+1	+1+1	+1+1
Channelword:	+0.9 -0.9	+0.7 +0.9	+0.9 +1.1	+1.7 +1.5
Decoded Modulated Message :	+1	+1	+1	+1

SOVA was proposed by [27] in 1989, which is similar to VA, but with slight modification. SOVA outputs a posteriori probabilities (APP). The main difference is in how the branch metric is calculated. An example of how these branch metrics are calculated is given in Figure 3.4 and Table 3-1. To obtain the APP, the SOVA decoder compares the received channelword with the modulated bits instead of the message bit as in the case of VA. For example, in the case of the channelword being +0.9 and -0.9 as in Table 3-1, and assuming that the first state of the Viterbi encoder started at 0 state, this channelword will have two possible next states (0 and 2) and could only have produced the channel bits +1 +1 or -1 -1. Therefore, the channelword will be multiplied by these possible channel bits and added together as follows: for the case of channel bits +1 +1 $\rightarrow (+0.9)(+1) + (-0.9)(+1) = 0$, while for the channel bits -1 -1 $\rightarrow (0.9)(-1) + (-0.9)(-1) = 0$. There we would see the branch metric as 0s for both transitions. The calculations will carry on in this manner until it reaches the final stage where all the channelwords were taken into account. The final decision is made, such that the path with the highest or largest accumulated probability value or branch metric value will be chosen to be the correct path.

3.3 BCJR/ MAP (Maximum A Posteriori) Algorithm

Decision rules can be classified as either maximum likelihood rule, where the databit corresponding to the received channelbit is the element with the highest likelihood. The BCJR algorithm (BCJRA) follows a rule that determines the a-posteriori probability $P(m_i = 1|[\mathbf{y}])$ or $P(m_i = -1|[\mathbf{y}])$, instead of determining the largest a-priori likelihood function, where the decision is made about the databit by looking at the sign of the of the received channel bit. In the process of calculating the a-posteriori probabilities, the decision m_i is based on the sequence of channelbits received $[\mathbf{y}]$. Therefore, the MAP algorithm, which is also known as the BCJR algorithm, was first introduced in 1974 by Bahl, Cocke, Jelinek and Raviv [24] as an optimal means for estimating the a-posteriori probabilities or APPs of the states and transitions of a finite-state Markov process observed over a discrete memoryless channel or DMC [24].

Unlike VA, the BCJRA is a forward/backwards recursive algorithm that minimizes the probability of BER. BCJR was initially not preferred for decoding convolutional codes due to its complexity in examining every possible path through the convolutional decoder trellis. The advantage of BCJR is in its abilities to not just provide the estimation bits, but also the information of how likely it has decoded these bits correctly (probabilistic information). This is why the turbo decoding scheme of [28] became very popular, now that this “probabilistic information” was reused to improve the BER performance and not just discarded as in the case of non-iterative schemes. The detailed mathematical description of the MAP algorithm will follow.

The idea of the MAP decoder is to make a decision on whether the sent databit is a +1 or a -1, where -1 replaced the bit 0 for modulation reasons. First we start by defining:

$$\frac{P(m_i = +1|\mathbf{y})}{P(m_i = -1|\mathbf{y})} \geq 1 \quad (3-3)$$

Equation (3-3) states that if the argument on the left side of the equality is greater than 1, then the hypothesis H1 is assumed to be true or else H2 holds true. H1 and H2 hypotheses are being used here for decision making, where H1 is the decision rule for databit +1 being true and databit -1 being true for H2 hypotheses. When using the well known Bayes rule,

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A) \quad (3-4)$$

Then Equation (3-3) becomes,

$$\frac{P(m_i = +1 \cap \mathbf{y})}{P(\mathbf{y})} >_{H1} \frac{P(m_i = -1 \cap \mathbf{y})}{P(\mathbf{y})} <_{H2} \quad (3-5)$$

The format of Equation (3-5) is useful when it comes to log-likelihood ratio calculations, which will be described in details in the next chapter. We will next present all the equations related to defining the BCJRA as a whole. We will derive the APPs calculations through a step by step procedure. There are three important values that contribute to obtaining the APPs, and they will be defined in the next 3 subsections that will follow. The equations that follow are the BCJRA proposed by [24], but the equations are taken from [23].

3.3.1 Forward Recursion

The forward recursion calculated through $\alpha_t(m)$, and is defined as the state joint probability. It tells us the probability that the state S is m at time t , when the received channel bits sequence is $[Y]$ from time 1 to time t .

$$\alpha_t(m) = Pr\{S_t = m; Y_1^t\} \quad (3-6)$$

It is necessary to add up all the probabilities of the state m for the received channel bits sequence Y_1^t defined above. Therefore, we expand Equation (3-6) to include the summation,

$$\begin{aligned} \alpha_t(m) &= \sum_{m'}^{2^{K-1}-1} Pr\{S_{t-1} = m'; S_t = m; Y_1^t\} \\ &= \sum_{m'}^{2^{K-1}-1} Pr\{S_{t-1} = m'; Y_1^{t-1}\} Pr\{S_t = m; Y_t | S_{t-1} = m'\} \\ &= \sum_{m'}^{2^{K-1}-1} \alpha_{t-1}(m') \Gamma(m', m) \quad \text{for } t = 1, 2, 3, \dots, L \end{aligned} \quad (3-7)$$

Normalizing $\alpha_t(m)$ by $\sum_{m'=0}^{N-1} \alpha_t(m')$ to avoid underflow Equation (3-7) becomes

$$\alpha_t(m) = \sum_{m'}^{N-1} \bar{\alpha}_{t-1}(m') \Gamma_t(m', m) \quad (3-8)$$

Where $\bar{\alpha}_t(m') = \alpha_t(m) / \sum_{m'=0}^{N-1} \alpha_t(m')$

$\alpha_t(m)$ is initialized as follows, $\alpha_0(0) = 1$, and $\alpha_0(m) = 0$, for $m \neq 0$, (at time $t=0$ and state m).

3.3.2 Backward Recursion

The backward recursion calculated through $\beta_t(m)$. It tells us the probability that the received channelbits sequence is Y_{t+1}^L , from time $t + 1$ to time L . With the backward recursion it is given that the state S is m at time t .

$$\beta_t(m) = Pr\{Y_{t+1}^L | S_t = m\} \quad (3-9)$$

Similarly, we obtain the final equatons as follows,

$$\begin{aligned} \beta_t(m) &= \sum_{m'=0}^{N-1} Pr\{S_{t+1} = m'; Y_{t+1}^L | S_t = m\} \\ &= \sum_{m'=0}^{N-1} Pr\{Y_{t+2}^L | S_{t+1} = m'\} Pr\{S_{t+1} = m'; S_t = m\} \\ &= \sum_{m'=0}^{N-1} \beta_{t+1}(m') \Gamma_{t+1}(m, m') \quad \text{for } t = 1, 2, \dots, L - 1 \end{aligned} \quad (3-10)$$

Normalizing $\beta_t(m)$ by $\sum_{m'=0}^{N-1} \beta_t(m')$ to avoid underflow Equation (3-10) becomes

$$\beta_t(m) = \sum_{m'=0}^{N-1} \bar{\beta}_{t+1}(m') \Gamma_{t+1}(m, m') \quad (3-11)$$

$\beta_t(m)$ is initialized similarly to that of $\alpha_t(m)$ as follows, $\beta_t(0) = 1$, and $\beta_t(m) = 0$, for $m \neq 0$, (at time t and state m).

3.3.3 State Transition Matrix

$\Gamma(m', m)$, which is the state transition probability matrix, and is defined as the conditional probability, with state S at time t is $S_t = m$ and the received channel bits sequence again at time t is Y_t , given the state $S_{t-1} = m'$ at time $t-1$.

$$\Gamma(m', m) = Pr\{S_t = m; Y_t | S_{t-1} = m'\} \quad (3-12)$$

By applying Bates' rule, we obtain,

$$\Gamma(m', m) = \sum_U Pr\{U_t | S_t = m; S_{t-1} = m'\} Pr\{S_t = m | S_{t-1} = m'\} Pr\{Y_t | X_t\} \quad (3-13)$$

Where,

$$Pr\{Y_t|X_t\} = \left(\frac{1}{\pi N_0}\right)^{n/2} \exp\left(-\frac{1}{N_0} \sum_{i=1}^n (y_{t,i} - x_{t,i})^2\right) \quad (3-14)$$

and it denotes the main transition probabilities of the received sequence $[Y_t]$ given the all possible transmitted databits $[X_t]$. The value of Equation (3-14) depends on the channel property and in this case it is the AWGN channel, for a rate of $1/n$ code.

The term $Pr\{S_t = m | S_{t-1} = m'\}$ in Equation (3-13) is the a priori information about the coded bits, and this value is usually obtained from the “other” decoder in iterative decoding. Therefore, in a system that is not turbo decoding and in the first iteration of turbo decoding, this value is set as 0.5, for equal probable of all the symbols.

Finally term $Pr\{U_t | S_t = m; S_{t-1} = m'\}$ in Equation (3-13) is a value of either a ‘1’ or a ‘0’. The value is ‘1’ when there is a transition between two states in the trellis of the encoder. Otherwise it will be a ‘0’.

3.3.4 APPs of the Symbols

Deriving the equations for the forward and backward feedbacks and the state transitions, allow us to carry forward and obtain what the BCJRA is meant to obtain, and that is the a posteriori probabilities (APPs). The estimated APPs of the states and the state transitions for the received channel bits sequence is defined using one of two equations as follows,

$$\lambda_t(m) = Pr\{S_t = m; Y_1^L\} = \frac{Pr\{S_t = m; Y_1^t\}}{Pr\{Y_1^L\}} \quad (3-15)$$

and

$$\sigma_t(m', m) = Pr\{S_{t-1} = m'; S_t = m; Y_1^L\} = \frac{Pr\{S_t = m; S_{t-1} = m'; Y_1^L\}}{Pr\{Y_1^L\}} \quad (3-16)$$

Where $Pr\{Y_1^L\}$, the probability of the received channel bits sequence, remains constant throughout the entire calculations. Therefore, Equations (3-15) and (3-16) respectively become,

$$\begin{aligned} \lambda_t(m) &= Pr\{S_t = m; Y_1^L\} = Pr\{S_t = m; Y_1^t\} \\ \lambda_t(m) &= Pr\{S_t = m; Y_1^t\} Pr\{Y_{t+1}^t | S_t = m; Y_1^t\} \\ \lambda_t(m) &= \alpha_t(m) Pr\{Y_{t+1}^L | S_t = m\} \\ \lambda_t(m) &= \alpha_t(m) \beta_t(m) \end{aligned} \quad (3-17)$$

and,

$$\begin{aligned} \sigma_t(m', m) &= Pr\{S_{t-1} = m'; S_t = m; Y_1^L\} = Pr\{S_t = m; S_{t-1} = m'; Y_1^L \sigma_t(m', m)\} \\ &= Pr\{S_{t-1} = m'; Y_1^{t-1}\} Pr\{S_t = m; Y_T | S_{t-1} = m'\} Pr\{Y_{t+1}^L | S_t = m\} \\ \sigma_t(m', m) &= \alpha_{t-1} \Gamma_t(m', m) \beta_t(m) \end{aligned} \quad (3-18)$$

Therefore, it can be seen from the equations above that the APPs rely on all 3 values, i.e. $\alpha_t(m)$, $\beta_t(m)$ and $\Gamma_t(m', m)$.

$\lambda_t(m)$ is usually used for NRC codes and $\sigma_t(m', m)$ is used for RSC codes. The transitions for NRC and RSC codes are shown in Figure 2.9 and Figure 2.8, respectively, Figure 3.5 illustrates the system diagram of the MAP decoding algorithm.

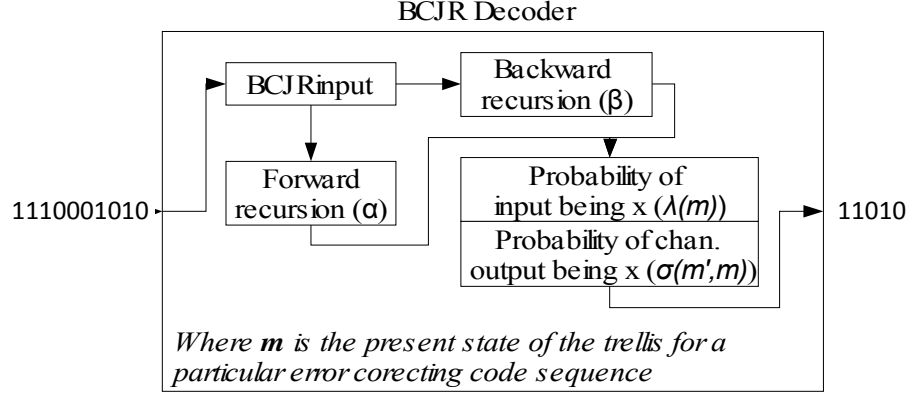


Figure 3.5 System Diagram of the MAP decoding Algorithm

For the final calculations in obtaining the APPs we may either choose to do the probabilities where there is a transition of '1' or '0', for the '1' transitions states,

$$Pr_{APP}[u_t = 1] = \frac{\sum_{m \in A} \lambda_t(m)}{\sum_{m \in \text{All States}} \lambda_t(m)} \quad (3-19)$$

where A is all the '1' transitions states, and

$$Pr_{APP}[u_t = 1] = \frac{\sum_{(m,m') \in B} \sigma_t(m', m)}{\sum_{(m,m') \in \text{All branches}} \sigma_t(m', m)} \quad (3-20)$$

where B is all '1' transitions branches. The decoder gives the hard outputs as an estimation of the input symbols, as follows,

$$\hat{u}_t = \begin{cases} 1, & \text{if } Pr_{APP}[u_t = 1] \geq Pr_{APP}[u_t = -1] \\ -1, & \text{if } Pr_{APP}[u_t = 1] < Pr_{APP}[u_t = -1] \end{cases} \quad (3-21)$$

3.4 Performance of DSRC Systems using Different Demapping and Decoding Schemes

As explained in Chapter 2, and based on the 802.11a standard, the mapping schemes available for DSRC system are BPSK, QPSK, 16QAM and 64 QAM. The detailed explanations of these mapping schemes can be found in [29] and [30].

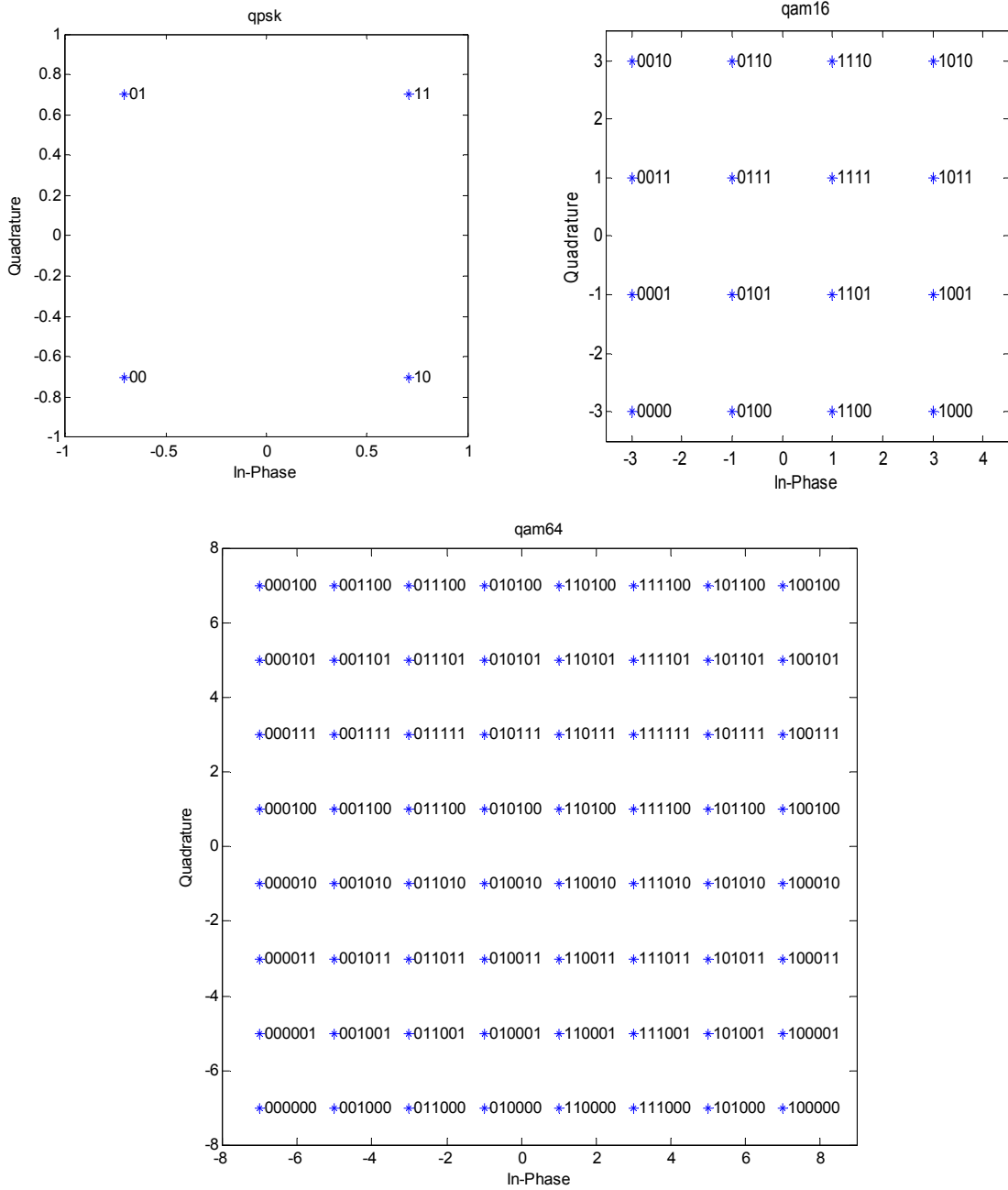


Figure 3.6 Gray-Coded of Constellation Points for QPSK, 16 QAM and 64 QAM Mappers

The received signal in subcarrier can be expressed as

$$\mathbf{y}_{n,sc} = \mathbf{h}_{n,sc} \cdot \mathbf{x}_{n,sc} + \mathbf{w}_{n,sc} \quad (3-22)$$

where $\mathbf{y}_{n,sc}$, $\mathbf{h}_{n,sc}$, $\mathbf{x}_{n,sc}$, $\mathbf{w}_{n,sc}$ denote the FFT of the received OFDM symbols, the channel response, the transmitted OFDM symbols, and the AWGN of the n^{th} OFDM symbol at the sc^{th} subcarrier, respectively. Here $\mathbf{x}_{n,sc}$ can be defined as the complex data (modulated) symbol \mathbf{x} at the n^{th} OFDM symbol and the sc^{th} subcarrier, where $\mathbf{x} = \mathbf{x}_I + j\mathbf{x}_Q$, i.e. \mathbf{x}_I is the in-phase component and \mathbf{x}_Q is the quadrature-phase

component for QPSK, 16 and 64 QAM data symbols. Note that for BPSK only the in-phase component exists. For QPSK the in-phase is mapped using 1 bit and so is its quadrature component. The difference with the 16 and 64 QAM mappers lies in the number of bits that represent their in-phase and quadrature components (4 bits for 16 QAM and 6 bits for 64 QAM). The figures below, Figure 3.1, demonstrate how the bit sequences are placed in the Gray-coded constellation points of QPSK, 16 QAM, and 64 QAM.

In the sections that follow, we will present three different demapping techniques that we will use in the DSRC system and compare their performances when used along side with both the standard digital Viterbi and BCJR decoders. Without loss of generality, we use QPSK and 16 QAM in this study to obtain the performance figures of the comparison of these three demappers under different environments, for example *Rayleigh* and *Rician* faded environments. Also, the DSRC system will be tested in varying SNR (both low and high SNR) and varying Doppler frequencies or velocities. The results can be extended to the 64 QAM modulation.

The purpose of a demapper is not only to undo what the mapper did, but to also assist the decoders (especially in the case of soft decoding) make better decisions by given the decoders probabilistic values as to what the message may have been after the effect of the signal compensator. The three demapping schemes are:

- 1- A demapper with similar concept as the 3rd one in this list (section 3.4.3 is introduced, but instead of having the curve lines defining the boundaries soft values, we'll have linear or constant lines.
- 2- We use the simplified LLR (Log-Likelihood Ratio) output derived in [31] as our second demapper, and
- 3- Finally, the proposed S and Π -Decision demapper of [32] will be our 3rd demapper.

3.4.1 Proposed Linear-based Demapper

In this section, a linear-based demapping scheme will be presented and then used in our soft decoding comparisons. The demapping scheme introduced here is used to assist both the soft-input BCJR and Viterbi decoders, and as will be shown from the results, this scheme outperforms the other two schemes when it comes to line of sight (LOS) exist and in the case of QPSK at relatively high SNR.

Earlier in this chapter, both hard-decision and soft-decision Viterbi were explained. Soft-decision decoder was shown to outperform the hard-decision decoder when it comes to minimizing the BER [20]. It is reported that an approximately 2dB coding gain is achieved using a soft-decision decoder over a hard-decision decoder [32].

Figure 3.6 demonstrates how the encoded bits sequence in the transmitter is mapped into the constellation points. Each constellation point in the Gray-coded mapper holds the in-phase and the quadrature-phase components (only the in-phase in the case of BPSK). For 16 QAM for example, from left to right the in-phase components are the first 2 bits of the 4-bit constellation points, and the last 2 bits are of the quadrature components. Therefore, a 4 bit encoded message bits sequence ($B_{I,1}, B_{I,2}, B_{Q,1}, B_{Q,2}$) are mapped according to the mapping schemes of Figure 3.6b for 16 QAM. B stands for the encoded message bit, I for the in-phase component and Q for the quadrature component. Also, it is observed that the I and Q

message sequence (encoded message sequence will be referred to as message sequence for the rest of this chapter) follow the same patterns in the constellation points, i.e. the 8-bit pattern of the I -components, that are composed of the first 2 bits from left to right of the constellation points have the same 8-bit pattern of the Q -components, that are composed of the last 2 bits from bottom to top of the constellation points (00 01 11 10). The message bits sequence ($B_{I,1}, B_{I,2}, B_{Q,1}, B_{Q,2}$) can be seen visually in Figure 3.7, where $B[I,Q,1]$ stand for the first bit for both the in-phase and quadrature components and $B[I,Q,2]$ stand for the second bit for both the in-phase and quadrature components. Again, these regions were chosen because of how the constellation points are placed in these Gray-coded modulation schemes.

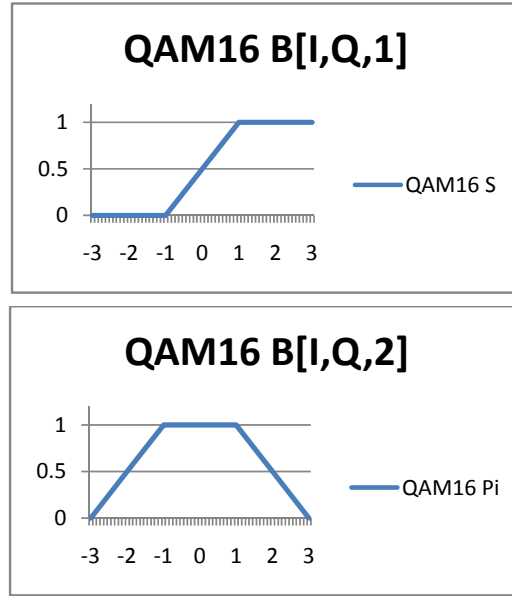


Figure 3.7 The demapping regions of all message bits sequence

Mathematically, the message bits sequence ($B_{I,1}, B_{I,2}, B_{Q,1}, B_{Q,2}$) are defined as follows:

$$B_{I,1} = \begin{cases} 0, & y_I[i] \leq -1 \\ \frac{(y_I[i] + 1)}{2}, & |y_I[i]| < 1 \\ 1, & y_I[i] > 1 \end{cases} \quad (3-23)$$

$$B_{I,2} = \begin{cases} 0, & y_I[i] \leq -3 \\ \frac{(y_I[i] + 3)}{2}, & -3 < y_I[i] < -1 \\ 1, & |y_I[i]| \leq 1 \\ \frac{-(y_I[i] - 3)}{2}, & 1 < y_I[i] < 3 \\ 0, & y_I[i] \geq 3 \end{cases} \quad (3-24)$$

$$B_{Q,1} = \begin{cases} 0, & y_Q[i] \leq -1 \\ (y_Q[i] + 1)/2, & |y_Q[i]| < 1 \\ 1, & y_Q[i] > 1 \end{cases} \quad (3-25)$$

$$B_{Q,2} = \begin{cases} 0, & y_Q[i] \leq -3 \\ \frac{(y_Q[i] + 3)}{2}, & -3 < y_Q[i] < -1 \\ 1, & |y_Q[i]| \leq 1 \\ -\frac{(y_Q[i] - 3)}{2}, & 1 < y_Q[i] < 3 \\ 0, & y_Q[i] \geq 3 \end{cases} \quad (3-26)$$

For the QPSK, the same principles apply, but we only have one bit representing the in-phase component and one bit for the quadrature component, and therefore the equations defined in (3-23) and (3-25) for $B_{I,1}$ and $B_{Q,1}$ will be the equation used for the QPSK schemes as B_I and B_Q , respectively.

3.4.2 Proposed LLR/Probabilistic Demapper

In [31], a set of equations were derived to define an approximation for the log likelihood ration (LLR). The equations were approximated using the LLR definition combined with transforming the intervals, where the constellation points lie, into binary elements of +1 and -1, in order for the LLR computation to take place (Note that Chapter 4.1 describes LLR computation in details). In [31] the mapping of these intervals is done as follows:

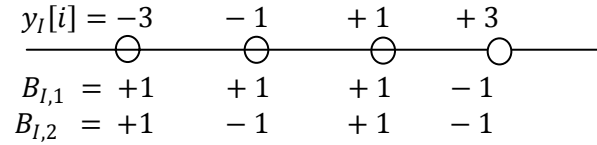


Figure 3.8 Mapping of 16 QAM In-Phase Symbols into Binary Elements

The same is done for the quadrature components. By definition the conditional LLR of a random variable B is defined as [33]:

$$L_B(B_{I,1}) = \ln \frac{P_B(B_{I,1} = +1/y_I[i])}{P_B(B_{I,1} = -1/y_I[i])} \quad (3-27)$$

Where $P_B(B_{I,1})$ denotes the probability of the random value B with elements $\{+1, -1\}$, where +1 is the “null” element and -1 is the +1 element. Again, a more detailed description of LLR can be found in the next chapter.

With the interval mapped into binary elements, the LLR in equation (3-27) can now be defined as:

$$L_B(B_{I,1}) = \ln \frac{P_B(B_{I,1} = +1/y_I[i]) + P_B(B_{I,1} = +3/y_I[i])}{P_B(B_{I,1} = -1/y_I[i]) + P_B(B_{I,1} = -3/y_I[i])} \quad (3-28)$$

It follows from [31], that at high signal to noise ratio (SNR), the approximation of (3-8) can be expressed as,

$$LLR(B_{I,1}) \cong \begin{cases} y_I[i], & |y_I[i]| \leq 2 \\ 2(y_I[i] - 1), & y_I[i] > 2 \\ 2(y_I[i] + 1), & y_I[i] < -2 \end{cases} \quad (3-29)$$

Where $y_I[i]$ denotes the received in-phase compensated signal of the data symbol $\mathbf{y}_{n,sc}$. The LLR was normalized to the standard deviation of $y_I[i]$. Similarly, we can get the LLR of $B_{I,2}$ to be [31],

$$LLR(B_{I,2}) \cong -|y_I[i]| + 2. \quad (3-30)$$

It can easily be verified that the same could be done for the quadrature components, i.e. the same could be done for bits $LLR(B_{Q,1})$ and $LLR(B_{Q,2})$, by replacing $y_I[i]$ to $y_Q[i]$. These LLRs will be de-interleaved and passed on to the decoder. There is one more thing that needs to be done, and that is, transferring these LLR values need to be translated into probabilistic values to be fed into the decoder's input, and that is done as follows [22]:

$$P(B_{I,1}|[y_I[i]]) = \frac{e^{\{[y_I[i]](LLR(B_{I,1}|y_I[i]))\}}}{1 + e^{\{[y_I[i]](LLR(B_{I,1}|y_I[i]))\}}} \quad (3-31)$$

The probability values for the other bits ($B_{I,2}$, $B_{Q,1}$, $B_{Q,2}$) can be done in a similar way.

For the QPSK demapper, the LLR is estimated as the signal compensated value $y_I[i]$ and that is then transformed into the probabilistic values before fed into the decoder.

3.4.3 S and Π -Decision Demapper

The S and Π -Decision demapper proposed in [32] also exploits and uses the principles that define the pattern of the message bits in a Gray-coded modulation scheme, as described in section 3.4.1. Therefore, due to the similarities of these two demappers, i.e. the demapper in section 3.4.1 with the S and Π -Decision demapper proposed in [32], we will only include the piece-wise equations that define the S and Π -Decision demapper's output. The message bits sequence are defined as the S and Π as follows [32]:

$$S(y_I[i]; \alpha, \beta, \gamma) = \begin{cases} 0, & y_I[i] \leq \alpha \\ 2 \left(\frac{y_I[i] - \alpha}{\gamma - \alpha} \right)^2, & \alpha \leq y_I[i] \leq \beta \\ 1 - 2 \left(\frac{y_I[i] - \gamma}{\gamma - \alpha} \right)^2, & \beta \leq y_I[i] \leq \gamma \\ 1, & y_I[i] \geq \gamma \end{cases} \quad (3-32)$$

$$\Pi'(y_I[i]) = \begin{cases} 0, & y_I[i] \leq -3 \\ S(y_I[i]; -3, -2, -1), & -3 \leq y_I[i] \leq -1 \\ 1, & -1 \leq y_I[i] \leq 1 \\ 1 - S(y_I[i]; 1, 2, 3), & 1 \leq y_I[i] \leq 3 \\ 1, & y_I[i] \geq 3 \end{cases} \quad (3-33)$$

Again, $y_I[i]$ denotes the received in-phase compensated signal of the data symbol $\mathbf{y}_{n,sc}$. The S and Π are just a different representation for the bits $(B_{I,1}, B_{I,2}, B_{Q,1}, B_{Q,2})$ defined in section 3.4.1. The plots that define the piece-wise equations defined above are shown in Figure 3.9 below.

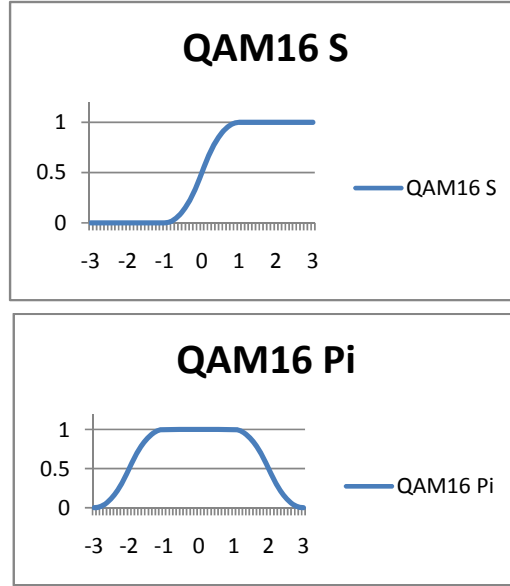


Figure 3.9 S and Π -Decision Rules for 16 QAM

For the QPSK, the same principles apply, but we only have one bit representing the in-phase component and one bit for the quadrature component, and therefore the equation defined in (3-32) as $S(y_I[i]; \alpha, \beta, \gamma)$ and $S(y_Q[i]; \alpha, \beta, \gamma)$ for the quadrature component will be the equations used for the QPSK schemes for both in-phase and quadrature components, respectively. Therefore, the Π -Decision rule is not included in the QPSK modulation scheme. That can be verified in how the constellation points are placed in a Gray-coded QPSK modulation scheme.

3.4.4 Simulation Results

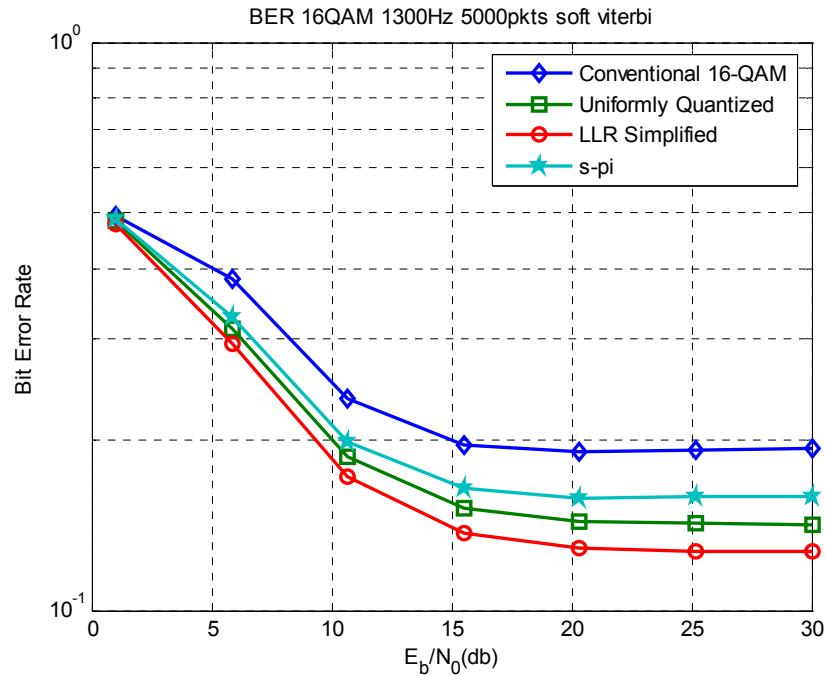
The different modulation schemes presented earlier were simulated using Matlab v7 software. The “Linear demapper” is the demapper that we proposed, which turned out to perform better than the other two demappers when it comes to high SNR in QPSK modulation scheme. The comparisons are done by applying the same conditions on all three demapping schemes in DSRC system.

One of the two other demappers, in particular the ‘‘Simplified LLR demapper’’ is a demapper proposed by [31] which allows a lesser complex demappers design compared to the more complicated MAP algorithm demapping scheme. Therefore, we took that method and adjusted it to accept the signal compensated symbol data and performed the simplified LLR algorithm on them, and then were fed to the decoders. As mentioned earlier, the purpose for these simulations is to allow us to understand or at least shift our attention from both the conventional decoder and the demapper to something else within the system if the performance of the DSRC system, when using the soft demapping and decoding schemes, doesn’t improve by much.

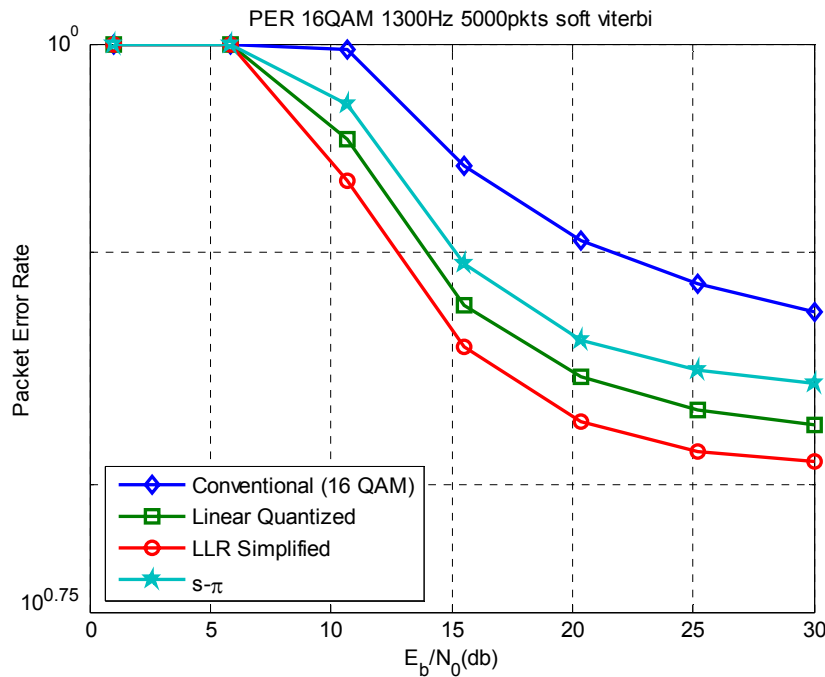
We have simulated the DSRC system under different channel environments, i.e. both Rayleigh faded (no-LOS) and Ricean faded (LOS) environments were considered. AWGN was also present in the channel. All three demappers presented in this chapter, alongside with different decoders, MAP-based (BCJR) and ML-based (soft-input Viterbi) decoders were compared with one another and the conventional DSRC system that uses the hard output demappers and the hard-input Viterbi decoder were the tested in the simulations. Table 3-2 provides the summary of the range of values of the variables used in the simulations.

Table 3-2: Simulation Values

Name	Abbreviation	Unit	Values
Signal-to-Noise Ratio	SNR	dB	[1, 2, 3, 6, 10, 15, 20, 25, 30]
Maximum Doppler Frequency (velocity)	f_m (v)	Hz (km/h)	[1300] [238]
Symbol Duration	T_s	μs	[8]
Packet Length	N_s	Symbols/packet	[60]
Rice Factor	K		[0, 1]
Modulation	m		[QPSK, 16QAM]
Decoding			[hard, soft, LLR]

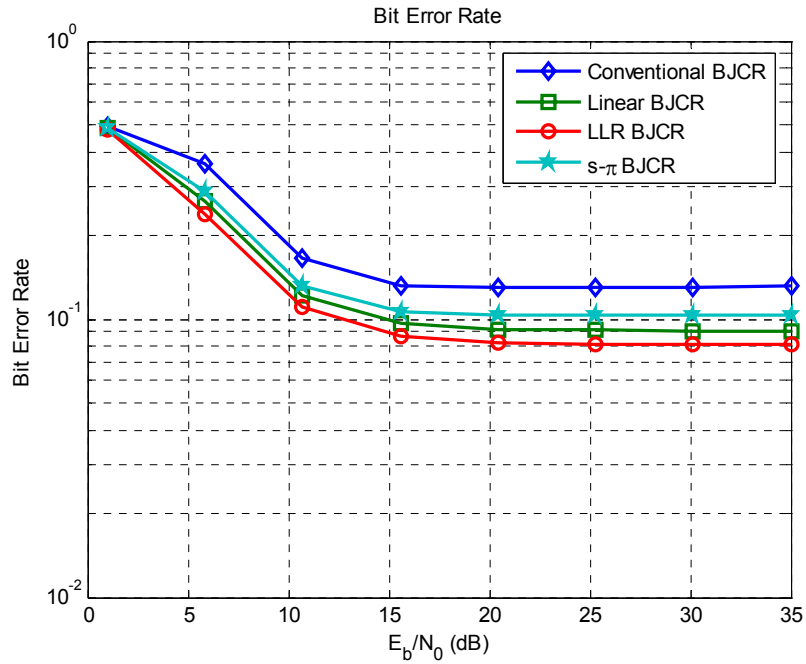


a) BER vs. SNR

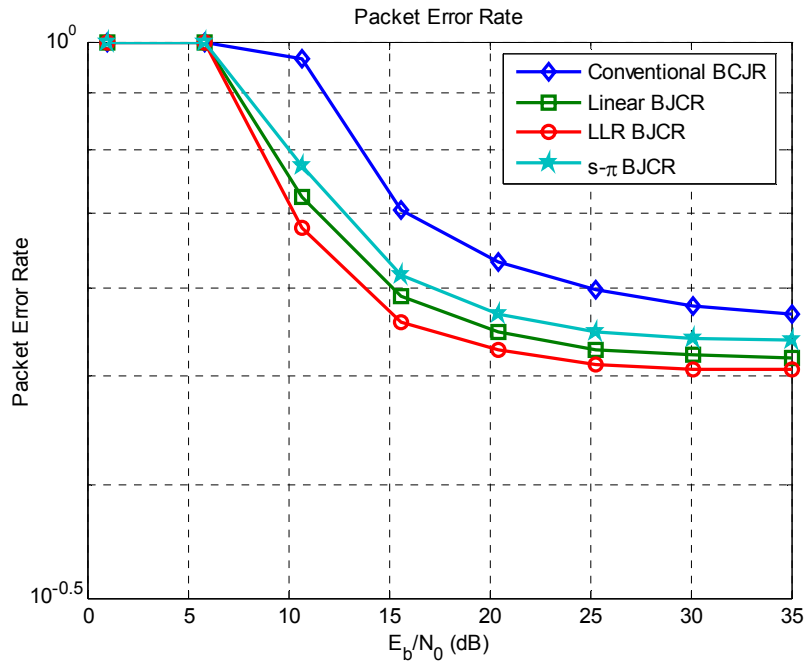


b) PER vs. SNR

Figure 3.10 16QAM: Conventional vs. Soft demappers using soft-input Viterbi under Rayleigh fading at 238 km/h



a) BER vs. SNR



b) PER vs. SNR

Figure 3.11 16QAM: Conventional vs. Soft demappers using BCJR under Rayleigh fading at 238 km/h under Rayleigh fading at 238 km/h

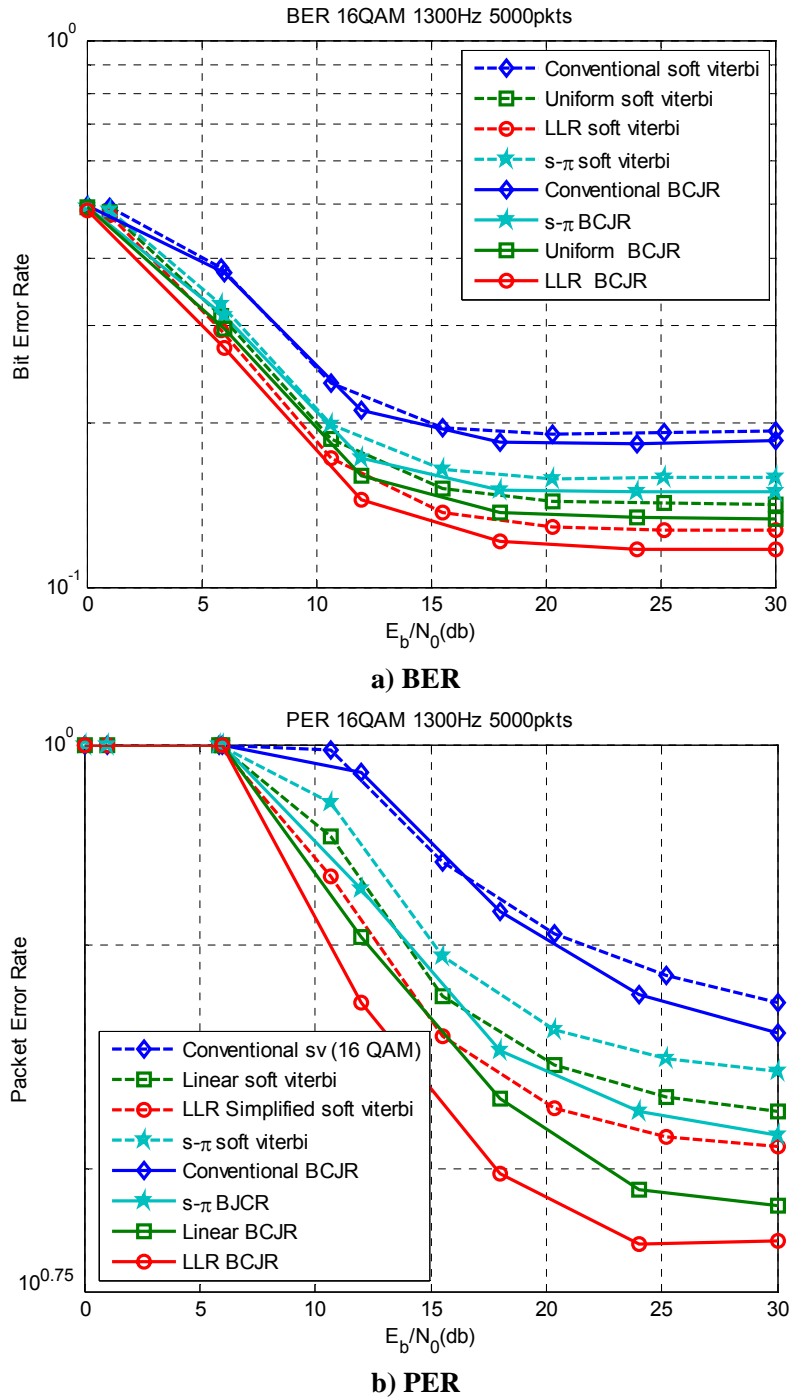
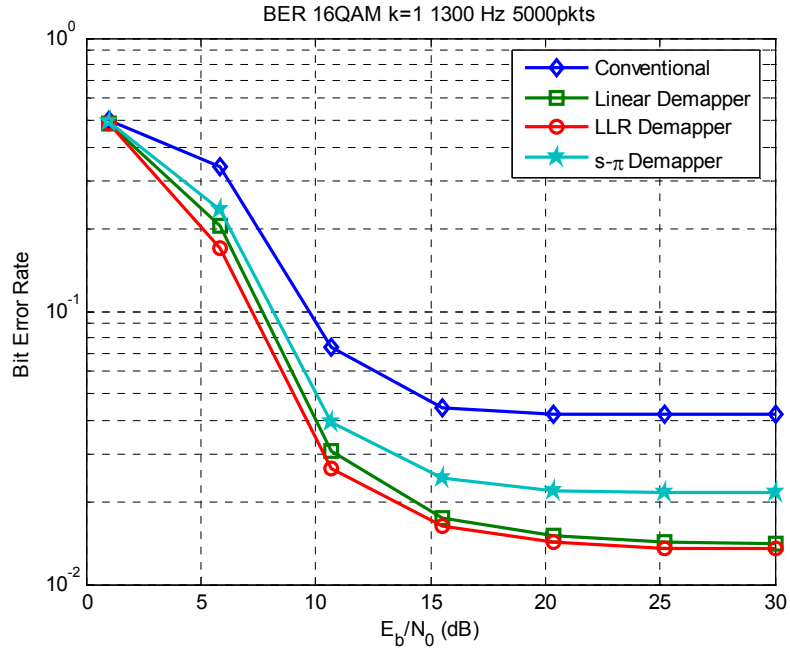
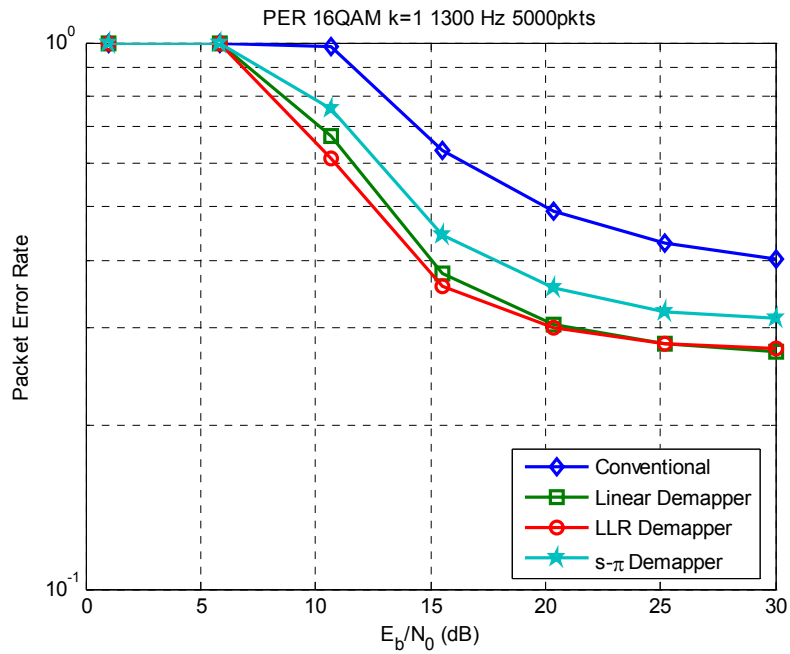


Figure 3.12 16QAM: Conventional vs. soft demappers using both soft-input Viterbi and BCJR under Rayleigh fading at 238 km/h under Rayleigh fading at 238 km/h

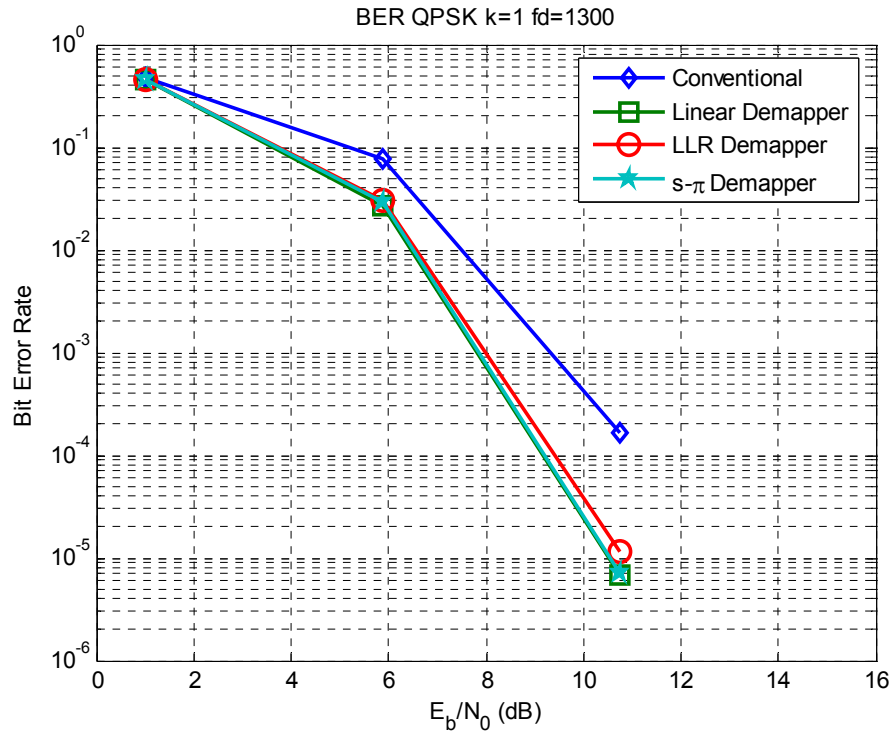


a) BER vs. SNR

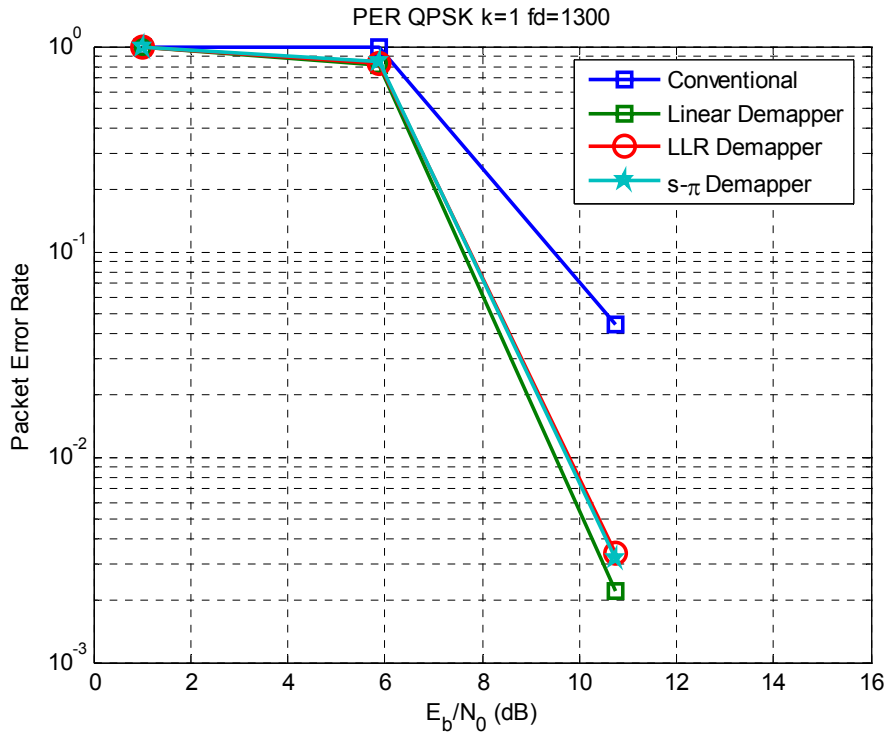


b) PER vs. SNR

Figure 3.13 16QAM: Conventional vs. Soft demappers using soft-input Viterbi under Ricean fading channel $K=1$



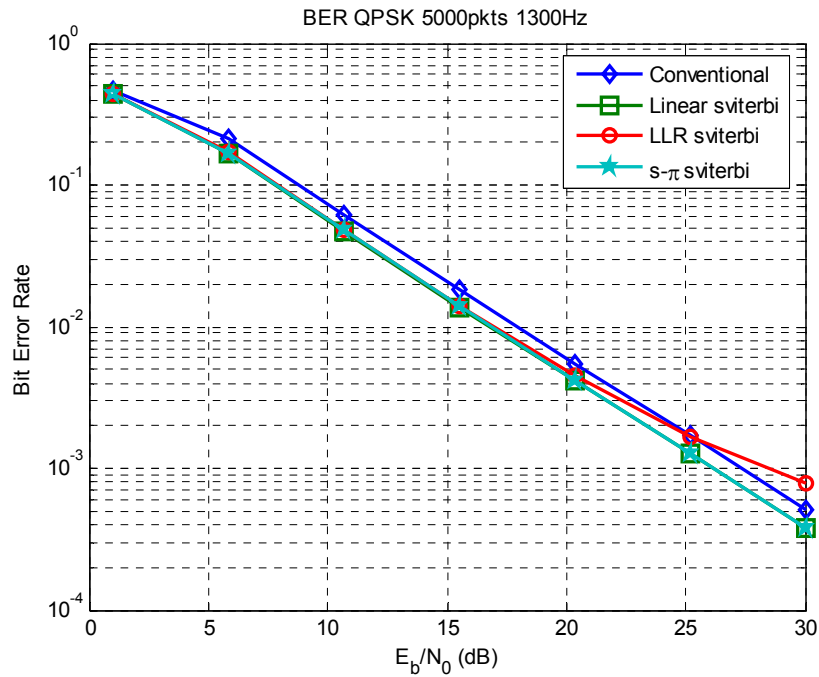
a) BER vs. SNR



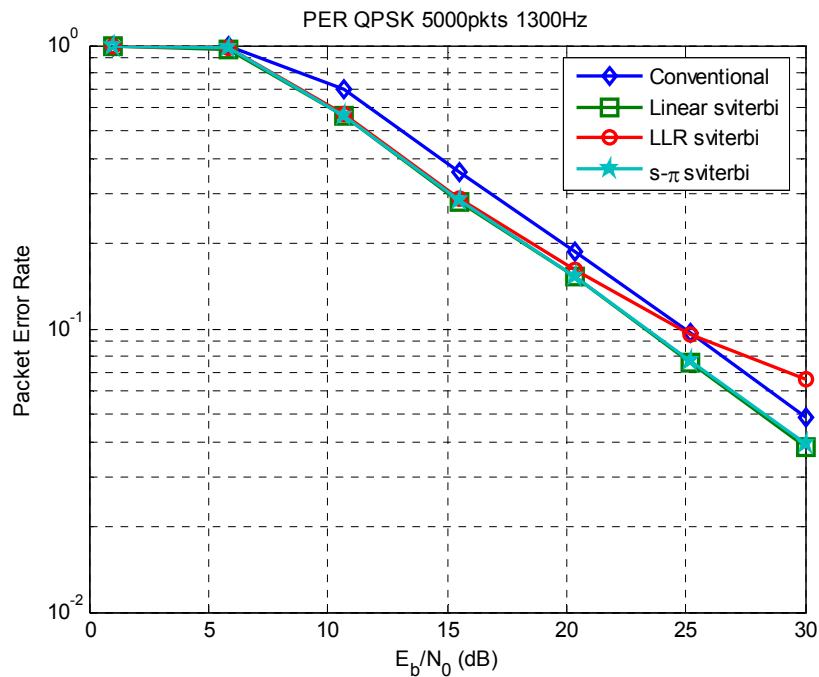
b) PER vs. SNR

Figure 3.14 QPSK: Conventional vs. Soft demappers using soft-input Viterbi under Ricean fading channel

$K=1$

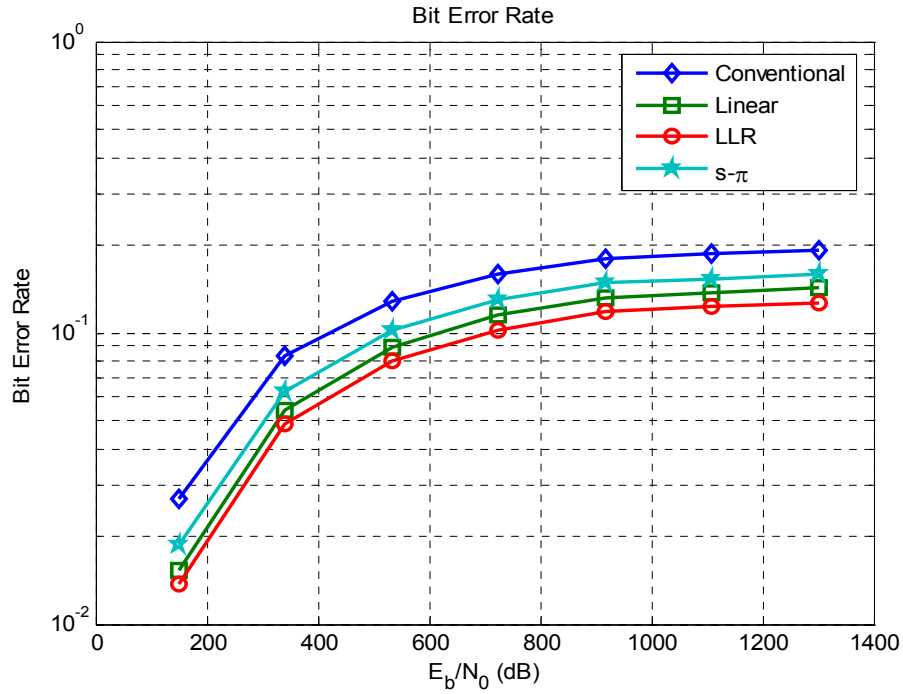


a) BER VS. SNR

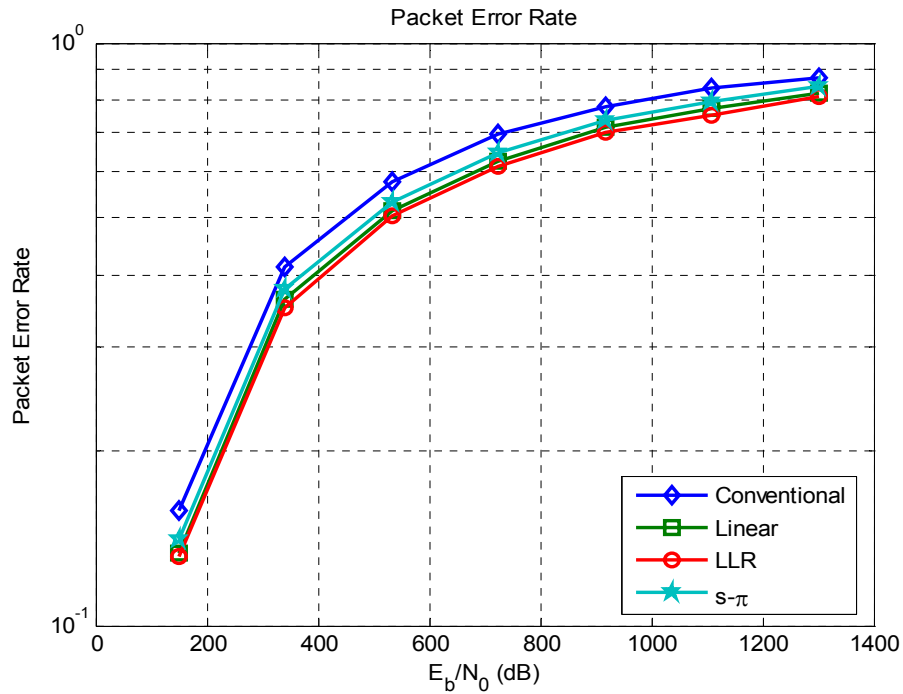


b) PER VS. SNR

Figure 3.15 QPSK: Conventional vs. Soft demappers using soft-input Viterbi under Rayleigh fading at 238 km/h

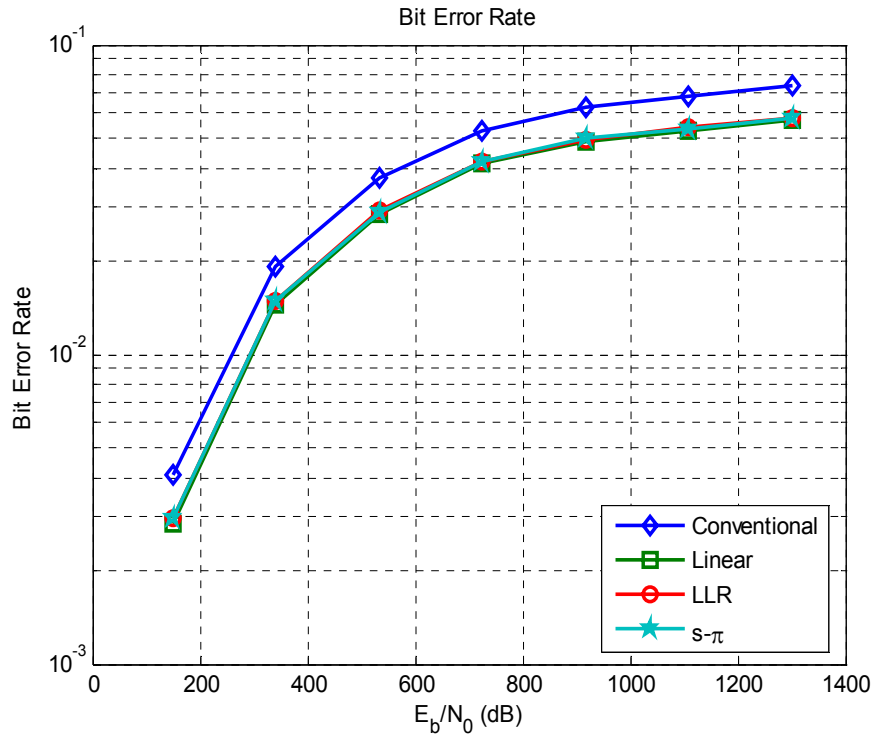


a) BER vs SNR for soft viterbi vs. conventional decoding

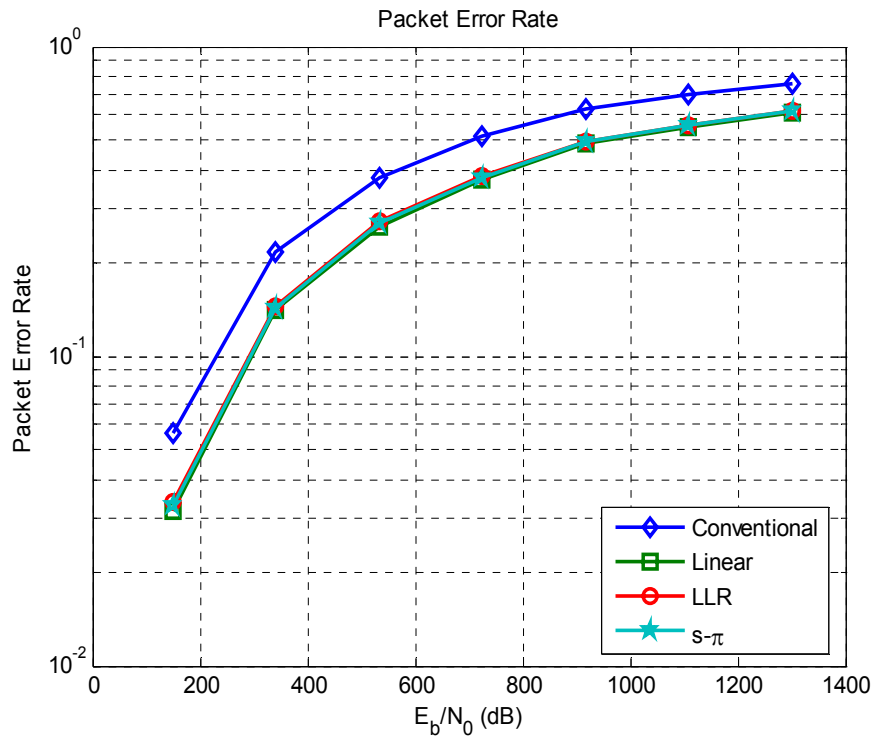


b) PER vs. SNR for soft viterbi vs. conventional decoding

Figure 3.16 16QAM velocity: Conventional vs. Soft demappers using soft-input Viterbi under Rayleigh fading varying velocities at 10dB

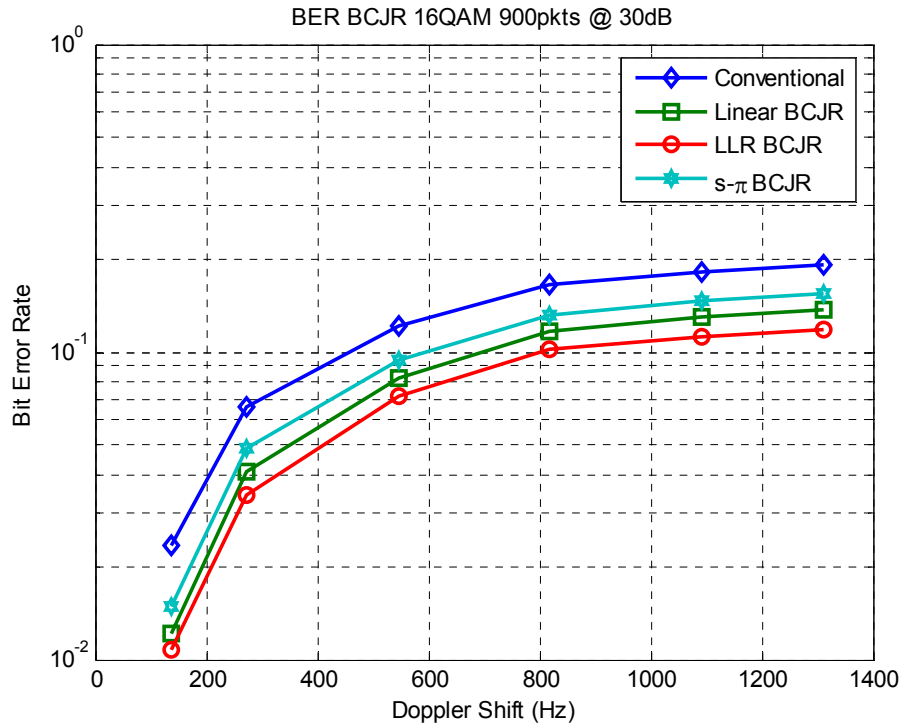


a) BER vs. Velocity

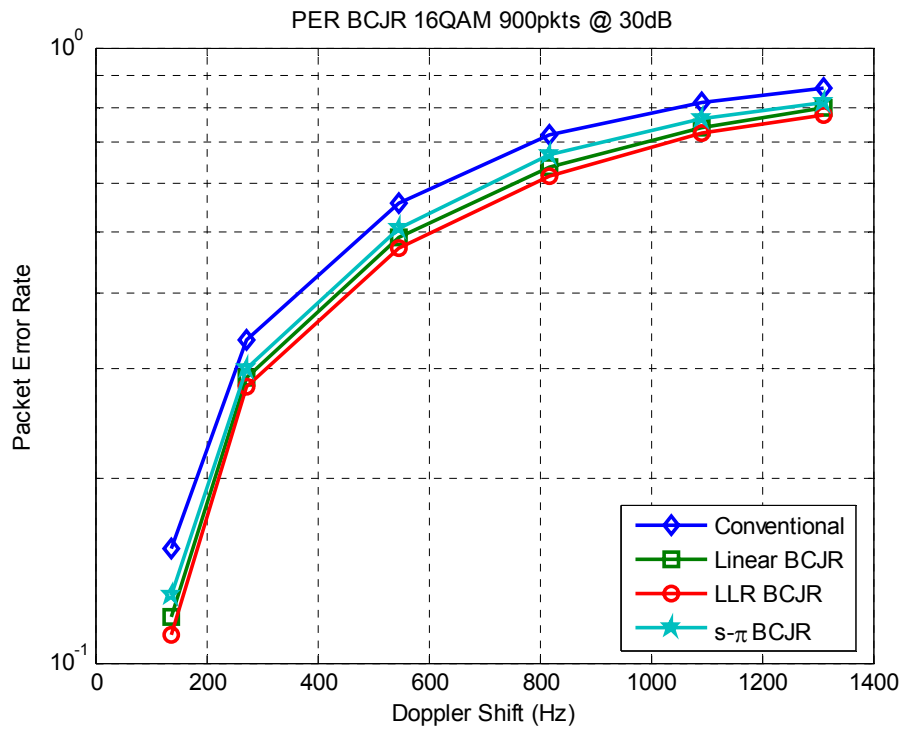


b) PER vs. Velocity

Figure 3.17 QPSK velocity: Conventional vs. Soft demappers using soft-input Viterbi under Rayleigh fading varying velocities at 30dB

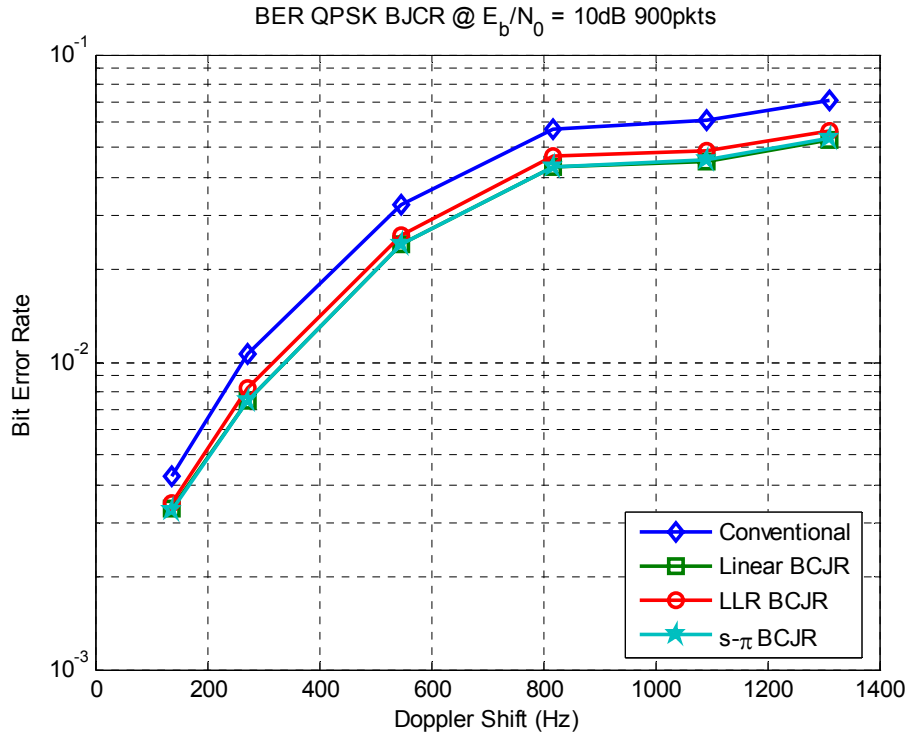


a) BER vs. Velocity

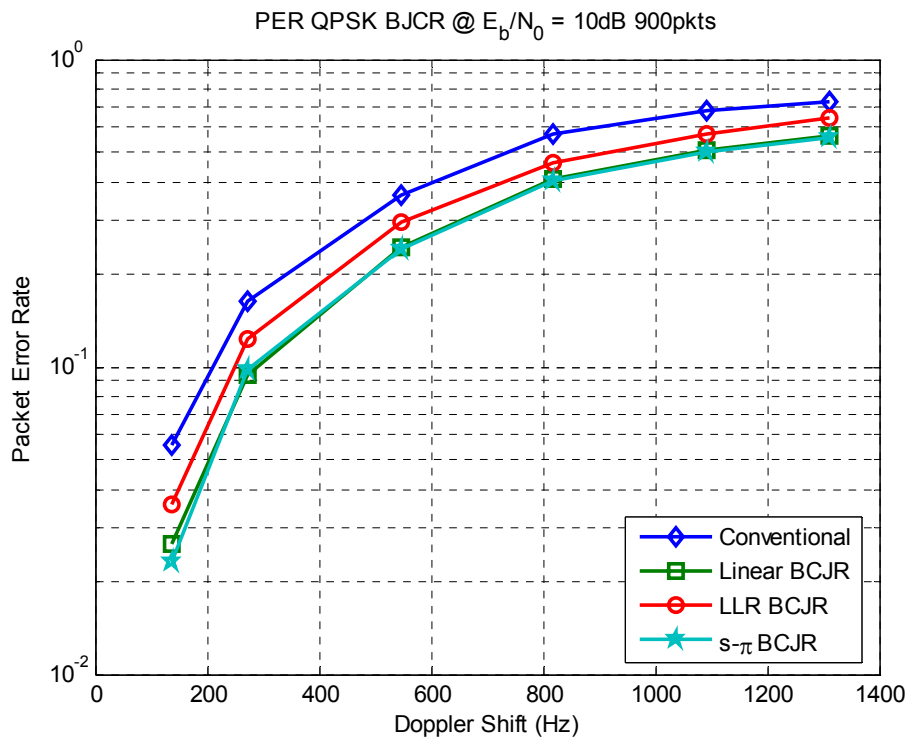


b) PER vs. Velocity

Figure 3.18 16QAM velocity: Conventional vs. Soft demappers using BCJR under Rayleigh fading varying velocities at 30dB

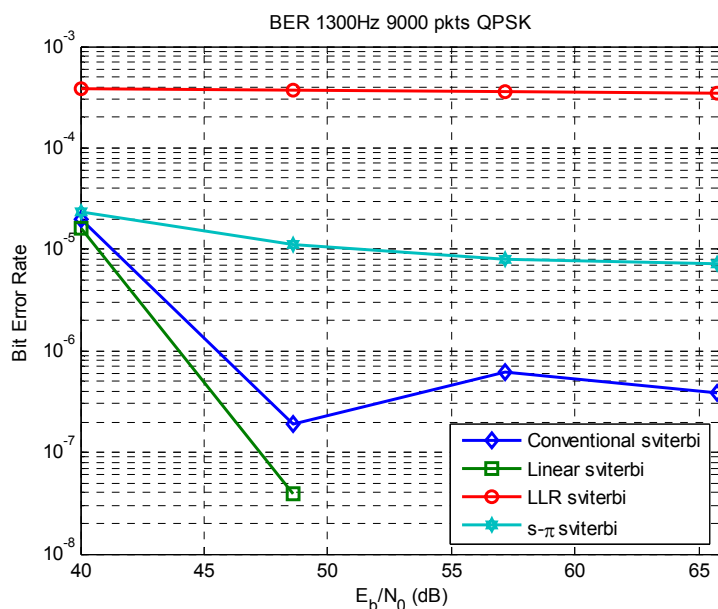


a) BER vs Velocity



b) BER vs Velocity

Figure 3.19 QPSK velocity: Conventional vs. Soft demappers using BCJR under Rayleigh fading varying velocities at 30dB



a) BER vs. SNR

Figure 3.20 QPSK: Conventional vs. Soft demappers using soft-input Viterbi under Rayleigh fading at varying high SNR

From all the simulation results, we can conclude that all the soft demappers performed better than the conventional DSRC system. The simulation results also tells us that the DSRC system suffers from irreducible error floor when the 16 QAM modulation is considered regardless of what kind of demapping scheme is used. We can also see from these results that the proposed Linear demapper outperformed both the $s-\pi$ and the LLR demappers when the systems were tested under QPSK modulation scheme and at high SNR. In fact both the $s-\pi$ and the LLR demappers, but not the Linear demapper, suffered irreducible error floor in the QPSK modulated system as seen in Figure 3.20 . The LLR demapper slightly outperformed the other two demappers when 16QAM modulation was used. As for the decoders' comparisons, the BCJR had a very slight improvement over the soft-input Viterbi in the case of 16 QAM only for SNR vs. BER plots (with $f_d = 1311\text{Hz}$). For the BER velocity plots, the soft-Viterbi was outperformed by the BCJR, especially at low Doppler frequencies (or velocities) for QPSK demapping and 30 dB SNR. At $f_d=1311$ Hz however, they both performed the same. What is interesting though is that when it came to 16 QAM both the soft Viterbi and BCJR performed very much similar.

All the simulation results suggest that the conventional DSRC system suffers from severe degradation in performance because of the effect of the channel's response on the transmitted symbols. Therefore, investigation should now be shifted on the remaining two algorithms, i.e channel estimation and equalization. The results obtained here may serve as a benchmark for future research.

Chapter 4

Performance Enhancement of DSRC Systems

In the previous chapter, a detailed derivation of the BCJR algorithm was presented. In this chapter, we combine what we've learned to derive the proposed DSRC receiver design using frequency-domain MAP equalization and MAP decoding scheme. This chapter starts with an efficient tool used in the proposed scheme called *log-likelihood ratio* LLR. After that, the proposed DSRC receiver will be designed by deriving the LLR output for the case of the frequency-domain MAP equalizer, and the LLR output of the MAP decoder as well. Next will be simulation results. Next chapter will be used to discuss the conclusions and future works.

4.1 Tools for (Iterative) Decoding of Binary Codes

The term iterative is used in the title of this section because the LLR is commonly used in turbo codes. LLR can also be used for non-iterative systems, as will be shown in our proposed system. Therefore, this section will be dedicated to present a very useful information measuring tool known as the *log-likelihood ratio* or simply LLR, and is commonly used in the computation of soft values when binary data are involved [33], [34] [35], [36].

Figure 4.1 below shows a *soft-input soft-output* (SISO) decoder. SISO decoders refer to decoders that accept a-priori information as their input and produce a-posteriori information as their output, usually in LLR format.

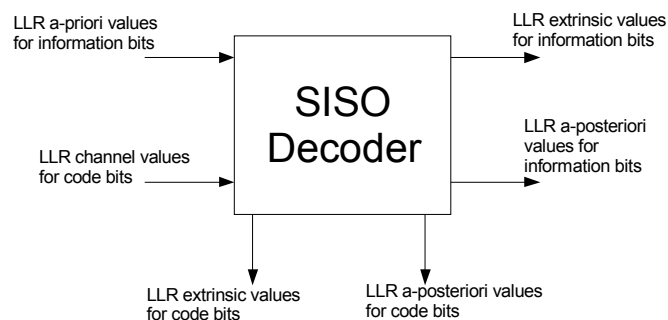


Figure 4.1 SISO Decoder

As can be seen in Figure 4.1, a SISO decoder can be separated into several inputs and outputs. The a-priori inputs are there only when turbo decoding/equalization is concerned. In a non-iterative system the a-priori values is set to ‘zero’ LLRs, because no feedback information is passed down between the decoders. Also, the extrinsic outputs (information bits or code bits) are not separated from the SISO decoder when no iteration between decoder/equalizer and equalizer/decoder is involved.

The notations used in this section will be different from the ones used in the proposed scheme, as the equations here are meant to present the general mathematical description of the LLR concept. If we let M be Galois field, GF(2) [22] with elements $\{+1, -1\}$, where +1 is the “null” element and -1 is the +1 element, then the LLR of the binary random variable M , $L_M(m_i)$, is defined as [33]:

$$L_M(m_i) = \ln \frac{P_M(m_i = +1)}{P_M(m_i = -1)} \quad (4-1)$$

Where $P_M(m_i)$ denotes the probability of the random value M . From now on, we will gradually not include the indices in the probabilities and the LLR equations, i.e. for example the subscripts M or Y or Y/M in $L_M(m_i)$ or $L_Y(Y_i)$ or $L_{Y/M}(\mathbf{y}/m_i)$, respectively will be gradually dropped (for example, equation (4.1) becomes $L(m_i) = \ln \frac{P(m_i=+1)}{P(m_i=-1)}$). The algorithm is in the natural logarithm, and the sign of the LLR is the hard decision, while its magnitude is its reliability measure (probabilistic). The natural logarithm, for the purpose of LLR, will be represented using the notations LLR, or L or \ln interchangeably. In the case where we have two binary random variables M and Y , and again using Bayes rule (as described in the previous chapter), the conditioned LLR will be:

$$\begin{aligned} \ln(m_i|\mathbf{y}) &= \ln \frac{P(m_i = +1|\mathbf{y})}{P(m_i = -1|\mathbf{y})} \\ \ln(m_i|\mathbf{y}) &= \ln \frac{P(m_i = +1)}{P(m_i = -1)} + \ln \frac{P(\mathbf{y}|m_i = +1)}{P(\mathbf{y}|m_i = -1)} \\ \therefore \ln(m_i|\mathbf{y}) &= L(m_i) + L(\mathbf{y}|m_i) \end{aligned} \quad (4-2)$$

Equation (4.2) is called the a-posteriori information of a SISO decoder with two components, namely the intrinsic information $L(m_i)$ and the extrinsic information $L(\mathbf{y}|m_i)$. It will soon be evident as the chapter unfolds, that this extrinsic information plays a very important role in turbo equalization setup.

4.1.1 From LLR to Probabilistic Values

The idea behind this section is to use the “incoming” information that is expressed in LLR format and transform it to regular probabilistic values.

A decoder/equalizer that uses MAP algorithm decode/equalize using probabilistic values. Therefore, if the incoming data is in LLR format, we can choose to convert it to probabilistic values (in the case where the decoder doesn’t use LLRs in its calculations) as follows:

Equation (4.1) becomes:

$$L_M(m_i) = \ln \frac{P_M(m_i = +1)}{P_M(m_i = -1)}$$

$$P(m_i) = e^{L_M(m_i)}, \text{ where } m_i = \mp 1$$

$$\text{and } P(m_i = -1) = 1 - P(m_i = +1)$$

With basic mathematics and from [33], we can derive the probabilistic values in terms of the LLRs:

$$e^{L_M(m_i)} = \frac{P(m_i = +1)}{1 - P(m_i = +1)} \quad (4-3)$$

$$P(m_i = +1) = \frac{e^{L_M(m_i)}}{1 + e^{L_M(m_i)}} = \frac{1}{1 + e^{-L_M(m_i)}} \quad (4-4)$$

$$P(m_i = -1) = \frac{e^{-L_M(m_i)}}{1 + e^{-L_M(m_i)}} = \frac{1}{1 + e^{L_M(m_i)}} \quad (4-5)$$

Similarly,

From (4.3)-(4.5), the expression for $P(m_i)$ (where $m_i = \mp 1$) becomes

$$P(m_i) = e^{L_M(m_i)} = \frac{e^{m_i L_M(m_i)}}{1 + e^{m_i L_M(m_i)}} \quad (4-6)$$

In order to separate the expression in (4-6) into two terms, one that is independent of m_i and the other isn't, in order for the term that is independent of m_i cancels out in the numerator and the denominator during the LLR calculation in the Gamma function, we simplify (4-6) as follows:

$$\begin{aligned} P(m_i) &= \frac{e^{m_i L_M(m_i)}}{1 + e^{m_i L_M(m_i)}} \\ &= \frac{e^{\frac{-L_M(m_i)}{2}}}{1 + e^{-L_M(m_i)}} * e^{\frac{m_i L_M(m_i)}{2}} = C * e^{\frac{m_i L_M(m_i)}{2}} \\ \text{where } C &= \frac{e^{\frac{-L_M(m_i)}{2}}}{1 + e^{-L_M(m_i)}} \end{aligned} \quad (4-7)$$

As can be seen the value "C" in (4.7) is the expression that is independent of m_i and the term $e^{\frac{-m_i L_M(m_i)}{2}}$ isn't. As mentioned earlier, the indices will eventually not be included in the equations.

Similarly, the derivation of the conditioned LLRs expressed in (4.2) can be expressed as:

$$\ln(m_i | \mathbf{y}) = L(m_i) + L(\mathbf{y} | m_i)$$

The probabilistic value for the first expression on the right hand side of the equation has already been derived in ((4-6) and (4-7)); similarly for the second expression:

$$L(\mathbf{y} | m_i) = \ln \frac{P(\mathbf{y} | m_i = +1)}{P(\mathbf{y} | m_i = -1)}$$

The probabilistic value is derived as follows:

$$\begin{aligned}
P(\mathbf{y}|m_i) &\xrightarrow{\text{yields}} \frac{e^{m_i L(\mathbf{y}|m_i)}}{1 + e^{m_i L(\mathbf{y}|m_i)}} \\
&= \frac{e^{\frac{-L(\mathbf{y}|m_i)}{2}}}{1 + e^{-L(\mathbf{y}|m_i)}} * e^{\frac{m_i L(\mathbf{y}|m_i)}{2}} = C_1 * e^{\frac{m_i L(\mathbf{y}|m_i)}{2}} \\
&\text{where } C_1 = \frac{e^{\frac{-L(\mathbf{y}|m_i)}{2}}}{1 + e^{-L(\mathbf{y}|m_i)}}
\end{aligned} \tag{4-8}$$

Table 4-1 below shows a numerical example of LLR. We can see that when the binary bits ‘0’ and ‘1’ are equal probable the LLR is ‘0’. What can also be derived from the table is the sign of the LLR. The sign tells us what bit was more probable, for example the LLR that corresponds to the $P_M(m_i = +1) = 0.2$ is -1.39 , which tells us that the bit was more likely a ‘0’ (recall: bit ‘0’ was mapped to ‘+1’ and bit ‘1’ was mapped to ‘-1’). That is, $L_M(m_i) = \ln \frac{P_M(m_i=+1)}{P_M(m_i=-1)} < 0$ happens only when $P_M(m_i = +1) < P_M(m_i = -1)$. Note that the LLRs shown here are for these probabilistic values only, i.e. if more precisions were used in the probabilistic values, the LLRs will have a wider value ranges, for example, if $P_M(m_i = +1) = 0.0001$, then the $L_M(m_i) = \ln \frac{P_M(m_i=+1)}{P_M(m_i=-1)} \cong -9.21$.

Table 4-1: Numerical Example of LLR

$P_M(m_i = +1)$	$P_M(m_i = -1)$	$L_M(m_i) = \ln \frac{P_M(m_i = +1)}{P_M(m_i = -1)}$
0.9	0.1	2.19722457733622
0.8	0.2	1.38629436111989
0.7	0.3	0.8472978603872
0.6	0.4	0.40546510810816
0.5	0.5	0
0.4	0.6	-0.40546510810816
0.3	0.7	-0.8472978603872
0.2	0.8	-1.38629436111989
0.1	0.9	-2.19722457733622

4.2 Proposed System Enhancement Using Frequency-Domain MAP Equalization

The previous section represented the tool for decoding binary codes, LLR. In Chapter 3, the MAP Algorithm used in a BCJR/MAP decoding/equalization was derived. This section will make use of the learned techniques to derive our proposed receiver design for DSRC systems.

Figure 4.2 below, shows the block diagram of the proposed receiver design that utilizes frequency-domain (F-D) MAP equalization.

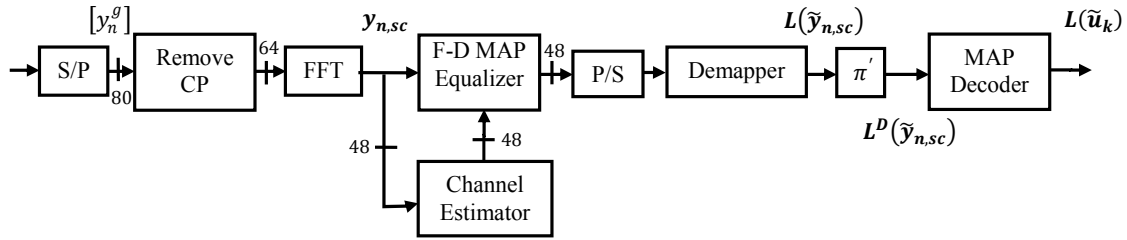


Figure 4.2 Proposed Receiver Design

Before talking about the equalizer's algorithm, it is rather useful to bring you back to chapter 2 where we talked about the channel model. Recall, the WAVE channel model takes in fading simulator as its channel coefficients. A lot of work has been carried out in the past when it comes to combating inter symbol interference (ISI). OFDM system is one popular technique used. MAP algorithm has also been used in equalization to give the a posteriori probabilities (APPs) about the encoded bits, which in turn is fed to a MAP-based decoder as a priori information to decode these bits into the message. Some of the authors who discussed MAP-based equalization in schemes like turbo equalization include [33], [26] to combat ISI. The equalizer works similar to the decoder, but with one exception and unlike the decoder, the channel coefficients have to be taken into account as they are needed in its branch metric calculations. In [33], the channel is modeled in time domain to combat ISI, but in our proposed scheme we want to combat inter carrier interference (ICI) and there is a good reason for that. OFDM is popular for its ability to mitigate ISI through the use of sufficient guard interval. In the case of a time-invariant channel, a one-tap equalizer per subcarrier suffices to equalize that channel [37]. However, in time-selective fading channels, when doppler shift is concerned, it is very useful to look at the channel model in frequency domain (F-D). In other words, it is important to remember that the induced doppler shift destroys the orthogonality between the subcarriers that make up the OFDM symbols of the DSRC system, which in turn causes ICI [14]. Next section will discuss the simplified channel model in frequency-domain (and its state diagrams) for time-varying fading multipath channel. After that, F-D MAP equalizer's algorithm will be explained in details for both time-domain and frequency-domain cases. MAP SISO decoder with LLR inputs will also be discussed. Then, we will use the information in our proposed system design to enhance the performance of the conventional DSRC systems. Finally for this chapter, a brief explanation of turbo codes will be presented [17], then that principle is applied in our proposed design as turbo equalization [38]. From now on, we will refer to our MAP equalization and MAP decoding algorithms as MAPEq and MAPdec respectively.

4.2.1 Simplified Channel Model in Frequency-Domain

Previously, it was mentioned that in the IEEE 802.11a standard the channel was assumed time-invariant, and therefore creating orthogonal subcarriers through the use of OFDM systems with sufficient guard interval is usually a sufficient technique needed to compensate the effect that the channel had upon the

transmitted signal without the need of very complicated equalization schemes. From Chapter 2, it can be seen that the compensation was as simple as just performing a bitwise component division operation between the channel response and the received symbols. However, in the case of a time-variant channel in order to mitigate the effects of the ICI caused by the existence of Doppler shift, the well known maximum a posteriori (MAP) algorithm technique [24] can be used to equalize the effects of the time-varying channel. It is necessary to know the channel coefficients of the time-discrete channel h_l , where l is the l -th path of the channel impulse response (CIR), to combat ISI. For instance, the systems in [34] and [26] use the MAP algorithm technique to do just that. In a DSRC system model however, it will make sense to combat the ICI in frequency domain. For performance enhancement purposes, we propose in this chapter a MAP-based equalizer in the frequency domain (F-D). Now that we are looking at the system in F-D, it is obvious that the MAP equalizer needs an equivalent channel model in F-D as well for its trellis description.

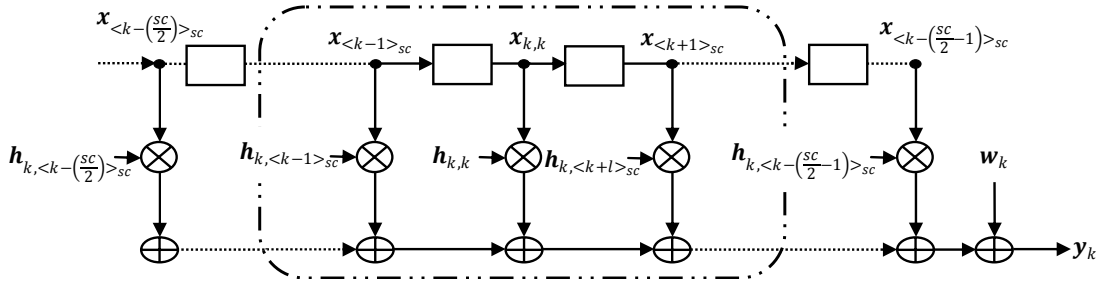


Figure 4.3 Simplified Frequency Domain Channel Model

Figure 4.3 shows the equivalent channel model in frequency domain [39]. Looking back at (2-8), and in terms of matrices, the FFT output of that relation is,

$$\mathbf{F}\mathbf{Y} = \mathbf{F}\{\mathbf{H} \cdot (\mathbf{F})^H \cdot \mathbf{X}\} + \mathbf{F}\mathbf{W} = \mathbf{H}\mathbf{X} + \mathbf{W} = \mathbf{Y} \quad (4-9)$$

The time-invariant-time-domain CIR (H) is a circulant matrix, while in frequency-domain the channel matrix (\mathbf{H}) is a diagonal matrix. However, in time-variant channels neither (H) is circulant nor is (\mathbf{H}) diagonal. The off-diagonal elements of (\mathbf{H}) are the representation of the ICI coefficients. Fortunately, (\mathbf{H}) has the highest power concentration of its most significant elements around the main diagonal [40]. A further discussion of the diagonal of the (\mathbf{H}) will follow shortly and can also be seen in Figure 4.4.

(\mathbf{H}) can be represented as follows [39],

$$\mathbf{H}_{sc_i, sc_j} = \frac{1}{N} \sum_{n=0}^{N-1} \left(\sum_{l=0}^{L-1} h_{n,l} \cdot e^{-j \cdot 2\pi \cdot l \cdot sc_j / N} \right) \cdot e^{-j \cdot 2\pi \cdot (sc_i - sc_j) \cdot n / N} \quad (4-10)$$

Where sc_i and sc_j are the addressing of the 2D matrix \mathbf{H}_{sc_i, sc_j} , and each take the size of the number of subcarriers $\{N\}$ in an OFDM symbol, i.e. sc_i and sc_j are the i -th row and j -th column of possible sc subcarriers ($0 < i < sc$, and $0 < j < sc$).

$$\begin{bmatrix} Y \\ y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_{sc} \end{bmatrix} = \begin{bmatrix} h_{0,0} & h_{0,1} & \dots & \dots & \dots & \dots & h_{0,sc_2-1} \\ h_{1,0} & h_{1,1} & \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & h_{2,1} & h_{2,2} & \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & h_{sc_1-3,sc_2-3} & h_{sc_1-3,sc_2-2} & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & h_{sc_1-2,sc_2-2} & h_{sc_1-2,sc_2-1} \\ h_{sc_1-1,0} & \dots & \dots & \dots & \dots & h_{sc_1-1,sc_2-2} & h_{sc_1-1,sc_2-1} \end{bmatrix} \begin{bmatrix} X \\ x_1 \\ x_2 \\ x_3 \\ \vdots \\ x_{sc} \end{bmatrix} + \begin{bmatrix} W \\ w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_{sc} \end{bmatrix} \quad (4-11)$$

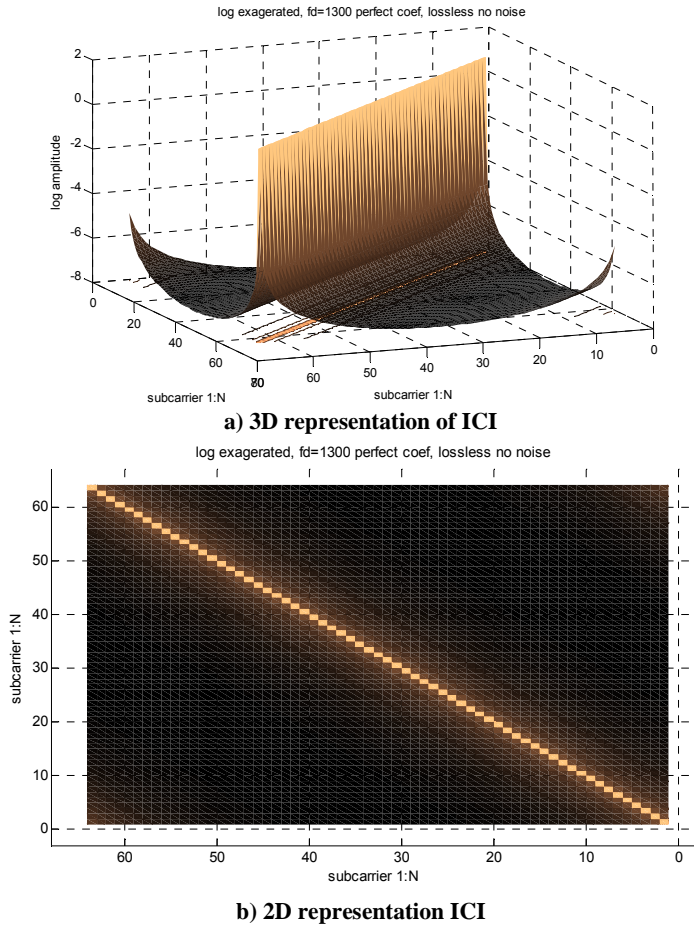


Figure 4.4 Subcarrier to Subcarrier ICI Contributions

Figure 4.4, represents the power amplitude of the subcarriers with respect to one another and the subcarrier to itself or simply the (H) matrix of one OFDM symbol under Rayleigh fading with $fd = 1300\text{Hz}$ (Doppler frequency). The amplitude in the 3-D plot was translated into log format to emphasize the amplitude difference between the subcarriers. Referring to that figure, it is useful to note that the most concentration of ICI power exist around that of the subcarrier in question. In other words, the neighbouring subcarriers have the highest level of interference between a subcarrier. In Figure 4.4, the brighter the colors in the 2-D plot, the higher the ICI concentration between subcarriers. In the 3-D plot, the higher concentration of ICI

between subcarriers exists at the peak of the image. Equation (4-11), also demonstrates how each subcarrier is affecting the transmitted data to produce the output, that is, the received OFDM symbol data in frequency domain. Due to the fact mentioned above, i.e. the ICI power concentration being around the “current” subcarrier in question, we can now further simplify the model by dropping the subcarriers that are relatively far from the “current” subcarrier, that is, we can express (4.9) or (4.11) in k^{th} symbol data \mathbf{y}_k as follows,

$$\mathbf{y}_k = \mathbf{v}_k + \mathbf{w}_k$$

$$\text{where } \mathbf{v}_k = \sum_{l=-f}^f \mathbf{h}_{k, \langle k-l \rangle_{sc}} \cdot \mathbf{x}_{\langle k-l \rangle_{sc}}$$
(4-12)

where $(2f + 1)$ is the total number of adjacent subcarriers used. The dashed rectangular lines in Figure 4.3 Simplified Frequency Domain Channel Model demonstrates a 3 subcarrier simplified model is chosen. Since the channel trellis in frequency-domain now depends only on the size of (f) instead of the multipath delay memory elements l , as in the case of the time-domain channel model, the complexity of the MAP equalizer is now q^{2f} number of states instead of q^l (where q is the constellation size of the modulation scheme used). In our study here, we will limit f to 1, which means that the complexity, when the number of states is concerned, will only depend on the modulation scheme used.

4.2.2 State Diagrams of the F-D Channel Model

This section shows the state diagram used to create the trellis of the simplified F-D channel model used in our BPSK modulated-message system. By simplified, and as described in the previous section, it means that $(2f + 1)$ with $f = 1$ is used to define the total number of adjacent subcarriers used.

Table 4-2: State Diagram for Simplified F-D Channel Model with BPSK Modulated Input

Modulated Input (x_k)	Present State (PS)	PS	Next State (NS)	Output (y_k)
+1 -1	+1+1 +1+1	0	+1+1 -1+1	$+\mathbf{h}_{k, \langle k-1 \rangle_{sc}} + \mathbf{h}_{k,k} + \mathbf{h}_{k, \langle k+1 \rangle_{sc}}$ $-\mathbf{h}_{k, \langle k-1 \rangle_{sc}} + \mathbf{h}_{k,k} + \mathbf{h}_{k, \langle k+1 \rangle_{sc}}$
+1 -1	+1-1 +1-1	1	+1+1 -1+1	$+\mathbf{h}_{k, \langle k-1 \rangle_{sc}} + \mathbf{h}_{k,k} - \mathbf{h}_{k, \langle k+1 \rangle_{sc}}$ $-\mathbf{h}_{k, \langle k-1 \rangle_{sc}} + \mathbf{h}_{k,k} - \mathbf{h}_{k, \langle k+1 \rangle_{sc}}$
+1 -1	-1+1 -1+1	2	+1+1 -1+1	$+\mathbf{h}_{k, \langle k-1 \rangle_{sc}} - \mathbf{h}_{k,k} + \mathbf{h}_{k, \langle k+1 \rangle_{sc}}$ $-\mathbf{h}_{k, \langle k-1 \rangle_{sc}} - \mathbf{h}_{k,k} + \mathbf{h}_{k, \langle k+1 \rangle_{sc}}$
+1 -1	-1-1 -1-1	3	+1+1 -1+1	$+\mathbf{h}_{k, \langle k-1 \rangle_{sc}} - \mathbf{h}_{k,k} - \mathbf{h}_{k, \langle k+1 \rangle_{sc}}$ $-\mathbf{h}_{k, \langle k-1 \rangle_{sc}} - \mathbf{h}_{k,k} - \mathbf{h}_{k, \langle k+1 \rangle_{sc}}$

The state diagram basically shows all the possible entries to the F-D channel trellis, in order to create the present states, next states and output to the channel model of Figure 4.3.

Table 4-2 shows the state diagram for our simplified frequency domain channel model with BPSK modulated input data to the channel. Using the same principles, it can be extended to other modulations schemes, QPSK and M-ary QAM modulations.

4.2.3 Frequency Domain MAP Equalization Model

Fig.4 demonstrates the block diagram for our proposed MAP equalization in the DSRC receiver design. This subsection will be dedicated to derive the MAP equalization algorithm that will be used in our proposed receiver design by making use of the reduced (lower complexity) non-diagonal matrix (\mathbf{H}) derived above for the equivalent frequency domain channel model with ICI elements. The idea behind both the MAP-based equalizer and the MAP-based decoder [26], are the same, and that is, to maximize the a posteriori probabilities (APP) $P(\mathbf{y}_k = \mathbf{y}|\mathbf{y}_n)$ given the received OFDM sequence $[\mathbf{y}_n]$

$$\begin{aligned} \tilde{\mathbf{y}}_k &= \arg \left\{ \max_{\mathbf{y} \in (-1,1)} P(\mathbf{y}_k = \mathbf{y}|\mathbf{y}_n) \right\} \\ &\xrightarrow{\text{yields}} p(s, s', [\mathbf{y}_n]) = p([\mathbf{y}_n]) \cdot p(s, s' | [\mathbf{y}_n]) \end{aligned} \quad (4-13)$$

where s and s' are the present state and the next state in all possible states \mathbf{S} of the channel/encoder trellis, respectively. As described in Chapter 2, a trellis description of a convolution encoder is very useful in decoding (e.g computing the APPs, in the case of MAP algorithm). Let the set of all valid branches in a trellis of the simplified frequency domain channel model of Figure 4.3 with $f = 1$ be \mathcal{S} . It can be given as,

$$\mathcal{S}(s, s') = \{(0,0), (0,2), (1,0), (1,2), (2,1), (2,3), (3,1), (3,3)\} \quad (4-14)$$

The sequence $[\mathbf{y}_n]$ can be separated into smaller segments in that n^{th} OFDM symbol,

$[\mathbf{y}_n] = \{[\mathbf{y}_1, \dots, \mathbf{y}_k], [\mathbf{y}_k], [\mathbf{y}_{k+1}, \dots, \mathbf{y}_{N_s}]\}$, and applying the joint probabilities, we obtain

$$p(s, s', [\mathbf{y}_n]) = p(s, [\mathbf{y}_1, \dots, \mathbf{y}_k]) \cdot p(s' | s) \cdot p(\mathbf{y}_k | s, s') \cdot p([\mathbf{y}_{k+1}, \dots, \mathbf{y}_{N_s}] | s') \quad (4-15)$$

The four terms on the right hand side of (4-15) include the forward and backward recursions

$$p(s, [\mathbf{y}_1, \dots, \mathbf{y}_k]) = \alpha_k(s') = \sum_{s \in \mathcal{S}} \alpha_{k-1}(s) \cdot \gamma_{k-1}(s, s') \quad (4-16)$$

$$p([\mathbf{y}_{k+1}, \dots, \mathbf{y}_{N_s}] | s') = \beta_k(s') = \sum_{s \in \mathcal{S}} \beta_{k+1}(s) \cdot \gamma_k(s', s) \quad (4-17)$$

where \mathcal{S} is all possible states in the trellis, and the initial states or values for both $\alpha_0(s)$ and $\beta_{N_s}(s)$ is equal to 1, which means that we assume all states are equal probable.

The second term or the gamma function $\gamma_k(s, s')$ is the only term from among the other two terms that will be different for both the equalizer and the decoder. For the equalizer, we have

$$\begin{aligned}
 & p(s'|s) \cdot p(\mathbf{y}_k | s, s') = \gamma_k(s, s') = \\
 & \begin{cases} p(\mathbf{x}_k = \mathbf{x}_{s, s'}) \cdot p(\mathbf{y}_k | \mathbf{v}_k = \mathbf{v}_{s, s'}), & (s, s') \in \mathcal{S} \\ 0, & \text{Otherwise} \end{cases}
 \end{aligned} \tag{4-18}$$

$\mathbf{x}_{s, s'}$ and $\mathbf{v}_{s, s'}$ represent the input and output symbols that correspond to s and s' states in the channel trellis in the frequency domain, respectively. The conditional probability in the $\gamma_k(s, s')$ function $p(\mathbf{y}_k | \mathbf{v}_k = \mathbf{v}_{s, s'})$ is:

$$\begin{aligned}
 & p(\mathbf{y}_k | \mathbf{v}_k = \mathbf{v}_{s, s'}) = e^{\left\{-\left(\frac{1}{2\sigma^2}\right) \cdot (\mathbf{y}_k - \mathbf{v}_k)^2\right\}} \\
 & \text{again, } \mathbf{v}_k = \sum_{l=-f}^f \mathbf{h}_{k, \langle k-l \rangle_{sc}} \cdot \mathbf{x}_{\langle k-l \rangle_{sc}}
 \end{aligned} \tag{4-19}$$

σ^2 is the variance of the AWGN in inphase and quadrature components. While $p(\mathbf{x}_k = \mathbf{x}_{s, s'})$ is the a priori information to the equalizer and is set to 0.5 when no iteration is taking place, that is, no extrinsic information is provided to the equalizer from the decoder. It is 0.5 because the values \mathbf{x}_k could take on the values of 0 or 1 at equal probability due to the assumption that they are independent and uniformly distributed. If the log-likelihood ratios (LLRs) are used, and the a priori given to the equalizer from the decoder is in that format, then the calculation of the $\gamma_k(s, s')$ function will be as follows:

$$\begin{aligned}
 & \gamma_k(s, s') = \\
 & e^{\left\{\frac{1}{2} \cdot LLR_e^i(\mathbf{x}_{k, v}) \cdot \mathbf{x}_{k, v}\right\}} \cdot e^{\left\{-\left(\frac{1}{2\sigma^2}\right) \cdot (\mathbf{y}_k - \mathbf{v}_k)^2\right\}} \\
 & \text{again, } \mathbf{v}_k = \sum_{l=-f}^f \mathbf{h}_{k, \langle k-l \rangle_{sc}} \cdot \mathbf{x}_{\langle k-l \rangle_{sc}}
 \end{aligned} \tag{4-20}$$

$LLR_e^i(\mathbf{x}_m) = 0$ at 0 iteration, and then at each iteration that follows it will take on the extrinsic values that come from the decoder.

From (4-13) and (4-15), and by taking all the possible branches in the trellis that take the input $\mathbf{x}_k = \mathbf{x}$, the output APP of the equalizer can now be defined as:

$$p(\mathbf{x}_k = \mathbf{x} | [\mathbf{y}_n]) = \sum_{\substack{\mathbf{x}_{s, s'} = \mathbf{x} \\ s, s' \in \mathcal{S}}} P(s, s' | [\mathbf{y}_n]) = \sum_{\substack{\mathbf{x}_{s, s'} = \mathbf{x} \\ s, s' \in \mathcal{S}}} \alpha_k(s') \cdot \beta_{k+1}(s) \cdot \gamma_k(s', s) \tag{4-21}$$

We included the demapping operation and the log-likelihood ratios (LLRs) as follows:

$$\begin{aligned}
 & LLR(\tilde{\mathbf{y}}_{k, v}) = \log \frac{p(\mathbf{y}_{k, v} = +\mathbf{1} | [\mathbf{y}_n])}{p(\mathbf{y}_{k, v} = -\mathbf{1} | [\mathbf{y}_n])} \\
 & = \sum_{\substack{\mathbf{y}_{k, v} = +\mathbf{1} \\ s, s' \in \mathcal{S}}} \alpha_k(s') \cdot \beta_{k+1}(s) \cdot \gamma_k(s', s) \Bigg/ \sum_{\substack{\mathbf{y}_{k, v} = -\mathbf{1} \\ s, s' \in \mathcal{S}}} \alpha_k(s') \cdot \beta_{k+1}(s) \cdot \gamma_k(s', s)
 \end{aligned} \tag{4-22}$$

$\mathbf{y}_{k,v}$ is the transmitted symbol \mathbf{y}_k at bit v (2 bits for QPSK and 1 bit for BPSK) in the frequency domain, and again $[\mathbf{y}_n]$ is the received OFDM symbol. The LLR is usually used to determine whether the $p(\mathbf{x}_k = +\mathbf{1})$ is larger, smaller or equal than that of its opponent $p(\mathbf{x}_k = -\mathbf{1})$,

$$\begin{aligned}
 & \text{if } LLR(\mathbf{y}_{k,v}) > 0 \\
 & \text{then } p(\mathbf{x}_k = +\mathbf{1} | [\mathbf{y}_n]) > p(\mathbf{x}_k = -\mathbf{1} | [\mathbf{y}_n]) \\
 & \text{elseif } LLR(\mathbf{y}_{k,v}) < 0 \\
 & \text{then } p(\mathbf{x}_k = +\mathbf{1} | [\mathbf{y}_n]) < p(\mathbf{x}_k = -\mathbf{1} | [\mathbf{y}_n]) \\
 & \text{else } p(\mathbf{x}_k = +\mathbf{1} | [\mathbf{y}_n]) = p(\mathbf{x}_k = -\mathbf{1} | [\mathbf{y}_n])
 \end{aligned} \tag{4-23}$$

These LLR values are then deinterleaved and then given to the MAP decoder as LLR inputs or a priori information.

We can also use the matrix operations as a description, for example the simplified channel trellis with $f = 1$, and by letting $p(\mathbf{y}_k | \mathbf{v}_k = \mathbf{v}_{s,s'}) = B$ in (4-18), and combining it to all the valid branches $\mathcal{S}(s, s')$ from (4-14), we obtain the term $p(\mathbf{y}_k | s, s')$ for $\gamma_k(s', s)$ in the following matrix format,

$$\{p(\mathbf{y}_k | s, s')\}_{s,s' \in \mathcal{S}} = \begin{bmatrix} B & 0 & B & 0 \\ B & 0 & B & 0 \\ 0 & B & 0 & B \\ 0 & B & 0 & B \end{bmatrix} \tag{4-24}$$

If we separate the transitions that contain the input 1 from the ones with input 0, we obtain:

$$\{p(\mathbf{y}_k | s, s')\}_{s,s' \in \partial(1)} = \begin{bmatrix} 0 & 0 & B & 0 \\ 0 & 0 & B & 0 \\ 0 & 0 & 0 & B \\ 0 & 0 & 0 & B \end{bmatrix}, \{p(\mathbf{y}_k | s, s')\}_{s,s' \in \partial(-1)} = \begin{bmatrix} B & 0 & 0 & 0 \\ B & 0 & 0 & 0 \\ 0 & B & 0 & 0 \\ 0 & B & 0 & 0 \end{bmatrix} \tag{4-25}$$

where $\partial(x_{s,s'})$ is the $x_{s,s'} = 1$ or -1 transition states.

Therefore $\gamma_k(s', s)$ is a matrix of $S \times S$ dimensions, represented by Γ_{k-1} . Let \mathbf{A}_k represent the matrix version of the $\alpha_k(s')$ function in (4-16), we note that \mathbf{A}_k is $1 \times S$ matrix. There we get,

$$\mathbf{A}_k = \overline{\mathbf{A}_{k-1}} \Gamma_{k-1} \tag{4-26}$$

$\overline{\mathbf{A}_{k-1}}$ is the transpose of matrix \mathbf{A}_{k-1} . Similarly, for the $\beta_k(s')$ function in (4-17), we get

$$\beta_k = \overline{\beta_{k+1}} \Gamma_k \tag{4-27}$$

4.2.4 Decoder Model

The MAP decoder takes these LLR values of each bit and then decodes the information. The decoding process is similar to that of the equalizer by providing its own a posteriori information at its output. The decoder may also include the extrinsic information about the coded bits as a priori feedback to the equalizer. The output of the decoder will make a decision based on the information bits \mathbf{u}_k using Equation (4-23) and the sign of the following equation similar to Equation (4-22),

$$LLR(\tilde{\mathbf{u}}_k) = \log \frac{p(\mathbf{u}_k = +1 | L^D(\tilde{\mathbf{y}}_{k,v}))}{p(\mathbf{u}_k = -1 | L^D(\tilde{\mathbf{y}}_{k,v}))} = \sum_{\substack{\mathbf{u}_k = +1 \\ s \in \mathcal{S}}} \lambda_k(s) / \sum_{\substack{\mathbf{u}_k = -1 \\ s \in \mathcal{S}}} \lambda_k(s)$$

$$\text{where } \lambda_k(s) = \alpha_k(s) \cdot \beta_k(s)$$

(4-28)

In other words, the forward and backward recursions are multiplied componentwise together to produce the maximum probability of the input being a 1 or a 0 at a specific instant of time. The summation in (4-28) sums the elements in the matrix that resulted from the component wise multiplication, for all transition 1 and transition 0 for the numerator and denominator, respectively to obtain the final LLR result. The LLR can then be used to make a ‘HARD’ decision on the equalized or decoded symbols or bits (decoder in this case) through its sign,

$$\hat{\mathbf{u}}_k = \begin{cases} 1, & LLR(\tilde{\mathbf{u}}_k | [\mathbf{y}_n]) \geq 0 \\ 0, & LLR(\tilde{\mathbf{u}}_k | [\mathbf{y}_n]) < 0 \end{cases} \quad (4-29)$$

The forward/backward calculations are the same for both the decoder and the equalizer with the exceptions of initial states conditions for the decoder is different than that of the equalizer, i.e.

$$\begin{cases} \alpha_0(0) \text{ and } \beta_{N_s}(0) = 1, \\ \alpha_0(s) \text{ and } \beta_{N_s}(s) = 0, & \forall s \neq 0 \end{cases} \quad (4-30)$$

Also, the $\gamma_k(s, s')$ function is different due to the different kinds of inputs to the soft-input soft-output (SISO) decoder.

$$\gamma_k(s, s') = e^{\sum_{i=1}^{\frac{1}{2}R} (\frac{1}{2}LLR(\tilde{\mathbf{y}}_{k,v}) \cdot \mathbf{y}_{k,v})} \cdot e^{\frac{1}{2}LLR(\mathbf{u}_k) \cdot \mathbf{u}_k} \quad (4-31)$$

R is the code rate ($R = 1/2$), $LLR(\tilde{\mathbf{y}}_{k,v})$ is the LLR input to the decoder that came from the equalizer. $LLR(\tilde{\mathbf{y}}_{k,v})$ contain the channel information. $LLR(\mathbf{u}_k)$ is the a priori information of the information bits \mathbf{u}_k , which is set to zero due to the fact that the random variables or information bits \mathbf{u}_k are assumed independent and uniformly distributed (hence equal probable in either taking on the values 1 or 0). The term $\left\{ \frac{1}{2} \cdot LLR(\tilde{\mathbf{y}}_{k,v}) \cdot \mathbf{y}_{k,v} \right\}$ in (4-31) is the conversion of the LLR value of the a priori information $LLR(\tilde{\mathbf{y}}_{k,v})$ into a probabilistic value. There’s a constant value that wasn’t included in that term, and that is because it will be cancelled at the decoder’s LLR output calculation in (4-28). The LLR conversion happened through the relation:

$$P(\mathbf{y}_{k,v} | [\mathbf{y}_n]) = \frac{e^{\{[\mathbf{y}_n](LLR(\mathbf{y}_{k,v} | [\mathbf{y}_n]))\}}}{1 + e^{\{[\mathbf{y}_n](LLR(\mathbf{y}_{k,v} | [\mathbf{y}_n]))\}}} = C * e^{\left\{ \frac{[\mathbf{y}_n](LLR(\mathbf{y}_{k,v} | [\mathbf{y}_n]))}{2} \right\}}$$

$$C = \frac{e^{\frac{-LLR(\mathbf{y}_{k,v} | [\mathbf{y}_n])}{2}}}{1 + e^{-LLR(\mathbf{y}_{k,v} | [\mathbf{y}_n])}} \quad (4-32)$$

C is the constant that doesn’t depend on $[\mathbf{y}_n]$ and cancels out in the output LLR calculation of a SISO decoder.

4.3 Simulation Results

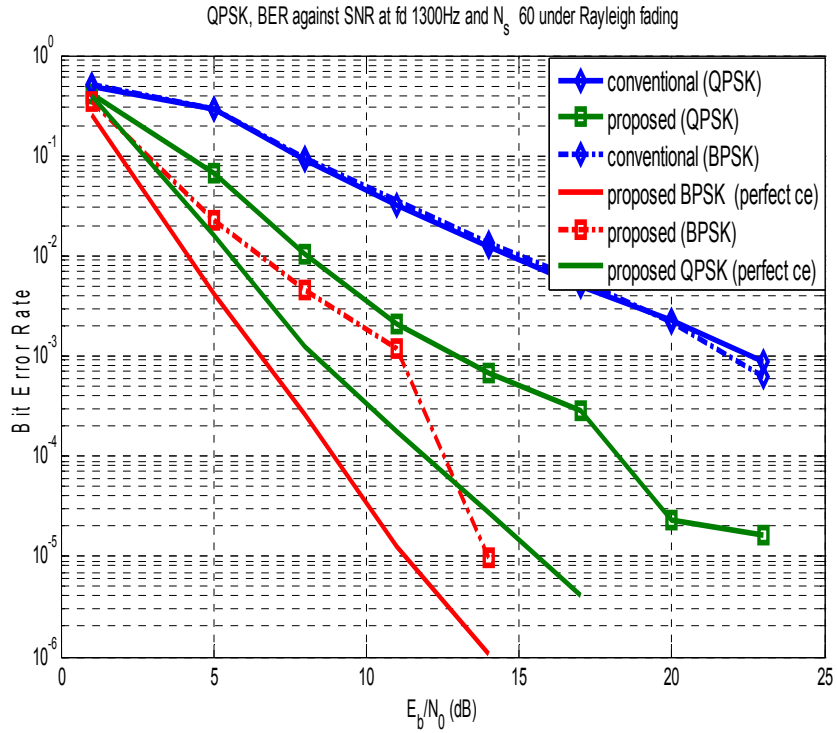
The performances of the proposed system against the conventional system for DSRC system are depicted from Figure 4.5 to Figure 4.10.

In each simulation, we simulated the BER and PER vs. the E_b/N_0 performances under the worst case scenario, i.e. No-LOS Rayleigh faded channel, was used. The relative velocity of around 238 Km/hr ($f_d = 1300$) was chosen, which was around the legal relative speed limits of vehicles. AWGN was also used as the noisy channel. All the simulations were carried out for the 60 ODFM symbols for each packet for 1200 packets. The modulation schemes used are BPSK and QPSK. The system was simulated under varying SNR or E_b/N_0 . The summary of the range of values of the variables used in the simulations can be found under Table 4-3.

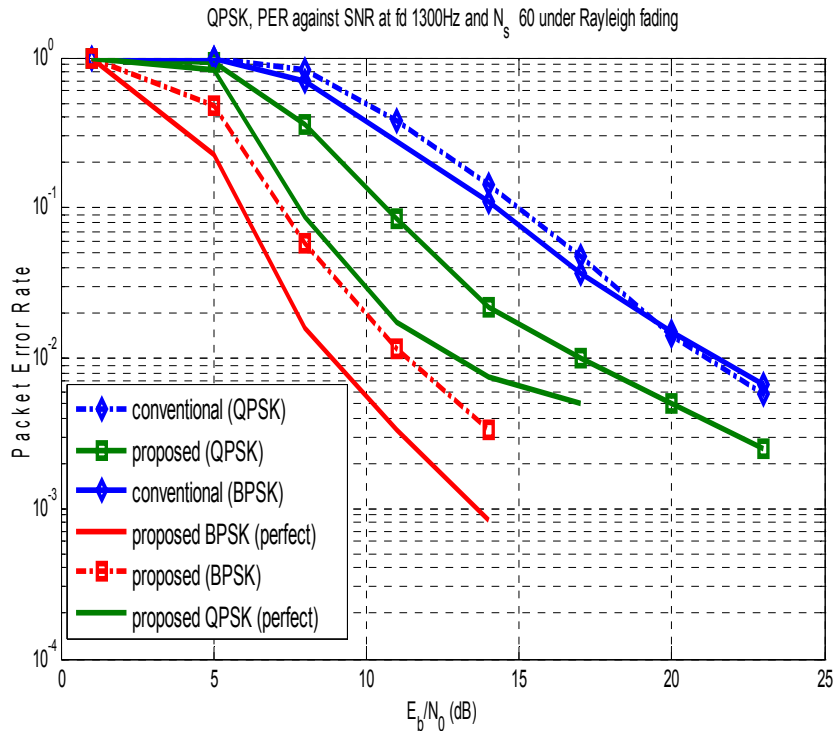
Table 4-3: Simulation Values

Name	Abbreviation	Unit	Values
Signal-to-Noise Ratio	SNR	dB	[1, 5, 8, 11, 14, 17, 20, 23]
Maximum Doppler Frequency (velocity)	f_m (v)	Hz (km/h)	[1300] [238]
Symbol Duration	T_s	μs	[8]
Packet Length	N_s	Symbols/packet	[60]
Rice Factor	K		[0]
Modulation	m		[BPSK, PSK]
Decoding			MAP Algorithm

What follows are the simulation results.

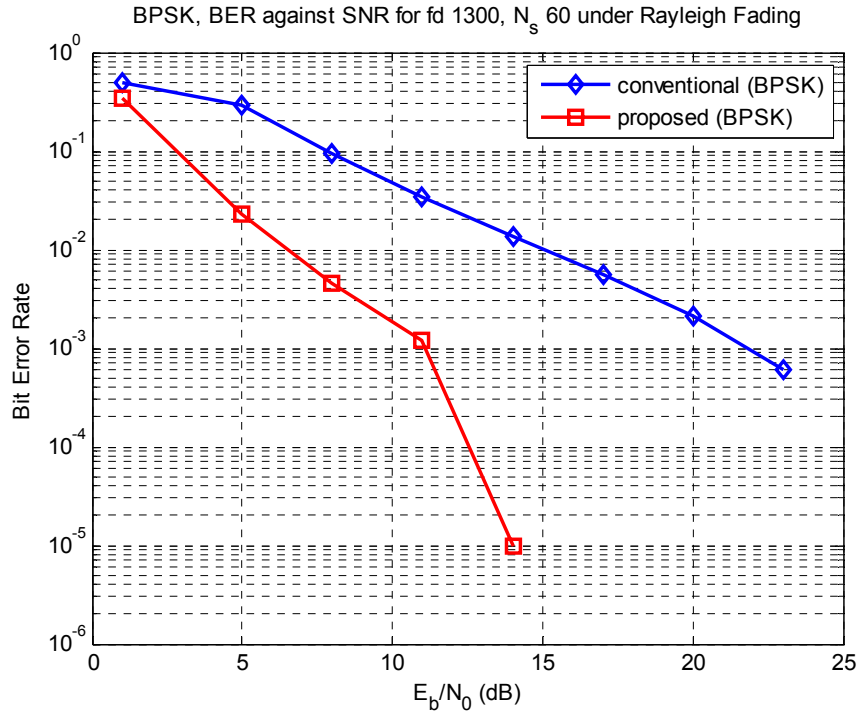
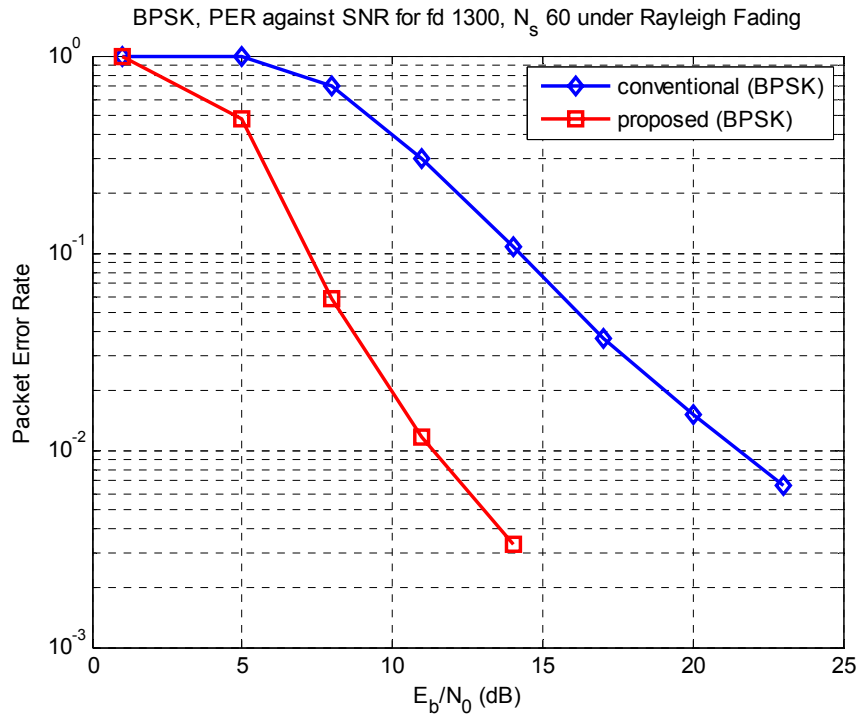


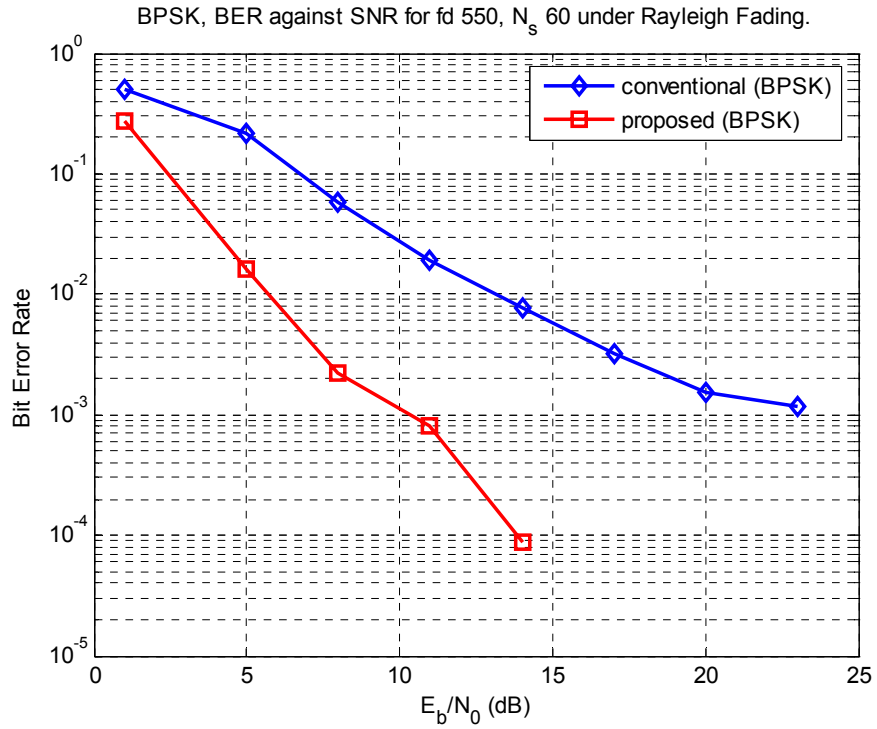
a) BER vs. SNR



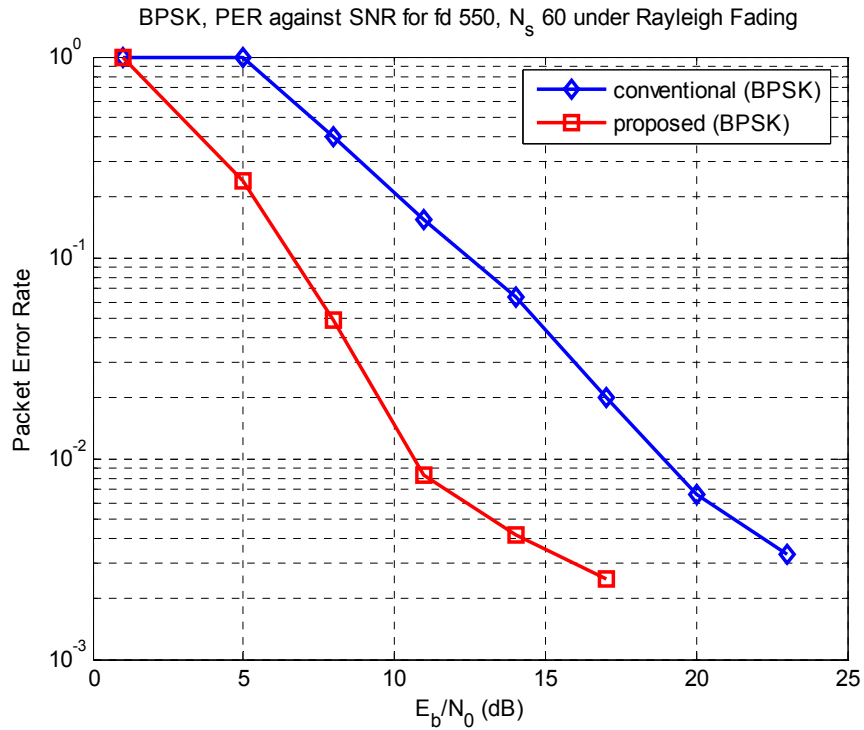
b) PER vs. SNR

Figure 4.5 Proposed vs. Conventional BPSK, QPSK modulation, with and without perfectly known coefficients at fd=1300Hz

**a) BER vs. SNR****b) PER vs. SNR**Figure 4.6 Proposed vs. Conventional BPSK modulation at $f_d=1300\text{Hz}$



a) BER vs. SNR



b) PER vs. SNR

Figure 4.7 Proposed vs. Conventional BPSK modulation at $f_d=550$ Hz

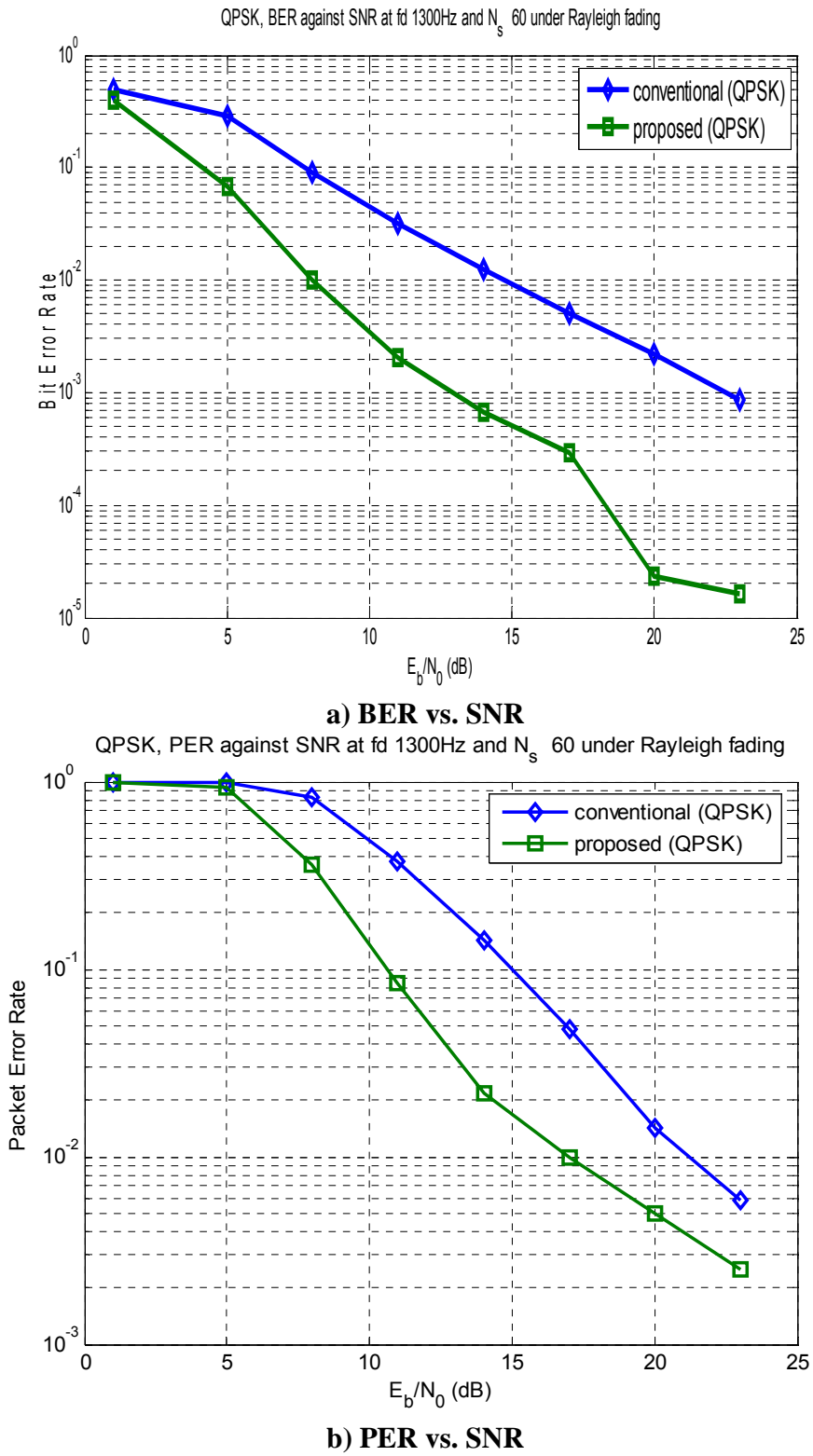
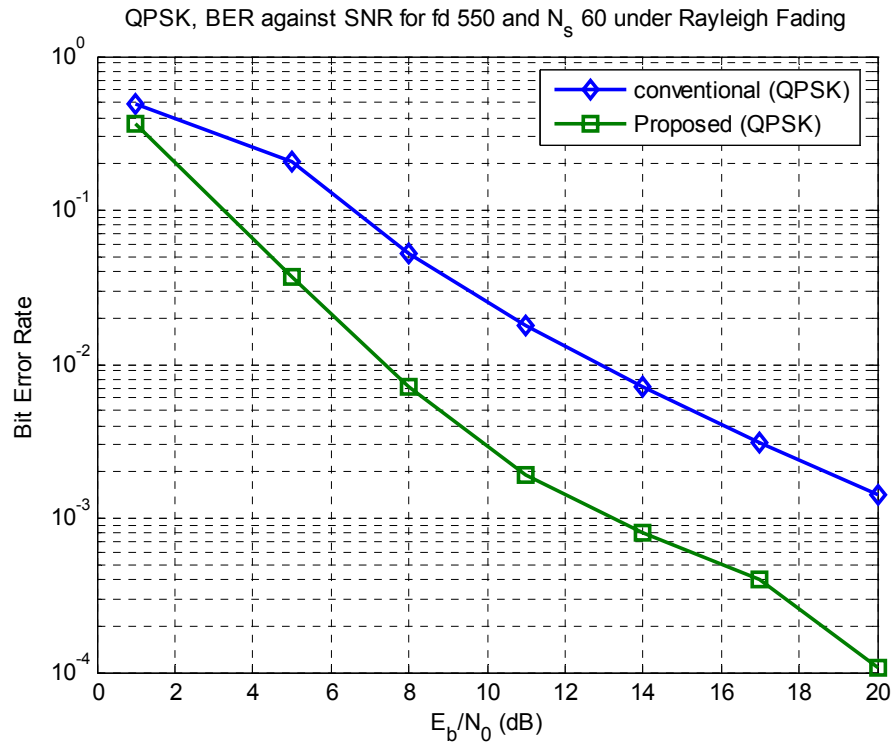
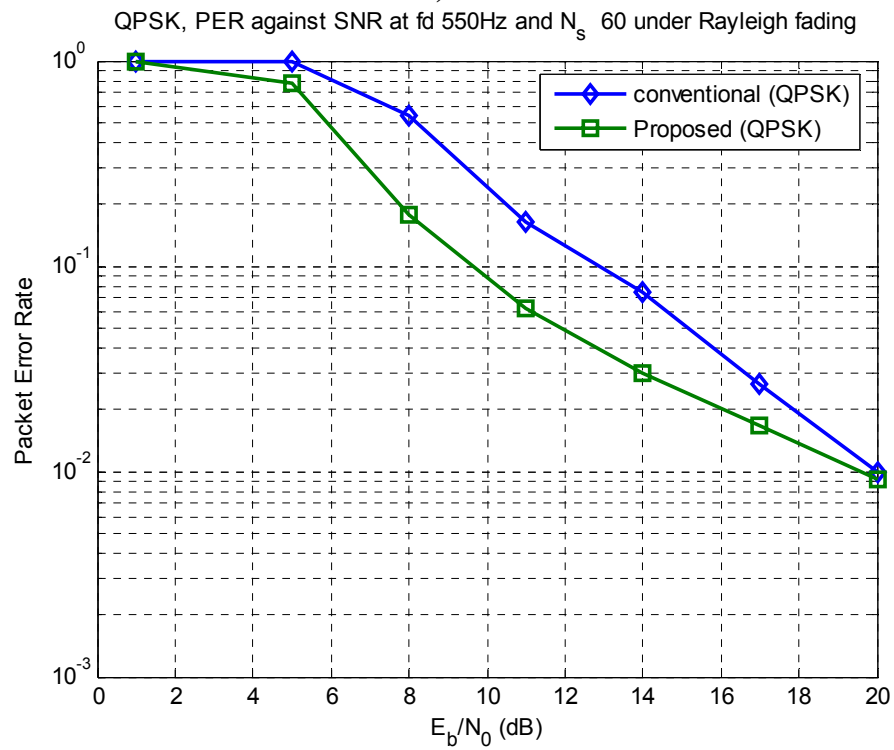


Figure 4.8 Proposed vs. Conventional QPSK modulation at fd=1300Hz



a) BER



b) PER

Figure 4.9 Proposed vs. Conventional QPSK modulation at fd=550Hz

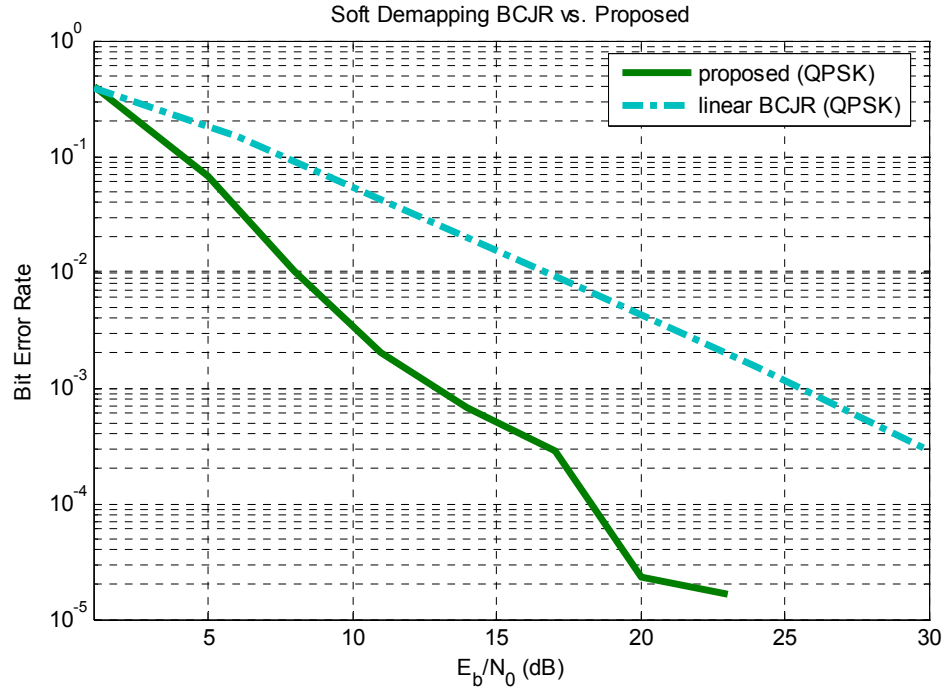


Figure 4.10 Soft Demapping BCJR vs. Proposed

The simulation results depict a considerably high improvement in performance using the proposed frequency-domain equalizer. From Figure 4.5, it can be seen that the proposed system with BPSK modulation (with doppler frequency of 1300Hz), and with perfectly known coefficients, had a performance gain of around 15 dB and 12 dB for the BER and PER performances, respectively. For QPSK with doppler frequency of 1300Hz (about the maximum relative highway street speeds), tested under conventional channel estimation, had a performance gain of about 3 dB as seen in Figure 4.8 under the PER plot. From Figure 4.7, BPSK with doppler frequency of 550Hz (about the maximum relative city street speeds) had a performance gain of around 6dB for PER performance. BPSK without perfect known coefficients had a performance gain of around 12 dB and 9 dB for BER and PER respectively, as seen in Figure 4.6. Also we can see that there were no errors after 14 dB with for BPSK with perfect coefficient knowledge or not and at $f_d = 1300$, as seen in Figure 4.5 and Figure 4.6. We have shown both QPSK and BPSK performances with both perfect knowledge of the channel coefficients and channel coefficients obtained from conventional channel estimation together in one plot, as seen in Figure 4.5. It can be seen that the QPSK proposed, although it did not perform better than the proposed BPSK, it still outperformed the conventional DSRC system by up to 14 dB and 7 dB with the perfect knowledge of channel coefficients case and for BER and PER, respectively.

The results shown in Figure 4.10 are the simulation comparisons between the proposed system of this chapter, i.e the frequency-domain equalizer in DSRC system, and the proposed Linear demapper found in chapter 3. The reason for this test is to show how well the proposed system equalization performed

compared with the same system that tries to use soft demapping and decoding techniques, which also helps with the error corrections process. The simulation environment was the same for the comparisons, even the decoders for both systems are the same. The simulation results demonstrate that our proposed scheme equalized the channel as expected, because the proposed system combated the ICI that exist naturally in time-varying channels.

In addition, all the simulation results show that our proposed system did not have errors at high SNR, which allows higher throughput possible. Higher throughput is critical when it comes to communicating important information to ensure road safety.

Chapter 5

Conclusions and Future Works

This thesis started with an investigative approach study of the DSRC system with respect to its performance when it was exposed to harsh outdoor environments. It was shown that the conventional DSRC system performed very poorly due to its inability to mitigate the effect of the channel response and the noisy channel that limited its performance in the first place. Its poor performance was directly related to its poor channel estimation design that only considered time-invariant channel. In addition, the signal compensator is similar to a one-tap equalizer, which is what usually a typical OFDM system uses to equalize the time-invariant channel. Therefore, the problem with that signal compensation design is that it only works when the assumption of the time-invariant nature of the channel is valid. The reason is that the time-variant channel does not produce a circulant time-domain or a diagonal frequency-domain channel matrices, instead it produces a non-diagonal channel matrix, where the non-diagonal components represent the inter carrier interference (ICI) coefficients. In other words, the orthogonality of the subcarriers in the OFDM system will be rendered invalid, which result in the interference of these subcarriers with one another.

These findings inspired our proposed receiver design to take into account these ICI coefficients and equalize the channel in the frequency-domain rather than the time-domain. We have shown that the proposed DSRC receiver design outperformed the conventional DSRC design by up to 15 dB and 12 dB against the same levels of BER and PER, respectively. This improvement in performance translates to an affordability of transmission of larger number of packets at relatively high velocities.

Furthermore, we investigated the DSRC system's performance when different soft decoding and soft demapping schemes replaced the conventional hard decision scheme. It was shown that although the system performed better with soft demapping and soft decoding schemes, the system still had a problem with the existence of the ICI coefficients caused by the channel. The usefulness of the frequency-domain MAP equalization was evident when we demonstrated through simulations and at worst case scenario, i.e. in Rayleigh faded channel, and with the receiver equalizer block being the only block that was changed in the system, the proposed scheme with the MAP equalizer outperformed the system with the soft demapping and soft decoding schemes by a substantial amount. In addition no error floor existed in the proposed design under the simulated conditions.

We also proposed an object-oriented simulation environment for OFDM systems, which can also be used for other communication systems. The proposed simulation environment maximized the diversity of specific analysis that can be compared.

The fact that the 5.9 GHz DSRC system is still undergoing the process of deployment, the findings in this thesis can be further expanded by exploiting or designing lesser complex algorithms for hardware implementation of the frequency-domain MAP equalizer or even exploiting other algorithms, which can accept the ICI coefficients in their designs to mitigate the effect of the channel response in time-varying channels. In addition to that, different channel estimation schemes would be worthwhile exploiting to help provide more accurate representation of the ICI coefficients to the frequency-domain equalizer. Turbo decoding was proven by many researchers to help produce BER performance close to the Shannon limit. Therefore, our proposed design can be further expanded and tested with the turbo equalization concepts implemented in its design.

References

- [1] D. Jiang and L. Delgrossi, "IEEE 802.11p: Towards an International Standard for Wireless Access in Vehicular Environments," *Vehicular Technology Conference*, pp. 2036-2040, May 2008.
- [2] M. Freitas. (2005) Overview of the U.S. DOT Vehicle Infrastructure Initiative.. [Online]. <http://ibtta.org/Events/pastpresdetail.cfm>
- [3] L. Armstrong. (2008, Dec.) Dedicated Short Range Communications. [Online]. <http://www.learmstrong.com/DSRC/DSRCHomeset.htm>
- [4] *IEEE Standard for Information technology—Telecommunications and information exchange between systems—Local and metropolitan area networks—Specific requirements*. 2007.
- [5] H. Abdulhamid, K. E. Tepe, and E. Abdel-Raheem, "Performance of DSRC systems using conventional channel estimation at high velocities," *Int. J. Electron. Commun. (AEU)*, vol. 61, pp. 556-561, 2007.
- [6] R. Chang and R. Gibby, "A Theoretical Study of Performance of an Orthogonal Multiplexing Data Transmission Scheme," *Communication Technology, IEEE Transactions on*, vol. 16, no. 4, pp. 529-540, Aug. 1968.
- [7] S. B. Weinstein and P. Ebert, "Data Transmission by Frequency-Division Multiplexing Using the Discrete Fourier Transform," *IEEE Transactions on Communication Technology*, vol. COM-19, no. 5, pp. 628-634, Oct. 1971.
- [8] M. Engles, *Wireless OFDM systems: how to make them work?*, illustrated ed. Springer, 2002.
- [9] O. Edfors, S. Magnus, J.-J. van de Beek, D. Landström, and F. Sjöberg, "An introduction to orthogonal frequency-division multiplexing," Luleå University of Technology, Sweden, Research Report, September 1996.
- [10] T. S. Rappaport, *Wireless Communications Principles and Practice*. Saddle River, NJ: Prentice Hall PTR, 1999.
- [11] W. C. Jakes, *Microwave mobile communications*, illustrated ed. Wiley, 1974.
- [12] G. L. Stüber, *Principles of mobile communication*, 2nd ed. Springer, 2001.
- [13] H. Abdulhamid, "Channel Estimation for 5.9 GHz DSRC applications," MASc Thesis, University of Windsor, Windsor ON, 1997.

- [14] M. Russel and G. L. Stuber, "Interchannel interference analysis of OFDM in a mobile environment," in *Vehicular Technology Conference, 1995 IEEE 45th*, vol. 2, 25-28 July 1995, pp. 820-824.
- [15] H. Harada, *Simulation and Software Radio for Mobile Communications*, illustrated ed. Artech House, 2002.
- [16] "IEEE P802.11p/D3.0, Draft Amendment for Wireless Access in Vehicular Environments (WAVE)," Jul. 2007.
- [17] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon limit error-correcting coding and decoding: Turbo codes," *Proc. 1993 IEEE International Conference on Communications*, pp. 1064-1070, May 1993.
- [18] Alcatel-Lucent. (2006, Nov.) Bell Labs Advances Intelligent Networks. [Online]. <http://www.bell-labs.com/news/2006/october/shannon.html>
- [19] P. R. John, *Signals: The Telephone and Beyond*, illustrated ed. United States of America: W.H. Freeman and Company, 1981.
- [20] A. J. Viterbi, "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," *IEEE Trans. Inform. Theory IT-13*, pp. 260-269, Jul. 1967.
- [21] H. Abdulhamid, E. Abdel-Raheem, and K. E. Tepe, "Channel tracking techniques for OFDM systems in wireless access vehicular environments," *Signal Processing and Its Applications, 2007. ISSPA 2007. 9th International Symposium on*, pp. 1-4, 12-15, Feb. 2007.
- [22] J. C. Moreira and P. G. Farrell, *Essentials of Error-Control Coding*. John Wiley & Sons, 2006.
- [23] K. E. Tepe, "Iterative decoding techniques for correlated Rayleigh fading and diversity channels," Rensselaer Polytechnic Institute, Troy, NY, USA, Report TR 2001-1, February 2001.
- [24] L. R. Bahl, J. Cocke, F. Jelinek, and J. Raviv, "Optimal Decoding of Linear Codes for minimizing symbol error rate," *IEEE Trans. Information Theory*, vol. 20, pp. 284-287, Mar. 1974.
- [25] L. Papke, P. Robertson, and E. Vilebrun, "Improved decoding with the SOVA in a parallel concatenated (Turbo-code) scheme," *Communications, 1996. ICC 96, Conference Record, Converging Technologies for Tomorrow's Applications. 1996 IEEE International Conference on*, vol. 1, pp. 102-106, Jun. 1996.
- [26] M. Tuchler, R. Koetter, and A. C. Singer, "Turbo equalization: principles and new results," *Communications, IEEE Transactions on*, vol. 50, no. 5, pp. 754-767, May 2002.
- [27] J. Hagenauer and P. Hoeher, "A Viterbi algorithm with soft-decision output and its applications," in *Proceeding of the IEEE, GLOBECOM*, 1989, pp. 1680-1686.
- [28] C. Berrou and A. Glavieux, "Near optimum error correcting coding and decoding: turbo-codes," *Communications, IEEE Transactions on*, vol. 44, no. 10, pp. 1261-1271, Oct. 1996.
- [29] W. C. Leon, *Digital and analog communication systems*, 2nd ed. Michigan: Macmillan, 1987.
- [30] J. G. Proakis, *Digital Communications*, 4th ed. McGraw-Hill, 2000.
- [31] F. Tosato and P. Bisaglia, "Simplified soft-output demapper for binary interleaved COFDM with application to HIPERLAN/2," *Communications, 2002. ICC 2002. IEEE International Conference on*, vol. 2, pp. 664-668, 2002.
- [32] J. Mar and C.-C. Kuo, "Performance Improvement of the DSRC System Using a Novel S and PI-

- Decision Demapper," in *Communications Workshops, 2008. ICC Workshops '08. IEEE International Conference on*, 19-23 May 2008, pp. 405-409.
- [33] J. Hagenauer, E. Offer, and L. Papke, "Iterative decoding of binary block and convolutional codes," *IEEE Transactions on Information Theory*, vol. 42, no. 2, pp. 429-445, Mar. 1996.
- [34] G. Bauch, H. Khorram, and J. Hagenauer, "Iterative Equalization and Decoding in Mobile Communications Systems," *Second European Personal Mobile Communications Conference 2. EPMCC '97*, pp. 307-312, 1997.
- [35] L. Hanzo, T. H. Liew, and B. L. Yeap, *Turbo coding, turbo equalisation, and space-time coding for transmission over fading channels*, illustrated, reprint ed. John Wiley and Sons, 2002.
- [36] C. Heegard and S. B. Wicker, *Turbo Coding*, illustrated ed. Springer, 1999.
- [37] Z. Wang and G. B. Giannakis, "Wireless multicarrier communications where Fourier meets Shannon," *IEEE Signal Processing Magazine*, vol. 17, no. 3, pp. 29-48, May 2008.
- [38] A. Drouillard, P. Picard, M. Jézéquel, C. Berrou, and A. Glavieux, "Iterative correction of intersymbol interference: turbo-equalization," *European Transactions on Telecommunications*, vol. 6, no. 5, special issue on turbo-decoding, pp. 507-511, Sept.-Oct. 1995.
- [39] L. Zhang, C. Zhao, J. Wu, and W. Wang, "WLCp1-12: MAP based Equalizer for OFDM Systems in Time-Varying Multipath Channels," *Global Telecommunications Conference, 2006. GLOBECOM '06. IEEE*, pp. 1-5, Dec. 2006.
- [40] P. Schniter, "Low-complexity equalization of OFDM in double selective channels," *IEEE Transactions, Signal Processing*, pp. 1002-1011, Apr. 2004.
- [41] A. J. Viterbi, *CDMA: Principles of Spread Spectrum Communication*, illustrated ed. Redwood City, CA, United States of America: Addison-Wesley, 1995.
- [42] G. Clark and J. Geist, "Punctured convolutional codes of rate(n-1)/n and simplified maximum likelihood decoding," *IEEE Trans. Information Theory*, vol. IT-25, pp. 97-100, Jan. 1979.
- [43] K. Fang and G. Leus, "Low-Complexity Block Turbo Equalization for OFDM Systems in Time-Varying Channels," *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, vol. 3, pp. III-445-III-448, 15-20 April 2007.
- [44] R. Otnes and M. Tuchler, "Low-complexity turbo equalization for time-varying channels," *Vehicular Technology Conference, 2002. VTC Spring 2002. IEEE 55th*, vol. 1, pp. 140-144, 2002.
- [45] Y.-S. Choi, P. J. Voltz, and F. A. Cassara, "On channel estimation and detection for multicarrier signals in fast selective Rayleigh fading channels," *Communications, IEEE Transactions on*, vol. 49, no. 8, p. 135, Aug. 2001.
- [46] K.-M. Lee, D.-S. Han, C. Jang-Jin, and C. Doo-Hyun, "Performance Of The Viterbi Decoder For Dvb-t In Rayleigh Fading Channels," *Consumer Electronics, 1998. ICCE. 1998 Digest of Technical Papers. International Conference on*, pp. 392-393, 2-4 Jun 1998.
- [47] M. F. Pop and N. C. Beaulieu, "Limitations of sum-of-sinusoids fading channel simulators," *Communications, IEEE Transactions on*, vol. 49, no. 4, pp. 669-708, Apr. 2001.
- [48] T. Aulin, "A modified model for the fading signal at a mobile radio channel," *Vehicular Technology, IEEE Transactions on*, vol. 28, no. 3, pp. 182-203, Aug. 1979.

Appendix A

Graphical User Interface (GUI) and Simulator for OFDM Systems

We propose a Graphical User Interface with an object oriented rapidly reconfigurable simulator of a generic Orthogonal Frequency Division Multiplexing communication system. Object-oriented systems can be designed to be reconfigurable more easily than non-Object-oriented systems. A GUI interface was developed to serve as a tool for a student or advanced professional as a means for investigating particular designs. Giving a GUI to a reconfigurable OFDM system allows for rapid investigation into relative performance analysis (under various conditions and configurations) that would not be as easily accessible otherwise. Additional components can be integrated with relative ease. Example benefits are provided and a variety of techniques are shown for testing OFDM systems on a reconfigurable simulator.

A.1 Introduction

OFDM Communication Systems can have very different configurations from each other, and depending on the transmission medium can also have very different performance characteristics. In order to test a proposed theory for communication advancement, either a full prototype or a simulation must be developed. This Appendix focuses on the design of a simulation environment as a first stage of a design implementation, and as a learning/research tool.

For the scope of this Appendix, receiver Design through Object Based testing is the focus.

Object oriented simulation environments allow for a fair comparisons for multiple receiver designs to be tested under the same transmission circumstances. This allows for relatively small simulations to carry significantly more meaning than compared with systems that employ, thus making it useful for getting reliable estimates of performance once all components have been verified.

The initial investment in simulator design is higher than that of a straight implementation. This fact is mitigated by the re-usability of the generic design and potential for rapid reconfiguration of the design.

Most communication systems are required to operate in different operating modes, having a simulator capable of reconfiguring on demand in a similar way is desirable as it gives ability to test the full system performance.

Object oriented simulators can be made compatible with other object oriented simulators with relative ease and hence can be reused as components when simulating other aspects of a communication system. Additionally, hardware components can be linked into the simulation itself with proper data networking.

A.2 Advantages

Some of the Advantages of using an object oriented simulation design are that it allows for freedom from the engineering team's perspective to approach a broad set of communication scenarios at reduced project cost due to component reuse.

A.2.1 ease of use

Good for novice users because they can use already existing blocks to experiment without having a programming overhead. Any user can make changes to settings using a GUI interface linked with the objects, which gives the advantage of needing less initial investment to start investigating or learning a particular OFDM system.

A.3 Simulation Environment

A.3.1 Environment overview

The Simulation Environment is designed to allow for many changes to be made to the system being simulated without having to change the code defining the simulator. This is due to the fact that the simulation parameters are separated from the system code. The simulation parameters and system configuration, is passed to the simulation setup level, which in-turn passes the system configuration as a global variable for the various components. External components can be either flexibly designed or specifically designed based on user requirements.

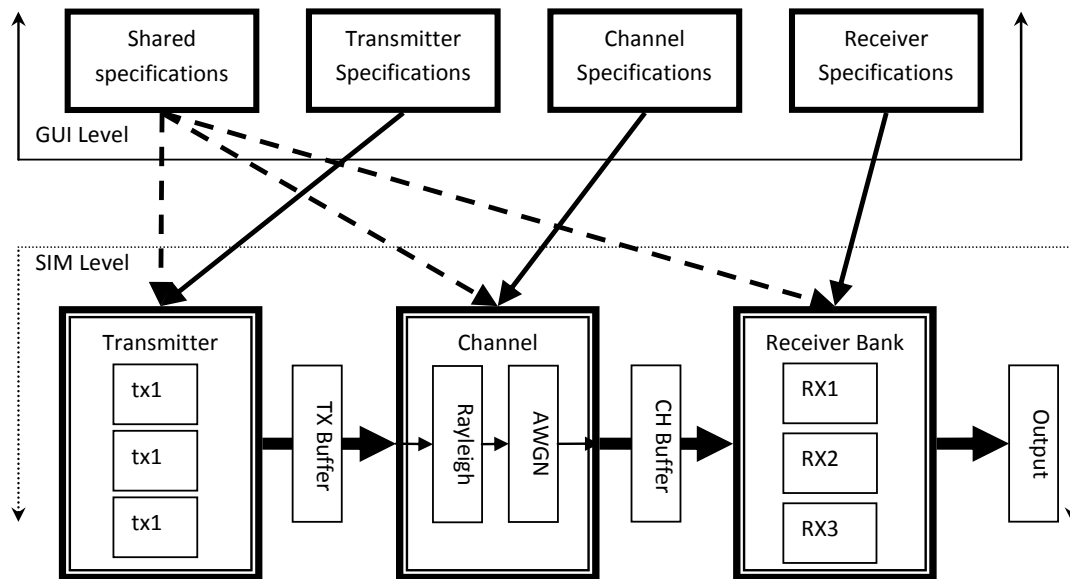


Figure A.1 Illustration of Information exchange between GUI and Simulator

Shared information is passed to each object in the simulator (i.e. fft length, # parallel transmission carriers, number of pilot symbols etc.). In addition, specific information for a particular object is passed as design specific information.

A.3.2 Simulation and System Parameters

The Simulation and System Parameters define what is being tested, and what configuration of system the testing is being done on respectively. For example to test a transmitter/receiver through a channel with varying E_b/N_0 , one would set a vector of changing E_b/N_0 .

A.3.3 Model-Based/Object-Oriented Engineering

Computer and Electrical Engineering is a broad field that is advancing the way we use technology, the speed at which new systems are developed is a metric for measuring good progress within this field. Dynamic Objects that work with each other have the advantage of being easily adapted to existing or new systems with little discrimination, and allow for creative implementations. A standards based transmitter can be coupled by a receiver that improves upon the standard receiver model, and also visa-versa. Using an object oriented simulator in the design process allows for flexibility and gives assistance in creating optimum designs through comparative analysis.

A.3.4 Simulation Configuration

The main specification of the Simulator can be completely defined in the GUI, with all the parameters being saved and sent to the simulator via a configuration file.

In order to give fair result comparisons with relatively little iteration, it is important to duplicate many aspects of the simulation on each pass. That is why each compared design should have the same channel distortions, of course this is not feasible in the real world, but through simulation and estimation of channel response, this can be achieved using pseudo random functions and using the same position of variation in algorithms such as the jakes Rayleigh fade model.

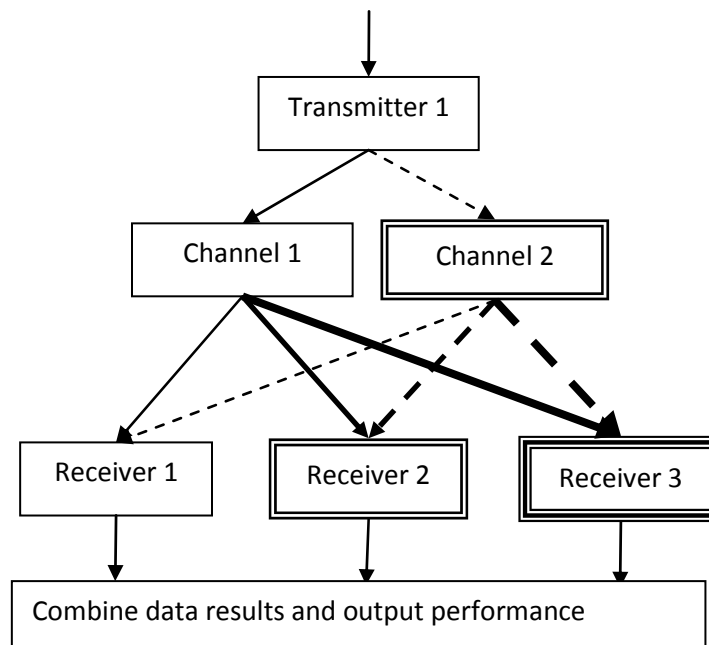


Figure A.2 Tiered Simulation with each path resulting in dataset for output

Tiered simulation is beneficial because it can give fast comparisons of different scenarios. Running different receivers through the same system and comparing results can lead to better analysis.

A.3.5 Iterative Case

For an iterative receiver, a mechanism must exist to fork the current receiver and still keep the data results from the previous iteration. This is achieved by inserting a duplicate receiver class into the receiver list class and feeding the required feedback iteration information. This allows for analysis at the multi simulation level, and additionally if feedback information is preserved from the receiver, further iterations can be added to an existing simulation data set for further analysis.

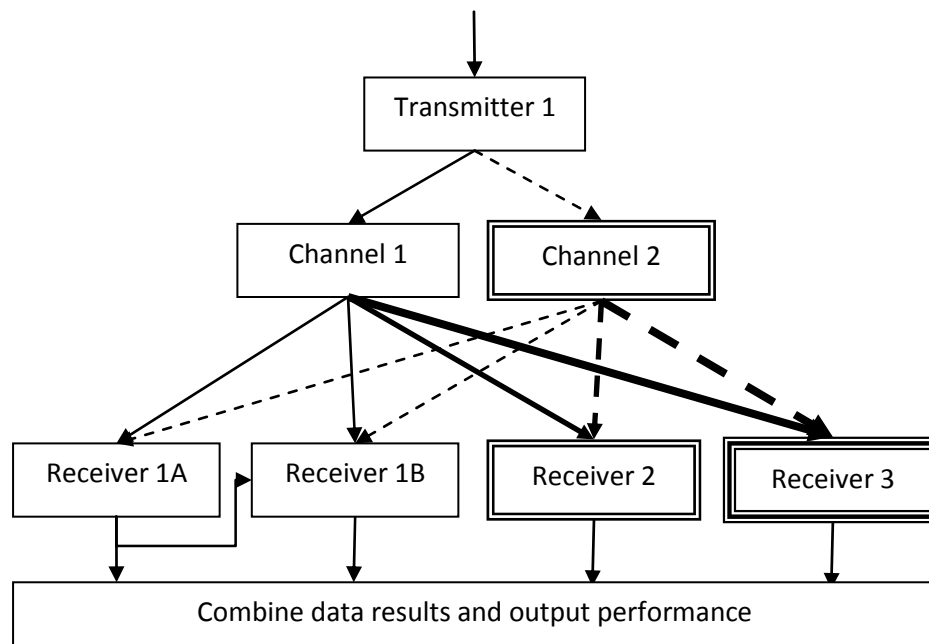


Figure A.3 Modified Tiered Simulation to include Iteration

Figure A.3 illustrates the simulation flow when an iterative receiver is added for comparison against all other receiver versions. To achieve this, the iterative receiver needs to have a feedback component and save the feedback information for the next iteration.

A.3.6 System Configuration

In order to enable the comparison of different implementations of various systems, the configurations run in each simulation block are expected to change according to settings defined by the user.

A.3.6.1 Constraints

It is assumed that the goal of the receiver is to be able to reproduce the original transmission information in its original format, and hence the receiver should output in the same format as the transmitter's input for comparison.

A.3.7 Simulation Settings in the GUI

The GUI contains settings regarding general flexibility available in a particular simulation. The main GUI in the implementation presented was developed using MATLAB Graphical User Interface Development Environment.

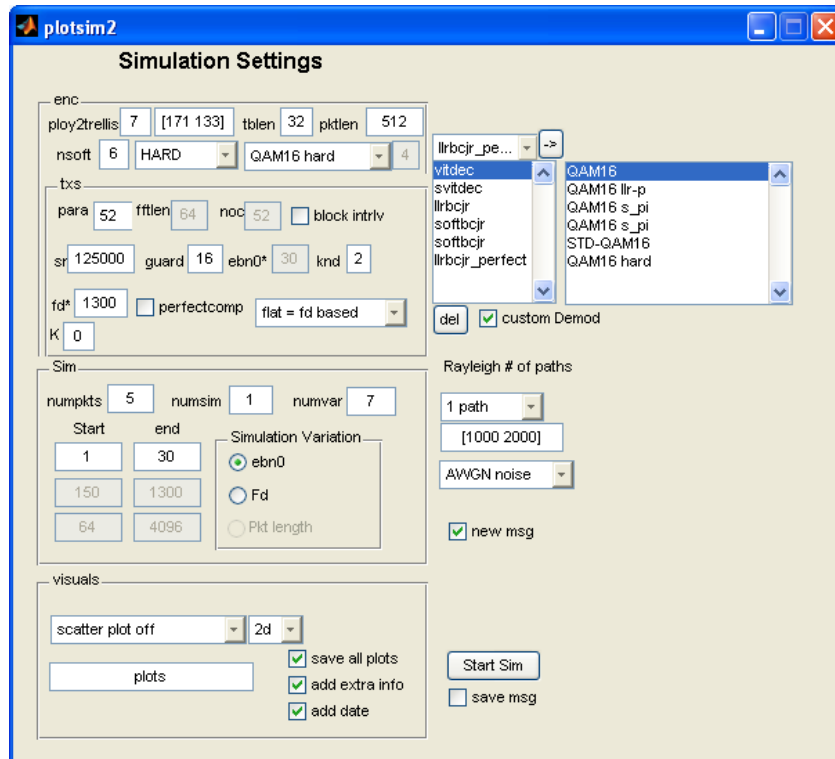


Figure A.4 Main GUI page

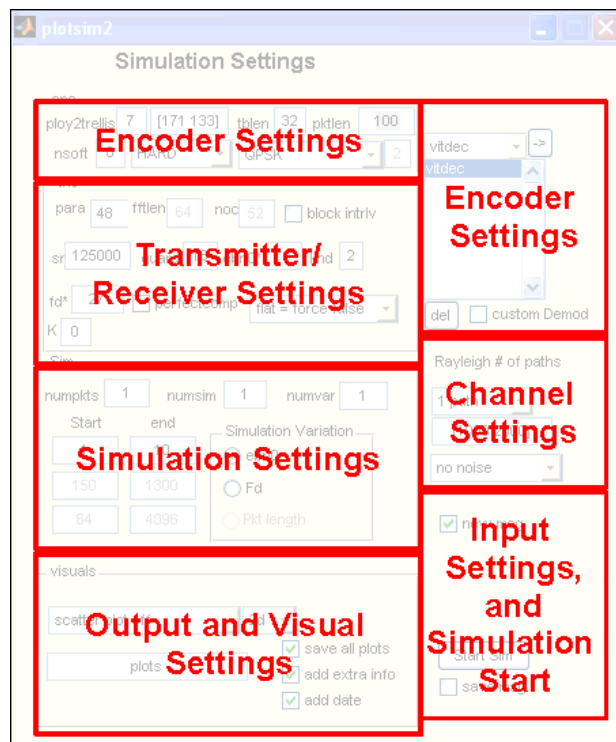


Figure A.5 GUI Layout Sections

Next we will describe each section of the GUI layout (Figure A.5) in more detail.

A.3.7.1 Encoder Settings

Here the trellis can be specified by specifying the constraint length, feedback loops (decimal binary representation of branches) and the trackback length (Figure A.6: trellis input). Additionally the packet length required is specified, which then translates to the number of OFDM symbols being simulated. Nsoft represents the number of $2^{nsoft} - 1$ decisions for the soft representation of bit values. This is and the modulation settings are available in different versions of BPSK, QPSK and QAM depending on the system being simulated during a run.

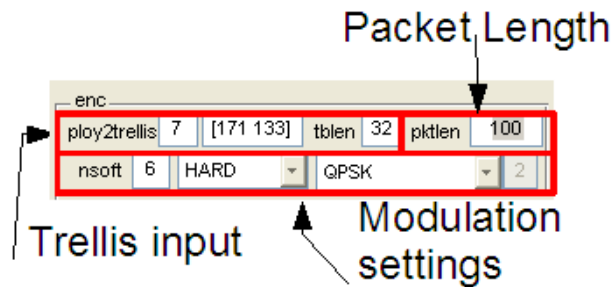


Figure A.6 Encoder Settings

Packet length is chosen and the number of OFDM symbols required to represent that quantity is $2 \times pktlen \times Rc$ etc. .

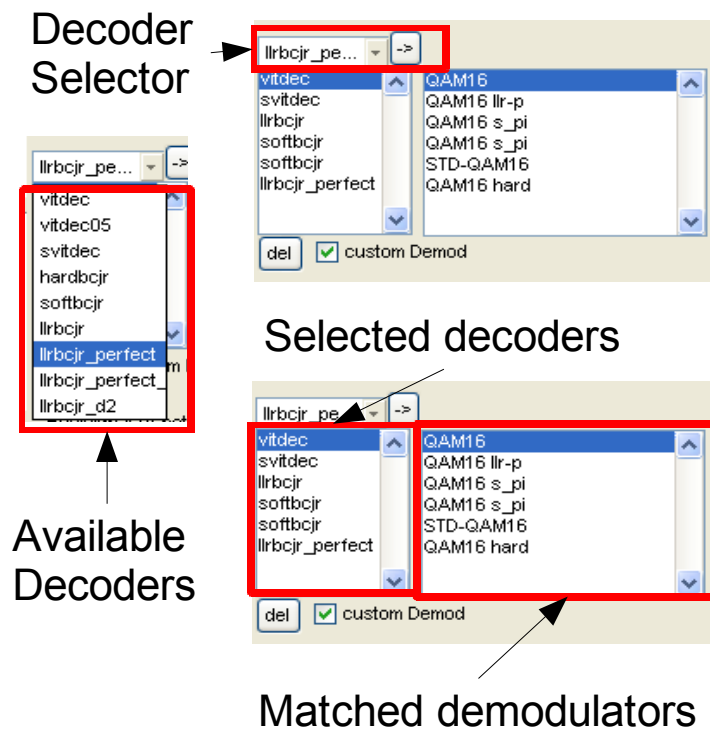


Figure A.7 Decoder Selection process

The available decoders are shown in a list that can be selected, and optionally assign an alternate demodulator scheme as well for further comparisons. For example, in Figure A.7, the matching demodulator for vitdec decoder is QAM16 which represents a hard decision demodulation suitable for the hard decision Viterbi decoder.

A.3.7.2 Transmitter/Receiver Settings

The Transmitter and Receiver settings store fundamental values that are essential to the interoperability of components of the system, to ensure input output size bounding.

Figure A.8 Transmitter and Receiver constants for interoperability between components

A.3.7.3 Simulation Settings

The simulation settings give a selection of different options to choose from when comparing different systems, example for varying ebn0, choose a vector from start to end with constant fd. Not all options are available by default because some options offer conflicting settings, or are not compatible with all components being simulated.

Start	end	Simulation Variation
1	10	<input checked="" type="radio"/> ebn0
150	1300	<input type="radio"/> Fd
64	4096	<input type="radio"/> Pkt length

Figure A.9 varying E_b/N_0

For varying ebn0, choose vector for ebn0, and a constant fd (i.e. it is seen in Figure A.9 that the fd is 27 in this case), notice that the fd vector selection is disabled, and the constant fd* option is enabled.

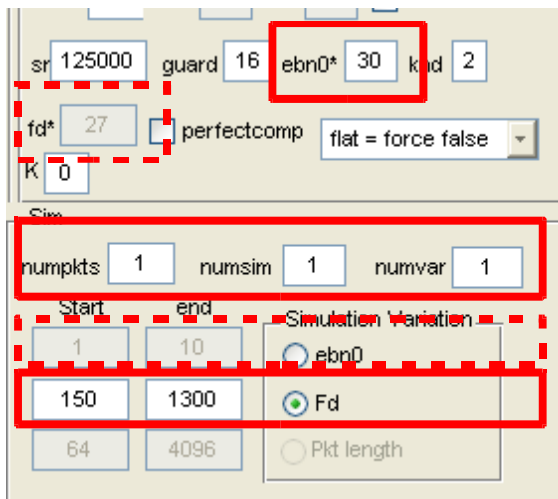


Figure A.10 varying Doppler shift

For varying f_d , choose vector for f_d , and a constant ebn_0 , notice that the ebn_0 vector and constant f_d selection is disabled to correspond to the new varying f_d case.

A.3.7.4 Output and Visual Settings

The output and visual settings define where to save the plotted information as well as what kind of intermediate information to display for analyzing internals (i.e. scatter plots).

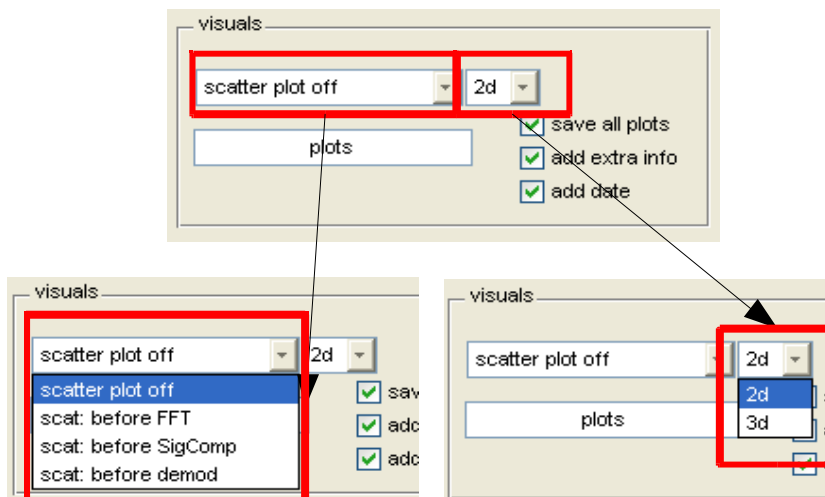


Figure A.11 output settings

The check boxes allow for automatic choice of save location based on overall system parameters, for easy review at a later time. Figure A.11 shows the expansion of some scatter plot options, also the save all plots, add extra info and add date options are enabled which enable the automatic saving of output in the “plots/” subfolder.

A.3.7.5 Channel Settings

The channel settings can be varied to check for response under different fade, and noise simulation components.

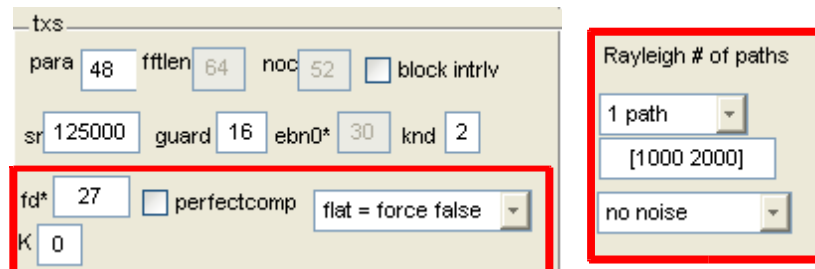


Figure A.12 configurable channel properties

Rayleigh fading can be selected, and a specular component can be added by adjusting the K (rice) factor.

The position of the Jakes Rayleigh fade can be adjusted by entering a matrix for the starting point of the path (i.e. [1000 2000]).

A.3.7.6 Input Settings

The input settings section allows for a message to be defined for use in the simulator. By default the simulator will use the same message if generating a new message, and if the user wants to run the same message on multiple computers, they can specify a *.mat file containing the variable msg for use in the compatible simulation.

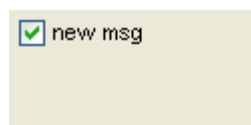


Figure A.13 default action is to create a new simulation message



Figure A.14 msg load section with new message disabled, showing name of the message to load message.

Figure A.14 shows that when disabling the new msg checkbox, the msg name to be loaded (relative to the current directory) is shown that can be changed to point to a specifically saved message.

Saving a message for use on another computer is enabled by marking the save msg check box, which will save the transmitted message (before encoding) in a file "msg.mat".

A.3.7.7 Different pilot settings

The pilot settings can be chosen at varying positions in order to allow for multiple scheme tests. Increasing number of pilot subcarriers decreases the throughput but still serve as good metrics when comparing error rates.

Pilots can be either whole subcarriers, or OFDM Symbols, or a combination of both for testing purposes.

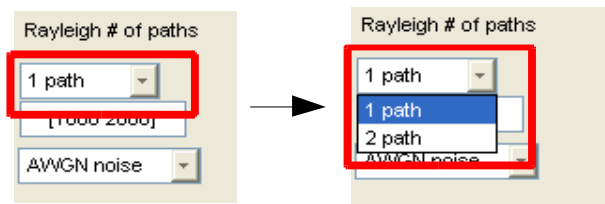


Figure A.15 Choose the number of paths

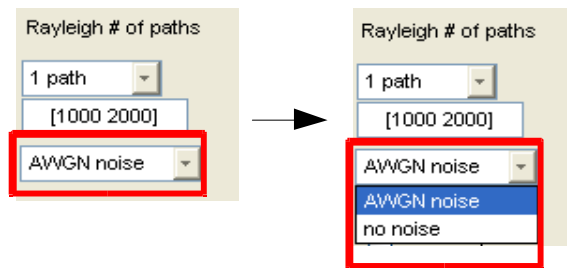


Figure A.16 choose the channel noise method

Figure A.15 shows the number of Rayleigh paths setup for the current simulator, and, Figure A.16 shows underneath that, the start of the Jakes Rayleigh fade counter for each path, and the available noise level options.

A.4 Output Comparisons and Plotting

Different comparisons can be made with relative ease through an interface, or through manual combining of data.

A.4.1 Plotting

When designing a communication system, having information such as the performance of an adaptive equalizer is very important, and measuring that performance directly is very useful as well. If a standard object oriented model is used, the task of analyzing is limited only by the software compatible the objects in questions.

Default plots are bit and packet error bar charts, and bit and packet error rates.

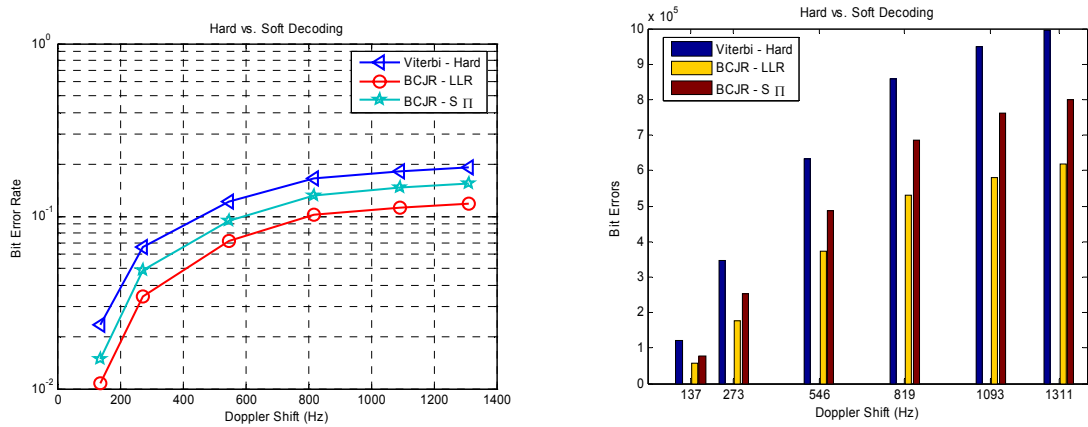


Figure A.17 Different Receivers with different demodulations schemes

A.4.1.1 Scatter plots

Scatter plots at various points of the simulation can be visualized through the drop down menu as seen in Figure A.11. Below is an example of a generated scatter plot.

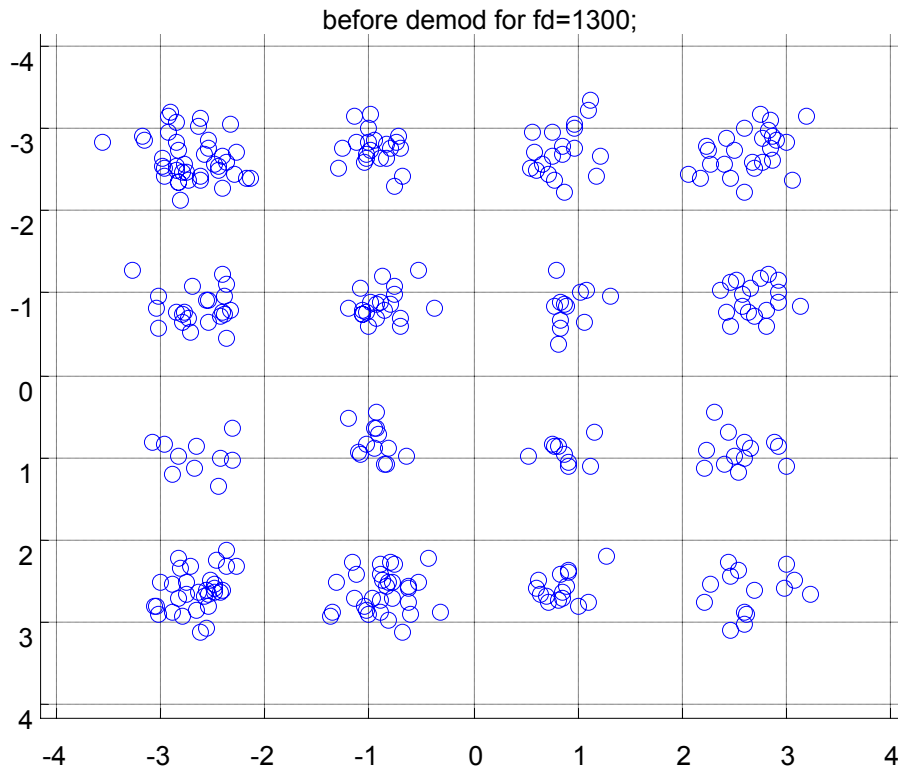


Figure A.18 QAM16 scatter plot at $E_b/N_0 = 20\text{dB}$

A.4.1.2 Alternative simulation combinations

The data generated by each simulation is saved as a data file that can be combined with other simulations. If using multiple computers to generate a simulation, each saved data can then be combined at the end in order to generate more refined/more accurate plots.

A.5 Other Consideration

A.5.1 Motivation

Motivation for this work is based on the time consumption involved in designing simulators of multiple alternatives of a system, and to make the most use of available resources for a researcher as possible through automatic redistribution of load.

A.6 Design

A.6.1 Language and Objects

Higher level languages generally allow for higher abstraction and give flexibility and re-usability in the form of objects. This gives the designer a pseudo block structure as a basis for the design.

The key to using this level of abstraction in an engineering system is creating scalable objects that have dynamic run-time size, and implementing a communication network between the components (objects) to express this configuration.

By creating engineered objects with flexibility inherent to their design, the designer is allowed to rapidly reconfigure a design and test it against known data, and hence allows for enhanced understanding of the topic.

Further: auto network multi-process of parallel systems (distributed simulating). Each packet run (block) can be run as a separate process and therefore can be run as a massively parallel process. This allows for high throughput simulations for rapid testing of theoretical parameters, with the amount of processing time dependent on (equation with packet processing time, iterations, number of processors available, speed of processors etc).

A.6.2 Graphical Design Around a Simulation

The Graphical User Interface is the primary interface between the computer simulation and the user. The GUI for the proposed Simulation environment is built separate from the simulator, but is used to create configuration objects (classes) that both define the simulation environment (i.e. what metrics will change in the simulation) and the configuration of the system being simulated. This separation allows for the information exchanges to be confined to configuration files that can be loaded and interpreted by the reconfigurable simulation.

A.6.3 Flexibility

Extensibility is important for testing new hardware components with minimal configuration changes; hence we included a feature for enabling complete replacement of entire components via file import. This can be done manually by modifying the default settings, or through the GUI menu.

A.6.4 Adding new components for simulating

The functionality of adding new components to the simulation is very useful when testing different versions of a design. Once the input/output of the designed components have been made compatible with the existing OFDM system, it is with relative ease that the new design can be fully integrated with the existing object oriented simulator.

A.6.5 Wrapper Object

Custom made components can be designed and added to the simulator via including a wrapper object which allows for a transition between the input/output of the custom object. This is to facilitate compatibility processing if required. Basically, a system can be made compatible with another by giving conversion instructions to another function, regardless of original programming style or even language. For Matlab implementations, there exists the option of using a mex wrapper for running assembled code specific to the processor.

A.6.6 Application to other Platforms

Although our implementation of this design was created using Matlab, it is important to note that the concepts involved in our generic OFDM simulator can be applied using other platforms alternatively such as C++, Java, IT++, Scilab, etc.

A.6.7 Simulation to Implementation

Once an appropriate design has been tested in the simulation environment successfully, the components settings can then be passed to an appropriate hardware test bench for further analysis (i.e. FPGA test design).

A.7 Performance

Because each Receiver variation is stored as a separate object, each object is calculated independently of the other, lending to parallel processing

A.7.1 Parallelism

In a Simulation, OFDM Transmission can be considered massively parallel due to the separation of the transmission into packets and OFDM Symbols. Time of a simulation is dependent on the number of parallel processors, the number of full transmission blocks being tested.

A.7.2 Alternative Improvements

Software defined radio techniques allow for reconfiguration of the receiver during reception which could be adapted using reconfigurable hardware such as FPGAs.

A.8 Further Reading

[1] Simulation and Software Radio for Mobile Communications, Hiroshi Harada, Published by Artech House, 2002, ISBN 1580530443, 9781580530446

[2] Veiverys, A.; Prasad Goluguri, V.; Le Moullec, Y.; Rom, C.; Olsen, O.; Koch, P., "A generic hardware-accelerated OFDM system simulator," *NORCHIP Conference, 2005. 23rd*, vol., no., pp. 62-65, 21-22 Nov. 2005

URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1596989&isnumber=33584>

[3] Hardware Link Level Emulator for System Level Simulations of WiMAX-like Systems

[4] D. Kotz, C. Newport, R. S. Gray, J. Liu, Y. Yuan, and C. Elliott, "Experimental evaluation of wireless simulation assumptions," in Proceedings of the 7th ACM international Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, Venice, Italy, 4-6 Oct. 2004.

[5] D. Eckhardt, P. Steenkiste, "Measurement and analysis of the error characteristics of an in-building wireless network," *ACM SIGCOMM Computer Communication Review*, v.26 n.4, Oct. 1996

Vita Auctoris

Nabih Jaber was born in London, England, on January 09, 1977. He received his B.A.Sc. degree in electrical engineering/communications option in 2006 from the University of Windsor. He is currently a candidate in the electrical and computer engineering M.A.Sc. program at the University of Windsor. His research interests include wireless and digital communications, information and coding theory, channel estimation and turbo equalization.