

**Low Density Parity Check (LDPC) Codes for Dedicated Short Range
Communications (DSRC) Systems**

by

Najmeh Khosroshahi
B.Sc., University of Tehran, Iran, 2007

A Thesis Submitted in Partial Fulfilment of the Requirements
for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

© Najmeh Khosroshahi, 2011
University of Victoria

All rights reserved. This thesis may not be reproduced in whole or in part by
photocopy or other means, without the permission of the author.

**Low Density Parity Check (LDPC) Codes for Dedicated Short Range
Communications (DSRC) Systems**

by

Najmeh Khosroshahi
B.Sc., University of Tehran, Iran, 2007

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor
(Department Electrical and Computer Engineering)

Dr. Mihai Sima, Departmental Member
(Department Electrical and Computer Engineering)

Supervisory Committee

Dr. T. Aaron Gulliver, Supervisor
(Department Electrical and Computer Engineering)

Dr. Mihai Sima, Departmental Member
(Department Electrical and Computer Engineering)

ABSTRACT

In this effort, we consider the performance of a dedicated short range communication (DSRC) system for inter-vehicle communications (IVC). The DSRC standard employs convolutional codes for forward error correction (FEC). The performance of the DSRC system is evaluated in three different channels with convolutional codes, regular low density parity check (LDPC) codes and quasi-cyclic (QC) LDPC codes. In addition, we compare the complexity of these codes. It is shown that LDPC and QC-LDPC codes provide a significant improvement in performance compared to convolutional codes.

Contents

| | |
|---|-------------|
| Supervisory Committee | ii |
| Abstract | iii |
| Table of Contents | iv |
| List of Tables | vii |
| List of Figures | viii |
| Acknowledgements | x |
| Dedication | xi |
| 1 Introduction | 1 |
| 1.1 Dedicated Short Range Communication (DSRC) | 2 |
| 1.1.1 FCC DSRC Frequency Allocation | 2 |
| 1.1.2 The ASTM DSRC Standard | 3 |
| 1.1.3 The DSRC Spectrum | 3 |
| 1.1.4 Current Activities | 5 |
| 1.2 DSRC and Error Control Coding | 5 |
| 1.3 The Objectives of this Thesis | 6 |
| 2 The DSRC Transceiver | 8 |
| 2.1 The DSRC Transmitter | 9 |
| 2.2 The DSRC Channel | 10 |
| 2.3 The DSRC Receiver | 10 |
| 2.4 Puncturing | 11 |
| 2.5 DSRC Scrambler Structure | 11 |
| 2.6 Orthogonal Frequency Division Multiplexing (OFDM) | 12 |

| | | |
|----------|---|-----------|
| 2.6.1 | Multiple Carriers for Data Transmission | 13 |
| 2.6.2 | Overlapping Sub-channels in Multicarrier Modulation | 15 |
| 2.6.3 | Discrete Fourier Transform (DFT) and Inverse DFT (IDFT) | 17 |
| 2.6.4 | Cyclic Prefix | 17 |
| 2.6.5 | OFDM in DSRC | 19 |
| 3 | Channel Model | 22 |
| 3.1 | Small-Scale Fading | 24 |
| 3.1.1 | Factors that Influence Small-Scale Fading | 24 |
| 3.1.2 | Slow and Fast Fading | 25 |
| 3.1.3 | Frequency-Flat and Frequency-Selective Fading | 26 |
| 3.1.4 | Modelling a Flat Fading Channel | 28 |
| 3.2 | The DSRC Channel | 30 |
| 3.2.1 | The DSRC Channel Model | 31 |
| 4 | Convolutional Coding | 32 |
| 4.1 | Convolutional Encoder | 33 |
| 4.1.1 | D-Transform Domain | 35 |
| 4.1.2 | Convolutional Code Representation | 36 |
| 4.2 | Minimum Free Distance of a Convolutional Code | 37 |
| 4.3 | Convolutional Decoding: The Viterbi Algorithm | 38 |
| 4.4 | Convolutional Decoding Complexity | 42 |
| 5 | Regular and Quasi-Cyclic LDPC Codes | 43 |
| 5.1 | Regular LDPC Codes | 44 |
| 5.1.1 | Linear Block Codes | 44 |
| 5.1.2 | Cyclic Codes | 46 |
| 5.1.3 | Quasi-Cyclic Codes | 48 |
| 5.1.4 | Regular LDPC Codes | 49 |
| 5.2 | Random LDPC Codes | 51 |
| 5.3 | QC-LDPC Code Construction | 53 |
| 5.3.1 | Distance Graph | 53 |
| 5.3.2 | Construction Algorithm for QC-LDPC Codes | 54 |
| 5.4 | The Sum Product Algorithm | 57 |
| 5.4.1 | Logarithmic SPA Decoding | 65 |
| 5.5 | Decoding Complexity | 66 |

| | | |
|----------|---------------------------|-----------|
| 6 | Simulation Results | 68 |
| 7 | Conclusions | 81 |
| 7.1 | Future Work | 82 |
| | Bibliography | 84 |

List of Tables

| | |
|--|----|
| Table 2.1 DSRC Physical Layer Parameters [4] | 8 |
| Table 5.1 The Received Vector and Corresponding values of f_j^x [10] | 60 |
| Table 5.2 Number of Operations per Iteration with SPA and Log-SPA Decoding | 66 |

List of Figures

| | | |
|-------------|--|----|
| Figure 1.1 | The national ITS architecture [1]. | 2 |
| Figure 1.2 | The open systems interconnection model [1]. | 3 |
| Figure 1.3 | The 5.9 GHz DSRC frequency plan with 10 MHz channels [1]. | 4 |
| Figure 1.4 | Coding gain for various signal to noise ratios [3]. | 6 |
| Figure 2.1 | The DSRC transmitter model. | 9 |
| Figure 2.2 | The DSRC receiver model. | 11 |
| Figure 2.3 | The DSRC scrambler structure [6]. | 12 |
| Figure 2.4 | The DSRC descrambler structure [6]. | 12 |
| Figure 2.5 | The multicarrier transmitter [3]. | 14 |
| Figure 2.6 | The multicarrier receiver [3]. | 15 |
| Figure 2.7 | Multicarrier modulation with overlapping subcarriers [3]. | 16 |
| Figure 2.8 | The multicarrier receiver for overlapping subcarriers [3]. | 16 |
| Figure 2.9 | A cyclic prefix of length μ [3]. | 18 |
| Figure 2.10 | The ISI between data blocks at the channel output [3]. | 18 |
| Figure 2.11 | The DSRC transmitter block diagram. | 19 |
| Figure 2.12 | The DSRC receiver block diagram. | 20 |
| Figure 3.1 | Combined path loss, shadowing, and narrowband fading obtained based on an empirical observation [3]. | 23 |
| Figure 3.2 | The reflected, diffracted, and scattered wave components [3]. | 24 |
| Figure 3.3 | Geometry associated with the Doppler shift [3]. | 26 |
| Figure 3.4 | Multipath signal reception in frequency-flat or frequency-selective fading [3]. | 27 |
| Figure 3.5 | Small-scale fading categories [7]. | 28 |
| Figure 4.1 | The convolutional encoder used in DSRC systems according to the IEEE 802.11a standard. | 33 |
| Figure 4.2 | Convolutional encoder example with rate 1/2 [10]. | 37 |

| | | |
|-------------|---|----|
| Figure 4.3 | The state diagram for the convolutional encoder in Fig. 4.2 [10]. | 37 |
| Figure 4.4 | The trellis diagram for the convolutional encoder in Fig. 4.2 [10]. | 38 |
| Figure 4.5 | Hard-decision decoding for the convolutional code in Fig. 4.2 [10]. | 40 |
| Figure 4.6 | Trellis diagram for the convolutional code in Fig. 4.2 with output values in polar format [10]. | 40 |
| Figure 4.7 | Soft-decision decoding of the convolutional code in Fig. 4.2 [10]. | 41 |
| Figure 5.1 | The Tanner graph of a (7,3) linear block code [9]. | 51 |
| Figure 5.2 | Distance graph and matrix representation of an LDPC code [13]. | 54 |
| Figure 5.3 | Graph representation of a (16, 2, 4) QC-LDPC code with girth eight [13]. | 56 |
| Figure 5.4 | Parity check matrix H of a (16, 2, 4) QC-LDPC code with girth eight [13]. | 56 |
| Figure 5.5 | Bipartite graph for the example given in [10]. | 61 |
| Figure 5.6 | Calculation of R_{12}^0 and R_{12}^1 [10]. | 63 |
| Figure 5.7 | Calculation of Q_{12}^0 and Q_{12}^1 [10]. | 64 |
| Figure 6.1 | BER for a $K = 6$ convolutional code with different code rates in an AWGN channel. | 72 |
| Figure 6.2 | BER for an LDPC (504,1008,3,6) code with different code rates in an AWGN channel. | 72 |
| Figure 6.3 | BER for a rate 1/2 convolutional code in three different channels. | 73 |
| Figure 6.4 | BER for an LDPC (504,1008,3,6) code in three different channels. | 73 |
| Figure 6.5 | BER for an LDPC (2000,4000,3,6) code in three different channels. | 74 |
| Figure 6.6 | BER for a QC-LDPC (504,1008,3,6) code in three different channels. | 75 |
| Figure 6.7 | BER for a QC-LDPC (2000,4000,3,6) code in three different channels. | 75 |
| Figure 6.8 | Performance result of three different coding methods in AWGN channel. | 76 |
| Figure 6.9 | Performance of three different coding methods in a Rayleigh channel. | 76 |
| Figure 6.10 | Performance of three different coding methods in a Rician channel. | 77 |
| Figure 6.11 | The effect of block fading on three different coding methods. . . | 78 |
| Figure 6.12 | The effect of an interleaver for a small parity check matrix size under block fading. | 79 |

ACKNOWLEDGEMENTS

I would like to state my immense gratitude to all who gave me the opportunity to complete this thesis. I am deeply indebted to my supervisor Prof. A. Gulliver who has been a source of inspiration, for his timely guidance in the conduct of my thesis, all his help, support, interest and his invaluable advice. It is a pleasure to express my gratitude to the Electrical Engineering Department of University of Victoria to provide me with the chance to commence my studies.

Lastly, and most importantly, I would like to express my profound heartfelt thanks to my beloved parents for their blessings, encouragement, help and wishes throughout my graduate studies.

Dedicated to

My Beloved Parents, my Sweet Sisters,
and my Dear Friends

Chapter 1

Introduction

Recently, wireless communications between vehicles and roadside equipment has attracted the attention of numerous organizations. Their goal is to improve transportation safety by developing intelligent transportation system (ITS) applications. One of the most important applications of ITS is dedicated short range communication (DSRC), which will play a key role in inter-vehicle communications. Fig. 1.1 [1] illustrates a DSRC system which allows wireless data exchange between roadside equipment at fixed location and emitters-receptors in moving vehicles. DSRC systems will not only increase traveller safety via applications such as collision warnings, but also reduce fuel consumption and pollution through applications such as Electronic Toll Collection (ETC), real-time traffic advisories, digital map updates, etc.

The impetus for the recent progress in DSRC systems results from the period 2002 to 2004 when seven automotive manufacturers worked with the United States Department of Transportation (USDOT) to evaluate vehicle safety applications. In the course of this project, communications equipment was recognized as a key system component, resulting in the development of the DSRC standards to support safety applications. Subsequently, from 2005 to 2006, automotive manufacturers developed the Emergency Electronic Brake Light application (EEBL) as the first inter-vehicle cooperative safety application. In 2006, five automotive manufacturers began a major vehicle safety communications project with the United States Department of Transportation (USDOT). They developed a Cooperative Intersection Collision Avoidance System [2] using infrastructure-to-vehicle communications to address intersection crashes that result from signal violations. The goal was to reduce the number of fatal accidents at intersections, where the collision probability is significant.

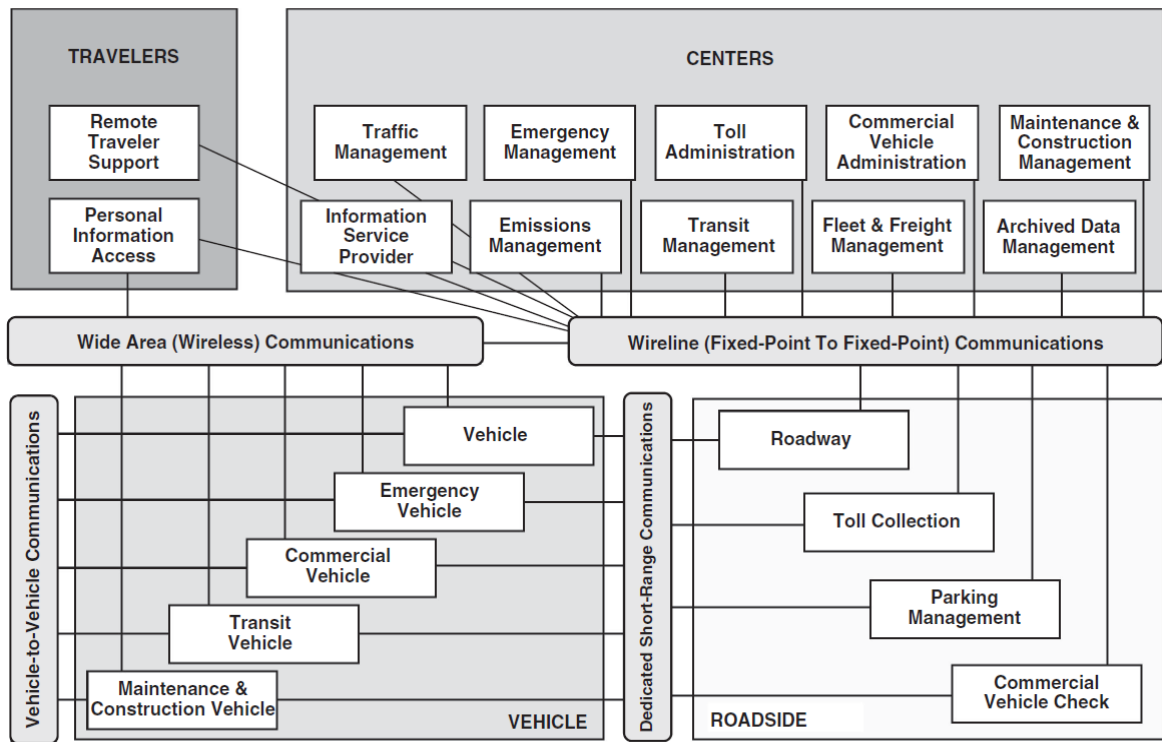


Figure 1.1: The national ITS architecture [1].

1.1 Dedicated Short Range Communication (DSRC)

1.1.1 FCC DSRC Frequency Allocation

In 1997, ITS appealed to the US Federal Communications Commission (FCC) to allocate seventy-five megahertz of spectrum in the 5.9 GHz band specifically for Dedicated Short Range Communication (DSRC) applications. In October 1999, the FCC accepted this request and allocated the requested bandwidth for DSRC applications. Subsequently, licensing and rules were established in 2003 for DSRC services.

The main equipment in a DSRC system are the On-Board Units (OBUs), a transceiver built in or on a vehicle, and Roadside Units (RSUs), a transceiver that is mounted along a road or pedestrian walkway. An RSU transmits data to OBUs or exchanges data with OBUs in its communication region.

1.1.2 The ASTM DSRC Standard

In order to provide an international standard for DSRC systems, the standards group chose the IEEE 802.11 specifications for the lower-layers, particularly the Medium Access Control (MAC) and Physical Layer (PHY). Based on the Open Systems Interconnection communication framework [1], the operation of DSRC devices can be categorized using a seven-layer profile, as shown in Fig. 1.2. The physical layer defines

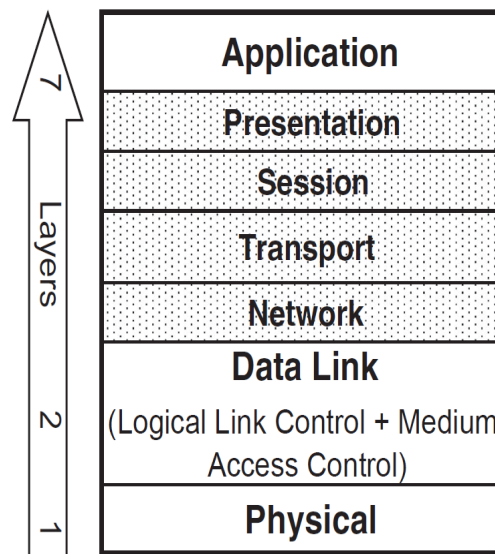


Figure 1.2: The open systems interconnection model [1].

the frequencies, modulation and coding for wireless communications. Since this thesis is devoted to the physical layer of DSRC systems, we do not consider the function of the other layers.

1.1.3 The DSRC Spectrum

To provide spectrum efficiency, the Federal Communications Commission (FCC) adopted channelization for the DSRC spectrum. Thus the DSRC spectrum is divided into 8 channels, one 5 MHz channel kept in reserve and seven 10 MHz channels. The identification of the DSRC channels dedication is illustrated in Fig. 1.3. One channel is used for control applications (Channel 178) and two other channels are set aside for public safety applications (Channels 184 and 172). The four remaining channels are available for both public and private short to medium distance communication applications. Channel 178 in the middle of the frequency band is the control

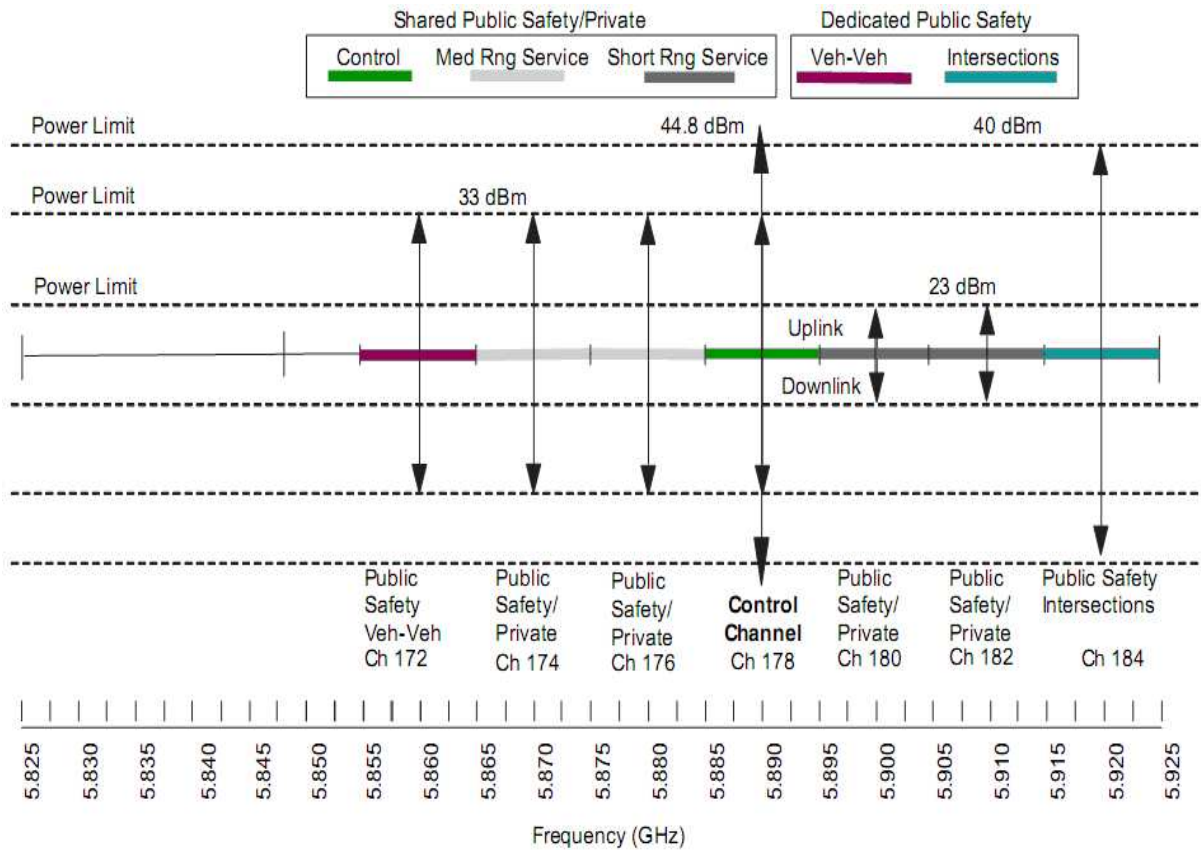


Figure 1.3: The 5.9 GHz DSRC frequency plan with 10 MHz channels [1].

channel. The basic concept is that a Road-Side Unit (RSU) broadcasts to On-Board Units (OBUs) on this channel 10 times per second according to the applications. The On-Board Unit listens to Channel 178, authenticates the RSU, executes safety applications first, switches channels and executes non-safety applications, then returns to listening to Channel 178. In 2006, the FCC designated Channel 172 (5.855-5.865 GHz), for vehicle-to-vehicle safety communications to support accident avoidance and safety of life and property applications. In addition, Channel 184 (5.915-5.925 GHz), is allocated to high power, long distance communications including road intersection collision mitigation. Allocating the first channel (Channel 172) for public safety applications reduces the probability of interference between other channels and this channel, thus enhancing system performance.

1.1.4 Current Activities

In the United States, there are many important activities in progress related to communications-based vehicle safety applications. Here we mention the two most significant developments.

Vehicle Infrastructure Initiative (VII)

The Vehicle Infrastructure Initiative (VII) will provide every vehicle manufactured in the U.S. with the equipment required for inter-vehicle and vehicle-to-infrastructure communications. This will not only reduce highway fatalities through safety applications, but also provide important features such as vehicle data, weather/road surface data, traveller information, and electronic tolls.

Cooperative Intersection Collision Avoidance System (CICAS)

The Cooperative Intersection Collision Avoidance System (CICAS) is a project to develop a vehicle-infrastructure communications system to improve intersection safety. In this project, vehicles will be equipped with an in-vehicle device that warns the driver either that it is probable that they will violate a traffic signal or stop sign, that it is unsafe to go through the intersection due to insufficient gaps in traffic, or that it is unsafe to make a left turn at a signal intersection.

1.2 DSRC and Error Control Coding

Error correction coding is employed in DSRC systems to reduce the probability of error in the data transmitted through the wireless channel. The bit error probability, P_b , for a coded system is the probability that a bit is decoded in error. One of the most significant properties of error correction coding is coding gain. The reduction in required signal to noise ratio (SNR) due to the coding technique for a given P_b is defined as the coding gain. This concept is illustrated in Fig. 1.4. Codes designed for high SNR channels can have a negative coding gain at low SNRs when the code redundancy does not provide sufficient performance gain to overcome the decrease in energy per bit. Negative coding gain can be avoided with proper code design for the target values of SNR.

The error probability versus SNR curve with or without coding has a waterfall shape at low to moderate SNRs. Without coding, this shape is maintained for all

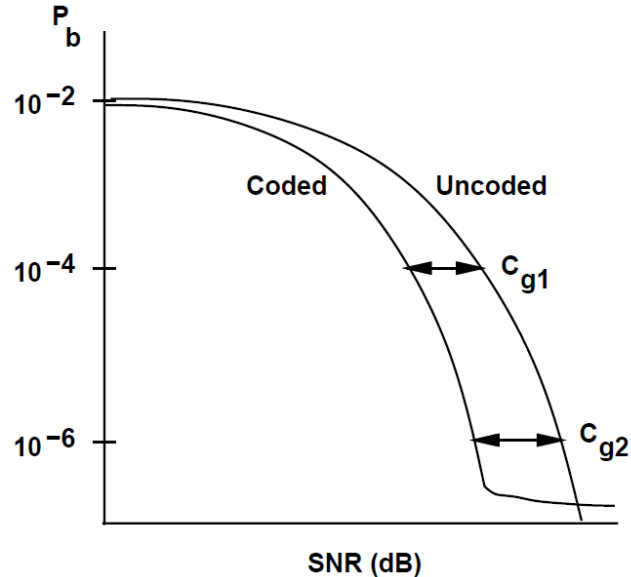


Figure 1.4: Coding gain for various signal to noise ratios [3].

SNRs, however for coded systems an error floor may appear at larger SNRs, as shown in Fig. 1.4. This error floor occurs at a threshold SNR which depends on the code design. For SNRs above this threshold, the error probability curve falls off faster. This is due to minimum distance error events which dominate code performance at high SNRs. Note that for all coding techniques, the error correction capability does not come for free. The performance improvement is paid for by increased complexity and either a decreased data rate or increased in signal bandwidth. For example, consider a code with n coded bits for every k uncoded bits. Thus the code converts an element of a k -dimensional space into an element of a larger n -dimensional space to allow for larger distances between codewords. On the other hand, if we assume R_b is the data rate in the channel, then the information rate for a code that uses n coded bits for every k uncoded bits is only $\frac{k}{n}R_b$. Thus coding decreases the data rate by the fraction k/n . Conversely, if we maintain the information rate constant and decrease the bit time by k/n , the result is an expanded signal bandwidth by n/k .

1.3 The Objectives of this Thesis

Considering the above discussion, this thesis focusses on Forward Error Correction (FEC) for the DSRC transceiver. In particular, the performance of DSRC systems with Low Density Parity Check (LDPC) Codes will be investigated. LDPC codes

are a type of block code, whereas convolutional codes have been proposed in the DSRC standard. Chapter 2 presents an introduction to the DSRC transmitter and receiver structures. The following chapters discuss convolutional and LDPC code construction, encoding and decoding structures, and code complexities. Then, the performance of a DSRC system with different channels, code rates, error control coding is simulated and compared. In addition, the effects of interleaving, puncturing and OFDM modulation are examined.

Chapter 2

The DSRC Transceiver

The DSRC physical layer frame including modulation and forward error correction, is similar to the IEEE 802.11a standard although the latter was designed for indoor wireless local area network (WLAN) applications. Therefore the system parameters are optimized for the indoor low-mobility propagation environment. The basic DSRC parameters are shown in Table 2.1, so the DSRC signal bandwidth is 10MHz which is half of the IEEE 802.11a bandwidth.

The most noticeable difference between the DSRC and IEEE 802.11a standard parameters is apparent when we consider vehicles moving at speeds up to 200 km/h , with communication ranges up to 1000 m . In this case not only the channel is very different from the IEEE 802.11a indoor low-mobility environment, but also it can be very hostile. In order to mitigate this problem, more powerful forward error correction (FEC) can be used, which is the goal of this thesis. In particular, we evaluate the performance of a DSRC system using LDPC codes rather than convolutional codes as specified in the DSRC standard.

Table 2.1: DSRC Physical Layer Parameters [4]

| | |
|-----------------------|-------------------------|
| Data Rate | 3,4,5,6,9,12,18,27 Mbps |
| Modulation | BPSK,QPSK,16-QAM,64-QAM |
| Coding Rate | 1/2, 2/3, 3/4 |
| Number of Subcarriers | 52 |
| Subcarrier Spacing | 156.25 KHz |
| Number of Pilot Tones | 4 |
| Guard Interval | $1.6\mu\text{sec}$ |
| OFDM Symbol Duration | $8\mu\text{sec}$ |
| Signal Bandwidth | 10 MHz |

2.1 The DSRC Transmitter

A block diagram of the DSRC transmitter is shown in Fig. 2.1. The input bit stream is first scrambled using the scrambling polynomial, $s(x) = x^7 + x^4 + 1$ as defined in the IEEE 802.11a standard. Then the data bits are encoded using a 64 state rate 1/2 convolutional code. Higher code rates are obtained by puncturing the convolutional encoder output.

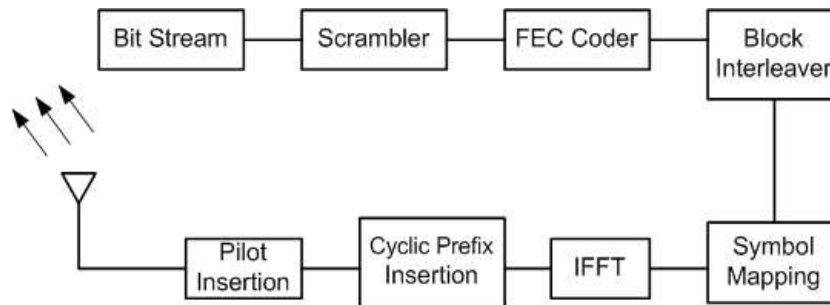


Figure 2.1: The DSRC transmitter model.

In order to reduce the effect of burst errors caused by the block fading channel¹, an interleaver is used to redistribute input bits in the time and/or frequency domains before transmission. A comprehensive explanation of fading channels will be given in the following chapters, but here we can briefly say that when vehicles move slowly, i.e., in congested urban areas, the channel coherence time is normally much larger than the packet transmit period, thus the channel can be assumed to be time invariant over a packet interval. According to the IEEE 802.11a standard, the encoded data is interleaved with a block size corresponding to the number of bits in a single OFDM symbol so in this case the interleaving depth depends on the modulation employed. For example, with BPSK modulation this amount is 48, while for QPSK it is 96. Data is transmitted on a set of orthogonal subcarriers after mapping to the desired modulation symbols. An inverse Fast Fourier Transform (IFFT) is performed to obtain the time domain orthogonal frequency-division multiplexing (OFDM) symbol. A cyclic prefix is added at the beginning of each OFDM symbol to combat the ISI introduced by the frequency selective fading channel. In addition, 4 pilot symbols are inserted in each OFDM symbol for synchronization at the receiver. Analysis of this aspect of the system is beyond the scope of this thesis, so we assume perfect time

¹In mobile communications, a channel in which the same multipath fading corrupts a number of consecutive bits is called block fading.

and frequency synchronization. In the next section, the OFDM system is examined in detail.

2.2 The DSRC Channel

In this section, we briefly introduce the channel model used in this thesis, as details of this model will be given in the following chapter. Although there have been numerous vehicle-to-vehicle communication studies, the mobile-to-mobile channel model for vehicular environments is not well understood. However, the analysis in [5] shows that we can assume a Rician fading distribution when the distance between two vehicles is less than $100m$, and a Rayleigh fading distribution when the distance is greater than $100m$.

A measure of the expected time duration over which the channel response is constant is called coherence time, T_c , which is inversely proportional to the Doppler spread of the channel, within a multiplicative constant. The channel is said to be slow fading if the symbol time, T_s , is much less than the coherence time, T_c , or equivalently the Doppler bandwidth, f_D , is much smaller than the signal bandwidth, $1/T_s$. Conversely, a channel is called fast fading if the symbol time is greater than the coherence time, or equivalently the Doppler bandwidth is greater than the signal bandwidth. According to the investigation in [5], the relative speed observed between two vehicles is typically less than $10 \text{ mi/h} = 16 \text{ km/h}$, so the corresponding Doppler spread is less than 100Hz . This occurs when two vehicles are travelling in the same direction. The corresponding channel coherence time in this model is around $10ms$, so it can be assumed that the channel fades independently for time durations greater than $10ms$. With a Rician channel model, there is a line of sight (LOS) component, and the ratio of the LOS component power to the Rayleigh scattered power (NLOS), is called the Rician parameter k . In [5], the Rician fading parameter is said to lie in the range $K = [0, 2]$. Since $K = 0$ corresponds to Rayleigh fading, we consider $K = 2$ for Rician fading.

2.3 The DSRC Receiver

The DSRC receiver is shown in Fig. 2.2. The reverse processes of the transmitter are performed on the receiver side. Timing and frequency estimation in addition to

channel estimation are disregarded since we assume perfect time and frequency synchronization. Then the first operation after receiving the signal is removing the guard interval (also called the cyclic prefix). A Fast Fourier Transform (FFT) is performed

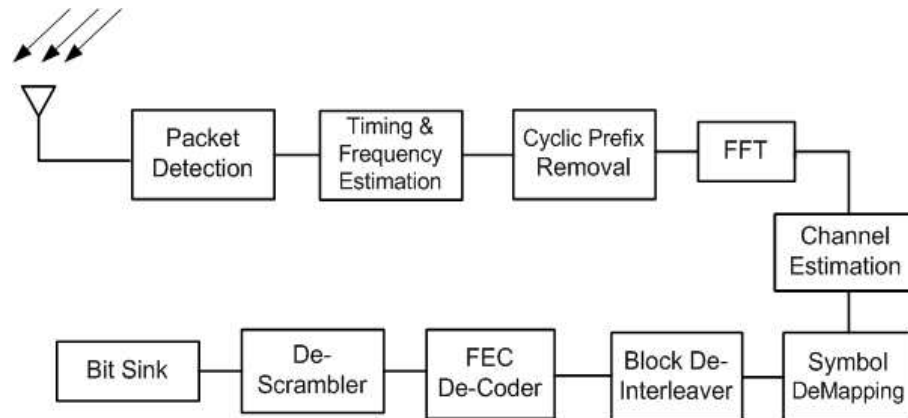


Figure 2.2: The DSRC receiver model.

on the received signal to transform it to the frequency domain and the resulting signal is mapped to bits and de-interleaved. This is used as the soft information input to the Viterbi decoder, and finally the decoded bits are descrambled.

2.4 Puncturing

In coding theory, puncturing is the procedure of eliminating a number of encoded bits added through Forward Error Correction (FEC). The effect of this procedure is identical to implementing an error-correction code with a higher rate, or less redundancy. On the other hand, the significant advantage of puncturing is that the same decoder can be used regardless of how many bits have been punctured; hence puncturing considerably enhances the flexibility of the system without increasing its complexity. In this thesis, we use puncturing to obtain different code rates for convolutional coding as the conventional FEC block.

2.5 DSRC Scrambler Structure

In telecommunications, a scrambler (also referred to as a randomizer), manipulates a data stream before transmission to eliminate long sequences consisting of only ‘0’ or ‘1’. This technique provides some level of automatic gain control in addition to

making the transmitted data more dispersed. This is important because if the power is concentrated in a narrow frequency band, it can interfere with adjacent channels and causes adjacent channel interference. The structures of the scrambler and descrambler

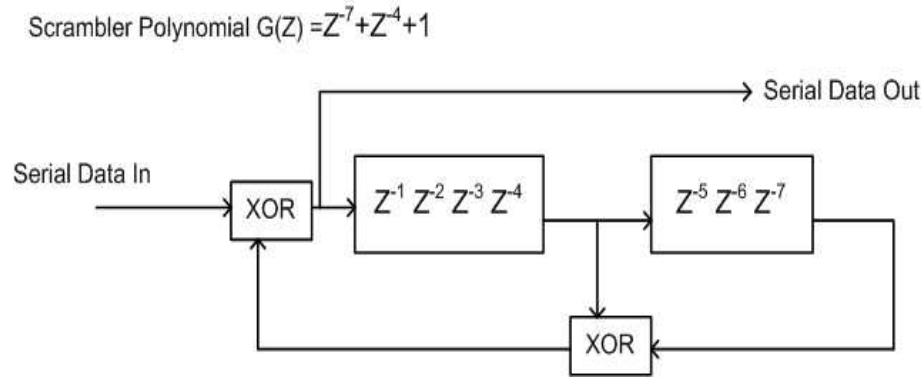


Figure 2.3: The DSRC scrambler structure [6].

of a DSRC system are shown in Fig. 2.3 and Fig. 2.4, respectively. A scrambler can be placed just before or after the FEC, but before the modulation.

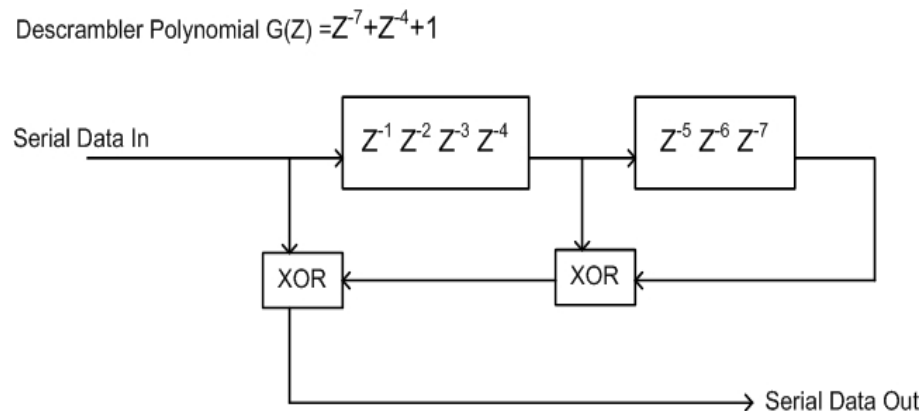


Figure 2.4: The DSRC descrambler structure [6].

2.6 Orthogonal Frequency Division Multiplexing (OFDM)

Orthogonal Frequency Division Multiplexing (OFDM) is considered as an effective technique to combat frequency selective fading in wireless applications. The basic objective is to divide the transmitted bit stream into a number of sub-streams and

send each over different orthogonal sub-channels. As a result, the data rate on each sub-channel is much less than the total data rate, therefore the bandwidth of each sub-channel is much less than the total system bandwidth. Thus, this technique converts frequency selective fading to approximately flat fading in each sub-channel, so the inter-symbol interference (ISI) on each sub-channel is generally negligible. This multicarrier modulation technique is the foundation of OFDM and allows the ISI to be eliminated through the use of a cyclic prefix. The main problems in multicarrier modulation that impair its performance are frequency offset² and timing jitter³, which degrade the orthogonality of the sub-channels.

2.6.1 Multiple Carriers for Data Transmission

As mentioned previously, the simplest multicarrier modulation structure splits the data stream into multiple sub-streams to be transmitted over orthogonal sub-channels at different subcarrier frequencies. The number of sub-streams has a key role as the sub-stream bandwidth should be less than the channel coherence bandwidth, or equivalently the symbol time on each sub-stream should be much greater than the delay spread of the channel. Then each sub-stream will not experience significant ISI. For a system with data rate R and bandwidth B , if the coherence bandwidth is assumed to be $B_c < B$, the signal experiences frequency-selective fading. With N parallel subsystem, each has a sub-channel bandwidth of $B_N = B/N$ and data rate $R_N \approx R/N$. For sufficiently large N , the sub-channel bandwidth $B_N = B/N \ll B_c$, which results in relatively flat fading on each sub-channel. On the other hand in the time domain, the symbol time T_N of the modulated signal in each sub-channel is proportional to the sub-channel bandwidth $1/B_N$, so $B_N \ll B_c$ implies that $T_N \approx 1/B_N \gg 1/B_c \approx T_m$ where T_m denotes the delay spread of the channel. As a consequence, if N is sufficiently large, the sub-channel symbol time will be much greater than the delay spread, and therefore each sub-channel experiences minimal ISI degradation.

Fig. 2.5 illustrates a multicarrier transmitter. If we assume raised cosine pulses for $g(t)$, we have a symbol time $T_N = (1 + \beta)/B_N$ for each sub-stream, where β is the roll-off factor of the pulse shape. In accordance with Fig. 2.5, the modulated

²The difference between the transmitter and receiver reference frequencies is called frequency offset.

³The time variation of a characteristic of a periodic signal in compare with a reference signal is known as jitter.

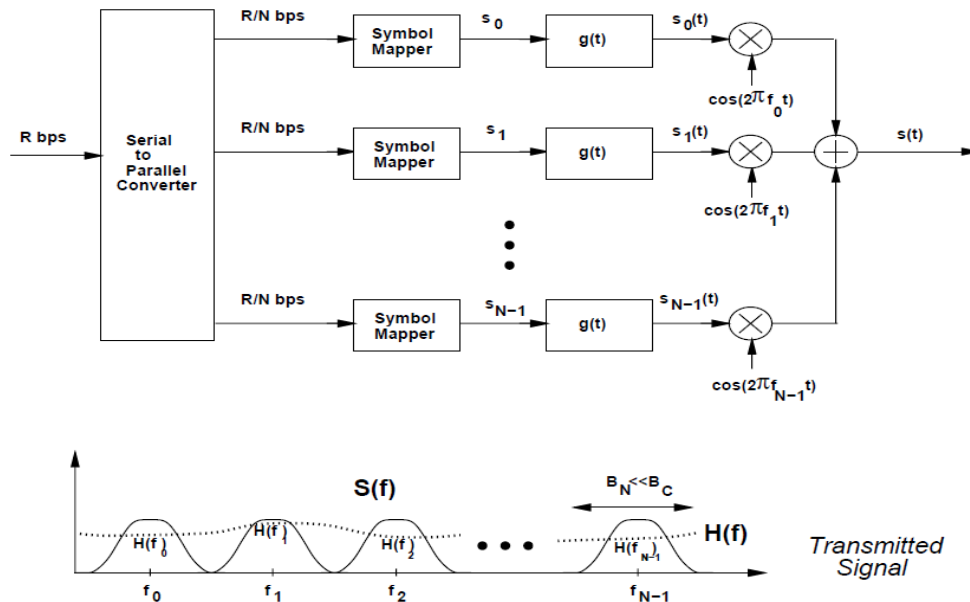


Figure 2.5: The multicarrier transmitter [3].

signals for the sub-channels are added to form the transmitted signal. The bandwidth for each sub-stream is B_N , giving a total bandwidth $N \times B_N = B$ and data rate $N \times R_N \approx R$. Based on Fig. 2.6, at the receiver narrowband filters separate the sub-streams. To obtain the original data stream, the outputs are passed through a parallel-to-serial converter. Although this simple method of multicarrier modulation is straightforward to understand, it has several significant drawbacks. One of the critical problems of implementing this technique is that there is no unlimited-time signal in practical situation. Thus in a realistic implementation, the sub-channels occupy a larger bandwidth than under ideal raised cosine pulse shaping. If we assume the additional required bandwidth due to time-limiting the signal is ε/T_N , then to prevent overlapping sub-channels there must be a frequency spacing of $(1 + \beta + \varepsilon)/T_N$ between the sub-channels so the total bandwidth for non overlapping sub-channels is

$$B = \frac{N(1 + \beta + \varepsilon)}{T_N} \quad (2.1)$$

Therefore, this form of multicarrier modulation is not only spectrally inefficient but also requires near-ideal and hence expensive low pass filters to maintain the orthogonality of the subcarriers at the receiver. The only reason this elementary scheme was mentioned was to explain the concept of OFDM and the effect on frequency selective fading.

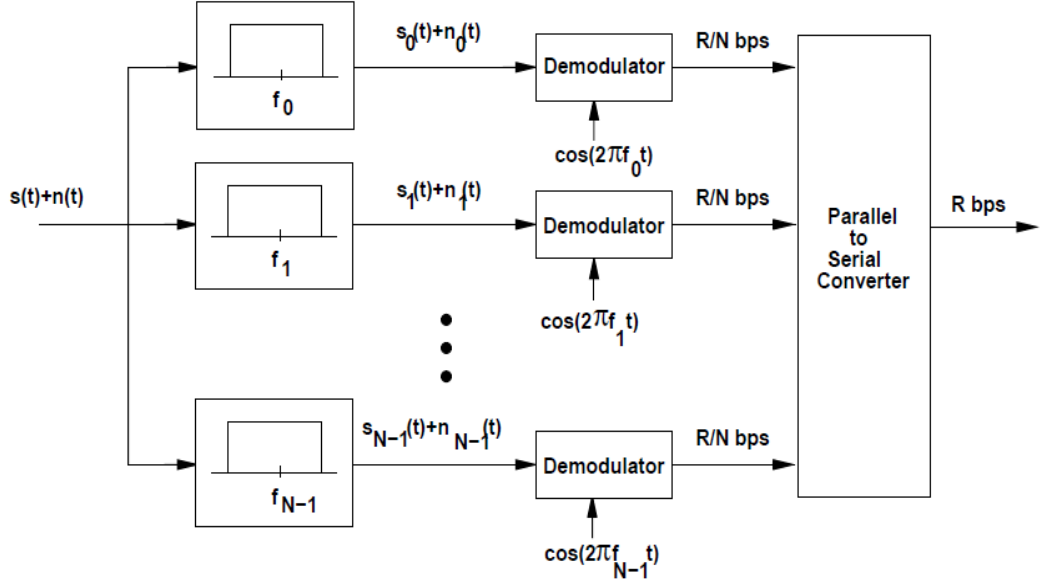


Figure 2.6: The multicarrier receiver [3].

2.6.2 Overlapping Sub-channels in Multicarrier Modulation

To have better spectral efficiency using multicarrier modulation, the sub-channels can be overlapped. In addition, the need for precise filtering should be eliminated. The subcarriers must still be orthogonal so that they can be separated by the demodulator in the receiver. The set of frequencies $\{\cos(2\pi(f_0 + i/T_N)t + \phi_i), i = 0, 1, 2, \dots\}$ approximately form a set of orthogonal basis functions over the interval of $[0, T_N]$ for any set of subcarriers with phase offsets of ϕ_i . The following calculation proves the orthogonality of this set

$$\begin{aligned}
 & \int_0^{T_N} \cos(2\pi(f_0 + i/T_N)t + \phi_i) \cos(2\pi(f_0 + j/T_N)t + \phi_j) dt \\
 &= \int_0^{T_N} 0.5 \cos(2\pi(i - j)t/T_N + \phi_i - \phi_j) dt + \int_0^{T_N} 0.5 \cos(2\pi(2f_0 + i + j)t/T_N + \phi_i + \phi_j) dt \\
 &\approx \int_0^{T_N} 0.5 \cos(2\pi(i - j)t/T_N + \phi_i - \phi_j) dt \\
 &= 0.5T_N \delta(i - j)
 \end{aligned}$$

In addition, it is shown in [3] that the minimum frequency separation required for the subcarriers to be orthogonal over the symbol interval $[0, T_N]$ is $1/T_N$.

Similar to the previous discussion, if we assume each sub-channel is modulated

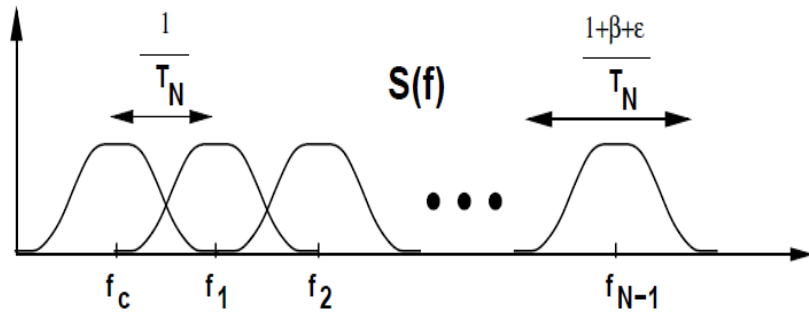


Figure 2.7: Multicarrier modulation with overlapping subcarriers [3].

using a raised cosine pulse shapes with roll off factor β , the total system bandwidth with overlapping sub-channels decreases considerably since β and ϵ affect the total system bandwidth only in the first and last sub-channels. Thus the total required bandwidth with this method is

$$B = \frac{(N + \beta + \epsilon)}{T_N} \approx \frac{N}{T_N} \quad (2.2)$$

As a consequence, with N large, the effect of β and ϵ on the total system bandwidth is insignificant and can be neglected, in contrast with the required bandwidth of $B = N(1 + \beta + \epsilon)/T_N$ when the sub-channels are not overlapped.

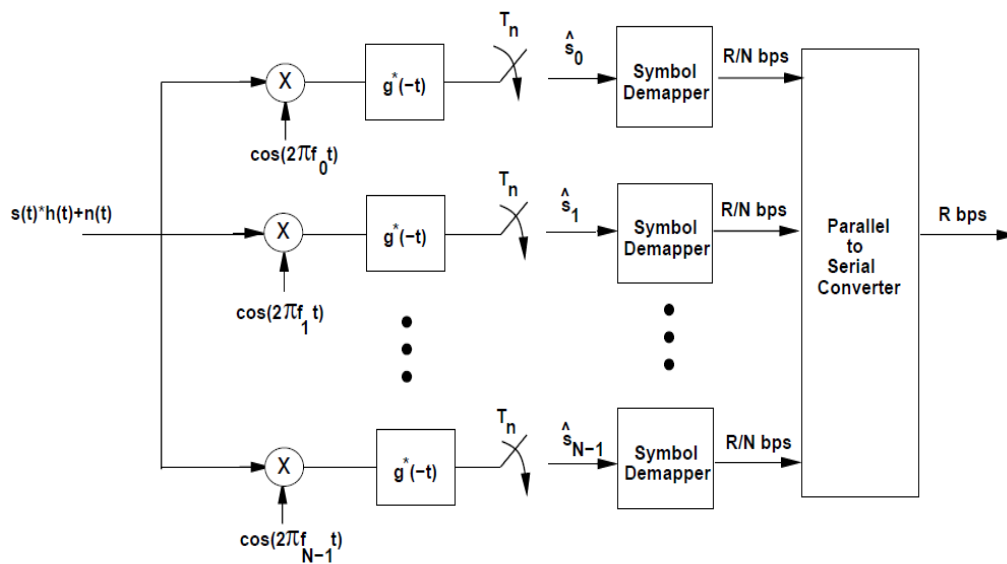


Figure 2.8: The multicarrier receiver for overlapping subcarriers [3].

The multicarrier receiver structure for overlapping sub-channels is shown in Fig. 2.8.

If the effects of the channel impulse response $h(t)$ and additive noise $n(t)$ are neglected, for a transmitted signal $s(t)$, the input to each symbol demapper in Fig. 2.8 is

$$\begin{aligned}
\hat{s}_i &= \int_0^{T_N} \left(\sum_{j=0}^{N-1} s_j g(t) \cos(2\pi f_j t + \phi_j) \right) g(t) \cos(2\pi f_i t + \phi_i) dt \\
&= \sum_{j=0}^{N-1} s_j \int_0^{T_N} g^2(t) \cos(2\pi(f_0 + j/T_N)t + \phi_j) \cos(2\pi(f_0 + i/T_N)t + \phi_i) \\
&= \sum_{j=0}^{N-1} s_j \delta(j - i) \\
&= s_i
\end{aligned}$$

On the other hand, if the channel and noise effects are included, the symbol in the i th sub-channel is scaled by the channel gain $\alpha_i = |H(f_i)|$ and corrupted by the noise sample, so $\hat{s}_i = \alpha_i s_i + n_i$, where n_i is additive white Gaussian noise (AWGN) with power $N_0 B_N$.

2.6.3 Discrete Fourier Transform (DFT) and Inverse DFT (IDFT)

The requirement to have separate modulators and demodulators on each sub-channel, as explained in the preceding sections, makes this modulation technique complex and expensive. In order to solve this problem, in the 1970's the Discrete Fourier Transform (DFT) and Inverse DFT (IDFT) were employed to simplify OFDM modulation. The details of the DFT and IDFT are beyond the scope of this thesis, and the following equations show mathematically how they are used to obtain the desired signal.

$$DFT\{x[n]\} = X[i] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi n i}{N}} \quad 0 \leq i \leq N-1 \quad (2.3)$$

$$IDFT\{X[i]\} = x[n] = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} X[i] e^{j \frac{2\pi n i}{N}} \quad 0 \leq n \leq N-1 \quad (2.4)$$

2.6.4 Cyclic Prefix

Consider a channel input sequence $x[n] = x[0], \dots, x[N-1]$ of length N and a channel impulse response $h[n] = h[0], \dots, h[\mu]$ of length $\mu+1 = T_m/T$, where T is the sampling

time which is equal to $\frac{1}{10MHz}$ in DSRC systems and T_m is the channel delay spread. The cyclic prefix typically employed for $x[n]$ is $\{x[N - \mu], \dots, x[N - 1]\}$, that is, it consists of the last μ values of the $x[n]$. For each input sequence of length N , these last μ samples are appended to the beginning of the sequence. This yields a new sequence $x[n]$, $-\mu \leq n \leq N - 1$, of length $N + \mu$, where $x[-\mu], \dots, x[N - 1] = x[N - \mu], \dots, x[N - 1], x[0], \dots, x[N - 1]$, as shown in Fig. 2.9. If the input $x[n]$ is

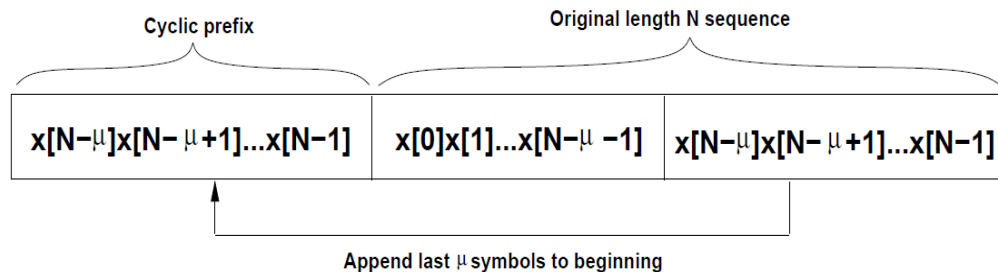


Figure 2.9: A cyclic prefix of length μ [3].

divided into blocks of size N , then with a cyclic prefix appended to each block the first μ samples of the channel output, $y[n]$, in a given block are corrupted by ISI as illustrated in Fig. 2.10. However, since $y[n]$ has length $N + \mu$, there is no need to recover the first μ samples in order to obtain $x[n]$, $0 \leq n \leq N - 1$, due to the redundancy. As a result, the cyclic prefix leads to elimination of the ISI between the

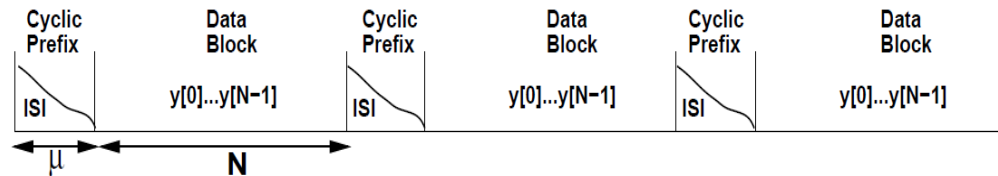


Figure 2.10: The ISI between data blocks at the channel output [3].

data blocks without any information loss. The cost of adding the cyclic prefix is a reduction of the data rate due to the μ redundant symbols. This can also be considered as wasted transmit power. If the prefix consists of all zero symbols, no power is used for transmission although data rate is still reduced by a factor of $N/(N + \mu)$. In addition, in this case the orthogonality of the sub-carriers is not preserved.

2.6.5 OFDM in DSRC

In DSRC systems, $N = 64$ subcarriers are employed, although only 48 are actually used for data transmission, with the outer 12 set to zero in order to reduce adjacent channel interference⁴, and 4 subcarriers are used as pilot symbols for channel estimation. In this thesis, the channel estimation is not considered since we assume the transmitter and receiver are synchronized.

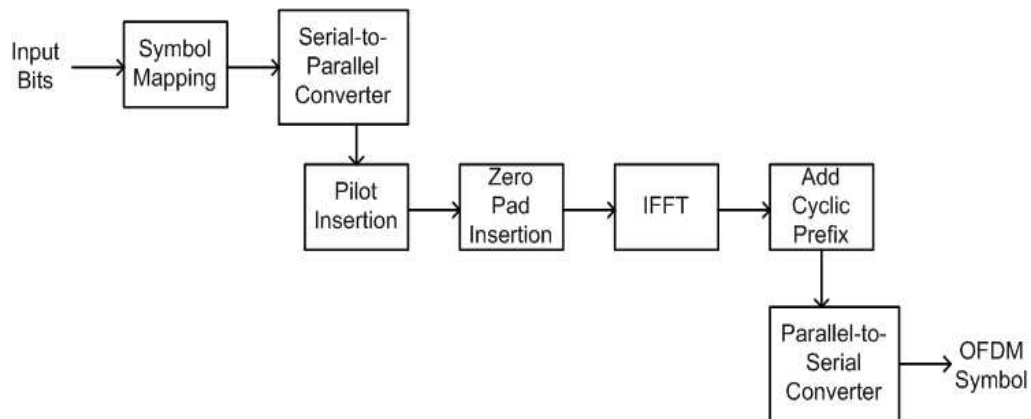


Figure 2.11: The DSRC transmitter block diagram.

The cyclic prefix consists of $\mu = 16$ samples, as a result the total number of samples associated with each OFDM symbol, including both data and the cyclic prefix, is 80. The same FEC and modulation are used for all the subcarriers at any given time, although BPSK, QPSK, 16QAM, or 64QAM modulation can be employed in a DSRC system. The modulation used in this thesis is BPSK as the performance under different modulation is not the main concern. The goal is to examine the performance of DSRC systems with different error correction coding including convolutional, LDPC and QC-LDPC codes. Convolutional coding is used in conventional DSRC systems with one of three possible coding rates: $R_c = 1/2$, $2/3$, or $3/4$. Since the bandwidth B (and sampling rate $1/T$), in DSRC systems is $10MHz$ and there are 64 subcarriers evenly spaced over this bandwidth, the subcarrier bandwidth is

$$B_N = \frac{10MHz}{64} = 156.25KHz \quad (2.5)$$

Since $\mu = 16$ and $T = \frac{1}{10MHz}$, the maximum delay spread in DSRC systems for which

⁴Adjacent channel interference (ACI) is extraneous power from a signal in an neighboring channel.

the ISI is removed is

$$T_m < \mu T = \frac{16}{10MHz} = 1.6\mu sec \quad (2.6)$$

Including both the OFDM symbol and cyclic prefix, there are $80 = 64 + 16$ samples per OFDM symbol time, so the symbol time per subchannel is

$$T_N = 80T = \frac{80}{10MHz} = 8\mu sec \quad (2.7)$$

In entire of this thesis, OFDM symbol duration is denoted as $T_s = 8\mu sec$ while the sampling time is shown by $T = \frac{1}{10MHz}$ considering DSRC physical layer parameters definition illustrated in the Table 2.1. The data rate per subchannel is $\log_2 M/T_N$ where $M = 2$ and $M = 6$ for BPSK and 16-QAM respectively. The minimum data rate for this system with BPSK modulation and $R_c = 1/2$, and taking into account that only 48 subcarriers actually carry usable data, is

$$R_{min} = 48 \times \frac{1}{2} \times \frac{1}{8 \times 10^{-6}} = 3Mbps \quad (2.8)$$

The maximum data rate corresponding to $R_c = 3/4$ and 64-QAM is

$$R_{max} = 48 \times \frac{3}{4} \times \frac{6}{8 \times 10^{-6}} = 27Mbps \quad (2.9)$$

These minimum and maximum data rates are the same as the DSRC physical layer parameters given in Table 2.1. In this thesis, we use $R_{min} = 3Mbps$ to analyze DSRC system performance which corresponds to BPSK modulation. The DSRC transmitter and receiver are shown in Figs. 2.11 and 2.12, respectively.

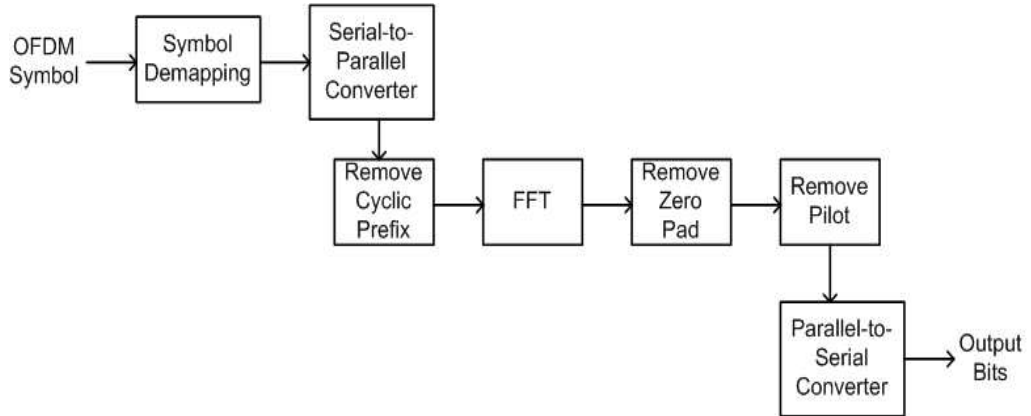


Figure 2.12: The DSRC receiver block diagram.

The effect of ISI on the DSRC systems is discussed in chapter 6.

Chapter 3

Channel Model

The unknown nature of obstructions in wireless transmissions introduces significant challenges for establishing reliable communications because of the existence of path loss, shadowing and multipath fading. Path loss is defined as the power dissipation due to the propagation channel from the transmitter. In general, path loss is considered the same at a given distance no matter what the characteristics of the path is. Therefore in modelling path loss, shadowing and fading are not concerned. Shadowing is an attenuation of the transmitted signal due to blockages such as buildings, trees, cars, etc. It is mainly dependant on the reflecting surfaces and scattering objects in the path of the transmitted signal, and can have a substantial effect on the received signal power.

From experimental results and observation, path loss and shadowing are large-scale propagation effects because they occur over large distances, i.e., 10-1000 meters. In comparison, multipath fading is specified over very short distances, i.e., on the order of the signal wavelength, so it is recognized as small-scale propagation effects. If a single pulse is sent from the transmitter over a multipath channel, the received signal will be a pulse train corresponding to a number of distinct multipath components, which may include a line of sight component. Thus, the time delay spread is an important attribute of multipath fading channels. This has a key role in modelling fading channels, and is defined as the time delay between the first received signal component, generally due to the line of sight (LOS) signal, and the last significant signal component received. As a result, if the delay spread is small in comparison with the inverse of the signal bandwidth, there will be little time spread in the received signal. On the other hand, if the delay spread is comparatively large, then there will be a significant time spread of the received signal which may cause considerable

signal distortion. The time-varying nature of the multipath channel is caused by movement of the transmitter or receiver, or reflecting objects between them. This causes changes in the transmission path over time, so the receiver will observe changes in the amplitudes, delays, and number of multipath components corresponding to the transmitted pulses.

In this chapter, we look at statistical models for the fading due to multipath components at the receiver. Three important multipath fading models are considered, namely Rician, Rayleigh and Nakagami. In this thesis, we will consider only Rician and Rayleigh multipath fading models to specify the DSRC channel, as they have been shown to provide an accurate characterization of the actual channels encountered. Fig. 3.1 illustrates the ratio of the received to transmitted power in dB versus the

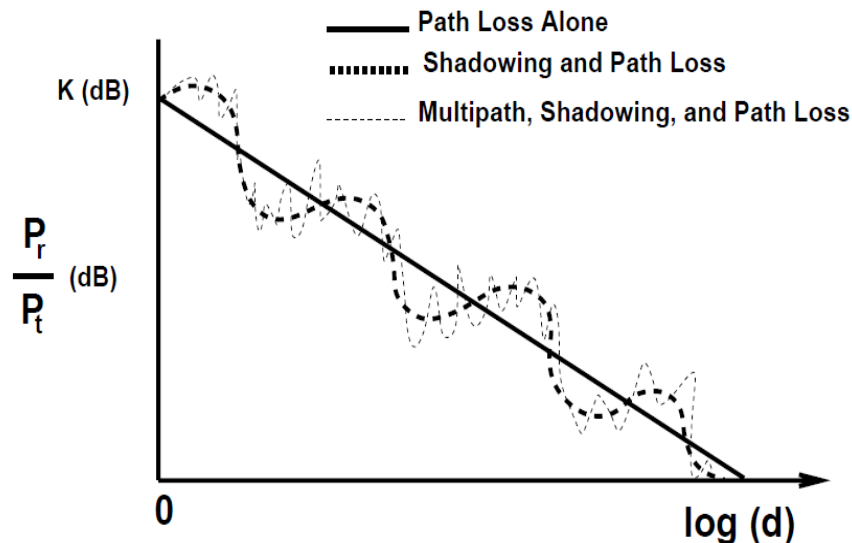


Figure 3.1: Combined path loss, shadowing, and narrowband fading obtained based on an empirical observation [3].

log of the distance. The combined effects of path loss, shadowing, and multipath fading are shown. Based on this diagram, we can conclude that path loss alone is the mean of the shadowing variations, while the combined shadowing and path loss is the average of the multipath fading variations.

Before proceeding to the next section, some terms are defined. A line of sight (LOS) signal is a direct signal received from the transmitter without any obstacles in the way. A reflected signal is a transmitted signal that is reflected by an obstacle, such as buildings or the terrain. A diffracted signal is the result of a signal striking object

edges. Scattering happens when objects in the signal path have dimension smaller than the wavelength. Fig. 3.2 illustrates these three types of signal propagation through a wireless channel.

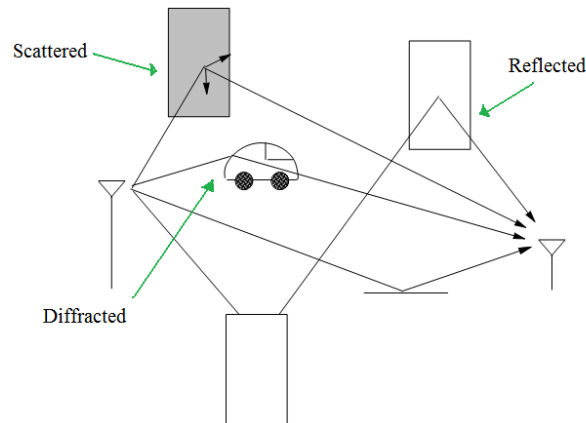


Figure 3.2: The reflected, diffracted, and scattered wave components [3].

3.1 Small-Scale Fading

In wireless communication channels, the main source of multipath signals is the existence of reflectors and scatterers which block the path between the transmitter and the receiver. Thus, multiple copies of the transmitted signal, each traversing a different path with different amplitude, phase and delay, are received, which leads to constructive and destructive interference. This can amplify or attenuate the signal power. The result is a possibility of failure of the communication link between the transmitter and receiver due to deep fades caused by strong destructive interference, with a corresponding severe drop in the signal-to-noise ratio (SNR).

3.1.1 Factors that Influence Small-Scale Fading

The main factor that influences small scale fading is multipath propagation, caused by reflecting objects and scatterers. There are other aspects besides the multipath propagation that should be taken into consideration, such as:

- Speed of the Mobile Transmitter or Receiver: The relative velocity between the transmitter and receiver results in a frequency shift, called the Doppler

frequency, of the multipath components. This can have a significant effect on the signals of outdoor wireless mobile channel.

- **Speed of Surrounding Objects:** If objects in the path of the signal are moving, there will be a time varying Doppler depending on the object speed. If this speed is greater than the transmitter or receiver speed then this effect dominates the small scale fading, otherwise, this can be ignored.
- **Signal Transmission Bandwidth:** The received signal experiences noticeable distortion if the transmitted signal bandwidth is greater than the bandwidth of the multipath channel. On the other hand if the transmitted signal has a narrow bandwidth compared to the channel, the amplitude of the signal will change rapidly, but the signal will not be distorted.

We can classify multipath fading into four groups, slow fading, and fast fading, narrowband or flat fading, and wideband or frequency selective fading. In the following sections, we briefly explain the characteristics of these fading categories.

3.1.2 Slow and Fast Fading

Slow and fast fading are caused by the time varying nature of multipath fading channels. The difference between slow and fast fading is a key aspect of statistical models of these channels, and consequently the performance evaluation of communication systems operating over wireless channels. This concept is associated with the coherence time T_c of the channel, which is a measure of the expected time duration over which the channel response is essentially constant. It is inversely proportional to the Doppler spread of the channel, within a multiplicative constant. Furthermore, the coherence time T_c is an estimate of when the correlation of the channel response at the same frequency but different time instants, is lower than a certain threshold. With this concise introduction of coherence time, the channel fading is assumed to be slow if the symbol duration, T_s , is much less than the channel coherence time, T_c , or equivalently the Doppler bandwidth, f_D , is much less than signal bandwidth, $1/T_s$. Alternatively the fading is fast if the symbol duration is equal or greater than the channel coherence time, or equivalently the Doppler bandwidth, f_D , is greater than the signal bandwidth, $1/T_s$. In slow fading, a particular fade level will involve many successive symbols, which causes burst errors, whereas in fast fading the fading is independent from symbol to symbol.

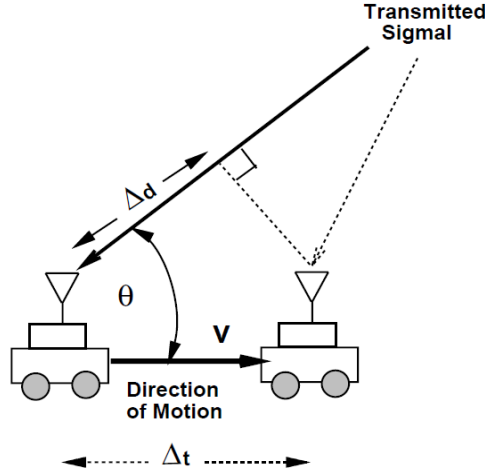


Figure 3.3: Geometry associated with the Doppler shift [3].

As stated above, Doppler is a frequency shift in the received signal which depends on the relative velocity of the vehicles, wavelength of the signal, and the angle of arrival of the received signal. According to the DSRC standard, all devices operate at a frequency of 5.9 GHz, so the Doppler frequency is given by

$$f_D = v \times \frac{\cos \theta}{\lambda} \quad \text{for DSRC} \rightarrow \text{systems} \quad f_D = v \times \frac{\cos \theta}{\frac{3 \times 10^8}{5.9 \times 10^9}} \quad (3.1)$$

3.1.3 Frequency-Flat and Frequency-Selective Fading

A mobile wireless channel can also be classified as narrowband (flat fading), or wideband (frequency selective fading), according to the relationship between the bandwidth of the transmitted signal and the frequency response of the channel. In flat fading, the coherence bandwidth of the channel is larger than the bandwidth of the signal; but in frequency selective fading, the coherence bandwidth of the channel is smaller than that of the signal. The coherence bandwidth is defined as the bandwidth such that the frequency response remains approximately constant, and can be approximated as $B_c \approx 0.2/\sigma_T$. σ_T is defined as the Root-Mean Square (RMS) delay spread which can be obtained from the delay profile of the channel. The coherence bandwidth and coherence time illustrate the same features of the channel response, but in the frequency and time domains, respectively.

When the signal is wideband, distortion appears due to the multipath delay spread. As a consequence, the duration of the received signal may increase con-

siderably, because if we transmit a short pulse signal of duration T , the received signal will have a duration of $T + T_m$, where T_m is the multipath delay spread defined as the time between the first received pulse (generally a LOS signal) and the last received pulse due to multiple transmission paths (an NLOS signal). This is illustrated in Fig. 3.4, where a pulse with time duration T is transmitted over a multipath fading channel. As shown in the upper right of Fig. 3.4, the multipath components

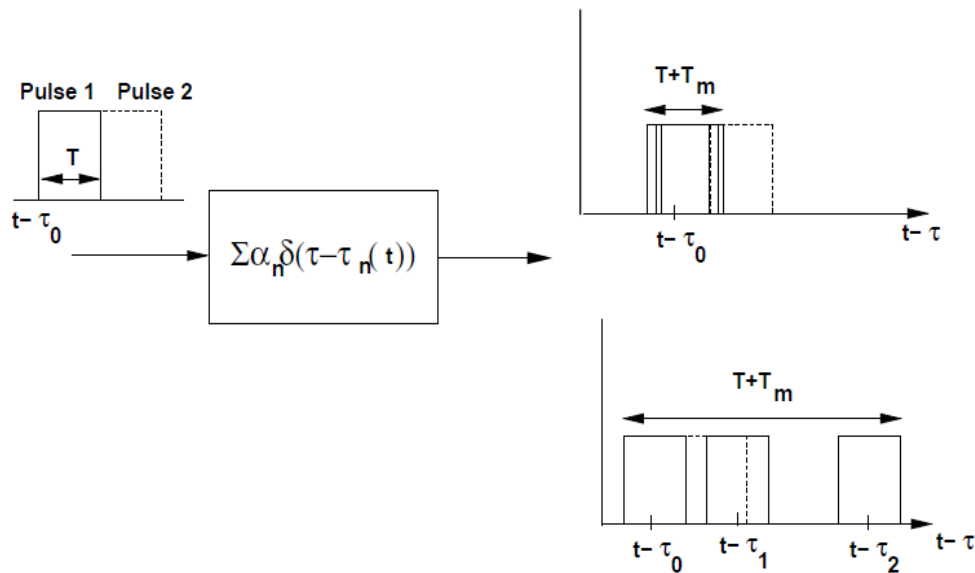


Figure 3.4: Multipath signal reception in frequency-flat or frequency-selective fading [3].

are received on top of each other. This occurs when the multipath delay spread is much less than the signal duration, $T_m \ll T$. In this case, the resulting interference leads to narrowband fading, in which case the interference between two transmitted signals can be ignored. Conversely, if the multipath delay spread is larger than the signal duration, $T_m \gg T$, then the multipath components for a single symbol can be resolved, as shown in the lower right of Fig. 3.4; however, unlike the previous situation, the multipath components may interfere with subsequent transmitted pulses leading to what is called inter symbol interference (ISI). This is shown by the dotted symbol in the figure, which is the next one to be transmitted.

Equalization, multicarrier modulation such as OFDM, and spread spectrum modulation are among the numerous methods used to mitigate the signal distortion due to multipath delay spread.

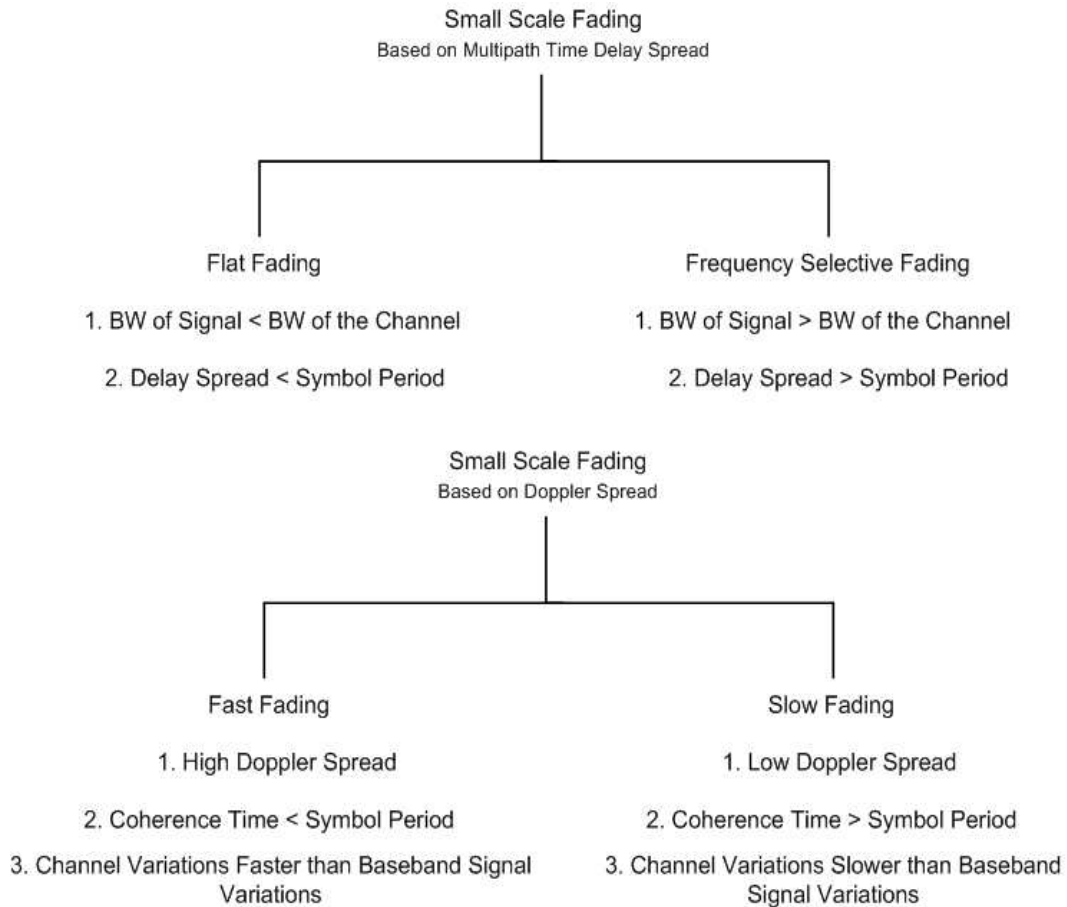


Figure 3.5: Small-scale fading categories [7].

3.1.4 Modelling a Flat Fading Channel

Multipath fading channel models depend on the nature of the randomly delayed, reflected, diffracted, and scattered signal components, which depend in turn on the characteristics of the environment. In wireless communication, the channel is mainly considered as the fast fading consequently responsible for the short-term signal variations. Among the multipath fading channels models, Rayleigh, Rician and Nakagami models are well known and widely employed.

In wireless outdoor applications, if there are a large number of reflecting objects, there is a great possibility of having just Non-Line of Sight (NLOS) received signals, especially if the distance between the transmitter and receiver is large. In this case, the small-scale fading is called Rayleigh fading, as the envelope of the superimposed received signals can be described statistically by a Rayleigh distribution. If a Line of Sight (LOS) signal also exists, the small scale fading envelope will have a Rician

distribution, so it is called Rician fading. In the following, the characteristics of these fading channels are presented.

Rayleigh Model

With a Rayleigh fading model it is assumed that the magnitude of the received signal will vary randomly, or fade, according to a Rayleigh distribution. This model is most applicable when there is no line of sight path between the transmitter and receiver. The complex low pass received signal consists of in-phase and quadrature components given by

$$r_{LP} = r_I(t) + jr_Q(t) \quad \theta = \arctan\left(\frac{r_Q(t)}{r_I(t)}\right) \quad (3.2)$$

where $r_I(t)$ and $r_Q(t)$ are uncorrelated zero-mean Gaussian random variables. Thus θ is a uniformly distributed random variable. In this case, the channel amplitude R is the radial component of two uncorrelated Gaussian random variables addition and so has a Rayleigh distribution

$$P_R(r) = \frac{2r}{\Omega} e^{-r^2/\Omega} \quad r \geq 0 \quad (3.3)$$

where $\Omega = E(R^2)$ is the average received power.

Rician Model

If the channel has a LOS component then $r_I(t)$ and $r_Q(t)$ in (3.2) are not zero-mean. In this case, the received signal is the superposition of complex Gaussian components and a LOS component, so that the signal envelope has a Rician distribution.

$$P_R(r) = \frac{2r}{\Omega} \exp\left[-\frac{(r^2 + s^2)}{\Omega}\right] I_0\left(\frac{2rs}{\Omega}\right) \quad r \geq 0 \quad (3.4)$$

where $\Omega = E(R^2)$ is the average power of the non-LOS component, and $s^2 = \alpha_0^2$ is the power of the LOS component. The function I_0 is the zero-th order modified Bessel function. The average received power in Rician fading is given by

$$P_{average \text{ received power}} = \int_0^\infty r^2 P_R(r) dr = s^2 + \Omega \quad (3.5)$$

The Rician distribution is often described in terms of a fading parameter K , defined as

$$K = \frac{s^2}{\Omega} \quad (3.6)$$

Thus, K is the ratio of the LOS component power to the NLOS power. $K = 0$ corresponds to not having LOS component, which is the Rayleigh fading model, whereas $K = \infty$ corresponds to no fading. The fading parameter K is therefore a measure of the severity of the fading. A small K implies severe fading, while a large K implies mild fading.

Nakagami Model

The Nakagami distribution is a more general fading channel model that can be used to represent Rician, Rayleigh, and even fading which is more severe than Rayleigh, depending on the model parameters. Thus this distribution is capable of describing a wide range of fading situations. The Nakagami distribution is given by

$$P_R(r; \mu, \omega) = \frac{2\mu^\mu r^{2\mu-1}}{\Gamma(\mu)\omega^\mu} e^{-\frac{\mu r^2}{\omega}} \quad (3.7)$$

where μ is a shape parameter, $\Gamma(\cdot)$ is the Gamma function, and $\omega = E(R^2)$ is an estimate of the average power in the fading envelope. When $\mu = 1$ the Nakagami distribution is the Rayleigh distribution.

3.2 The DSRC Channel

In recent years, the safety applications of vehicle-to-vehicle communications has made wireless inter-vehicle communications studies a significant research field. Specifically, Dedicated Short Range Communications (DSRC) [8] is a standard developed to support the design of wireless devices for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. Navigational maps and roadway toll payments are considered to be the major DSRC applications. To date, based on the literature available, a precise understanding of mobile-to-mobile channels modeling in vehicular environments is lacking due to the many different scenarios that can be defined. Even though prototypes of DSRC devices are already available, there is no widely accepted channel model for vehicular systems. DSRC channel modeling consists of two important components, large scale path loss and small scale fading. The former

is used to determine the average received signal strength at a particular distance from the transmitter as shown in the Fig. 3.1, whereas small scale fading generally involves the detailed modeling of multipath fading statistics, power delay profile, and Doppler spectrum.

In this section, we introduce the model of a DSRC channel based on the investigation carried out in [5]. Their channel model is based on empirical measurements using devices compatible with the defined DSRC standard. The Nakagami distribution was used as the default statistical channel model to analyze the received signal strength (RSS) from empirical measurements. It was chosen because it can be used to model a wide-range of fading channels properties. A Kolmogorov-Smirnov (K-S) test was used to validate the assumption that the Nakagami distribution is a good fit to the measured RSS values. This test simply compare the measured values with a reference probability distribution. While the empirical fading distribution can be approximated by a Nakagami distribution, their results show that the fading can also be approximated by a Rician distribution within a range of 100 m , and by a Rayleigh distribution beyond this distance. The advantage is that these latter models are much simpler, and so these will be employed in the remainder of the thesis.

3.2.1 The DSRC Channel Model

The experiments in [5] used 200 byte data packets transmitted every 100 ms between pairs of moving in the same direction vehicles on a freeway near Detroit. These results show that the relative speed between two vehicles is typically below 10 $mi/h = 16 km/h$, so the associated Doppler spread of the channel is almost 100 Hz . The corresponding channel coherence time is around 10 ms , so it can be assumed that the channel fades independently for time durations greater than 10 ms . In view of the fact that packets are sent at intervals of 100 ms , the channel sampling period is much larger than the channel coherence time, so the fading can be considered to be independent.

Chapter 4

Convolutional Coding

Convolutional coding is employed for the error control coding in the current DSRC physical layer structure. The main difference between linear block coding (including LDPC codes) and non-block coding (including convolutional codes) is that in the former, the encoder output is in block form while in the latter the encoder output is in sequence form generated from the entire input information sequence. A convolutional encoder has memory, so that at a given time, the encoder output is a function of the input at that time and also a number of previous inputs. This is the fundamental difference between convolutional and LDPC codes.

Similar to other coding techniques, in convolutional coding, a given message sequence creates a distinct encoded sequence so that the decoder at the receiver can use the received sequence to extract the message. A convolutional encoder can have a Finite Impulse Response (FIR) or Infinite Impulse Response (IIR) structure. An IIR encoder contains feedback, whereas an FIR encoder has a feed-forward structure. In this thesis we consider an FIR convolutional encoder as defined in the current DSRC standard.

A convolutional encoder with rate of $R = k/n$ and memory order of K can be considered as a k -input, n -output linear sequential circuit with memory K . This means that an input stays in the encoder for an additional K time units after first entering the circuit. A convolutional code with parameters n , k and K will be denoted as $C_{conv}(n, k, K)$. With convolutional codes, increasing K typically leads to better error correction (and thus performance), but also greater encoder and decoder complexity.

Convolutional codes were originally introduced by Elias in 1955 [9] as an alternative to block codes. Shortly afterward, Wozencraft and Reiffen proposed sequential decoding as an efficient method to decode convolutional codes with large constraint

lengths. A less efficient but simpler decoding method called threshold decoding was proposed by Massey in 1963. This accomplishment spawned a number of practical applications of convolutional codes, specifically in digital transmission over telephone, satellite, and radio channels. A maximum likelihood (ML) decoding algorithm was proposed by Viterbi. This enabled the implementation of soft decision decoding for convolutional codes with small constraint lengths, and as a consequence convolutional coding was extensively deployed in satellite communication systems in the 1970s. In 1976, Ungerboeck and Csajka introduced Trellis Coded Modulation (TCM), a technique which combines modulation with convolutional coding. Because of these developments, convolutional codes now play a significant role in wireless communication systems, allowing for high speed data transmission even over bandwidth limited channels.

4.1 Convolutional Encoder

With convolutional codes, the encoder receives a k -tuple m_i of message elements as input and generates an n -tuple c_i of coded elements as output, at a given time i . This output is a function of the input m_i and the K preceding inputs m_j . The

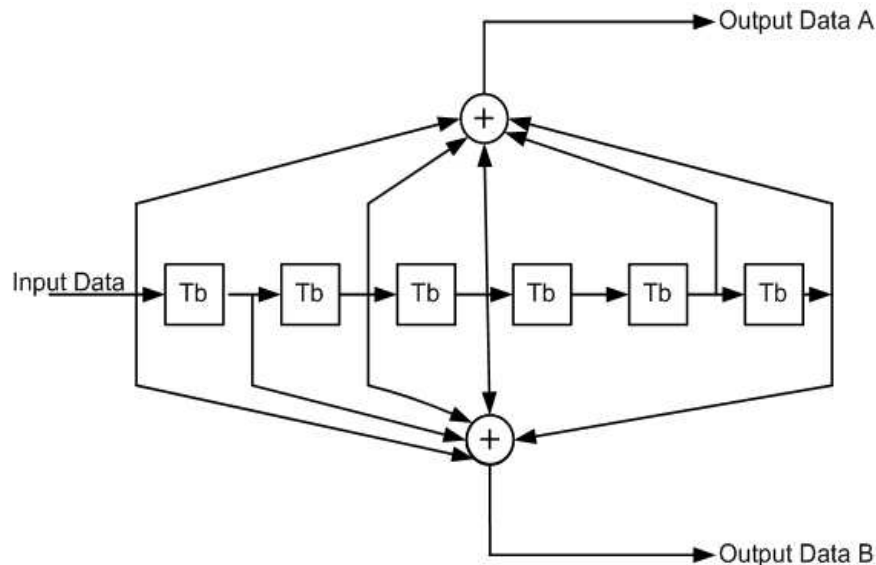


Figure 4.1: The convolutional encoder used in DSRC systems according to the IEEE 802.11a standard.

convolutional code structure for DSRC systems is shown in Fig. 4.1. The code rate

for this code, defined as the ratio the number of input elements k to the number of output elements n , is $R_c = 1/2$.

The convolutional encoder in Fig. 4.1 is a Finite State Sequential Machine (FSSM) that operates over the binary field $GF(2)$ where the input message k -tuple is simply one bit, m , and at each time step i the encoder generates an output of two bits $c_i^{(1)}$ and $c_i^{(2)}$. The input sequence $m_i = (m_0, m_1, m_2, \dots)$ creates two output sequences $c^{(1)} = (c_0^{(1)}, c_1^{(1)}, c_2^{(1)}, \dots)$ and $c^{(2)} = (c_0^{(2)}, c_1^{(2)}, c_2^{(2)}, \dots)$ which are shown as Output Data A and Output Data B in Fig. 4.1. Theoretically, the two output sequences can be obtained via a convolution operation between the input sequence and the impulse responses of the two encoder outputs. These impulse responses can be obtained using a unit impulse input sequence $m = (1, 0, 0, \dots)$ to get the outputs $c_i^{(1)}$ and $c_i^{(2)}$. Since a convolutional encoder has K memory elements ($K = 6$ in the example), an impulse response can span no more than $K + 1$ time units, so for the example we have

$$g^{(1)} = (g_0^{(1)}, g_1^{(1)}, g_2^{(1)}, \dots, g_K^{(1)}) \quad g^{(2)} = (g_0^{(2)}, g_1^{(2)}, g_2^{(2)}, \dots, g_K^{(2)}) \quad (4.1)$$

The constraint length is defined as $K + 1$, and so is 7 for DSRC systems. It is the maximum number of time units that a given bit of the input sequence can influence the output.

The convolution operation is not practical for generating the output sequences. Instead, the impulse response vectors are used to describe the connections in the encoder structure. If a given bit in the impulse response vector is ‘1’, the corresponding memory element is connected to the output. For the DSRC code, the encoded sequences can be defined via convolution as

$$c^{(1)} = u * g^{(1)} \quad c^{(2)} = u * g^{(2)} \quad (4.2)$$

and from the code structure in Fig. 4.1 the impulse responses are

$$g^{(1)} = (1011011) \quad g^{(2)} = (1111001) \quad (4.3)$$

the output sequences can be obtained by multiplying the generator polynomials with the input sequence polynomial. For the DSRC code this is

$$G(D) = [1 + D^2 + D^3 + D^5 + D^6 \quad 1 + D + D^2 + D^3 + D^6] \quad (4.10)$$

After the last input sequence bit enters the encoder, K zeros must be input to empty the shift registers. The reason for this is to end the encoded sequence in the all-zeros state and produce a codeword from a linear code.

Based on the analysis in [10], for a given convolutional code $C_{conv}(n, k, K)$, the input vector is a sequence of kL information bits. For the DSRC code, $k = 1$, so this amount is just L bits. The encoded sequence contains $N = nL + nK = n(L + K)$ bits. The nK additional bits are used to zero the encoder memory (or shift register). An input of k bits generates an output of n bits, as the rate of a convolutional code $C_{conv}(n, k, K)$ is k/n . However, more precisely, for a given finite input sequence of length L , the corresponding output sequence contains $n(L + K)$ bits. Thus, the code rate is more accurately given by

$$R_c = \frac{kL}{n(L + K)} \xrightarrow{\text{DSRC code } C_{conv}(2,1,6)} R_c = \frac{L}{2(L + 6)} \quad (4.11)$$

This number tends to k/n for a sufficiently large input sequence of length $L \gg K$. This is typical of most coded sequences and so $R_c = k/n$ is used in this thesis.

4.1.2 Convolutional Code Representation

There are three ways to express a convolutional code: tree diagram, state diagram, and trellis diagram. The tree diagram characterizes the encoder in the form of a tree where each branch shows a different encoder state and the corresponding encoder output. The state diagram is a graph showing the different states of the encoder with the possible state transitions and corresponding encoder outputs. The trellis diagram can be defined using the tree representation, but it is simplified by merging nodes in the tree corresponding to the same encoder state. Since the code trellis is used for decoding, it will be explained more clearly via an example.

Consider the convolutional code structure in Fig. 4.2. The corresponding state and trellis diagrams are shown in Figs. 4.3 and 4.4, respectively. The current state of the encoder is defined as the contents of the K stage shift register. The next encoder state can be obtained by shifting the register contents to the right with the input

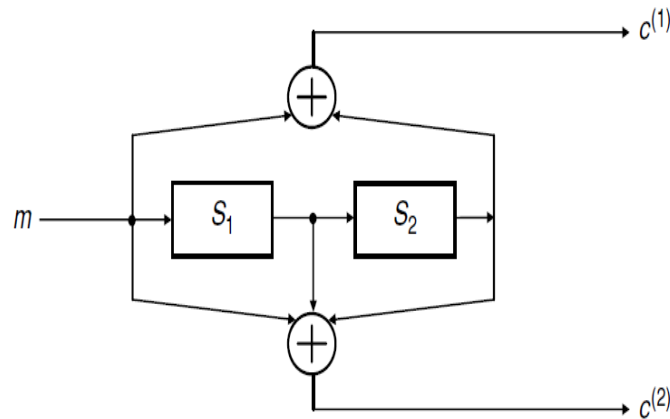


Figure 4.2: Convolutional encoder example with rate 1/2 [10].

placed in the left most position. An appealing feature of representing a convolutional

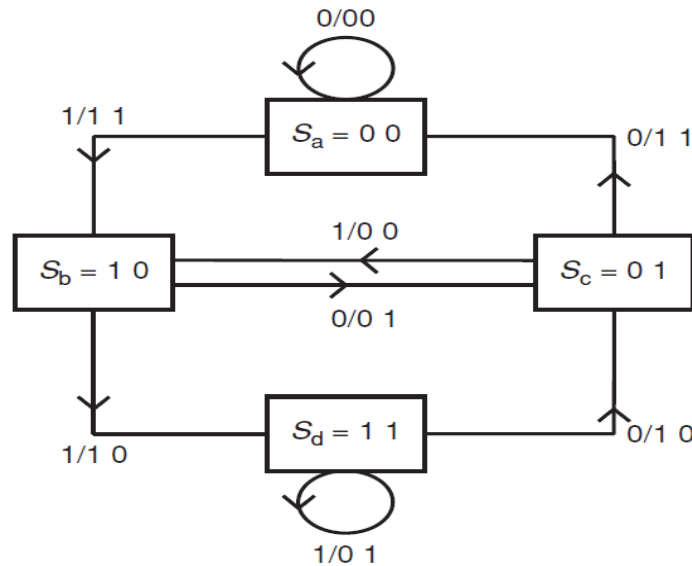


Figure 4.3: The state diagram for the convolutional encoder in Fig. 4.2 [10].

code based on the encoder shift register state is the finite number of states, so that eventually the states must repeat. It is also useful in building the trellis diagram.

4.2 Minimum Free Distance of a Convolutional Code

One of the most important parameters for a convolutional code is the minimum free distance. This indicates the minimum distance between two specific code vectors and

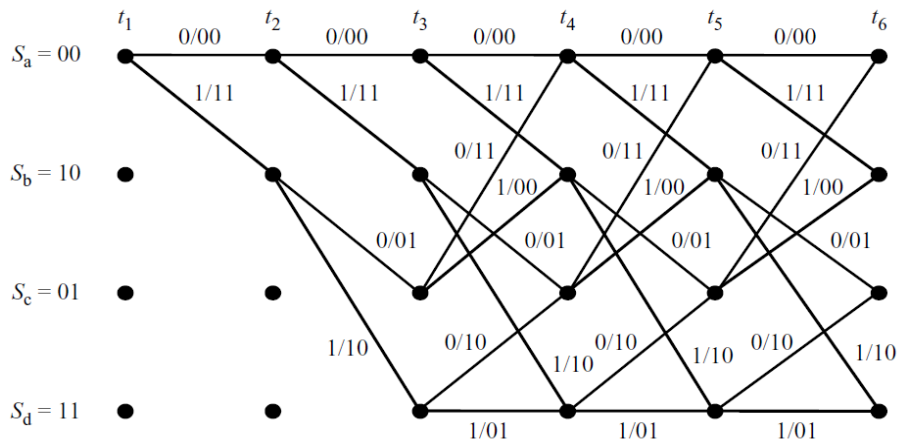


Figure 4.4: The trellis diagram for the convolutional encoder in Fig. 4.2 [10].

is expressed as

$$d_f = \min \{d(c_i, c_j) : m_i \neq m_j\} \quad (4.12)$$

where c_i and c_j are the two code sequences corresponding to the message sequences m_i and m_j , respectively. The term ‘free’ denotes that there is no limitation on the trellis or state diagram path which is chosen to determine the minimum distance. For linear convolutional and block codes, determining the minimum distance between any two code sequences is the same as determining the minimum weight of the non-zero code sequences

$$d_f = \min \{w(c_i \oplus c_j) : m_i \neq m_j\} = \min \{w(c) : m \neq 0\} \quad (4.13)$$

The maximum number of errors that the code can correct is given by

$$t = \left\lfloor \frac{d_f - 1}{2} \right\rfloor \quad (4.14)$$

4.3 Convolutional Decoding: The Viterbi Algorithm

The Viterbi Algorithm (VA) is the most commonly employed decoding technique for convolutional codes, as it is a maximum likelihood algorithm. The main concern in implementing this algorithm is the number of calculations that have to be done to obtain the maximum likelihood codeword. The trellis diagram of a convolutional code

is the foundation of the Viterbi decoding algorithm, and calculating the cumulative distance in the trellis is the critical operation. The cumulative distance (Hamming distance in hard decision decoding (HDD) and Euclidean distance in soft decision decoding (SDD)), is the distance between the received sequence at time t_i at a given state of the trellis, and a code sequence that arrives at the same state at the same instant in time. In order to find the sequence with the minimum cumulative distance, the distance calculations are done for all states in the trellis diagram, and for each state, the sequence with the lowest distance is kept. Once the last state is reached, the final remaining sequence with the minimum cumulative distance is the maximum likelihood sequence. The decoding can be either SDD or HDD. HDD requires that fractional values of the received code words bits be discarded, as decisions about the received bits are made based on a threshold. This is called hard decision decoding because the continuous values of the received sequence are translated into discrete values. On the other hand, an SDD takes advantage of the actual values of the received sequence in decoding, and therefore SDD provides better performance.

In this thesis, SDD is employed for the DSRC convolutional code. Therefore, this decoding technique is explained via an example based on [10]. The convolutional encoder in this example is $C_{conv}(2, 1, 2)$ with the same code rate as the DSRC convolutional code.

Example:

Consider the convolutional encoder illustrated in Fig. 4.2, with the trellis diagram shown in Fig. 4.4. The first step in both SDD and HDD is to find the cumulative distances between the received sequence and the equivalent encoder outputs for each branch in the trellis.

Define:

Message sequence: $m = 10101$

Encoded (transmitted) sequence: $c = (+1 \ +1 \ -1 \ +1 \ +1 \ +1 \ -1 \ -1 \ +1 \ +1)$

Received sequence:

$s_r = (+1.35 \ -0.15 \ -1.25 \ +1.40 \ -0.85 \ -0.10 \ -0.95 \ -1.75 \ +0.5 \ +1.30)$

For hard decision decoding, the threshold is set to zero because the transmitted sequence is in polar format (± 1). After converting from polar format to binary, the input to the trellis decoder is the sequence (10 01 00 00 11). This sequence has

three errors compared to the binary sequence at the encoder output (11 01 11 00 11). As shown in Fig. 4.5, an error event occurs in the trellis decoder at time instant t_4 , resulting in an erroneous codeword at the decoder output. In this figure, the survivor paths (those remaining after a choice is made), are shown in bold. Fig. 4.6 shows the

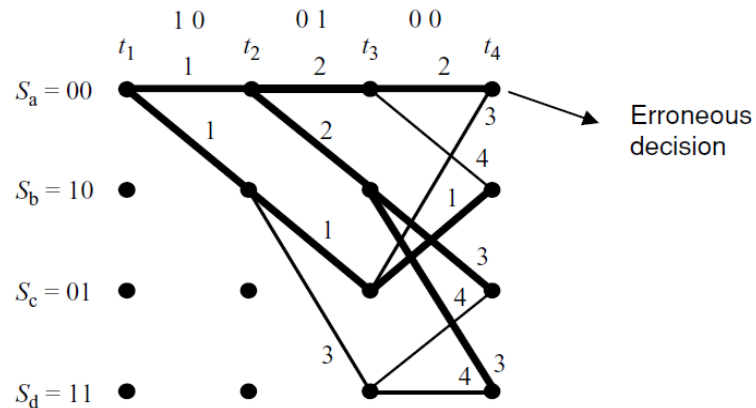


Figure 4.5: Hard-decision decoding for the convolutional code in Fig. 4.2 [10].

trellis diagram with the output values in polar format and the input values omitted. This is more useful for illustrating soft-decision decoding. The squared Euclidean

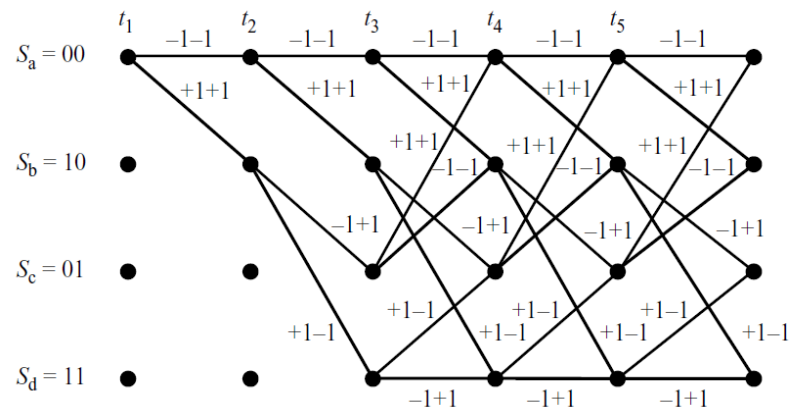


Figure 4.6: Trellis diagram for the convolutional code in Fig. 4.2 with output values in polar format [10].

distance is used as a metric to calculate the minimum cumulative distance of the paths on the trellis diagram. This is defined as the sum of the squared distances involved in the path considered

$$d_{(i-1)}^2 (s_{r(i-1)}, c_k) = \sum_{j=1}^n (s_{r(i-1)}^{(j)} - c_k^{(j)})^2 \quad (4.15)$$

Based on the above equation, the Euclidean distance is calculated by adding the square of the distance between two corresponding bits of the sequences. The Euclidean distance for a given path in the trellis is obtained by adding the Euclidean distances of the corresponding trellis branches. For the example, the Euclidean distance for the first branch in Fig. 4.7 is 6.245, as $(+1.35, -0.15)$ is the received sequence and $(-1, -1)$ is the expected received sequence based on the encoder output in bipolar format in the Fig. 4.6. For a path with U state transitions in the trellis, the cumulative squared distance is

$$d_U^2 = \sum_{v=1}^U (d_v^2 (s_{r(v)}, c_k)) \tag{4.16}$$

The first Euclidean distance in our example is

$$d_1^2[(+1.35, -0.15), (-1, -1)] = (1.35 + 1)^2 + (-0.15 + 1)^2 = 6.245$$

According to Fig. 4.7, the most probable sequence after decoding corresponds to the path through the trellis with minimum cumulative squared distance, which is equal to 7.204. To minimize the memory required for decoding, decisions should be made

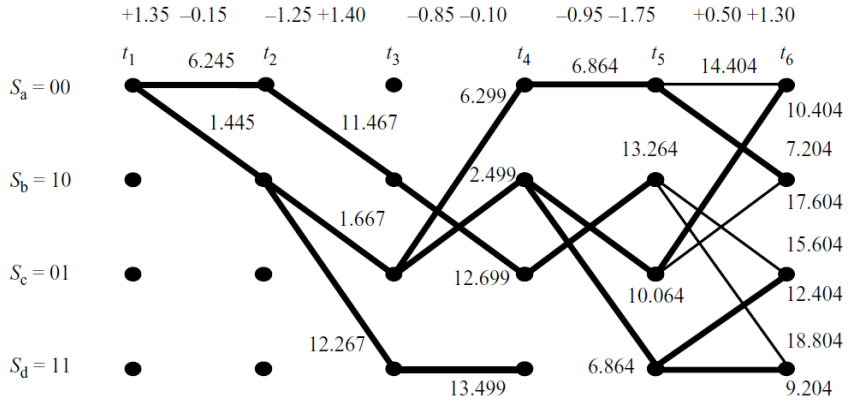


Figure 4.7: Soft-decision decoding of the convolutional code in Fig. 4.2 [10].

as early as possible regarding the most likely path in the trellis. For a sufficiently long message sequence (and therefore encoded sequence and received sequence), based on the results in [10], a decoding decision can be made at time t_i if the survivor sequences have been determined at time $t_i + J$. It has been shown that there is minimal loss in performance if J is equal to five times the constraint length of the code, i.e., $J = 5 \times (K + 1)$.

4.4 Convolutional Decoding Complexity

The complexity of the convolutional decoding is a major concern for implementation. Based on the analysis in [11], the complexity of convolutional decoding, C , is measured as the number of branch metrics computed per decoded bit. Specifically, a binary encoder which takes in k information bits per unit time has a trellis with 2^k branches entering or leaving each state, and its decoder must output k information bits per unit time. Therefore, the complexity estimate for the convolutional decoder is given by $C = S \times (2^k)/k$ where $S = 2^K$, K is the number of shift registers in the encoder, and k is the number of input bits per unit time. For the DSRC convolutional code, $S = 2^6$, $k = 1$ and $K = 6$. The complexity of decoding n bits is $C_t = n \times S \times (2^k)/k$. Therefore the complexity for the DSRC code is defined as

$$C_t = n \times S \times (2^k)/k \quad \text{for the DSRC code} \quad \rightarrow \quad C_t = n \times 2^6 \times 2, \quad n = 1008 \text{ or } 4000$$

Chapter 5

Regular and Quasi-Cyclic LDPC Codes

The theoretical bounds on coding performance established by Shannon in 1949 have led to many practical error-correction schemes including turbo codes, which were invented by Berrou, Glavieux and Thitimajshima in 1993. Three years later, MacKay and Neal rediscovered a class of codes which was first introduced by Gallager in 1962, namely Low Density Parity Check (LDPC) codes. These codes have been shown to have near-ideal performance. Gallager's work was ignored by coding researchers for over thirty years because of the lack of digital technology to implement LDPC encoders and decoders. Another major break though occurred in 1981 when Tanner provided a new graphical representation of linear block codes. The work of Tanner was also ignored for 14 years until the late 1990s when iterative decoding using a graphical representation was established. Gallager codes, part of the class of LDPC codes, are considered in this thesis. They are linear block codes constructed by designing a sparse parity check matrix H which has a key role in encoding the message blocks and decoding the codewords. A sparse binary parity check matrix contains relatively few '1's and many '0's. Although Gallager proposed LDPC codes, he did not provide a method for constructing good codes. Subsequently, Kou, Lin, and Fosshier introduced a systematic construction, namely quasi-cyclic LDPC codes. The first proposed LDPC code construction considered a sparse parity check matrix H with a fixed number of '1's per row and column. This is called a regular LDPC codes. If the number of '1's per row and column varies, it is called an irregular LDPC code. Only regular codes are considered here, In accordance with [10], by considering the

Bit Error Rate (BER), turbo code performance is similar to that with LDPC codes. Both employ pseudo-random techniques in the code design. In turbo codes, a random interleaver is employed, while in LDPC codes, the construction of the sparse parity check matrix H is random. In general, LDPC codes have some advantages over turbo codes. First, LDPC codes do not require a long interleaver to achieve good performance. Interleaver design and implementation can be difficult. Second, LDPC decoding is not based on the trellis diagram which makes the decoding procedure less complex in compare with turbo codes decoding scheme. In this chapter, we focus on the construction, encoding and decoding of LDPC codes, including quasi-cyclic LDPC codes. The decoder complexity is also analysed.

5.1 Regular LDPC Codes

Since LDPC codes are linear block codes, in this section linear block codes are explained.

5.1.1 Linear Block Codes

With a binary linear block code, a message is grouped into blocks of k bits, which are called the message bits. Each block of k bits is encoded into a longer block of $n > k$ bits which is called the codeword or the coded bits. Typically, $n - k$ redundant bits (called parity check bits) are added to the message bits to create a codeword. There are many ways of creating these parity bits, but in all cases they must be such that the message bits can be recovered by applying the inverse operation. A block code is denoted by $C_b(n, k)$ with code rate $R_c = k/n$, which is the ratio of the message bits to the coded bits. In order to make linear block codes, some mathematical structures are now introduced.

Group, Field, Vector Space and Vector Subspace

Definition: A **group** (G, \cdot) is a set of objects G on which a binary operation \cdot is defined. $a \cdot b \in G$ for all $a, b \in G$. The operation must satisfy the following requirements:

1. Associativity: $a \cdot (b \cdot c) = (a \cdot b) \cdot c$

2. Identity: there exists $e \in G$ such that for all $a \in G$,
 $a \cdot e = e \cdot a = a$ e : identity element of G
3. Inverse: for all $a \in G$, there exists a unique element, $a^{-1} \in G$ such that:
 $a \cdot a^{-1} = a^{-1} \cdot a = e$ a^{-1} : inverse of a
4. A group is said to be commutative if it also satisfies:
for all $a, b \in G$, $a \cdot b = b \cdot a$

Definition: A **field** $(F, +, \cdot)$ is a set of objects F on which two binary operations $+$ and \cdot are defined. F is said to be a field if and only if:

1. $(F, +)$ is a commutative group under $+$ with additive identity '0'.
2. $(F^* = F - \{0\}, \cdot)$ is a commutative group over \cdot with multiplicative identity '1'.
3. The operation \cdot distributes over $+$ $\longrightarrow a \cdot (b + c) = (a \cdot b) + (a \cdot c)$

Definition: A **vector space** is a set of vectors which is closed under vector addition and scalar multiplication defined over a field F . A subset of vectors in a vector space V which satisfy all the vector space conditions is called a subspace of the vector space V . A binary block code of length n with 2^k codewords is a linear block code $C_b(n, k)$ if the codewords form a vector subspace of dimension k , by considering V_n as the vector space of all the vectors of length n with components in the field $GF(2)$.

Generator and Parity Check Matrices

Since $C_b(n, k)$ is a vector subspace S of V_n , there will be k linearly independent codewords g_0, g_1, \dots, g_{k-1} , such that all codewords can be obtained as a linear combination of these vectors

$$c = m_0 \cdot g_0 \oplus m_1 \cdot g_1 \oplus \dots \oplus m_{k-1} \cdot g_{k-1} \quad (5.1)$$

According to (5.1), a generator matrix can be defined as

$$G = \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = \begin{bmatrix} g_{00} & g_{01} & \dots & g_{0,n-1} \\ g_{10} & g_{11} & \dots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \dots & g_{k-1,n-1} \end{bmatrix} \quad (5.2)$$

so that the codewords can be generated as follows

$$\begin{aligned}
c = m \circ G &= (m_0, m_1, \dots, m_{k-1}) \circ \begin{bmatrix} g_{00} & g_{01} & \dots & g_{0,n-1} \\ g_{10} & g_{11} & \dots & g_{1,n-1} \\ \vdots & \vdots & & \vdots \\ g_{k-1,0} & g_{k-1,1} & \dots & g_{k-1,n-1} \end{bmatrix} \\
&= (m_0, m_1, \dots, m_{k-1}) \circ \begin{bmatrix} g_0 \\ g_1 \\ \vdots \\ g_{k-1} \end{bmatrix} = m_0 \cdot g_0 \oplus m_1 \cdot g_1 \oplus \dots \oplus m_{k-1} \cdot g_{k-1}
\end{aligned} \tag{5.3}$$

where \circ denotes inner product between vectors or matrices and \cdot represents scalar multiplication in the field $GF(2)$. From (5.3), the k linearly independent rows of G completely define the code. The dual vector subspace S_d is the set of vectors orthogonal to the subspace S . This dual vector subspace S_d is generated by $n - k$ linearly independent rows of the matrix H known as the parity check matrix. Derived from the characteristics of the dual vector space, matrix H is accounted as the generator matrix of the dual vector space. The matrix H defines the dual code $C_{bd}(n, n - k)$ which is the dual subspace of the code $C_b(n, k)$. Since the vectors in S and S_d are orthogonal

$$G \circ H^T = 0 \quad \xrightarrow{\text{so that}} \quad c \circ H^T = m \circ G \circ H^T = 0 \tag{5.4}$$

Thus c is a codeword if and only if the inner product of c with H is zero. The generator and parity check matrices can be put in systematic form so that every codeword can be separated into k information symbols and $n - k$ parity check symbols. In this case, encoding consists of adding $n - k$ check symbols to the message symbols, as shown below.



5.1.2 Cyclic Codes

A vector $c = (c_0, c_1, \dots, c_{n-1})$ can be represented in polynomial format as $c(X) = c_0 + c_1X + \dots + c_{n-1}X^{n-1}$. Consider a right-shift rotation of this vector. A linear block code is a cyclic code if the i th cyclic rotation of a codeword is also a codeword,

defined as

$$c^{(i)} = (c_{n-i}, c_{n-i+1}, \dots, c_{n-1}, c_0, c_1, \dots, c_{n-i-1})$$

As with any linear block code, the sum of any two codewords of a cyclic code is also a codeword. Thus, a code C is cyclic if C is linear and a cyclic shift of any codeword is also a codeword.

Generator and Parity Check Matrices

Define $GF(q)[x]$ as the set of polynomials with coefficients from $GF(q)$

$$c(X) = c_0 + c_1X + \dots + c_{n-1}X^{n-1} \quad c_i \in GF(q)$$

In this thesis, only $q = 2$ is considered, so c_i is either 0 or 1. Now let R_n be the set of polynomials in $GF(2)[x]$ modulo $x^n - 1$ denoted as $GF(2)[x]/x^n - 1$, $\langle f(x) \rangle$ be the subset of R_n consisting of all multiples of $f(x)$ modulo $x^n - 1$, where $f(x)$ is in R_n

$$\langle f(x) \rangle = \{r(x)f(x) \mid r(x) \in R_n\}$$

Then C is a cyclic code if and only if it satisfies the following conditions:

1. There is a unique polynomial $g(x)$ with smallest degree in C which is known as the generator polynomial.
2. $C = \langle g(x) \rangle$
3. $g(x) \mid x^n - 1$

If the generator polynomial $g(x)$ has degree $n-k$, then the the message block can have length k , so the number of codewords is $|C| = q^k = 2^k$. Therefore every codeword is of the form $c(x) = m(x) \cdot g(x)$ where $c(x)$, $m(x)$, and $g(x)$ have degrees $n-1$, $k-1$, and $n-k$, respectively. A linear cyclic code $C_{cyc}(n, k)$ has generator polynomial

$$g(x) = g_0 + g_1x + \dots + g_{n-k-1}x_{n-k-1} + g_{n-k}x_{n-k}$$

In order to represent the code using a generator matrix, the cyclic nature of the code can be exploited. Consider the generator polynomial coefficients, $(g_0, g_1, g_2, \dots, g_{n-k-1}, g_{n-k})$, as a vector and as the first row of the generator matrix. Shifting the first row one position to the right, the second row of the generator matrix can be obtained. Repeating this k times produces a generator matrix G of dimension $k \times n$ as illustrated

below

$$G = \begin{bmatrix} g_0 & g_1 & g_2 & \dots & g_{n-k} & 0 & 0 & \dots & 0 \\ 0 & g_0 & g_1 & \dots & g_{n-k-1} & g_{n-k} & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & g_0 & g_1 & g_2 & \dots & g_{n-k} \end{bmatrix} \quad (5.5)$$

$h(x)$ is defined as the unique monic polynomial of degree k such that multiplication with $g(x)$ gives $x^n - 1$. Thus $h(x)$ is also the generator polynomial of a cyclic code C' (not necessarily the dual code of C generated by $g(x)$). For a codeword $c(x)$, we have the following relationship

$$h(x)c(x) = h(x)g(x)m(x) = (x^n - 1)m(x) = 0 \pmod{x^n - 1}$$

Thus $h(x)$ has a role similar to that of the parity check matrix H . To obtain H , the inner product of a codeword with the corresponding parity check matrix should equal zero, which requires using

$$h^*(x) = x_k h(x^{-1}) = h_k + x h_{k-1} + \dots + x_k h_0$$

$$H = \begin{bmatrix} h_k & h_{k-1} & h_{k-2} & \dots & h_0 & 0 & 0 & \dots & 0 \\ 0 & h_k & h_{k-1} & \dots & h_1 & h_0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \dots & h_k & h_{k-1} & h_{k-2} & \dots & h_0 \end{bmatrix} \quad (5.6)$$

5.1.3 Quasi-Cyclic Codes

A quasi-cyclic (QC) code is a linear block code C in which a cyclic shift of any codeword in C by t positions is also a codeword in C . Thus if $t = 1$, we obtain a cyclic code. For a quasi-cyclic code, a generator matrix G can be obtained by taking t circular shifts of the previous rows. From [12], the generator matrix G for a quasi-cyclic code can be decomposed into circulant¹matrices. This characteristic of G for a quasi-cyclic code C means that the corresponding parity check matrix H is also composed of circulant matrices. The encoder can be designed to exploit this structure to obtain a low cost hardware implementation. The general form of a $[k \times n]$ generator matrix for the quasi-cyclic code with $t = 2$ is shown below. It can

¹A is a circulant matrix if it is a square $[p \times p]$ matrix in which each row is a right cyclic shift of the previous row. The number of non-zero entries in the rows or columns of A is a constant.

be defined by the first row of its generator matrix G , $[g_0, g_1, \dots, g_{n-1}]$.

$$G = \begin{bmatrix} g_0 & g_1 \cdots & g_{n-4} & g_{n-3} & g_{n-2} & g_{n-1} \\ g_{n-2} & g_{n-1} & g_0 & g_1 \cdots & g_{n-4} & g_{n-3} \\ g_{n-4} & g_{n-3} & g_{n-2} & g_{n-1} & g_0 \cdots & g_{n-5} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ g_2 & g_3 \cdots & \cdots & g_{n-1} & g_0 & g_1 \end{bmatrix} \quad (5.7)$$

5.1.4 Regular LDPC Codes

This section describes the characteristics of LDPC codes and a general method to construct them. A regular LDPC code satisfies the four following conditions:

1. The parity check matrix H has a fixed number of ‘1’s per row, denoted by ρ .
2. The parity check matrix H has a fixed number of ‘1’s per column, denoted by γ .
3. The number of common ‘1’s per column and per row is at most one. This is a necessary condition to avoid short cycles in the corresponding bipartite graph.
4. The number of ‘1’s per row and per column is small compared to the code length.

LDPC codes can be classified into two categories, random LDPC codes and structured LDPC codes, based on the construction method used to generate the parity check matrix. As a general rule, random LDPC codes have slightly better BER performance in comparison with structured LDPC codes, however this benefit is achievable with an implementation that is more complex and subsequently more expensive. A random matrix construction method is employed in this thesis since it provides better performance with tolerable complexity. The random parity check matrices were selected from those constructed by Mackay².

Tanner Graph

A graph $G = (V, E)$ is defined by a set of vertices $V = \{v_1, v_2, \dots\}$ and a set of edges $E = \{e_1, e_2, \dots\}$. The vertices are represented as points and each edge is a

²<http://www.inference.phy.cam.ac.uk/mackay/codes/data.html>

line joining two vertices. A graph with a finite number of vertices and edges is called a finite graph. To better explain the graphical representation of a code, a number of definitions are given below. The edge connecting two vertices is defined as the incident edge with these vertices. The degree of a vertex v_i , denoted $d(v_i)$, is the number of edges that are connected to vertex v_i . Two vertices are called adjacent if they are connected by an edge; similarly two edges are called adjacent if they have a common vertex. By taking into account that the number of the edges in a specific path is called length of the path, a cycle is defined as a closed path in which no vertex (excluding the initial and the final vertices), appears more than once. The length of the shortest cycle in a graph is called the girth of the graph. A graph $G = (V, E)$ is called a bipartite graph if its vertex set V can be divided into two disjoint subsets V_1, V_2 in which no two vertices from the same subset are connected. Thus, every edges in E joins a vertex in V_1 to a vertex in V_2 . A bipartite graph therefore has no self-loop, and if it contains cycles, they must all have even length. Similar to the trellis diagram for convolutional codes, the Tanner graph has a key role in representing linear block codes such as LDPC codes. The Tanner graph G_T is a bipartite graph where the first set of vertices V_1 denoted v_0, v_1, \dots, v_{n-1} is the code-bit vertices (also called variable or symbol nodes) representing the n -bit codeword, and the second set of vertices V_2 is the parity check nodes, consisting of J vertices representing parity checks such as the rows of the parity check matrix. The Tanner graph representing a regular LDPC code is called a regular Tanner graph due to the fact that the symbol nodes have the same degree equal to the column weight γ of the parity check matrix. In addition, the parity check nodes have the same degree equal to the row weight ρ of the parity check matrix. It can be shown that the girth of the Tanner graph G_T representing a regular LDPC code is more than 4, and since the cycle length of a bipartite graph is always even, the girth is at least 6. The girth is important because with iterative decoding, short cycles or small girth limits the decoding performance, as will be demonstrated in the following sections. The following example illustrates the Tanner graph representation for a $(7, 3)$ linear block code. In this matrix, parity

check node j is connected to symbol node i if $H_{ji} = 1$.

$$H = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.8)$$

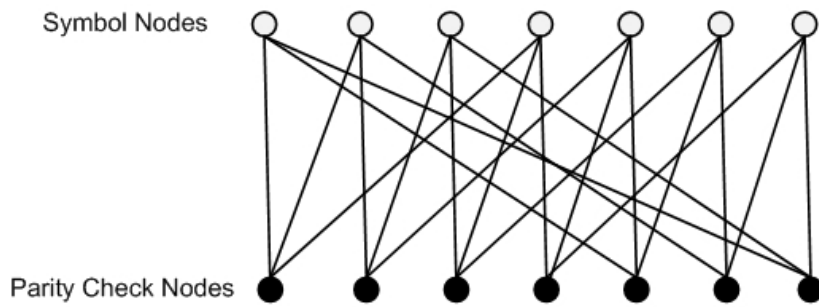


Figure 5.1: The Tanner graph of a $(7,3)$ linear block code [9].

5.2 Random LDPC Codes

As mentioned previously, there are two methods of generating the parity check matrix of an LDPC code. The first method is a geometric construction which is not considered in this thesis. Due to the fact that all the parity check matrices of LDPC codes used in simulating the DSRC system are derived from the second method employs a pseudorandom number generator to determine the location of the ‘1’s in H while satisfying the conditions given in the previous section. Given the LDPC code length n and code rate k/n , the column weight γ and the number of parity check matrix rows J are chosen to provide a suitable matrix H . It is necessary that $J \geq n - k$, but typically equality is used in the construction, so that the following condition is satisfied:

$$\gamma \times n = \rho \times (n - k) \quad (5.9)$$

Therefore, if n is divisible by $n - k$ then ρ is a multiple of γ , i.e., $\rho = \gamma n / (n - k)$. As a result, the matrix can be constructed with the column weight of γ and row weight of ρ . On the other hand, if n is not divisible by $n - k$ then the code is not regular because

$$\gamma \times n = \rho(n - k) + b \quad (5.10)$$

where b and ρ are the remainder and the quotient with $0 < b < n - k$, respectively. Equation 5.10 can be rearranged to give

$$\gamma \times n = (n - k - b)\rho + b(\rho + 1) \quad (5.11)$$

This results in an irregular LDPC code which could have two row weights ρ and $\rho + 1$. For example, the top b rows could have weight $\rho + 1$ and the bottom $n - k - b$ rows weight ρ .

The random construction of H by means of the random technique is carried out one column at a time such that the constraints listed above are satisfied. For $1 \leq i \leq n$, a binary $(n - k)$ -tuple of weight γ is chosen as a candidate column h_i at the i -th step and added to the partial parity check matrix, so that

$$H_{i-1} = [h_1, h_2, \dots, h_{i-1}] \quad (5.12)$$

Column h_i must satisfy the following constraints [9]

1. Choose h_i randomly from the remaining binary $(n-k)$ -tuples that have neither been used in H_{i-1} nor rejected in the previous steps.
2. Check whether h_i has more than one component in common with any column in H_{i-1} ; if not, go to the next step; otherwise, reject h_i and go back to the first step to choose another candidate column.
3. Add h_i to H_{i-1} to form a temporary partial parity check matrix H_i . Check the row weights of H_i ; if the top b rows of H_i have weight less than or equal to $\rho + 1$ and the bottom $n - k - b$ rows of H_i have weight less than or equal to ρ then choose this matrix as H_i and go to the first step to continue the construction process. Otherwise, go to the first step to choose another candidate column.

The parity check matrix construction algorithm continues until a matrix H with n columns is obtained that satisfies all constraints. If $b = 0$, a regular LDPC code is

obtained. In order to reduce the possibility of algorithm failure, n , k and γ should be chosen such that the total number of binary $(n - k)$ -tuples

$$\binom{n - k}{\gamma},$$

is much larger than the code length n .

If the parity check matrix H obtained has row rank $n - k$, then the null space³ of H generates an (n, k) LDPC code with code rate k/n . Otherwise, the result is an (n, k') LDPC code with $k' > k$ and rate $k'/n > k/n$. Based on the analysis in [9], the code rate R has a lower bound of

$$R \geq 1 - \frac{\gamma}{\rho} \quad (5.13)$$

5.3 QC-LDPC Code Construction

In this section, QC-LDPC code construction is presented based on the distance graph algorithm in [13]. It can produce QC-LDPC code parity check matrices with flexible values of column and row weight. This construction is used to obtain the QC-LDPC codes employed in this thesis.

5.3.1 Distance Graph

The distance graph is another way of representing an LDPC code based on its parity check matrix H . A distance graph is a non-bipartite connected graph in which the vertices represent rows, and edges represent columns of the associated matrix. It can be defined by the number of vertices J , the smallest cycle length or girth g and the average vertex degree represented by ρ . If there are two rows in H which have a '1' in the same column, then there is a single edge between the corresponding vertices in the graph, in other words connected vertices in the graph represent rows that have a '1' in the same column of H . Fig. 5.2 shows a distance graph with five vertices and girth three. The number of '1' entries in a row (ρ) is equal to the number of edges of the corresponding vertex. Based on the analysis in [13], the size of the parity check matrix H of a QC-LDPC code based on the distance graph conception is $(J, J\rho/\gamma)$

³In linear algebra, the kernel or null space of a matrix A is the set of all vectors x for which $Ax = 0$

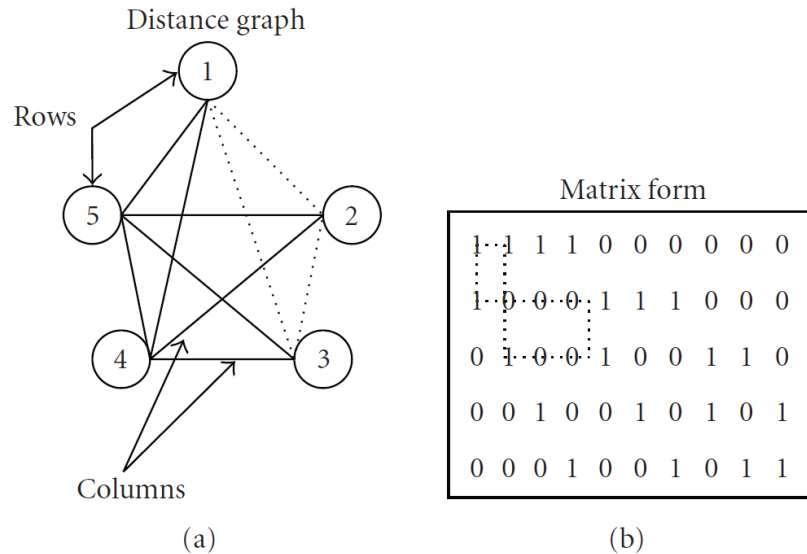


Figure 5.2: Distance graph and matrix representation of an LDPC code [13].

where J is the number of vertices in the graph or equivalently the number of the rows in H , and ρ and γ are the row and column weights of H , respectively. Consequently, defining three parameters J, γ, ρ is sufficient to identify the code. Therefore, the notation (n, γ, ρ) is often used to denote a QC-LDPC code, where n is the number of code length equal to $J\rho/\gamma$ and $1 - \gamma/\rho$ is the code rate.

A useful property of distance graphs is that a cycle of length g in the graph corresponds to a cycle of length $2g$ in H . An explanation for this statement is that the girth can be calculated using either the vertices or edges, whereas in the matrix a cycle alternates between rows and columns, therefore, the graph cycle represents half of the matrix cycle.

5.3.2 Construction Algorithm for QC-LDPC Codes

In this section, a recently proposed algorithm for constructing QC-LDPC codes is presented [13]. All QC-LDPC code parity check matrices used in this thesis were constructed based on this algorithm. Note that this method is limited in terms of its ability to produce codes with arbitrary girth, rate, and length. Similar to LDPC codes, QC-LDPC codes also can be created based on either random or structured methods. In [13], the authors take advantage of the flexibility in random search methods to construct semi-structured codes by adding additional constraints to random search algorithms. As mentioned in the preceding section, this algorithm is based

on the distance graph representation.

Constructing a distance graph for a QC-LDPC code can be achieved by dividing the vertices, representing rows in H , into equal groups which represent sub-matrices in H . Two vertices of the graph are connected if only they are in different groups. In addition, for the purpose of constructing a quasi-cyclic code, all vertices in a group are connected to the vertices in the other group in sequential order to obtain cyclically shifted identity sub-matrices. Thus if vertices v_x and v_y are connected, then v_{x+a} and v_{y+a} are also connected. However, in order to create a QC-LDPC code with a desired girth g , vertices must be selected and connected at a desired distance from each other. The steps of the algorithm to construct QC-LDPC codes are given below [13].

1. Divide the rows into j' equal groups of size p , $(RG_1, \dots, RG_{j'})$. If the number of rows is unknown or not given, start with a theoretical minimum number of rows if known, otherwise start with a group size of ρ (row-weight).

Let r_x denote row x .

\bigcup_{r_x} is the set of rows within distance g from r_x .

2. Pair row groups such that each group appears ρ times. There are $\rho j'/\gamma$ row group pairs⁴, $(RGP_1, \dots, RGP_{\rho j'/\gamma})$.

3. For $t = 1$ to $\rho j'/\gamma$ {
 - $RG_{ref} = RGP_t(1)$
 - select $r_i \in RG_{ref}$
 - For $y = 2$ to γ {
 - sequentially or randomly search for $r_x \in RGP_t(y)$, where $r_x \notin \bigcup_{r_i}$
 - else algorithm fails for this particular selected r_x ,
 - For $z = 1$ to p {
 - r_{i+z} is connected to r_{x+z} if $r_{x+z} \notin \bigcup_{r_{i+z}}$,
 - else algorithm fails for this particular selected r_x

4. Use the resulting distance graph to form an LDPC parity-check matrix.

In this algorithm, adding r_x to the group of r_i in the illustrated step is required when the column weight of the desired parity check matrix is larger than two or ($\gamma > 2$),

⁴In this thesis, $j' = \gamma$ so there are just ρ group pairs

so the next r_x which is chosen from $RGP_t(y)$ while $y > 2$, should not be in the set of rows within distance g from all previously selected rows of r_x . It is worth mentioning that the group of r_i is reset after moving to the next row group pairs.

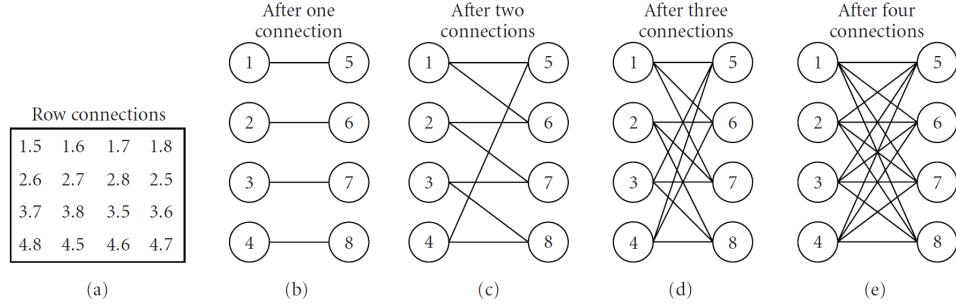


Figure 5.3: Graph representation of a $(16, 2, 4)$ QC-LDPC code with girth eight [13].

An LDPC code with parameters $(16, 2, 4)$ and desired girth eight was constructed using the algorithm given above, and this code is shown in Fig. 5.3. There are two groups of size 4. This size was chosen based on the required row-weight of H . The first group has rows 1 to 4 and the second group has rows 5 to 8. Since there are only two groups and there must be $\rho j' / \gamma = 4$ group pairs in which each group should appear $\rho = 4$ times, the groups pairings are $[1\ 2], [1\ 2], [1\ 2], [1\ 2]$. In the first connection, row 5 is found to satisfy the girth requirement of four from row 1. The rest of the first group of vertices is then sequentially connected to the rest of the second group of vertices. This process is repeated as shown in the figure. The resulting graph has eight vertices, a vertex degree equal to four, and a girth of four. Fig. 5.4 shows the corresponding parity check matrix for this code.

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | | | 1 | | | | 1 | | | | 1 | | | |
| | 1 | | | | 1 | | | | 1 | | | | 1 | | |
| | | 1 | | | | 1 | | | | 1 | | | | 1 | |
| | | | 1 | | | | 1 | | | | 1 | | | | 1 |
| 1 | | | | | | | | | | 1 | | | | 1 | |
| | 1 | | | | 1 | | | | | | 1 | | | | 1 |
| | | 1 | | | | 1 | | | | 1 | | | | | 1 |
| | | | 1 | | | | 1 | | | | 1 | | | | |

Figure 5.4: Parity check matrix H of a $(16, 2, 4)$ QC-LDPC code with girth eight [13].

The complexity of the algorithm to construct a parity check matrix of a QC-LDPC code is on the order of the number of rows of H , or equivalently on the order of the number of vertices $\mathbf{O}(J)$ [13]. If the number of the groups horizontally

and vertically in H are equal to the row and column weights, respectively, then the maximum possible girth is twelve [14]. In addition, if the number of row groups (and consequently column-weight of H) is equal to two, then the resulting QC-LDPC code has a minimum girth of eight.

5.4 The Sum Product Algorithm

A commonly employed decoding algorithm for LDPC codes is the sum product algorithm, which is a belief propagation algorithm. This decoding procedure is based on interchanging and updating information according to the Tanner graph of the code. Similar to other decoding techniques, the major objective of this method is to find a codeword based on the received word which satisfies the parity check equations, i.e., $c \circ H^T = 0$.

In the sum product algorithm, there are two key parameters, R_{ij}^x and Q_{ij}^x . R_{ij}^x is defined as the information that each parity check node h_i sends to a connected symbol node d_j , which is defined as its parent node. It is the probability of satisfying the parity check equation of node h_i when the parent symbol node d_j is in state x . Q_{ij}^x is defined as the information that each symbol node d_j sends to its connected parity check node h_i , which is defined as its child node. An additional parameter f_j^x is defined as the probability that the j th symbol is equal to x based on the channel model.

The sum product algorithm is initialized by setting the values of Q_{ij}^x to f_j^x . Then information is exchanged between the symbol (parent) and the parity check (child) nodes. This exchange leads to calculation of the first values of R_{ij}^x according to [10]

$$R_{ij}^x = \sum_{\mathbf{d}:d_j=x} \mathbf{P}(h_i|\mathbf{d}) \prod_{k \in N(i) \setminus j} Q_{ik}^{dk} \quad (5.14)$$

where $N(i)$ denotes the set of parent symbol nodes connected to the parity check node h_i , and $N(i) \setminus j$ denotes the same symbol node set excluding the parent symbol node d_j . The probability $\mathbf{P}(h_i|\mathbf{d})$ for a given vector \mathbf{d} is either 1 or 0 depending on whether or not the vector \mathbf{d} satisfies the parity check equation of node h_i .

As stated previously, Q_{ij}^x indicates if symbol node d_j is in state x according to the information provided by its children parity check nodes. This probability for node d_j

can be calculated by

$$Q_{ij}^x = \alpha_{ij} f_j^x \prod_{k \in M(j) \setminus i} R_{kj}^x \quad (5.15)$$

where $M(j)$ denotes the set of children parity check nodes connected to the parent symbol node d_j , $M(j) \setminus i$ denotes the same parity check node set excluding the child parity check node h_i , and α_{ij} is the normalizing constant so that $\sum_x Q_{ij}^x = 1$ is satisfied. Calculation of the Q_{ij}^x using (5.15) is followed by calculation of the R_{ij}^x using (5.14), and the estimate of d_j is given by

$$\hat{d}_j = \underbrace{\arg \max}_x f_j^x \prod_{k \in M(j)} R_{kj}^x \quad (5.16)$$

The sum product algorithm is an iterative decoding method in which data is exchanged between symbol nodes and parity check nodes until the estimated symbols satisfy the parity check equations or an iteration limit is reached. More precisely, if \hat{d} found via (5.16) satisfies $\hat{d} \circ H^T = 0$, then \hat{d} is a valid code vector c . If the iteration limit is reached and \hat{d} does not satisfy the parity check equations, then \hat{d} is passed on to the data sink as the best estimate of the transmitted codeword. In order to make the sum product algorithm more comprehensible, an example from [10] is given below.

Example:

Consider the linear LDPC code $C_b(12, 4)$ with code rate $R_c = 1/3$. The parity check matrix H is a fairly sparse matrix with dimensions 8×12 , and is given by

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

The corresponding systematic generator matrix G is

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

Encoding is done using the generator matrix, so the code vector c is generated by multiplying the message vector m by the generator matrix G , $c = m \circ G$. Then the code vector satisfies the syndrome equations $c \circ H^T = 0$.

The message vector for this example is $m = (1 \ 0 \ 0 \ 0)$, and generates the codeword vector

$$c = (1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0)$$

which in polar format is

$$t = (+1 \ +1 \ +1 \ +1 \ +1 \ -1 \ -1 \ -1 \ +1 \ -1 \ -1 \ -1).$$

Additive white Gaussian noise (AWGN) with standard deviation $\sigma = 0.8$ is added to t to produce the received vector

$$r = (+1.3129 \ +2.6584 \ +0.7413 \ +2.1745 \ +0.5981 \ -0.8323 \ -0.3962 \ -1.7586 \\ +1.4905 \ +0.4084 \ -0.9290 \ +1.0765)$$

If a hard-decision is made on r , there will be two errors at positions 10 and 12

$$(1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 0 \ 1)$$

The values of f_j^x representing the probability of d_j being in state x^5 , can be calculated based on the channel model. For an AWGN channel, these probabilities are determined using a Gaussian probability density function and are given by

$$f_j^0 = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-(r_j+1)^2/(2\sigma^2)} \quad (5.17)$$

and

$$f_j^1 = \frac{1}{\sqrt{2\pi}\sigma} \exp^{-(r_j-1)^2/(2\sigma^2)} \quad (5.18)$$

⁵In this thesis, x is binary and so can be either 1 or 0.

Table 5.1: The Received Vector and Corresponding values of f_j^x [10]

| j | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | ... |
|---------|---------|---------|---------|---------|---------|---------|---------|---------|-----|
| r | +1.3129 | +2.6584 | +0.7413 | +2.1745 | +0.5981 | -0.8323 | -0.3962 | -1.7586 | ... |
| t | +1 | +1 | +1 | +1 | +1 | -1 | -1 | -1 | ... |
| f_j^0 | 0.0076 | 0.0000 | 0.0467 | 0.0002 | 0.0678 | 0.4878 | 0.3751 | 0.3181 | ... |
| f_j^1 | 0.4619 | 0.0582 | 0.4733 | 0.1697 | 0.4396 | 0.0362 | 0.1088 | 0.0013 | ... |

The resulting values (truncated to four decimal places) are given in Table 5.1. From the Tanner graph given in Fig. 5.5 or the parity check matrix, the parity check equations for the example can be written as

$$c_2 \oplus c_4 \oplus c_6 \oplus c_7 \oplus c_8 \oplus c_{12} = 0 \quad (5.19)$$

$$c_1 \oplus c_3 \oplus c_4 \oplus c_9 = 0 \quad (5.20)$$

$$c_2 \oplus c_5 \oplus c_7 \oplus c_{12} = 0 \quad (5.21)$$

$$c_1 \oplus c_4 \oplus c_{10} \oplus c_{11} = 0 \quad (5.22)$$

$$c_3 \oplus c_5 \oplus c_6 \oplus c_{10} = 0 \quad (5.23)$$

$$c_1 \oplus c_3 \oplus c_7 \oplus c_8 \oplus c_{11} = 0 \quad (5.24)$$

$$c_2 \oplus c_6 \oplus c_8 \oplus c_9 \oplus c_{10} = 0 \quad (5.25)$$

$$c_5 \oplus c_9 \oplus c_{11} \oplus c_{12} = 0 \quad (5.26)$$

Each row of the parity check matrix H corresponds to a parity check equation and thus to a parity check node in the Tanner graph, while each bit of the code vector corresponds to a symbol node in the Tanner graph. Thus in the example, the children parity check nodes of symbol node 2 are parity check nodes 1, 3 and 7, whereas the parent symbol nodes of parity check node 1 are symbol nodes 2, 4, 6, 7, 8 and 12.

The sum product decoding algorithm is initialized by setting Q_{ij}^0 and Q_{ij}^1 equal to f_j^0 and f_j^1 , respectively. Q_{ij}^0 and Q_{ij}^1 are then sent to the parity check nodes as shown in Fig. 5.6.

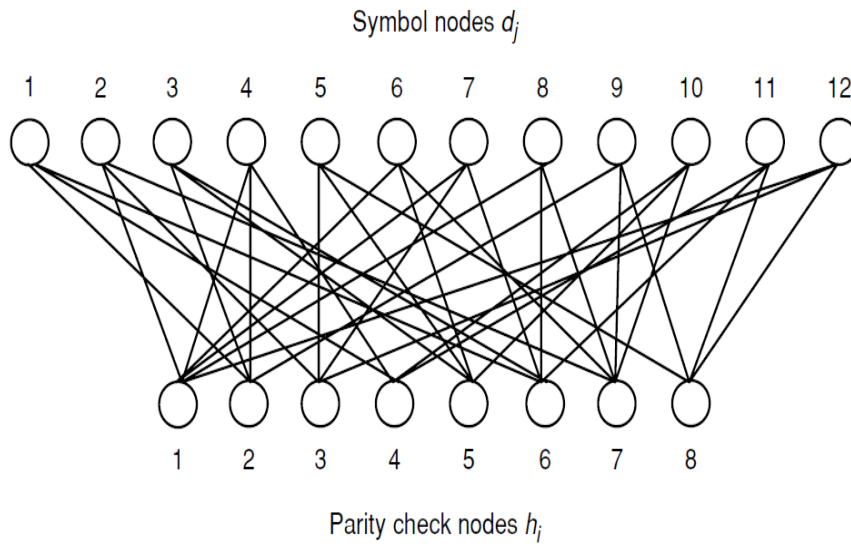


Figure 5.5: Bipartite graph for the example given in [10].

$$\begin{aligned}
 Q_{12}^0 &= 0.0000 & Q_{12}^1 &= 0.0582 \\
 Q_{14}^0 &= 0.0002 & Q_{14}^1 &= 0.1697 \\
 Q_{16}^0 &= 0.4878 & Q_{16}^1 &= 0.0362 \\
 Q_{17}^0 &= 0.3751 & Q_{17}^1 &= 0.1088 \\
 Q_{18}^0 &= 0.3181 & Q_{18}^1 &= 0.0013 \\
 Q_{1\ 12}^0 &= 0.0172 & Q_{1\ 12}^1 &= 0.4964 \\
 \\
 Q_{21}^0 &= 0.0076 & Q_{21}^1 &= 0.4619 \\
 Q_{23}^0 &= 0.0467 & Q_{23}^1 &= 0.4733 \\
 Q_{24}^0 &= 0.0002 & Q_{24}^1 &= 0.1697 \\
 Q_{29}^0 &= 0.0039 & Q_{29}^1 &= 0.4132 \\
 \\
 Q_{32}^0 &= 0.0000 & Q_{32}^1 &= 0.0582 \\
 Q_{35}^0 &= 0.0678 & Q_{35}^1 &= 0.4396 \\
 Q_{37}^0 &= 0.3751 & Q_{37}^1 &= 0.1088 \\
 Q_{3\ 12}^0 &= 0.0172 & Q_{3\ 12}^1 &= 0.4964
 \end{aligned}$$

The next step after calculating the first values of Q_{ij}^x is determining the estimates of R_{ij}^0 and R_{ij}^1 . The values of R_{ij}^x are the estimates which are sent from the parity check nodes to the symbol nodes according to Fig. 5.7. To clarify this, consider R_{12}^0 which is the estimate that parity check node 1 sends to its parent symbol node 2.

According to the Tanner graph, if the parity check equation for child parity check node 1, calculated as $c_2 \oplus c_4 \oplus c_6 \oplus c_7 \oplus c_8 \oplus c_{12} = 0$, is satisfied then symbol 2 is in state $c_2 = 0$. In this case, there are 16 combinations of the bits c_4, c_6, c_7, c_8 and c_{12} with an even number of 1's⁶ that can satisfy the parity check equation at parity node 1. The probabilities associated with each combination are added to calculate the estimate R_{12}^0 as shown below

$$\begin{aligned}
R_{12}^0 = & Q_{14}^0 Q_{16}^0 Q_{17}^0 Q_{18}^0 Q_{1,12}^0 + Q_{14}^0 Q_{16}^0 Q_{17}^0 Q_{18}^1 Q_{1,12}^1 + Q_{14}^0 Q_{16}^0 Q_{17}^1 Q_{18}^0 Q_{1,12}^0 + \\
& Q_{14}^0 Q_{16}^0 Q_{17}^1 Q_{18}^1 Q_{1,12}^1 + Q_{14}^0 Q_{16}^1 Q_{17}^0 Q_{18}^0 Q_{1,12}^0 + Q_{14}^0 Q_{16}^1 Q_{17}^0 Q_{18}^1 Q_{1,12}^1 + \\
& Q_{14}^0 Q_{16}^1 Q_{17}^1 Q_{18}^0 Q_{1,12}^0 + Q_{14}^0 Q_{16}^1 Q_{17}^1 Q_{18}^1 Q_{1,12}^1 + Q_{14}^1 Q_{16}^0 Q_{17}^0 Q_{18}^0 Q_{1,12}^0 + \\
& Q_{14}^1 Q_{16}^0 Q_{17}^0 Q_{18}^1 Q_{1,12}^1 + Q_{14}^1 Q_{16}^0 Q_{17}^1 Q_{18}^0 Q_{1,12}^0 + Q_{14}^1 Q_{16}^0 Q_{17}^1 Q_{18}^1 Q_{1,12}^1 + \\
& Q_{14}^1 Q_{16}^1 Q_{17}^0 Q_{18}^0 Q_{1,12}^0 + Q_{14}^1 Q_{16}^1 Q_{17}^0 Q_{18}^1 Q_{1,12}^1 + Q_{14}^1 Q_{16}^1 Q_{17}^1 Q_{18}^0 Q_{1,12}^0 + \\
& Q_{14}^1 Q_{16}^1 Q_{17}^1 Q_{18}^1 Q_{1,12}^1 = 0.0051
\end{aligned}$$

In the same way R_{12}^1 is estimated assuming that its corresponding parity check equation, which is $c_2 \oplus c_4 \oplus c_6 \oplus c_7 \oplus c_8 \oplus c_{12} = 0$, is satisfied, when symbol 2 is in state $c_2 = 1$. There are again 16 combinations (odd number of '1's) of the bits c_4, c_6, c_7, c_8 and c_{12} that can satisfy this parity check equation, and the probabilities associated with each combination are added to calculate the estimate R_{12}^1 as shown below

$$\begin{aligned}
R_{12}^1 = & Q_{14}^0 Q_{16}^0 Q_{17}^0 Q_{18}^0 Q_{1,12}^1 + Q_{14}^0 Q_{16}^0 Q_{17}^0 Q_{18}^1 Q_{1,12}^0 + Q_{14}^0 Q_{16}^0 Q_{17}^1 Q_{18}^0 Q_{1,12}^0 + \\
& Q_{14}^0 Q_{16}^0 Q_{17}^1 Q_{18}^1 Q_{1,12}^1 + Q_{14}^0 Q_{16}^1 Q_{17}^0 Q_{18}^0 Q_{1,12}^0 + Q_{14}^0 Q_{16}^1 Q_{17}^0 Q_{18}^1 Q_{1,12}^1 + \\
& Q_{14}^0 Q_{16}^1 Q_{17}^1 Q_{18}^0 Q_{1,12}^0 + Q_{14}^0 Q_{16}^1 Q_{17}^1 Q_{18}^1 Q_{1,12}^1 + Q_{14}^1 Q_{16}^0 Q_{17}^0 Q_{18}^0 Q_{1,12}^0 + \\
& Q_{14}^1 Q_{16}^0 Q_{17}^0 Q_{18}^1 Q_{1,12}^1 + Q_{14}^1 Q_{16}^0 Q_{17}^1 Q_{18}^0 Q_{1,12}^0 + Q_{14}^1 Q_{16}^0 Q_{17}^1 Q_{18}^1 Q_{1,12}^1 + \\
& Q_{14}^1 Q_{16}^1 Q_{17}^0 Q_{18}^0 Q_{1,12}^0 + Q_{14}^1 Q_{16}^1 Q_{17}^0 Q_{18}^1 Q_{1,12}^1 + Q_{14}^1 Q_{16}^1 Q_{17}^1 Q_{18}^0 Q_{1,12}^0 + \\
& Q_{14}^1 Q_{16}^1 Q_{17}^1 Q_{18}^1 Q_{1,12}^1 = 0.0020
\end{aligned}$$

The procedure of interchanging information between symbol and parity check nodes in the sum product algorithm as an iterative decoding technique is illustrated in Figs. 5.6 and 5.7. In this process, the node that is updated does not participate in the calculation. The complexity and delay are two important parameters of SPA

⁶Based on the parity check equation, the binary addition of the bits involved must equal zero, so in order to satisfy this condition, the number of bits(including symbol node $d_j = 2$) with value '1' must be even.

decoding which depend on the row and column weight of the parity check matrix, as will be discussed later. This is because, the greater the number of ‘1’s in each row of the parity check matrix, the larger the number of combinations of bits needed to calculate R_{ij}^0 and R_{ij}^1 .

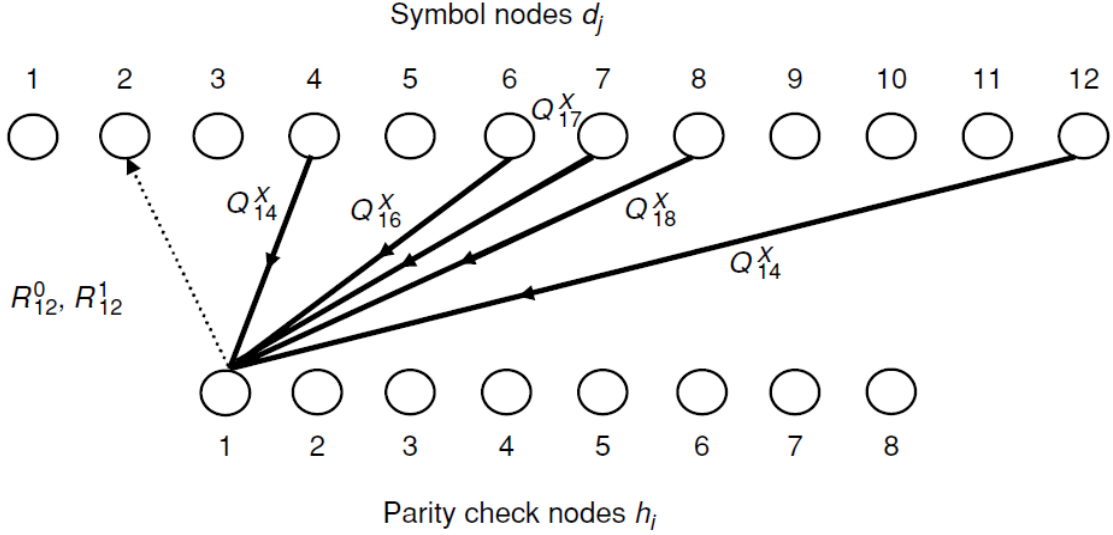


Figure 5.6: Calculation of R_{12}^0 and R_{12}^1 [10].

Once the R_{ij}^x have been calculated, the first decoder decision \hat{d}_j can be obtained using (5.16). For the example, the bits after the first iteration are

$$\hat{d}_1 \Rightarrow \begin{cases} \hat{0} \rightarrow f_1^0 \times R_{21}^0 \times R_{41}^0 \times R_{61}^0 = 1.13 \times 10^{-8} \\ \hat{1} \rightarrow f_1^1 \times R_{21}^1 \times R_{41}^1 \times R_{61}^1 = 4.85 \times 10^{-6} \end{cases} \\ \Rightarrow \text{'1'}$$

$$\hat{d}_2 \Rightarrow \begin{cases} \hat{0} \rightarrow f_2^0 \times R_{12}^0 \times R_{32}^0 \times R_{72}^0 = 1.58 \times 10^{-10} \\ \hat{1} \rightarrow f_2^1 \times R_{12}^1 \times R_{32}^1 \times R_{72}^1 = 4.06 \times 10^{-8} \end{cases} \\ \Rightarrow \text{'1'}$$

In the same way, all values of \hat{d}_j can be calculated. The first estimate of the decoded vector is obtained by choosing the bit corresponding to the larger values, which gives

$$\hat{d} = (1 \ 1 \ 1 \ 1 \ 0 \ 1 \ 1 \ 0 \ 1 \ 0 \ 0 \ 0)$$

This vector contains three errors, so it does not satisfy the parity check equations.

The next decoding iteration starts with the calculation of Q_{ij}^0 and Q_{ij}^1 based on the current values of R_{ij}^0 and R_{ij}^1 using (5.15). For example, Q_{12}^0 is the estimate that parent symbol node 2 sends to child parity check node 1, calculated by forming the product of the estimates R_{k2}^0 of all its children parity check nodes excluding the child node 1 which is being updated. In a similar way, Q_{12}^1 can be calculated using the estimates R_{k2}^1 . The notation k denotes all parity check nodes that are connected to the desired parent symbol node j (node 2 in this example), excluding the parity check node i (node 1 in this example). The values of Q_{ij}^0 and Q_{ij}^1 are then normalized using α_{ij} in order to satisfy $Q_{ij}^0 + Q_{ij}^1 = 1$.

The values of Q_{12}^0 and Q_{12}^1 are

$$Q_{12}^0 = \alpha_{12} f_2^0 R_{32}^0 R_{72}^0$$

and

$$Q_{12}^1 = \alpha_{12} f_2^1 R_{32}^1 R_{72}^1$$

and the normalizing constant α_{ij} is

$$\alpha_{12} = \frac{1}{f_2^0 R_{32}^0 R_{72}^0 + f_2^1 R_{32}^1 R_{72}^1}$$

Fig. 5.7 shows the flow of information and the nodes participating in the calculation of Q_{12}^0 and Q_{12}^1 for the example. Each decoding iteration, the updated values of the

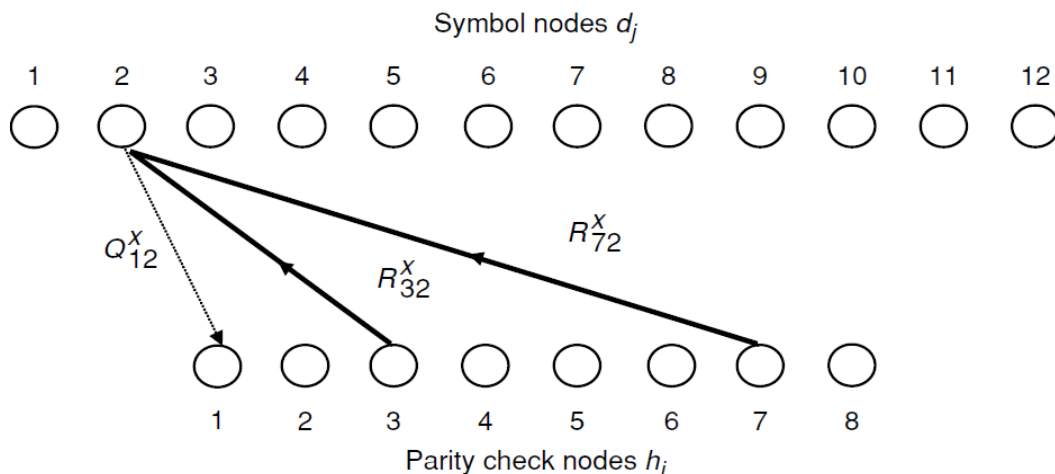


Figure 5.7: Calculation of Q_{12}^0 and Q_{12}^1 [10].

coefficients are employed.

In the example, a code vector satisfying all parity check equations is obtained after three iterations, which shows that the SPA algorithm is able to correct the two errors that hard-decision decoding could not. The minimum Hamming distance of the LDPC linear block code in this example is $d_{min} = 4$. Thus with hard-decision decoding, any error pattern of size $t = 1$ can be corrected. Soft-decision decoding allows for the correction of more than t errors, as the above example has shown.

Based on the analysis in [10], a given block code will perform better than uncoded transmission if the product $R_c(t + 1)$ satisfies the condition $R_c(t + 1) > 1$. For the example, the code rate is $R_c = 1/3$, so $R_c(t + 1) = (1/3) \times 2 = 0.667$, so this simple code does not provide an advantage over uncoded transmission. This is not surprising because it is a very short code. Poor performance with this simple code can also be predicted from the fact that its Tanner graph, shown in Fig. 5.5, contains several short cycles of length 4, and a small girth degrades the performance of iterative soft-decision decoding algorithms such as the sum product algorithm.

5.4.1 Logarithmic SPA Decoding

The purpose of all decoding techniques including the sum product decoding algorithm is to find a decoded vector d , which is an estimate of the transmitted code vector c which satisfies

$$d \circ H^T = 0.$$

As previously explained, f_j^x is defined based on the channel characteristics. The calculation of this parameter for an AWGN channel and the SPA algorithm was presented earlier. The corresponding expressions for the Log-SPA algorithm are

$$f_j^1 = \frac{1}{1 + e^{-\frac{2Ay_j}{\sigma^2}}} \quad (5.27)$$

and

$$f_j^0 = 1 - f_j^1. \quad (5.28)$$

In the above equations, y_j is the channel output at time j , and it is assumed that the transmitted bits are sent in polar format with the amplitudes $\pm A$. Here, the normalized polar format ± 1 is employed. To make the sum product algorithm less complex, a logarithmic representation is used to implement the logarithmic sum-

| | + | * |
|---------|--|-----------------------------------|
| SPA | $(3\rho - 4)Jq(q - 1) + \rho J(q - 1)$ | $(3\rho - 4)Jq^2 + J\gamma\rho q$ |
| Log-SPA | $2(3\rho - 4)J(q - 1)^2 + \rho J(\gamma - 1)(q - 1)$ | 0 |

Table 5.2: Number of Operations per Iteration with SPA and Log-SPA Decoding

product algorithm (LogSPA). Even after considering the need to calculate logarithms, the decoding algorithm is significantly simplified because products and quotients are converted into additions and subtractions. The performance is the same as with the sum-product algorithm, the LogSPA algorithm is employed for the simulations in this thesis.

5.5 Decoding Complexity

Based on the analysis performed in [15], the number of multiplications and additions using the two versions of the sum-product algorithm are shown in Table 5.2.

The LDPC code is assumed to be $C(J, n, \gamma, \rho)$ where (J, n) represents the size of the parity check matrix and (γ, ρ) are the column and row weights of the corresponding parity check matrix H . The expressions in Table 5.2 are for one decoding iteration, so they must be multiplied by the number of iterations to obtain the decoding complexity. The Log-SPA algorithm is employed in this thesis, and the number of iterations used is 15. The LDPC codes considered here are $C_{LDPC}(504, 1008, 3, 6)$, $C_{LDPC}(2000, 4000, 3, 6)$ and $C_{QC-LDPC}(504, 1008, 3, 6)$, and $C_{QC-LDPC}(2000, 4000, 3, 6)$, so the decoding complexities are as follows:

Log-SPA decoding complexity calculations which is applied to this thesis:

For $C(504, 1008, 3, 6) \Rightarrow$

$$15 \times \{2(3 \times 6 - 4) \times 504 + 6 \times 504(3 - 1)\} = 15 \times 20160 \text{ additions}$$

For $C(2000, 4000, 3, 6) \Rightarrow$

$$15 \times \{2(3 \times 6 - 4) \times 2000 + 6 \times 2000(3 - 1)\} = 15 \times 80000 \text{ additions}$$

As we mentioned before the LDPC encoding complexity is $O(n^2)$. By considering the large amount of n which the length of the codeword the encoding complexity

of LDPC codes can be significant while the encoding complexity of QC-LDPC code based on the investigations performed in [13] is in the order of the number of the rows in the correspondence parity check matrix or $O(J)$ according to the section 5.3.2. As the result, QC-LDPC encoding techniques have lower complexity than the regular LDPC codes, while they can provide similar performance as will be shown in the following chapter.

The decoding scheme of low-density parity-check (LDPC) codes allows a high degree of parallelism, which makes it very suitable for high data rate applications such as DSRC communications [16]. In addition quasi-cyclic (QC) LDPC codes have received significant attention due to their efficient hardware implementation and good performance [16]. This reduction in hardware complexity (and power consumption) is obtained by exploiting the regularity of the parity check matrix structure, and results in more efficient decoding than ordinary LDPC codes; however, the Log-SPA decoding technique is applied to both LDPC and QC-LDPC codes in this thesis which still shows better performance of QC-LDPC code in compare with LDPC code. Therefore the decoding complexity of QC-LDPC implemented to this thesis simulations can be calculated the same as LDPC code as illustrated in above.

Chapter 6

Simulation Results

This chapter presents simulation results for the DSRC system employing convolutional and LDPC coding. The performance improvement is investigated for AWGN, Rayleigh and Rician channels, and with LDPC and QC-LDPC codes of different sizes. A pseudo-random number generator (PRNG) is employed to generate the binary digits for transmission. A uniform distribution is assumed, so that ‘0’ and ‘1’ are each generated with probability 0.5. After generating a random binary sequence, the next step is encoding this sequence using either an LDPC or convolutional code. In LDPC and QC-LDPC coding, the inner product of the message bits with the corresponding generator matrix G produces the codeword block. For a convolutional code, the shift register structure or the trellis diagram can be employed for encoding.

In general, there are three types of interleaving, block, convolutional, and random. In the conventional DSRC system, a block interleaver is used to decrease the effects of deep fading. The depth of this interleaver depends on the modulation employed. For instance, with BPSK modulation a 6×8 block interleaver is implemented which operates by grouping a block of 48 bits into 6 rows and 8 columns. The interleaving is performed by inputting via rows and outputting via columns.

In the first part of this chapter, the performance is considered without interleaving. This is because the errors are assumed to be independent in AWGN channel, and also the main objective of this thesis is to compare the performance of the different coding techniques in DSRC systems. In the second part of the chapter, the robustness of the coding techniques against deep fading is considered. The deep fading channel has almost the same fading amplitude for all codeword bits. This highlights the importance of interleaving, and its effect on decoding performance.

Symbol mapping is done following the block interleaver. BPSK modulation is

employed commonly as the simplest modulation scheme to compare different coding techniques, and as a consequence the specific modulation method is not a concern. BPSK maps ‘0’ to -1 and ‘1’ to $+1$. The data rate is $3Mbps$, and was calculated using the results in Chapter 2.

As described in Chapter 2, every 48 bits from the symbol mapping produces one 80 bit OFDM symbol including 4 pilots bits, 12 zero bits and a 16 bit cyclic prefix. Six of the zero bits are inserted at the beginning and at end of the symbol to reduce adjacent channel interference. The pilots are placed in bit positions [12 26 40 54]. According to the standard, the pilot bits are transmitted with double the power of the information bits due to their significant role in channel estimation, and frequency and timing synchronization. However, as mentioned previously, channel estimation is disregarded so perfect time and frequency synchronization is assumed. The DSRC channel is assumed to be a slow fading channel. From [7], the multipath delay spread can be used to quantify the severity of the ISI, and is given by

$$\bar{\tau} = \frac{\sum_i P(\tau_i) \cdot \tau_i}{\sum_i P(\tau_i)} \quad (6.1)$$

$$\bar{\tau}^2 = \frac{\sum_i P(\tau_i) \cdot \tau_i^2}{\sum_i P(\tau_i)} \quad (6.2)$$

$$\sigma_\tau = \sqrt{\bar{\tau}^2 - (\bar{\tau})^2} \quad (6.3)$$

where $P(\tau_i)$ denotes the relative power derived from the power delay profile, σ_τ is the RMS delay spread, τ_i is the delay spread, and i is the path number. As explained in Chapter 3, if the symbol duration is less than the RMS delay spread the channel is called frequency-selective, and the severity of the ISI can be considerable.

According to the investigations performed in [4] to characterize WAVE¹ properties, the minimum RMS delay spread for DSRC systems is $10ns$, which corresponds to two vehicles separated by 10 to 30 meters and moving together through a tunnel. The maximum RMS delay spread is $512ns$, and is usually associated with intersection traffic. In addition, the path loss exponent is in the range $1.4dB - 3.5dB$ for LOS, and $2.8dB - 5.9dB$ for non-LOS channels, for measurement distances of $30m - 300m$. The mean RMS delay spreads, for this scenario, are within $29ns - 102ns$ and $22ns - 88ns$, respectively.

In addition to RMS delay spread and path loss, Doppler can be significant in

¹Wireless Access Vehicular Environment

DSRC systems. The maximum Doppler frequency in a DSRC system is about $2000Hz$ assuming a maximum relative speed of $385km/h$ [4], while the minimum Doppler frequency is $10Hz$ corresponding to a speed of $2km/h$, which happens in congested areas such as intersections; however, Doppler frequency in the range of $10Hz \sim 100Hz$ is typical.

Based on the parameters obtained in various studies, in this chapter, a 3-path fading channel is considered with average path gains $P_G = [0 \text{ dB}, -2 \text{ dB}, -4 \text{ dB}]$ and delays $\tau = [0, 0.2 \mu\text{sec}, 0.5 \mu\text{sec}]$, respectively, along with a Doppler frequency of $10Hz$.

As we mentioned previously, consistent with [4], the path loss exponent is generally in the range $1.4dB - 5.9dB$. Correspondingly, in our simulations we considered the path loss exponents of 3-path fading channel as $P_L = [0 \text{ dB}, 2 \text{ dB}, 4 \text{ dB}]$. Derived from the literature, path loss is defined as the negative of path gain while both are determined in the scale of dB

$$P_L(dB) = P_t(dB) - P_r(dB) \quad \longrightarrow \quad P_G(dB) = P_r(dB) - P_t(dB) \quad (6.4)$$

In (6.4), the transmitted power and the received power are denoted as P_t and P_r respectively in dB scale; However, in order to calculate RMS delay through (6.3), we need to find the power delay profile $P(\tau)$. By considering that transmission power in DSRC systems is usually less than $33 \text{ dBm} = 3 \text{ dB}$ and the definition of the path loss based on (6.4), the power delay profile can be computed as $P(\tau) = [3 \text{ dB}, 1 \text{ dB}, -1 \text{ dB}] = [10^{0.3} \text{ W}, 10^{0.1} \text{ W}, 10^{-0.1} \text{ W}]$. Subsequently, from (6.3) the RMS delay spread is $190ns$ which can be calculated as bellow:

$$\bar{\tau} = \frac{10^{0.1} \times 0.2 \times 10^{-6} + 10^{-0.1} \times 0.5 \times 10^{-6}}{10^{0.3} + 10^{0.1} + 10^{-0.1}} = 0.160 \times 10^{-6} \quad (6.5)$$

$$\bar{\tau}^2 = \frac{10^{0.1} \times (0.2 \times 10^{-6})^2 + 10^{-0.1} \times (0.5 \times 10^{-6})^2}{10^{0.3} + 10^{0.1} + 10^{-0.1}} = 0.0615 \times 10^{-12} \quad (6.6)$$

$$\sigma_\tau = \sqrt{\bar{\tau}^2 - (\bar{\tau})^2} = \sqrt{0.0615 \times 10^{-12} - (0.160 \times 10^{-6})^2} \simeq 190ns \quad (6.7)$$

The coherence time is given by

$$T_c = \sqrt{\frac{9}{16\pi f_D^2}} = \frac{0.423}{f_D} \quad (6.8)$$

By considering Doppler frequencies in the range $10Hz - 2000Hz$, the channel coherence time using (6.8) is in the range $0.0002sec - 0.04sec$ with a symbol duration of $T_s = 8\mu sec$. If $T_s \ll T_c$ or $10 \times T_s \simeq T_c$, the channel is slow fading which is the case here since $T_c = 0.04sec$ is much larger than $T_s = 8\mu sec$. On the other hand, with an RMS delay spread in the range $10ns - 512ns$, the symbol duration before applying OFDM technique $T = \frac{1}{10MHz}$ is either less than or in the range of the RMS delay spread, $T \simeq \sigma_\tau$, so the DSRC wireless channel would have been considered to be frequency-selective fading if OFDM technique had not been employed; however, frequency-selective fading effect is mitigated via applying OFDM scheme leading to have symbol duration $T_s = 8\mu sec$. In this case frequency-selective fading turns into frequency-flat fading for each OFDM symbol transmitting over the DSRC wireless channel.

In this chapter, the performance of the coded DSRC system is investigated based on the bit error rate (BER) for a given signal to noise ratio (SNR). Figs. 6.1, and 6.2 show the performance of the convolutional code $C_{conv}(2, 1, 6)$ and the LDPC (504, 1008, 3, 6) code for code rates $1/2, 2/3, 3/4$. The last two code rates were obtained by puncturing some of the parity bits after encoding the data. Some performance degradation is expected due to the higher code rate with puncturing, as there are fewer parity bits. Figs. 6.1, and 6.2 show that the LDPC code with rate $2/3$ is about $0.5 dB$ better than the convolutional code with the same code rate at $BER = 10^{-4}$. Figs. 6.3, 6.4, and 6.5 show that for an AWGN channel, the LDPC code provides an improvement of approximately $1 dB$ for LDPC (504, 1008, 3, 6), and approximately $1.5 dB$ for LDPC (2000, 4000, 3, 6), over convolutional coding at $BER = 10^{-4}$. The corresponding values remain the same for the Rayleigh and Rician channels. Thus it can be concluded that LDPC coding, even with a small block length of 1008, provides significantly better performance than convolutional coding. This performance improvement is very important considering that transmission power in DSRC systems is usually less than $33 dBm(2W)$ (but can be up to $44.8 dBm(30W)$ for public safety applications).

Figs. 6.6 and 6.7 show the performance of QC-LDPC codes of block lengths 1008

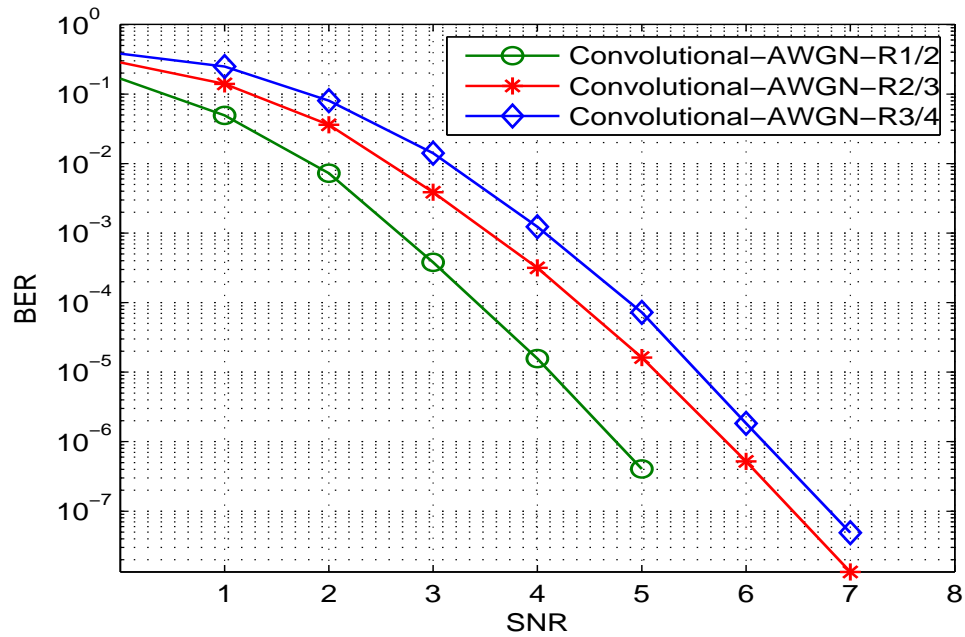


Figure 6.1: BER for a $K = 6$ convolutional code with different code rates in an AWGN channel.

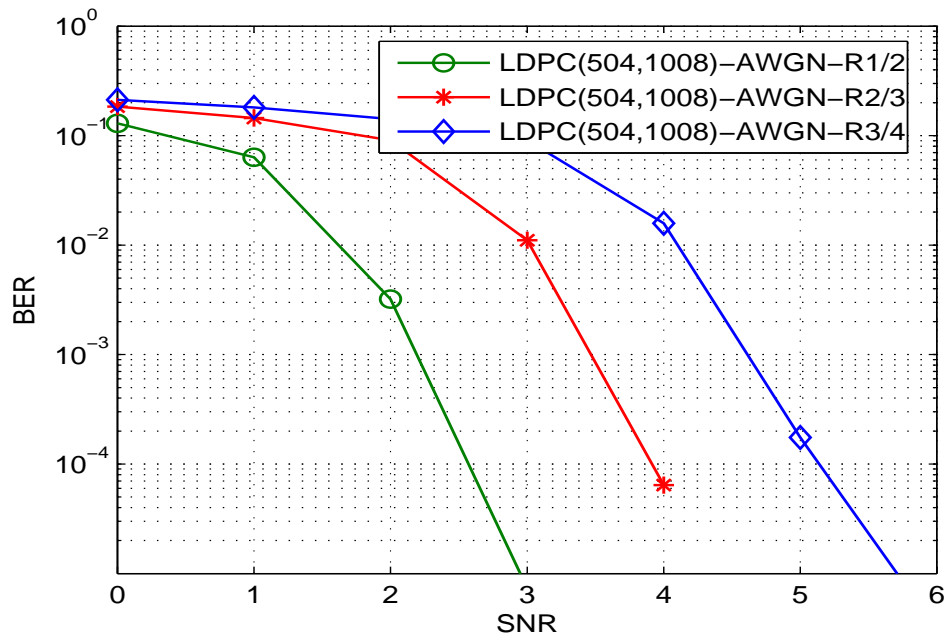


Figure 6.2: BER for an LDPC (504,1008,3,6) code with different code rates in an AWGN channel.

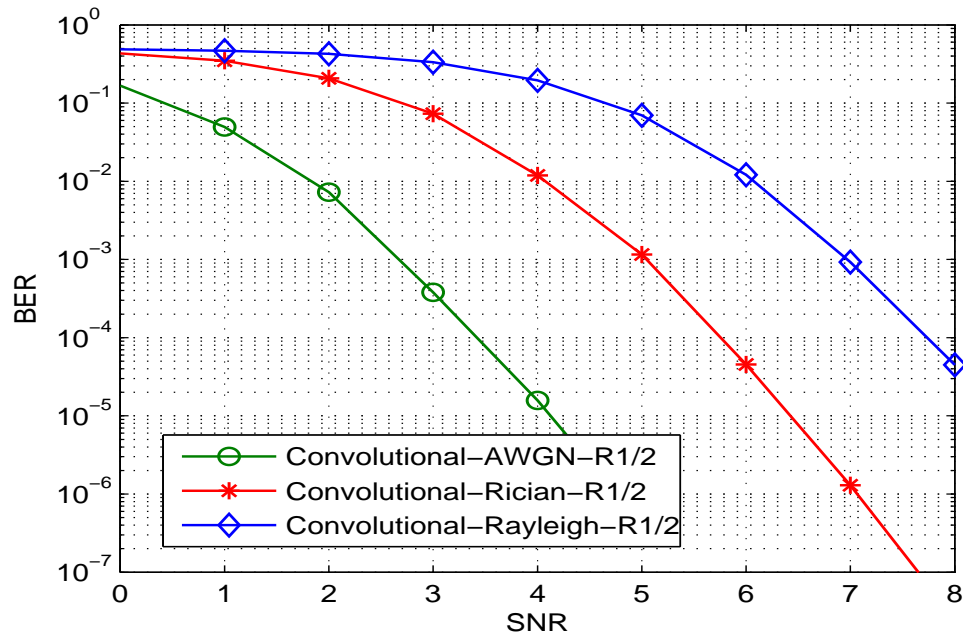


Figure 6.3: BER for a rate 1/2 convolutional code in three different channels.

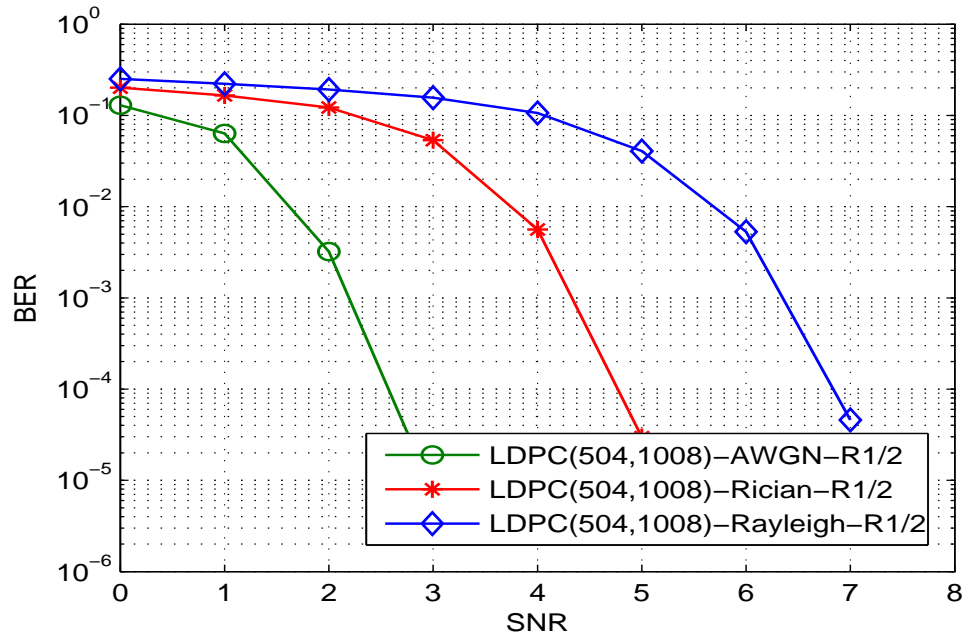


Figure 6.4: BER for an LDPC (504,1008,3,6) code in three different channels.

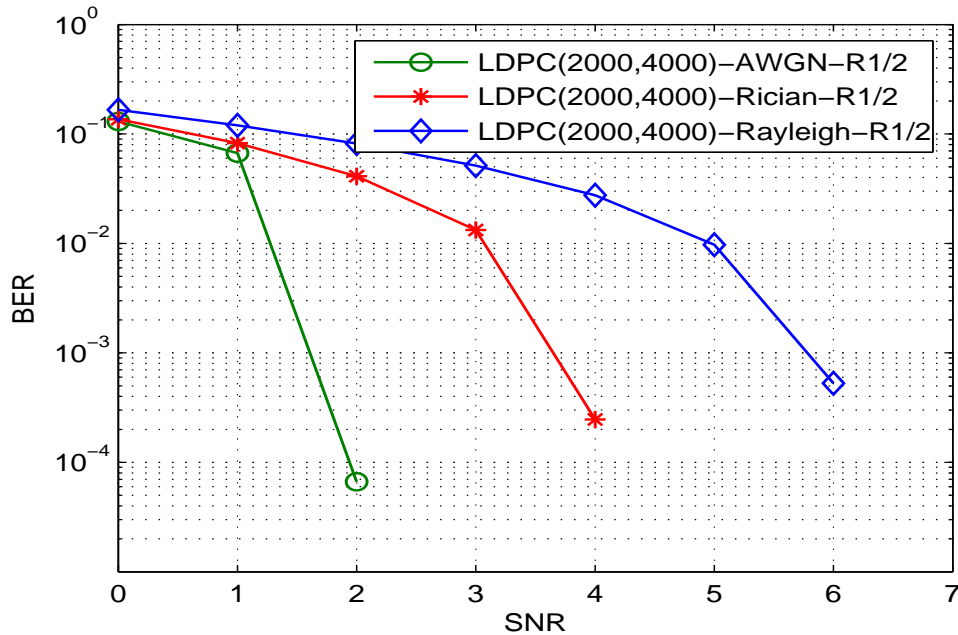


Figure 6.5: BER for an LDPC (2000,4000,3,6) code in three different channels.

and 4000 in two different types of fading channels. As both figures show the block length of the QC-LDPC code is chosen the same as LDPC code; however The performance of the QC-LDPC codes is similar to that of the LDPC codes in an AWGN channel, but there is a considerable difference in Rician and Rayleigh fading channels.

Fig. 6.8 shows the performance of the DSRC system in an AWGN channel with convolutional, LDPC and QC-LDPC codes. This shows that the LDPC codes are much better than the convolutional code, as expected, and there is almost no difference between the LDPC and QC-LDPC codes in AWGN channel. However, the QC-LDPC code has much lower complexity. Figs. 6.9 and 6.10 present the performance for Rayleigh and Rician channels, respectively. With a Doppler frequency of $10Hz$, the wireless channel is characterized as slow fading. The performance difference between the QC-LDPC codes with girth 6 and 8 is not significant, thus by having less complicated error correction coding technique we can achieve the same performance. For the Rayleigh fading channel at the BER of 10^{-3} , the performance improvement using LDPC (504, 1008, 3, 6) over the convolutional code is approximately 1 dB and this increases to 1.5 dB for LDPC (2000, 4000, 3, 6). The corresponding values are almost identical for the Rician fading channel by considering

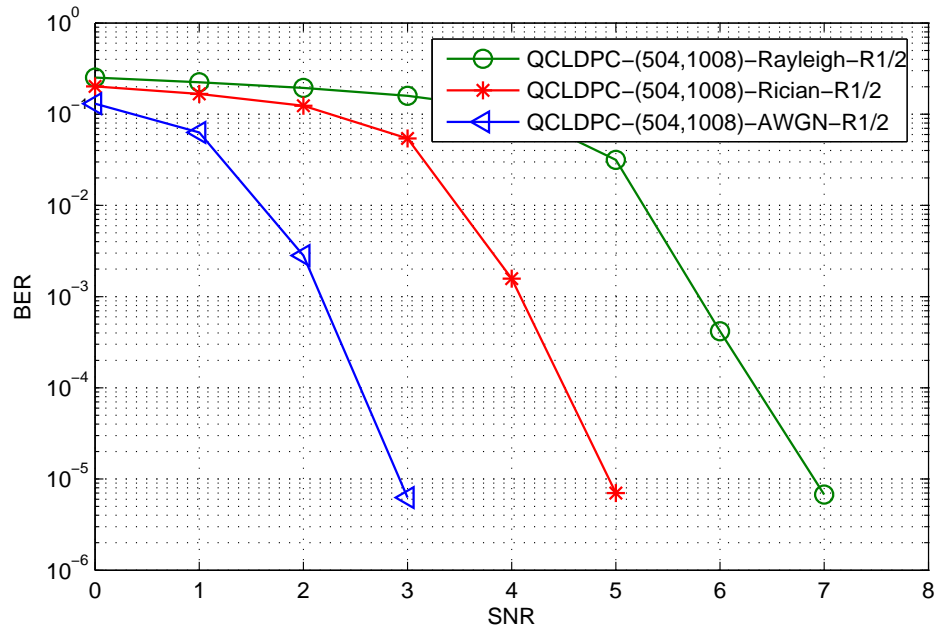


Figure 6.6: BER for a QC-LDPC (504,1008,3,6) code in three different channels.

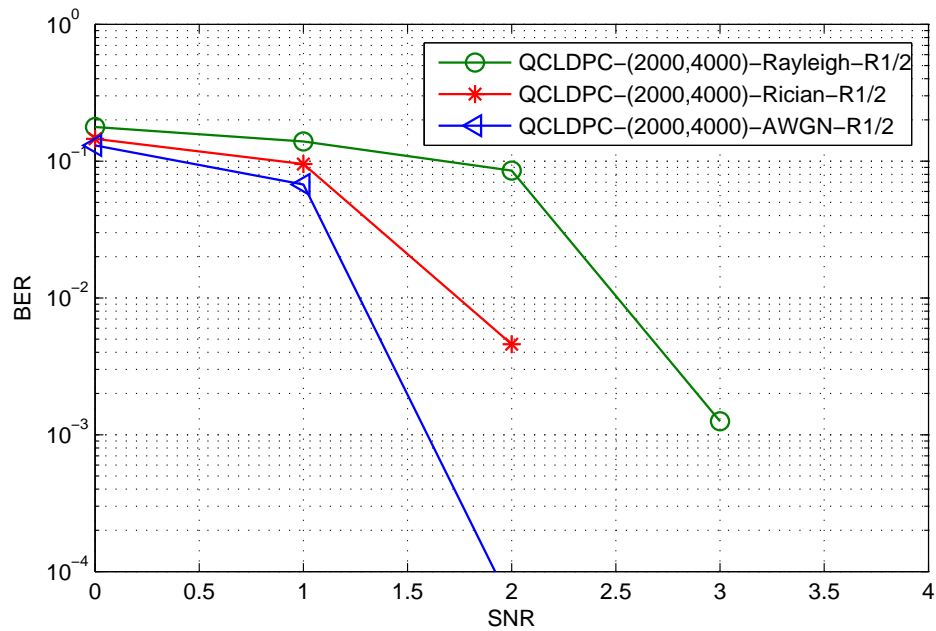


Figure 6.7: BER for a QC-LDPC (2000,4000,3,6) code in three different channels.

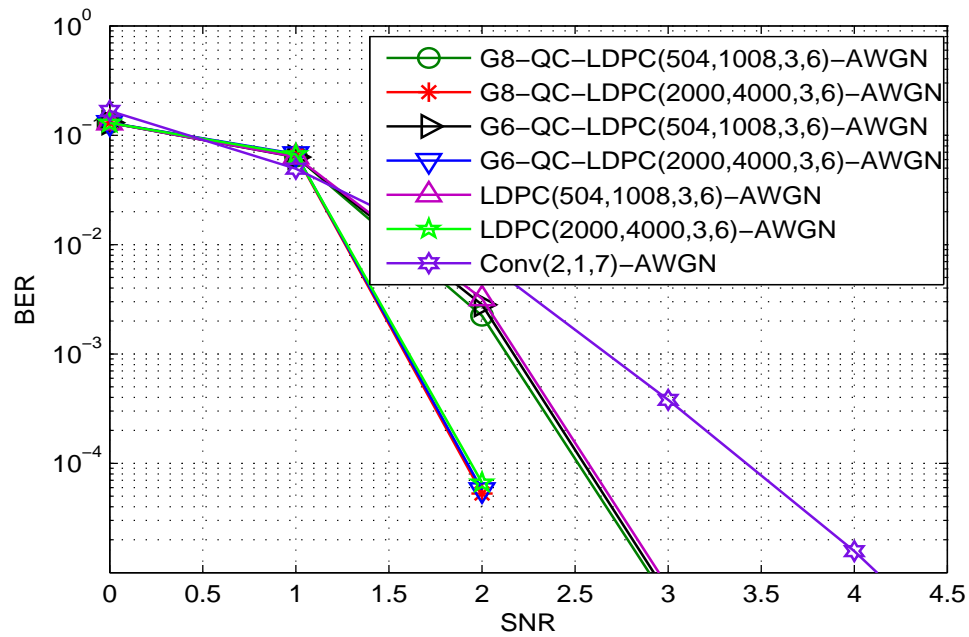


Figure 6.8: Performance result of three different coding methods in AWGN channel.

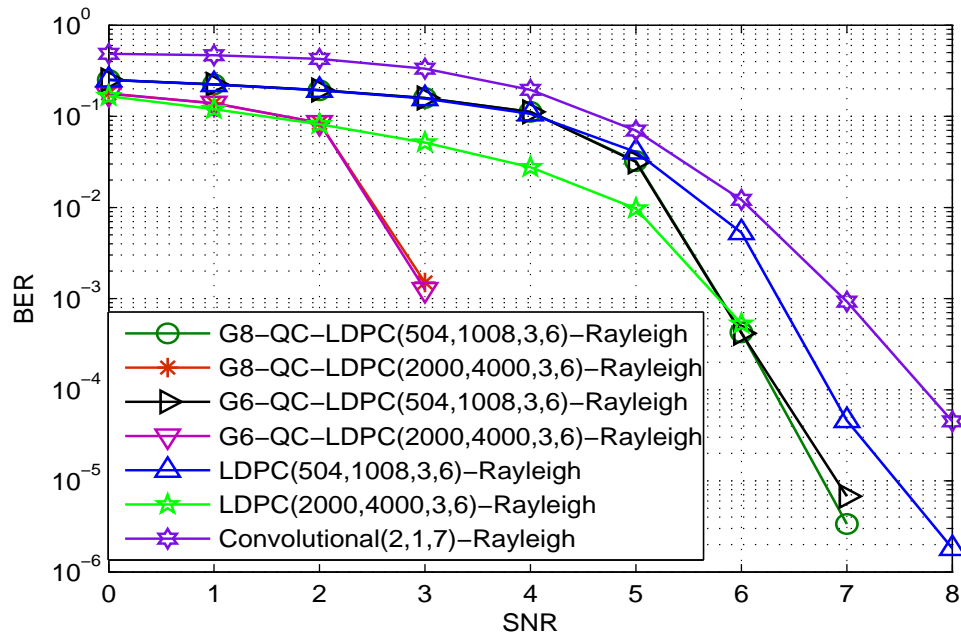


Figure 6.9: Performance of three different coding methods in a Rayleigh channel.

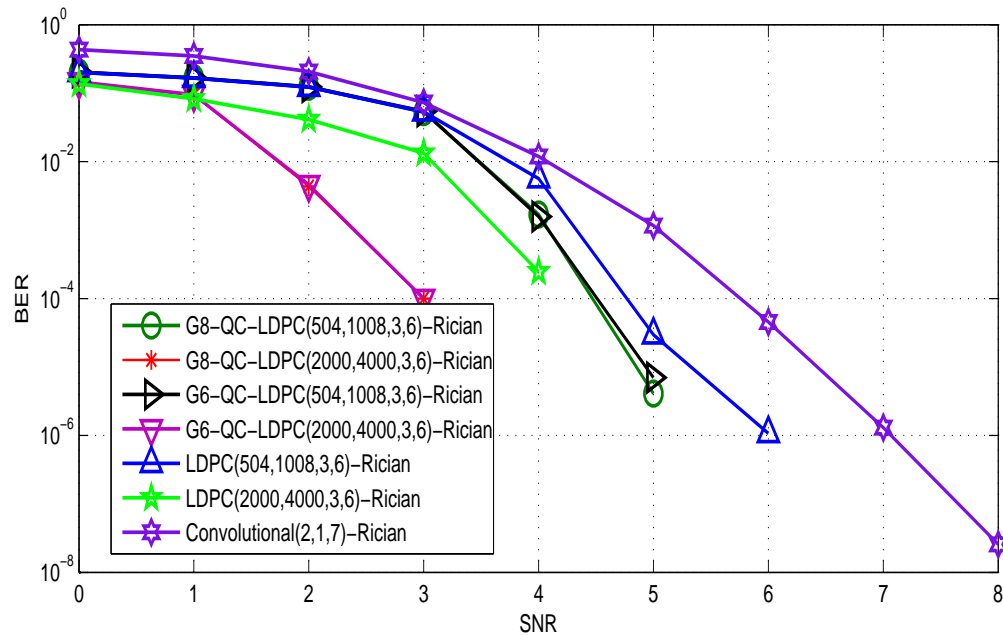


Figure 6.10: Performance of three different coding methods in a Rician channel.

$BER = 10^{-4}$. In a Rayleigh fading channel, an improvement of approximately 1.5 dB for the QC-LDPC (504, 1008, 3, 6) code and approximately 4 dB for the QC-LDPC (2000, 4000, 3, 6) code, over convolutional coding at $BER = 10^{-3}$ can be achieved. The corresponding values for a Rician channel are 1.5 dB and 2.8 dB , respectively at $BER = 10^{-4}$. This is significant in view of the fact that the QC-LDPC codes have a simpler implementation and lower complexity compared to the LDPC codes. As a result, approximately 3 dB better performance can be obtained in multi-path fading channels by applying a QC-LDPC code rather than a convolutional code. It is worth noting that the improvement with QC-LDPC codes over LDPC codes increases as the number of decoding iterations increases. For instance, with two iterations, the performance of the LDPC and QC-LDPC codes is almost the same, but the difference is very significant after 15 iterations.

As described earlier, a block fading channel with almost the same fading coefficient for all transmitted bits in a codeword is considered to examine the performance with burst errors, which occur in slow fading channels. As illustrated in Fig. 6.11, the QC-LDPC code has the greatest robustness against burst errors while the LDPC code is second, and the convolutional code has the least resistance to burst fading. By taking Rayleigh fading channel into account, LDPC code reaches its error floor

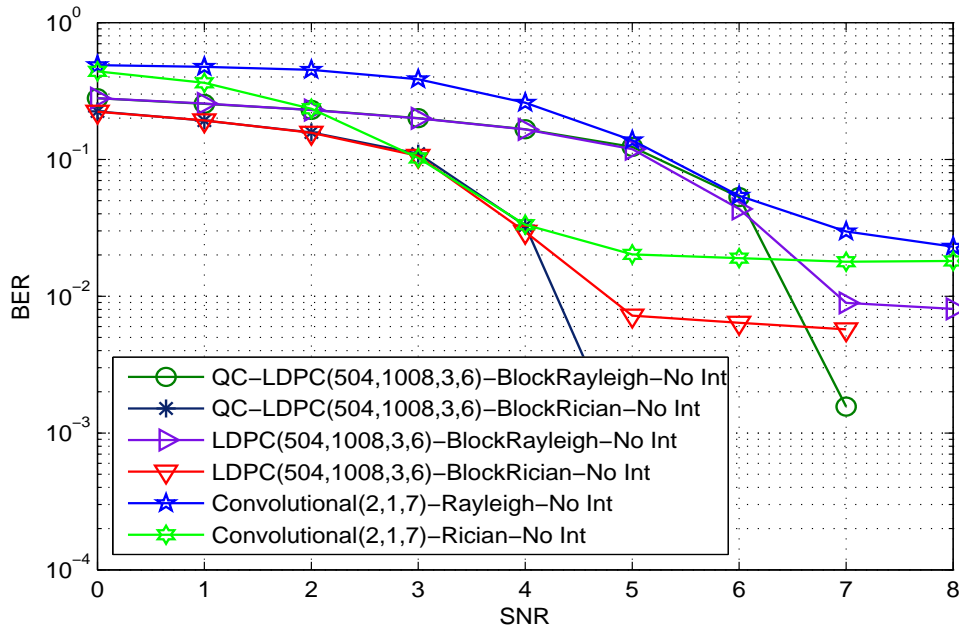


Figure 6.11: The effect of block fading on three different coding methods.

at $BER \simeq 10^{-2}$ while for the convolutional code this value is $BER \simeq 2 \times 10^{-2}$. Fig. 6.12 shows the effect of an interleaver on code performance. A random interleaver is employed which has a longer delay than the block interleaver specified in the DSRC standard. The most interesting effect of using an interleaver in the block fading channel is that unlike in an independent fading channel, the performance of the QC-LDPC code is approximately the same as the LDPC code. Therefore in block fading, both the LDPC and QC-LDPC codes show similar behaviour with an interleaver, however, the LDPC code seems more sensitive to the interleaver than QC-LDPC code based on the simulations results. Comparing the performance of the simulated block fading with independent fading for a specific fading channel (e.g. Rician), leads to the conclusion that this interleaver cannot entirely eliminate the block fading effects. The interleaver applied to this part of our simulations rearranges the input elements via a pseudo-random permutation. The number of input bits for the interleaver is the same for all three coding techniques.

As it is illustrated in the Fig. 6.8, there is no significant difference between the performance of the LDPC and QC-LDPC codes in AWGN; however, the better performance of QC-LDPC gets highlighted in presence of fading channel which is the point of interest of this thesis. The reason for this declaration is that the QC-LDPC

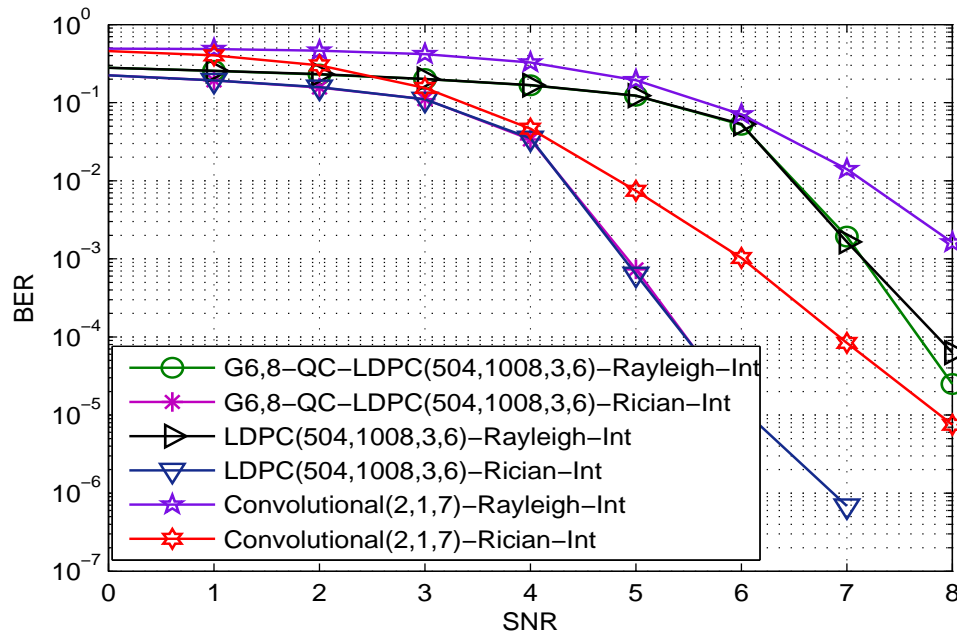


Figure 6.12: The effect of an interleaver for a small parity check matrix size under block fading.

codes as a subset of the more general class of LDPC codes is showing better performance in the more severe wireless channel. Therefore, there is not only performance penalty by restricting the LDPC codes to only QC-LDPC codes but also the better performance in particular defined DSRC channels can be achieved while the coding procedure complexity is definitely reduced.

In our thesis simulations, the DSRC channel is assumed to be slow fading that can be confirmed by considering the amount of the Doppler frequency which is set to its minimum amount $f_D = 10 \text{ Hz}$ as we described formerly. Accordingly, having a closer look at the slow fading characteristics can lead us to justify the better performance of QC-LDPC code in compare with LDPC code in the slow fading channel. A slow fading channel affects a particular number of transmitted bits with the same fading coefficient. The channel simulated in the last part of our thesis in order to show the effect of the interleaver, is characterized as block fading in which a specific number of transmitted bits is affected by the exact same fading coefficient.

The minimum distance can have a key role in coding performance. Based on [17] the minimum free distance of the convolutional code $C_{conv}(2, 1, 6)$ is equal to $D_f = 10$, while for the LDPC code $(504, 1008, 3, 6)$ the minimum distance between codewords

is $D_{min} = 34$ [18]. According to Chapter 5, the parity check matrix generation procedures for both LDPC codes and QC-LDPC codes are different.

The minimum distance of the codes in our simulations was not determined, in view of the fact that QC-LDPC parity check matrix generation is based on semi-random process. However, we can claim that the better performance in a slow fading channel by the QC-LDPC code in comparison with the LDPC code is related to its minimum distance. This is confirmed by Figs. 6.11 and 6.12.

As shown in Fig. 6.11, the error floor initially occurs for the convolutional code and then occurs for the LDPC code. In this figure the channel is considered very slow. The reason behind the error floor for different coding techniques is the minimum distance error events which dominate code performance at high SNRs. A better minimum distance for a QC-LDPC code explains its robustness in slow fading.

Chapter 7

Conclusions

In this thesis, the performance of the DSRC inter-vehicle communication standard was investigated with regular LDPC and QC-LDPC codes for forward error correction (FEC) in the physical layer. The simulation results presented in the previous chapter showed that LDPC and QC-LDPC codes provide a significant performance improvement over convolutional codes in both AWGN and fading channels.

The performance difference between QC-LDPC and LDPC codes in an AWGN channel is not as considerable as in multi-path fading channels. As shown in Chapter 6, between QC-LDPC and LDPC codes, not only can the former provide better performance in comparison with the latter, but also the complexity of QC-LDPC codes is lower due to its simple structure. However, the computational complexity of the LDPC (2000, 4000, 3, 6) and (504, 1008, 3, 6) codes is 2.3 times that of the convolutional code in the DSRC standard. Consequently the complexity is in the same range of the convolutional code or even less for the LDPC codes with the smaller parity check matrix size. If the performance improvement in fading channels is considered, the QC-LDPC (504, 1008, 3, 6) code provides at least 1.5 *dB* better performance than convolutional coding, and the corresponding value for the LDPC (504, 1008, 3, 6) code is 1 *dB*. For the larger block lengths, the performance improvement is even greater. The QC-LDPC (2000, 4000, 3, 6) and LDPC (2000, 4000, 3, 6) codes are at least 3 *dB* and 1.5 *dB* better, respectively, at a BER of 10^{-5} .

As the US FCC has designated 75 *MHz* of bandwidth in the 5.9 *GHz* band to support DSRC communications with a maximum permissible transmit power level of only 33 *dBm* [19], QC-LDPC and LDPC codes provide an attractive trade off between performance and computational complexity. In particular, the low complexity and very good performance of QC-LDPC codes can be exploited to save power.

7.1 Future Work

Based on the investigations and analysis performed in [20], low-density parity-check (LDPC) codes can provide performance very close to the Shannon limit, specifically for irregular LDPC codes with a large block size [9].

LDPC encoding is important for low complexity implementation, and is a topic for future research because in general the generator matrix for these code cannot be readily obtained. The encoding technique used in this thesis is based on the conventional systematic generator matrix. This matrix was obtained from the parity check matrix using Gaussian elimination, which has complexity of $O(n^3)$ [21]. The systematic form of the parity check matrix $H' = [P, I]$ was achieved by row permutations and some column permutations (if necessary) [9], from which the generator matrix $G = [I_k, P^T]$ was constructed. This method is applicable to any block code, but it does not retain the sparseness of the LDPC codes, so the encoding complexity can be approximated as $O(n^2)$, since it is essential a matrix multiplication.

There has been some research on improving LDPC encoding technique since the block length n is typically very large. One interesting approach that can be considered to make DSRC systems more efficient is suggested in [21]. The idea is to transform the parity check matrix H into an approximate lower triangular (ALT) form using row and column permutations only (but without any row additions), which preserves the sparseness of the matrix. Based on the analysis in [21], the encoding complexity can be reduced to $O(n + g^2)$, where g is the number of rows in H that cannot be put in triangular form by row and column permutations. If g is small, the encoding complexity is essentially linear, but this comes with a trade-off between encoding complexity and performance.

Implementing enhanced or improved decoding techniques for LDPC codes can be considered as another future work to improve the performance of DSRC systems. For example, in [22] a new LDPC decoder called Layered Belief-Propagation (LBP) is presented. In LBP, sequential message-passing schedules are used to update the nodes sequentially in contrast to Belief-Propagation that uses flooding (updates all variable nodes simultaneously using the previously generated check-to-variable messages, and then updates all check nodes simultaneously using the previously generated variable-to-check messages). Several studies including [22] show that sequential scheduling not only improves convergence in terms of number of iterations, but also outperforms the traditional flooding scheduling for a large number of iterations. The results in [23]

and [24] show that LBP converges twice as fast as flooding because the messages are updated using the most recent information available as opposed to updating several messages with the same out-dated information. LBP has a computation complexity per iteration which is 1.5 times simpler than BP.

Another method which can be considered to improve DSRC systems is the use of an LDPC convolutional code (LDPC-CC). LDPC-CCs have recently been introduced specifically for applications which require the ability to operate on random lengths of data, such as the Ethernet frame format. LDPC-CC encoding is simpler than for LDPC codes, as the complexity is between $O(n)$ and $O(n^2)$ [25]. An LDPC-CC encoder is very similar to a convolutional encoder in that the outputs at time t are a function of the present input and the encoder state [26]. The reason that this coding is attractive for DSRC systems is that it can be considered a middle ground between the current convolutional code in the standard and LDPC codes. In [26], it is argued that LDPC-CC codes are better suited to frame based communication systems such as the Ethernet Frame Format (*IEEE* 802.3) than their block counterparts because they can accommodate random sized frames, have simple encoding structures and can provide enhanced error protection for the first and last bytes of a frame.

Irregular LDPC codes can also be used to improve DSRC systems, as they can provide excellent error-correction capabilities close to the Shannon limit [20]. However, the major drawback of irregular LDPC codes versus regular LDPC codes is its lower decoding convergence [27]. The different bit node degrees in irregular LDPC codes contribute significantly to the decoding performance, but bit nodes with small degrees converge slower than ones with larger degrees.

Apart from improving DSRC system performance by improving the error control coding, employing a different channel model can also be considered in future work. The DSRC channel model used in this thesis was derived from the analysis performed in [5], which is based on Rician and Rayleigh fading channel models. However, a Nakagami distribution can also be used to characterize the physical properties of the DSRC fading channel, and it may be more suitable for some applications.

Bibliography

- [1] Y. Liu, F. Dion, and S. Biswas, “Dedicated Short-Range Wireless Communications for Intelligent Transportation System Applications: State of the Art,” *Transportation Research Record: Journal of the Transportation Research Board*, No. 1910, pp. 29–37, 2005.
- [2] M. Shulman, and R. Deering, “Vehicle Safety Communications in The United States,” No. 07–0010
- [3] A. Goldsmith, *Wireless Communications*, University of Cambridge, 2005.
- [4] J. Yin, T. ElBatt, G. Yeung, B. Ryu, S. Habermas, H. Krishnan, and T. Talty, “Performance Evaluation of Safety applications over DSRC Vehicular Ad Hoc Networks,” *Proc. ACM Int. Workshop on Vehic. Ad hoc Networks*, pp. 1–9, Oct. 2004.
- [5] J. Yin, G. Holland, T. ElBatt, F. Bai, H. Krishnan “*DSRC Channel Fading Analysis from Empirical Measurement*,” *ChinaCom. First Int. Conf. on Communications and Networking*, No. 10, pp.1–5, Oct. 2006.
- [6] IEEE802.11a, Institute of Electrical and Electronics Engineers (IEEE) International standard,
- [7] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall PTR, 2007.
- [8] ASTM Int., “Standard Specification for Telecommunications and Information Exchange Between Roadside and Vehicle Systems - 5.9 GHz Band Wireless Access in Vehicular Environments (WAVE) / Dedicated Short Range Communications (DSRC) Medium Access Control (MAC) and Physical Layer (PHY) Specifications,” ASTM E 2213–03, West Conshohocken, PA, 2003.

- [9] S. Lin, and D. J. Costello, *Error Control Coding*, John Wiley and Sons, 2004.
- [10] J. C. Moreira, P. G. Farrell, *Essential of Error-Control Coding*, Prentice-Hall, 2006.
- [11] D. W. Matolak, and S. G. Wilson, “Variable-Complexity Trellis Decoding of Binary Convolutional Codes,” *IEEE Transactions on Communications*, Vol. 44, No. 2, pp. 121–126, Feb. 1996.
- [12] C. Spagnol, W. Marnane, “A Class of Quasi-Cyclic LDPC Codes over $GF(2^m)$,” *IEEE Transactions on Communications*, Vol. 57, No. 9, pp. 2425–2427, Sep. 2009
- [13] G. Malema, and M. Liebelt, “Quasi-Cyclic LDPC Codes of Column-Weight Two Using a Search Algorithm,” *EURASIP Journal on Advances in Signal Processing*, Vol. 4, No. 45768 , pp. 1–8 , Dec. 2007.
- [14] M. P. C. Fossorier, “Quasi-Cyclic Low-Density Parity-Check Codes from Circulant Permutation Matrices,” *IEEE Transactions on Information Theory*, Vol. 50, No. 8, pp. 1788–1793, 2004.
- [15] H. Wymeersch, H. Steendam, and M. Moeneclaey, “*Log-domain Decoding of LDPC Codes over $GF(q)$* ,” *IEEE Int. Conf. Communications*, Vol. 2, No. 6, pp. 772–776, June 2004.
- [16] D. Oh, and K. K. Parhi, “Efficient Highly-Parallel Decoder Architecture for Quasi-Cyclic Low-Density Parity-Check Codes,” *IEEE Int. Symp. on Circuits and Systems*, No. 5, pp. 1855–1858, May 2007.
- [17] L. Mroueh, S. Rouquette-Leveil, G. Rekaya-Ben Othman, and J. C. Belfiore “On the Performance of the Golden Code in BICM-MIMO and in IEEE 802.11n Cases.” No. 11, pp.1544–548, Nov. 2007.
- [18] X. Y. Hu, and M. P. C. Fossorier, “*On the Computation of the Minimum Distance of Low-Density Parity-Check Codes*,” *IEEE Int. Conf. Communications*, Vol. 2, No. 6, pp.767–771, June 2004.
- [19] K. Hong, D. Xing, V. Rai, and J. Kenney, “Characterization of DSRC Performance as a Function of Transmit Power,” *ACM Int. Workshop on Vehic. Ad hoc Networks*, Sep. 2009.

- [20] T. J. Richardson, M. A. Shokrollahi, and R. L. Urbanke, “Design of Capacity-Approaching Irregular Low-Density Parity Check Codes,” *IEEE Transactions on Information Theory*, Vol. 47, No. 2, pp. 619–637, Feb. 2001.
- [21] H. Qi, and N. Goertz, “Low-Complexity Encoding of LDPC Codes: A New Algorithm and its Performance,”
- [22] Y. M. Chang, A. I. V. Casado, M. C. F. Chang, and R. D. Wesel, “*Lower-Complexity Layered Belief-Propagation Decoding of LDPC Codes*,” *IEEE Int. Conf. on Communications*, No. 5, pp. 1155–1160, May 2008.
- [23] E. Yeo, P. Pakzad, B. Nikolic, and V. Anantharam, “*High Throughput Low-Density Parity-Check Decoder Architectures*,” *Proc. Global Conf. on Communications*, Vol. 5, No. 11, pp. 3019-3024, Nov. 2001.
- [24] A. I. V. Casado, M. Griot, and R. D. Wesel, “*Informed Dynamic Scheduling for Belief-Propagation Decoding of LDPC Codes*,” *IEEE Int. Conf. on Communications*, No. 6, pp. 932-937, June 2007.
- [25] C. Schlegel, and L. Perez, “Trellis and Turbo Coding,” *IEEE*, 2004.
- [26] S. Bates, Z. Chen, and X. Dong; “*Low-Density Parity-Check Convolutional Codes for Ethernet Networks*,” *IEEE Pacific Rim Conf.*, No. 8, pp 85–88, Aug. 2005.
- [27] X. Chen, H. Yu, and Y. Xu “*Optimized Decoding Schedule for Irregular LDPC Codes*,” *IET Int. Communications Conf.*, No. 12, pp 261–264, Dec. 2009.