AN ABSTRACT OF THE THESIS OF

Yun Yang for the degree of Master of Science in Electrical and Computer Engineering presented on June 10, 2014

Title: Decoding Channel Coded Higher-order QAM Signals over Gaussian and Rayleigh Fading Channels

Abstract approved:

_____________________________________________________

Huaping Liu

In modern digital communication systems, error correction codes (ECC) are widely used and play an important role. The main effect of ECCs is to reduce the transmission error caused by channel noise, thereby protecting data and increasing the quality of information transmission. In addition, high spectral efficiency is desired in recent and future communication systems. In order to achieve the maximum spectral efficiency, higher-order modulation schemes were chosen in this thesis. However, for given bandwidth, higher-order modulations are very sensitive to noise and channel conditions. Channel coding was applied to decrease the data distortion brought by higher-order modulation schemes, and a coding rate close to 1 is highly preferred to ensure high spectral efficiency. We designed several system combinations of different modulation schemes, coding rates, decoders and channels to compare the required Eb/N0 (SNR) to achieve the same spectral efficiency and target BER. As a conclusion which we verified in this thesis, MAP decoder is the optimal decoder, especially when higher coding rate is applied with higher-order QAM with worse channel conditions.
Decoding Channel Coded Higher-order QAM Signals over Gaussian and Rayleigh Fading Channels

by

Yun Yang

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Master of Science

Presented June 10, 2014
Commencement June, 2015

APPROVED:

__________________________
Major Professor, representing Electrical and Computer Engineering

__________________________
Director of the School of Electrical Engineering and Computer Science

__________________________
Dean of the Graduate School

I understand that my thesis will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my thesis to any reader upon request.

__________________________
Yun Yang, Author
ACKNOWLEDGEMENTS

First, my deepest and most sincere appreciation to Dr. Huaping Liu as my major professor, his guidance and advice has been very important and helpful. Without his advising, I couldn’t complete this thesis work. Also, I’d like to express the same appreciation to all my committee members, Dr. Thinh Nguyen, Stephen Redfield and Dr. Abi Farsoni, with their timely help I was able to set up the final examination.

I must also thank all the group members in Dr. Liu’s group, Ruiqing Ye, Wei Liu, Tianzhu Qiao, Younghoon Whang and others. They provided me with very helpful information during my work and studies. The same to all my friends at OSU and Corvallis, Mr. Alan Coffman, Ms. Piper Jones, Shan Xue, RuiRui Zhang, Hui Feng, Lei Guo, Hua Yue, Jinzhou Cao, Chao Shi and Maureen McMahon, who warmly helped and support my life in Corvallis.

Also, the greatest appreciation goes to all my team members in Stress and Stability test team of Mobile and Wireless Group in Broadcom Corporation, especially to my manager Khurram Hafeez, my colleagues Qiong Shi, Ling Lou and others. Their strong supports for two years have given me the best work experience.

Last, the special appreciation for my parents, with their selfless dedications, I was able to survive in the US these years.

Thank you to everyone who helped me during this time.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 Introduction ........................................................................................................ 1</td>
</tr>
<tr>
<td>1.1 Goal of the Thesis .............................................................................................. 1</td>
</tr>
<tr>
<td>1.2 Thesis Outline .................................................................................................... 2</td>
</tr>
<tr>
<td>2 Digital Communication System ............................................................................. 3</td>
</tr>
<tr>
<td>2.1 System Overview .................................................................................................. 3</td>
</tr>
<tr>
<td>2.2 Coding and Decoding .......................................................................................... 5</td>
</tr>
<tr>
<td>2.2.1 Error Control Techniques ................................................................................ 5</td>
</tr>
<tr>
<td>2.2.2 Types of Code .................................................................................................. 6</td>
</tr>
<tr>
<td>2.2.3 Modification of Codes ...................................................................................... 7</td>
</tr>
<tr>
<td>2.3 Modulation and Demodulation ............................................................................ 7</td>
</tr>
<tr>
<td>2.3.1 Modulation ....................................................................................................... 8</td>
</tr>
<tr>
<td>2.3.2 Spectral Efficiency and Modulation Efficiency .............................................. 8</td>
</tr>
<tr>
<td>2.3.3 Digital Modulation ......................................................................................... 9</td>
</tr>
<tr>
<td>2.3.3.1 Phase-shift Keying (PSK) ............................................................................ 10</td>
</tr>
<tr>
<td>2.3.3.2 Amplitude-Shift Keying (ASK) .................................................................. 10</td>
</tr>
<tr>
<td>2.3.3.3 Quadrature Amplitude Modulation (QAM) ............................................... 11</td>
</tr>
<tr>
<td>2.3.3.3.1 Constellation Diagrams for QAM ....................................................... 12</td>
</tr>
<tr>
<td>2.3.3.3.2 QAM Bits per Symbol ........................................................................... 18</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (Continued)

2.3.3.3.3 QAM Noise Margin .......................................................................................................................... 19
2.3.3.3.4 High Order QAM Application ............................................................................................................. 20

2.4 Channels ....................................................................................................................................................... 21

2.4.1 Gaussian Channel ........................................................................................................................................ 21
2.4.2 Fading Channel .......................................................................................................................................... 21

2.4.2.1 Nakagami-m Fading Channel .................................................................................................................. 22
2.4.2.2 Rayleigh Fading Channel ....................................................................................................................... 22

2.5 Interleaving .................................................................................................................................................... 25

2.5.1 S-random Interleaving ............................................................................................................................... 26
2.5.2 Block Interleaving ....................................................................................................................................... 26
2.5.3 Prime Interleaving ...................................................................................................................................... 26
2.5.4 The Dilemma of Interleaving ...................................................................................................................... 26

3 Error Correcting Codes (Channel Coding) .................................................................................................... 28

3.1 Channel Encoding ......................................................................................................................................... 28

3.1.1 Convolution Code ...................................................................................................................................... 28
3.1.2 Convolution Encoder ................................................................................................................................. 28
3.1.3 Mother Code for High Rate Convolutional Code ....................................................................................... 29
3.1.4 Punctured Convolution Code .................................................................................................................... 29
3.1.5 Optimum Choice of Puncturing Pattern ..................................................................................................... 29

3.2 Optimal Decoding of Convolution Codes .................................................................................................. 31

3.2.1 Maximum Likelihood (ML) Decoding ......................................................................................................... 32
### TABLE OF CONTENTS (Continued)

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>3.2.2 Viterbi Decoding Algorithm</td>
<td>32</td>
</tr>
<tr>
<td>3.2.2.1 Hard-Decision Viterbi Decoding Algorithm</td>
<td>33</td>
</tr>
<tr>
<td>3.2.2.2 Soft-Decision Viterbi Decoding Algorithm</td>
<td>33</td>
</tr>
<tr>
<td>3.2.3 Maximum a posteriori Probability (MAP)</td>
<td>34</td>
</tr>
<tr>
<td>3.2.3.1 BCJR Algorithm</td>
<td>35</td>
</tr>
<tr>
<td>3.2.3.1.1 Computing Branch Metrics</td>
<td>35</td>
</tr>
<tr>
<td>3.2.3.1.2 Calculating Forward Metrics</td>
<td>35</td>
</tr>
<tr>
<td>3.2.3.1.3 Calculating Backward State Metrics</td>
<td>36</td>
</tr>
<tr>
<td>3.2.3.1.4 Computing Extrinsic L-values</td>
<td>36</td>
</tr>
<tr>
<td>3.2.3.2 Modifications of MAP Algorithm</td>
<td>36</td>
</tr>
<tr>
<td>3.2.3.2.1 Log-MAP Algorithm</td>
<td>36</td>
</tr>
<tr>
<td>3.2.3.2.3 Max-Log-MAP Algorithm</td>
<td>37</td>
</tr>
<tr>
<td>3.2.4 Difference of Path Search</td>
<td>38</td>
</tr>
<tr>
<td>3.2.5 Complexity Comparisons for Each Algorithm</td>
<td>40</td>
</tr>
<tr>
<td>4 Systems Comparison and Results Analysis</td>
<td>41</td>
</tr>
<tr>
<td>4.1 System Selection for Determinate Spectral Efficiency</td>
<td>41</td>
</tr>
<tr>
<td>4.2 Tradeoff among Different Coding Rates</td>
<td>43</td>
</tr>
<tr>
<td>4.3 System over Rayleigh Fading Channel</td>
<td>45</td>
</tr>
<tr>
<td>5 Conclusion</td>
<td>47</td>
</tr>
<tr>
<td>Bibliography</td>
<td>48</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>Basic Structure of a Digital Communication System</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>Gray Coding</td>
<td>12</td>
</tr>
<tr>
<td>2.3</td>
<td>64 QAM Constellation</td>
<td>13</td>
</tr>
<tr>
<td>2.4</td>
<td>64 QAM Constellation over AWGN with SNR =10 (lower SNR)</td>
<td>14</td>
</tr>
<tr>
<td>2.5</td>
<td>64 QAM Constellation over AWGN with SNR =30 (higher SNR)</td>
<td>15</td>
</tr>
<tr>
<td>2.6</td>
<td>256 QAM Constellation</td>
<td>16</td>
</tr>
<tr>
<td>2.7</td>
<td>256 QAM Constellation over AWGN with SNR =10 (lower SNR)</td>
<td>17</td>
</tr>
<tr>
<td>2.8</td>
<td>256 QAM Constellation over AWGN with SNR =30 (higher SNR)</td>
<td>18</td>
</tr>
<tr>
<td>2.9</td>
<td>Rayleigh Fading Channel</td>
<td>25</td>
</tr>
<tr>
<td>3.1</td>
<td>Rate 1/2 Constraint Length K=7 Convolution Encoder</td>
<td>29</td>
</tr>
<tr>
<td>3.2</td>
<td>Comparisons between Log-MAP, Max-Log-MAP and SOVA</td>
<td>39</td>
</tr>
<tr>
<td>3.3</td>
<td>Relationship between MAP, Log-MAP, Max-Log-MAP and SOVA</td>
<td>39</td>
</tr>
<tr>
<td>4.1</td>
<td>System Result for Spectral Efficiency = 4(bits/s)/Hz</td>
<td>42</td>
</tr>
<tr>
<td>4.2</td>
<td>System Results Comparison for Spectral Efficiency = 5.33 (bits/s)/Hz</td>
<td>43</td>
</tr>
<tr>
<td>4.3</td>
<td>Tradeoff among Different Coding Rates for 256-QAM</td>
<td>44</td>
</tr>
<tr>
<td>4.4</td>
<td>Performance for 1024-QAM with rate 15/16 over AWGN</td>
<td>45</td>
</tr>
<tr>
<td>4.5</td>
<td>Performance for System over Rayleigh Fading Channel</td>
<td>46</td>
</tr>
</tbody>
</table>
LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Bit Rates of Different Forms of QAM and PSK</td>
<td>19</td>
</tr>
<tr>
<td>3.1 Best PCC with Best Puncturing Pattern</td>
<td>30</td>
</tr>
<tr>
<td>3.2 Optimum Puncturing Matrixes for (133,171)</td>
<td>31</td>
</tr>
<tr>
<td>3.3 Complexity Comparisons for Each Algorithm</td>
<td>40</td>
</tr>
<tr>
<td>4.1 Spectral Efficiency for Higher Order QAM at Different Coding Rate</td>
<td>41</td>
</tr>
</tbody>
</table>
Chapter 1 Introduction

1.1 Goal of the Thesis

Error correction codes (ECCs) have been broadly used in modern digital communication and storage systems to increase the quality of information transmission by protecting data and reducing the error caused by channel noise [1]. One of the two main types of ECCs is the convolutional code. In communication systems, convolutional codes are mainly used for real-time error correction and can be applied to non-binary modulations, such as quadrature amplitude modulation (QAM) [3]. In order to support a certain specified coding rate and to decrease the decoding complexity, puncturing and de-puncturing techniques are often used. In this thesis, codes of various rates are constructed based on a (2, 1, 7) convolutional code as a mother code, and “puncturing” is used to design codes with rates close to 1.

If a high data rate achievement is required, higher-order modulation schemes are often used. For example, 64-QAM is applied in IEEE 802.11a.e. For most application scenarios that require a high spectral efficiency, higher-order modulation is not applied with codes of low rates. However, not providing any channel encoding is often not acceptable. If spectral efficiency is of importance, higher-order modulation can be used with high-rate codes.

In order to accurately decode higher-order QAM with a high data rate, we studied two optimal decoders. The Viterbi algorithm is known to be the main decoding method for convolutional codes over noisy digital communication channels, which was introduced by Andrew Viterbi in 1967 [4]. The basic idea of Viterbi decoding algorithm is to acquire the most likely decoding sequence for an input data stream, which is the Maximum Likelihood (ML) decoding algorithm. Later on, a soft-decision
Viterbi decoder was presented [32], which takes into account the \textit{a priori} probability of input symbols. Numerous existing analyses are base on this. Therefore we first reviewed and applied it in this thesis. Furthermore, we applied a modified maximum \textit{a posteriori} probability (MAP) decoder, which determines the most likely transmitted bits in order to obtain better performance at the bit level [2]. We expected to see whether there are any differences between these two decoding algorithms along with required spectral efficiency and Eb/N0 for target BER.

1.2 Thesis Outline

The basic construction of a modern digital communication system is reviewed in Chapter 2, which includes information about channel coding and decoding, modulation and demodulation, and transmission channels. This was also used as the system model in this thesis. Chapter 3 introduces the relevant backgrounds about error correcting codes (ECC), such as convolutional encoding, maximum likelihood decoding and MAP decoding algorithm, which were applied here. In Chapter 4, the simulation results and analyses of designed systems are presented. After we compare the different observations, the optimal systems are selected, depending on diverse conditions and situations. The last chapter provides a conclusion.
Chapter 2 Digital Communication System

2.1 System Overview

Nowadays, digital communication systems are becoming more popular for transmitting data information. Figure 2.1 shows the simple and basic structure of a digital communication system [2]. In a digital system, the information is produced as source signal, which is represented by a symbol sequence of finite discrete messages (digital signal) or infinite continuous messages (analog signal). First, the source signal goes through the source encoder, to convert into digital signal. Then, a channel encoder codes the information sequence so that it can recover the correct information after passing through the channel. The information sequence is protected by error correction codes such as convolutional codes. The digital modulator maps the binary sequence onto analog signal waveforms in order to be efficiently transmitted over the communication channel. The modulator acts as an interface between the digital signal and the channel.

The communication channel is the physical medium that is used to send the signal from the transmitter to the receiver. The channel attenuates the transmitted signal and introduces noise. The attenuation is generally caused by energy absorption and scattering in the propagation medium. The noise is generated in a random manner by many possible mechanisms such as ambient heat in the transmitter/receiver hardware and the propagation medium hardware-induced transients, co-channel and adjacent-channel interference from other communication systems, or climatic phenomena. The most commonly assumed noise models are the additive white Gaussian noise (AWGN) model and Rayleigh fading channel.
As the receiver of the digital communications system, the digital demodulator processes the channel-corrupted transmitted waveform and recovers a sequence of digital values from the waveforms, then feeds it into the channel decoder. The decoder reconstructs the original information sequence based on the knowledge of the code used by the channel encoder and the redundancy contained in the received data. Channel decoders can be ML decoders such as Viterbi decoders, and MAP decoders, etc. Then, the source decoder decompresses the data and retrieves the original information. The probability of having error in the output sequence is a function of the code characteristics, the type of modulation, and of channel characteristics such as noise and interference level, etc. There is a trade-off between the power of transmission and the bit error rate. Researchers are trying to minimize the power consumption while maintaining a reliable communication. This raises a need for stronger codes with more error correction abilities.

Figure 2.1 Basic Structure of a Digital Communication System

In this thesis, as our focus was not on the source encoder and decoder, we provided an input signal with randomly generated 0s and 1s to simulate that the signal
is encoded by a source encoder. The decoding data from the channel decoder was directly compared with the input signal to get bit error rate performance for analyzing.

2.2 Coding and Decoding

In the 1948 paper “A Mathematical Theory of Communication” by Shannon, a metric that can quantify the information was introduced [5]. This theory provides a way to determine the minimum possible number of symbols necessary for the error-free representation of a given message. A longer message containing the same information is said to have redundant symbols. These can lead to the definition of three distinct types of coding: source coding, secrecy coding and channel coding.

Source coding is used to remove the uncontrolled redundancy from the information symbols. It reduces the symbol throughput requirement placed upon the transmitter.

Secrecy coding encrypts the information so that it cannot be understood by anyone else who is not the intended recipient.

Channel coding is used to format the transmitted information so that it can increase its immunity to noise. This is achieved by adding controlled redundant information into the transmitted information stream, allowing the receiver to detect and possibly correct those errors.

As we mentioned before, in a communication system, all three types of these codes are used to increase the reliability and performance of the system. However, in this thesis, our research only focused on channel coding in the designed system.

2.2.1 Error Control Techniques

There are two main types of Error Control techniques: Error Correcting
schemes and Error Detecting schemes [34]. The Error Correcting schemes include a 
Backward Error Correction (BEC) and Forward Error Correction (FEC). In the Error 
Detection schemes, the Repetition Codes, the Parity Check Codes, the Checksum, the 
Cyclic Redundancy Check (CRC), and the Automatic Repeat reQuest (ARQ) are 
included. Here, we will briefly introduce one Error Control technique which is the 
Forward Error Correction.

The main idea of Forward Error Correction (FEC) is that the sender encodes 
their message in a redundant way by using an error-correcting code (ECC). The 
redundancy allows the receiver to detect a limited number of errors that may occur 
anywhere in the message, and often to correct these errors without retransmission. 
FEC gives the receiver the ability to correct errors without needing a reverse channel 
to request retransmission of data, but at the cost of a fixed, higher forward channel 
bandwidth. FEC is therefore applied in situations where retransmissions are costly or 
impossible, such as one-way communication links, and when transmitting to multiple 
receivers in multicast. FEC information is usually added to mass storage devices to 
enable recovery of corrupted data, and is widely used in modems.

2.2.2 Types of FEC Code

The two basic types of FEC codes are Block codes and Convolution codes. In a 
block code all code words have the same length, and the encoding for each possible 
data message can be statically defined. In a convolution code, the code word produced 
depends on both the data message itself and a given number of previously encoded 
messages: the encoder changes its state with every message processed. The length of 
the code words is usually constant. We can further distinguish among linear codes, 
cyclic codes and systematic codes.
Linear and cyclic block codes are the most commonly used codes in data communication protocols. In a linear code, every linear combination of valid code words (such as a modulo-2 sum) produces another valid code word. A cyclic code is a code in which every cyclic shift of a valid code word also produces a valid code word.

A systematic code, finally, is a code in which each code word includes the data bits from the original message unaltered, either followed or proceeded by a separate group of check bits.

In all cases, the code words are longer than the data words on which they are based. If the number of original bits is \(d\) and the number of additional bits is \(e\), the ratio \(d/(d + e)\) is called the code rate. Improving the quality of a code often means increasing its redundancy and thus lowering the code rate. To reduce the channel error rate by a factor of 5.102 by forward error control, for instance, may require a code with a code rate of 0.5 or less.

2.2.3 Modification of Codes

Extension: A \((n, k, d)\) systematic linear code can be modified to a \((n + 1, k, \geq d)\) linear code by adding an overall parity-check bit. If \(d\) is odd, then we can get a \((n + 1, k, d+ 1)\) code.

Shortening: A \((n, k)\) systematic linear code can be modified to a \((n − l, k − l)\) linear code by deleting \(l\) message bit. The minimum distance of a shortened code is not less than the original code.

Puncturing: A \((n, k)\) systematic linear code can be modified to a \((n − l, k)\) linear code by deleting \(l\) parity bit. The minimum distance of a punctured code is not greater than the original code.

2.3 Modulation and Demodulation
2.3.1 Modulation

Modulation is the process of conveying a message signal. For example, a digital bit stream or an analog audio signal can be physically transmitted inside another signal. Modulation of a sine waveform is used to transform a baseband message signal into a pass-band signal. For example, low-frequency audio signal can be transformed into a radio-frequency signal (RF signal). In radio communications, cable TV systems or the public switched telephone network, electrical signals can only be transferred over a limited pass-band frequency spectrum with specific (non-zero) lower and upper cutoff frequencies. Modulating a sine-wave carrier makes it possible to keep the frequency content of the transferred signal as close as possible to the center frequency (typically the carrier frequency) of the pass-band. There are two types of modulation, analog modulation and digital modulation.

2.3.2 Spectral Efficiency and Modulation Efficiency

In addition, if higher spectral efficiency is desired, higher-order modulation schemes will be preferred [31]. A coding rate close to 1 is essential to ensure that the spectral efficiency of the system is extremely close to the upper bound.

The spectral efficiency of a digital communication system is measured in bit/s/Hz or, less frequently but unambiguously, in (bit/s)/Hz. It is the net bitrate (useful information rate excluding error-correcting codes) or maximum throughput divided by the bandwidth in hertz of a communication channel or a data link. Alternatively, the spectral efficiency may be measured in bit/symbol, which is equivalent to bits per channel use (bpcu), implying that the net bit rate is divided by the symbol rate (modulation rate) or line code pulse rate.

Spectral efficiency is typically used to analyze the efficiency of a digital
modulation method or line code, sometimes in combination with a forward error correction (FEC) code and other physical layer overhead. In the latter case, a “bit” refers to a user data bit; FEC overhead is always excluded.

The modulation efficiency in bit/s is the gross bitrate (including any error-correcting code) divided by the bandwidth.

An upper bound for the attainable modulation efficiency is given by the Nyquist rate or Hartley’s law as follows: For a signaling alphabet with M alternative symbols, each symbol represents $N = \log_2 M$ bits. N is the modulation efficiency measured in bit/symbol or bpcu. In the case of baseband transmission (line coding or pulse-amplitude modulation) with a baseband bandwidth (or upper cut-off frequency) B, the symbol rate cannot exceed 2B symbols/s in view to avoid inter-symbol interference. Thus, the spectral efficiency cannot exceed 2N (bit/s)/Hz in the baseband transmission case. In the pass-band transmission case, a signal with pass band bandwidth W can be converted to an equivalent baseband signal (using under sampling or a super heterodyne receiver), with upper cut-off frequency W/2. If double-sideband modulation schemes such as QAM, ASK, PSK are used, this results in a maximum symbol rate of W symbols/s, and in that the modulation efficiency cannot exceed N (bit/s)/Hz. If digital single-sideband modulation is used, the pass-band signal with bandwidth W corresponds to a baseband message signal with baseband bandwidth W, resulting in a maximum symbol rate of 2W and an attainable modulation efficiency of 2N (bit/s)/Hz.

2.3.3 Digital Modulation

Modulation refers to the representation of digital information in terms of analog waveforms that can be transmitted over physical channels. The most common
digital modulation techniques are: Phase-shift keying (PSK), Frequency-shift keying (FSK), Amplitude-shift keying (ASK), Quadrature amplitude modulation (QAM), and Orthogonal frequency-division multiplexing (OFDM).

2.3.3.1 Phase-shift Keying (PSK)

PSK is a digital modulation scheme that conveys data by changing, or modulating, the phase of a reference signal. PSK uses a finite number of phases. Each is assigned a unique pattern of binary digits. Usually, each phase encodes an equal number of bits. Each pattern of bits forms the symbol that is represented by the particular phase. The demodulator, which is designed specifically for the symbol-set used by the modulator, determines the phase of the received signal and maps it back to the symbol it represents, thus recovering the original data. This requires the receiver to be able to compare the phase of the received signal to a reference signal — such a system is termed coherent (and referred to as CPSK). Alternatively, instead of operating with respect to a constant reference wave, the broadcast can operate with respect to itself. Changes in the phase of a single broadcast waveform can be considered the significant items. In this system, the demodulator determines the changes in the phase of the received signal rather than the phase (relative to a reference wave) itself. Since this scheme depends on the difference between successive phases, it is termed differential phase-shift keying (DPSK). DPSK can be significantly simpler to implement than ordinary PSK since there is no need for the demodulator to have a copy of the reference signal to determine the exact phase of the received signal (it is a non-coherent scheme). In exchange, it produces more erroneous demodulation.

2.3.3.2 Amplitude-Shift Keying (ASK)
Amplitude-shift keying (ASK) is a form of amplitude modulation that represents digital data as variations in the amplitude of a carrier wave. ASK uses a finite number of amplitudes, each assigned a unique pattern of binary digits. Usually, each amplitude encodes an equal number of bits. Each pattern of bits forms the symbol that is represented by the particular amplitude. The demodulator, which is designed specifically for the symbol-set used by the modulator, determines the amplitude of the received signal and maps it back to the symbol it represents, thus recovering the original data. The frequency and phase of the carrier are kept constant. Like AM, ASK is also linear and sensitive to atmospheric noise, distortions, and propagation conditions on different routes in PSTN, etc. Both ASK modulation and demodulation processes are relatively inexpensive.

The simplest and most common form of ASK operates as a switch, using the presence of a carrier wave to indicate a binary one and its absence to indicate a binary zero. This type of modulation is called on-off keying, and is used at radio frequencies to transmit Morse code.

2.3.3.3 Quadrature Amplitude Modulation (QAM)

Quadrature amplitude modulation (QAM) is both an analog and a digital modulation scheme. It conveys two analog message signals, or two digital bit streams, by changing (modulating) the amplitudes of two carrier waves, using the amplitude-shift keying (ASK) digital modulation scheme or amplitude modulation (AM) analog modulation scheme. The two carrier waves, usually sinusoids, are out of phase with each other by 90° and are thus called quadrature carriers or quadrature components — hence the name of the scheme. The modulated waves are summed, and the resulting waveform is a combination of both phase-shift keying (PSK) and amplitude-shift
keying (ASK), or (in the analog case) of phase modulation (PM) and amplitude modulation (AM).

In the digital QAM case, a finite number of at least two phases and at least two amplitudes are used. PSK modulators are often designed using the QAM principle, but are not considered QAM since the amplitude of the modulated carrier signal is constant. QAM is used extensively as a modulation scheme for digital telecommunication systems. Arbitrarily high spectral efficiencies can be achieved with QAM by setting a suitable constellation size, limited only by the noise level and linearity of the communications channel.

By moving to a higher-order constellation, it is possible to transmit more bits per symbol. However, if the mean energy of the constellation is to remain the same (by way of making a fair comparison), the points must be closer together and are thus more susceptible to noise and other corruption. This results in a higher bit error rate and so higher-order QAM can deliver more data less reliably than lower-order QAM, for constant mean constellation energy. Using higher-order QAM without increasing the bit error rate requires a higher signal-to-noise ratio (SNR) by increasing signal energy, reducing noise, or both.

2.3.3.3.1 Constellation Diagrams for QAM

The symbol, which each location on the constellation represents, is chosen using a technique known as Gray Coding. Gray Coding ensures that any adjacent location will only be one bit different, reducing the effect of an error.
Adjacent Locations only differ by one bit in Figure 2.2. By adding more levels to I and Q channels, higher data rates can be carried. The higher the number of levels, the more effect there will be from noise or interference.

64 QAM uses 8 levels in the I direction and 8 levels in the Q direction for a total of 8 squared or 64 symbols. Since QAM is a form of double sideband pass-band transmission, the spectral efficiency cannot exceed 6 (bit/s)/Hz.
If the signal is passing through a noisy channel, AWGN for example, below 64 QAM will be observed according to different SNR (Figure 2.4 and Figure 2.5).

Figure 2.4 64 QAM Constellation over AWGN with SNR =10 (lower SNR)
Figure 2.5 64 QAM Constellation over AWGN with SNR =30 (higher SNR)
256 QAM uses 16 levels in the I direction and 16 levels in the Q direction for a total of 16 squared, or 256 symbols. Each symbol can represent eight bits. Since QAM is a form of double sideband pass-band transmission, the spectral efficiency cannot exceed 8 (bit/s)/Hz. A 256 QAM signal that can send eight bits per symbol is very spectrally efficient. However, the symbols are very close together and are thus more subject to errors due to noise and distortion. Such a signal may have to be transmitted with extra power (to effectively spread the symbols out more) and this reduces power efficiency as compared to simpler schemes.

We also sent the 256 QAM signal through the AWGN channel to obtain below noisy constellations by different SNR levels (Figure 2.7 and Figure 2.8).
Figure 2.7: 256 QAM Constellation over AWGN with SNR = 10 (lower SNR)
2.3.3.3.2 QAM Bits per Symbol

The advantage of using QAM is that it is a higher order form of modulation and as a result it is able to carry more bits of information per symbol. By selecting a higher order format of QAM, the data rate of a link can be increased.

Table 2.1 below gives a summary of the bit rates of different forms of QAM and PSK. To understand and compare different modulation format efficiencies, it is important to first understand the difference between bit rate and symbol rate. The signal bandwidth for the communications channel needed depends on the symbol rate,
not on the bit rate. Symbol rate = bit rate / the number of bits transmitted with each symbol, where bit rate is the frequency of a system bit stream. Take, for example, a radio with an 8 bit sampler, sampling at 10 kHz for voice. The bit rate, the basic bit stream rate in the radio, would be eight bits multiplied by 10K samples per second or 80 Kbits per second.

<table>
<thead>
<tr>
<th>Modulation</th>
<th>Bits per Symbol</th>
<th>Symbol Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPSK</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>QPSK</td>
<td>2</td>
<td>½</td>
</tr>
<tr>
<td>8PSK</td>
<td>3</td>
<td>1/3</td>
</tr>
<tr>
<td>16 QAM</td>
<td>4</td>
<td>¼</td>
</tr>
<tr>
<td>32 QAM</td>
<td>5</td>
<td>1/5</td>
</tr>
<tr>
<td>64 QAM</td>
<td>6</td>
<td>1/6</td>
</tr>
<tr>
<td>128 QAM</td>
<td>7</td>
<td>1/7</td>
</tr>
<tr>
<td>256 QAM</td>
<td>8</td>
<td>1/8</td>
</tr>
</tbody>
</table>

Table 2.1 Bit Rates of Different Forms of QAM and PSK [2]

2.3.3.3 QAM Noise Margin

While higher order modulation rates are able to offer much faster data rates and higher levels of spectral efficiency for the radio communications system, this comes at a price. The higher order modulation schemes are considerably less resilient to noise and interference.

As a result of this, many radio communications systems now use dynamic adaptive modulation techniques. They sense the channel conditions and adapt the modulation scheme to obtain the highest data rate for the given conditions. As signal to noise ratios decrease, errors will increase along with re-sends of the data, thereby
slowing throughput. By reverting to a lower order modulation scheme, a more reliable link can be made with fewer data errors and re-sends.

2.3.3.3.4 High Order QAM Application

The main advantage is increased capacity, or higher throughput. However, capacity improvement diminishes with every higher modulation step, so the real capability of a higher-order modulation alone to address the objective of increasing capacity is very limited. Other techniques will be needed.

64-QAM and 256-QAM are often used in digital cable television and cable modem applications. In the United States, 64-QAM and 256-QAM are the mandated modulation schemes for digital cable as standardized by the SCTE in the standard ANSI/SCTE 07 2000. In the UK, 64-QAM is used for digital terrestrial television whilst 256-QAM is used for Freeview-HD.

Communication systems designed to achieve very high levels of spectral efficiency usually employ very dense QAM constellations. For example, current Home plug AV2 500-Mbit power line Ethernet devices use 1024-QAM and 4096-QAM, as well as future devices using ITU-T G.hn standard for networking over existing home wiring (coaxial cable, phone lines and power lines); 4096-QAM provides 12 bits/symbol. Another example is VDSL2 technology for copper twisted pairs, whose constellation size goes up to 32768 points.

Ultra-high capacity Microwave Backhaul Systems also use 1024-QAM. With 1024-QAM, Adaptive Coding and Modulation (ACM), and XPIC, Vendors can obtain Gigabit capacity in a single 56 MHz channel.

In this thesis, 64-QAM, 256-QAM, 1024-QAM are applied to ensure the high spectral efficiency provided by the system.
2.4 Channels

A channel is used to convey an information signal; for example a digital bit stream, from one or more senders (or transmitters) to one or more receivers. A channel has a certain capacity for transmitting information, often measured by its bandwidth in Hz or its data rate in bits per second.

2.4.1 Gaussian Channel

A Gaussian Channel has a discrete input alphabet and a continuous output alphabet range \((-\infty, \infty)\). The channel adds noise to symbols. Since the noise is a Gaussian random variable with zero mean and variance \(\sigma^2\), the resulting probability density function (pdf) of the received random variables \(z\) conditioned on the symbols \(u_k\) (the likelihood of \(u_k\)), can be written as:

\[
p(z \mid u_k) = \frac{1}{\sigma \sqrt{2\pi}} \exp\left[\frac{-(z-u_k)^2}{2\sigma^2}\right], \text{ for all } z, \text{ where } k = 1, 2, \ldots, M
\]

When the demodulator output consists of a continuous alphabet or its quantized approximation (with greater than two quantization levels), the demodulator is said to make soft decisions. In the case of a coded system, the demodulator feeds such quantized code symbols to the decoder. Since the decoder then operates on the soft decision made by the demodulator, decoding with a Gaussian channel is called soft-decision decoding.

The Additive White Gaussian Noise (AWGN) is used in this thesis for ideal channel cases. Here, we simply add the white Gaussian noise to the transmission signal to simulate the AWGN channel condition.

2.4.2 Fading Channel
In wireless communications, fading is deviation of the attenuation affecting a signal over certain propagation media. The fading may vary with time, geographical position or radio frequency, and is often modeled as a random process. A fading channel is a communication channel comprising fading. In wireless systems, fading may be either due to multipath propagation, referred to as multipath induced fading, or be due to shadowing from obstacles affecting the wave propagation, and sometimes referred to as shadow fading.

2.4.2.1 Nakagami-m Fading Channel

An alternative statistical model for the envelope of the channel response is the Nakagami-m distribution. In contrast to the Rayleigh fading distribution, which has a single parameter that can be used to match fading channel statistics, the Nakagami is a two parameter distribution, involving the parameter $m$ and the second moment $\Omega = E(R^2)$. As a consequence, this distribution provides more flexibility and accuracy in matching the observed signal statistics. The Nakagami-m distribution can be used to model fading channel conditions that are either more or less severe than the Rayleigh distribution, and it includes the Rayleigh distribution as a special case ($m=1$).

2.4.2.2 Rayleigh Fading Channel

Rayleigh fading is a statistical model for the effect of a propagation environment on radio signal, such as that used by wireless devices. It assumes that the magnitude of a signal that has passed through such a transmission medium will vary randomly, or fade, according to a Rayleigh distribution.

$$p_r(r) = \frac{2r}{\Omega} e^{-r^2/\Omega}, \quad r \geq 0, \text{ where } \Omega = E(R^2)$$
Often, the gain and phase elements of a channel's distortion are conveniently represented as a complex number. In this case, Rayleigh fading is exhibited by the assumption that the real and imaginary parts of the response are modeled by independent and identically distributed zero-mean Gaussian processes so that the amplitude of the response is the sum of two such processes.

The requirement that there be many scatters present means that Rayleigh fading can be a useful model in heavily built-up city centers where there is no line of sight between the transmitter and receiver and many buildings and other objects attenuate, reflect, refract, and diffract the signal. Experimental work in Manhattan has found near-Rayleigh fading there. In tropospheric and ionosphere signal propagation, the many particles in the atmospheric layers act as scatterers and this kind of environment may also approximate Rayleigh fading. If the environment is such that, in addition to the scattering, there is a strongly dominant signal seen at the receiver, usually caused by a line of sight, then the mean of the random process will no longer be zero, varying instead around the power-level of the dominant path. Such a situation may be better modeled as Rician fading.

Note that Rayleigh fading is a small-scale effect. There will be bulk properties of the environment such as path loss and shadowing upon which the fading is superimposed.

How rapidly the channel fades will be affected by the movement speed of receiver and/or transmitter. Motion causes Doppler shift in the received signal components.

There are Rayleigh Fading generation methods such as Clarke model. The mathematical reference model of the flat Rayleigh fading channel is the Clarke Model.
The fading channel comprised of propagation paths; the low-pass fading process is shown as follows:

\[ C(t) = E_0 \sum_{n=1}^{N} b_n \left[ \exp \left( j \times (\omega_d t \times \cos a_n + \phi_n) \right) \right] \]  

(1)

In equation (1), \( E_0 \) is a scaling constant, \( b_n \), \( a_n \) and \( \phi_n \) are respectively the random path gains. The approximation to assume \( b_n \) is real valued (1) can be written as follows:

\[ C(t) = X_c(t) + jX_s(t) \]

\[ X_c(t) = E_0 \sum_{n=1}^{S_0} b_n \cos(\omega_d t \cos a_n + \phi_n) \]

\[ X_s(t) = E_0 \sum_{n=1}^{S_0} b_n \sin(\omega_d t \cos a_n + \phi_n) \]  

(2)

\[ \omega_d = 2 \times \pi \times f_d \]

To apply Rayleigh Fading Channel in this thesis, we assumed incoming waves \( M \), and send incoming waves in every quadrature of real part of the channel and imaginary part of the channel, and then we performed normalization on those coefficients. Equalization is needed before sending noisy signal to de-modulator after Rayleigh Fading Channel.

In the case of our Rayleigh fading channel, the function takes the maximum Doppler shift as 25Hz, and signal sampling time sequence is 1e-4 seconds, the plots of the channel are:
Interleaving is the reordering of data that is to be transmitted so that consecutive bytes of data are distributed over a larger sequence of data to reduce the effect of burst errors. It greatly increases the ability of error protection codes to correct for burst errors. Index mapping function by an interleaving is given by:

\[ \pi(i) = j, 0 \leq i, j < N \]

There are several types of interleaving: S-random interleaving, block interleaving, prime interleaving and the Dilemma of interleaving.
2.5.1 S-random Interleaving

It generates a random interleaving test for all \( i_1 - S < i_2 < i_1 \), if an index choice does not satisfy the condition, choose a new index. An interleaving has spreading factor \( (S, T) \), if \( (i_2 - i_1) < S \) implies \( \pi(i_2) - \pi(i_1) \geq T \).

2.5.2 Block Interleaving

In order to use the block interleaving, the \( m \times n \) matrix is used, where is \( N = m \cdot n \).

There are two ways to use the block interleaving. The first way requires the following two steps. First, information bits are written into a matrix row-wise. Second, information bits are read from a matrix column-wise.

\[
\pi(i) = \frac{ki(i+1)}{2} \mod N, \quad k \text{ is an odd constant.}
\]

The second way is a quadratic interleaving.

\[
\pi(i) = ni + \left\lfloor \frac{i}{m} \right\rfloor \mod N, \quad N = m \cdot n.
\]

This technique is applied in this thesis by using MATLAB command “RESHAPE”.

2.5.3 Prime Interleaving

Information bits are written into a \( m \times n \) matrix row-wise. The rows are circularly shifted by prime numbers distinct to each row. Information bits are read from a matrix column-wise.

2.5.4 The Dilemma of Interleaving
Interleaving with high regularity has good spreading factors, but dense distance spectrums. On the other hand, interleaving with low regularity has a sparse distance spectrum, but poor spreading factors.
Chapter 3 Error Correcting Codes (Channel Coding)

3.1 Channel Encoding

In digital communications, a channel code is a broadly used term mostly referring to the forward error correction code and interleaving in communication and storage where the communication media or storage media is viewed as a channel. The channel code is used to protect data sent over it for storage or retrieval even in the presence of noise (errors).

3.1.1 Convolution Code

A convolutional code is a type of error-correcting code in which each k-bit information symbol (each k-bit string) to be encoded is transformed into a n-bit symbol, where k/n is the code rate (n ≥ k) and the transformation is a function of the last K information symbols, where K is the constraint length of the code.

3.1.2 Convolution Encoder

In this thesis, the industry standard rate 1/2 convolutional code with constraint length 7 was used, defined by the following diagram (Figure 3.1).

The encoder structure is described by a pair of binary numbers with the same length as the code’s constraint length that specify the connections from the delay cells to modulo-2 addition nodes. The binary number for the upper addition node is 1011011. A 1 indicates that the bit in the corresponding delay cell (reading from left to right) is sent to the addition node, and a 0 indicates that the bit is not sent. The binary number for the lower addition node is 1111001. To convert these two binary numbers to octal gives the pair [133, 171]. You can enter this pair into the block's mask by typing poly2trellis (7, [133 171]) in the field for Trellis Structure.
3.1.3 Mother Code for High Rate Convolutional Code

The punctured codes used in this thesis are of rates 2/3, 3/4, 8/9, 9/10, 13/14, and 15/16 which is close to 1. They are obtained from a rate 1/2-mother code.

For any of these codes, the encoder is the rate 1/2 mother code followed by a puncturing mechanism, and at the receiver the decoder operates on the trellis of the mother code and uses the same deleting map as in the encoder in computing the path metrics [7].

3.1.4 Punctured Convolution Code

In 1979, the punctured convolutional code (PCC) was introduced to make the high rate codes from low rate ones simply [8]. High rate PCCs \( R = \frac{l}{nl - m} \) are produced from \( R = \frac{1}{n} \) low rate convolutional code (mother code) by being periodically \( nl \) bits punctured \( m \) bits.

3.1.5 Optimum Choice of Puncturing Pattern

If the original low-rate code is given, the high-rate punctured code depends on the puncturing pattern \( (P) \) [9]. \( P \) is a matrix with binary elements defined below: \( p_{ij} \)}
is 0, if symbol $i$ of every $j$th branch is punctured; $p_{ij}$ is 1, if symbol $i$ of every $j$th branch is retained.

Best codes with best puncturing matrix for memory $M=2, 3, 4$ are provided in Table 3.1.

<table>
<thead>
<tr>
<th>Original code</th>
<th>Rate 2/3</th>
<th>Punctured Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v$</td>
<td>$g_1, g_2$</td>
<td>$d_1, a_{d_1}, a_{d_2}, \ldots, a_{d_k}$</td>
</tr>
<tr>
<td>2</td>
<td>5,7</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>15,17</td>
<td>10,11</td>
</tr>
<tr>
<td>4</td>
<td>23,35</td>
<td>100</td>
</tr>
</tbody>
</table>

Table 3.1 Best PCC with Best Puncturing Pattern [9]
However, we applied code (133,171) in this thesis which has M=6. Therefore, for rate ½ convolutional code of generator polynomial (133,171) have the best puncturing pattern, as shown below (Table 3.2). These puncturing patterns are applied in this thesis for system results comparison.

<table>
<thead>
<tr>
<th>code</th>
<th>rate</th>
<th>deleting matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>2/3</td>
<td>0.667</td>
<td>1 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0</td>
</tr>
<tr>
<td>3/4</td>
<td>0.75</td>
<td>1 1 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0 1</td>
</tr>
<tr>
<td>7/8</td>
<td>0.875</td>
<td>1 1 1 1 0 1 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0 0 0 1 0 1</td>
</tr>
<tr>
<td>8/9</td>
<td>0.889</td>
<td>1 1 1 1 0 1 0 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0 0 0 1 0 1 1</td>
</tr>
<tr>
<td>9/10</td>
<td>0.9</td>
<td>1 1 1 1 0 1 1 1 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0 0 0 1 0 0 1</td>
</tr>
<tr>
<td>13/14</td>
<td>0.929</td>
<td>1 1 0 1 0 0 0 0 0 1 1 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0 1 0 1 1 1 1 0 0 0</td>
</tr>
<tr>
<td>15/16</td>
<td>0.938</td>
<td>1 1 1 0 0 1 0 1 1 0 1 0</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 0 0 1 1 0 1 0 0 1 1 0</td>
</tr>
</tbody>
</table>

Table 3.2 Optimum Puncturing Matrixes for (133,171) [9] [33]

3.2 Optimal Decoding of Convolution Codes

There are several algorithms for decoding convolutional codes. For relatively small values of $K$, the Viterbi algorithm is universally used as it provides maximum likelihood performance and is extremely easy to parallel. Viterbi decoders are thus
easy to implement in VLSI hardware and in software on CPUs with SIMD instruction sets.

Longer constraint length codes are more practically decoded with any of several sequential decoding algorithms, of which the Fano algorithm is the best known. Unlike Viterbi decoding, sequential decoding is not maximum likelihood, but its complexity increases only slightly with constraint length, allowing the use of strong, long-constraint-length codes. Viterbi-decoded codes are usually concatenated with large Reed-Solomon error correction codes that steepen the overall bit-error-rate curve and produce extremely low residual undetected error rates.

Both Viterbi and sequential decoding algorithms return hard decisions: the bits that form the most likely code word. An approximate confidence measure can be added to each bit by use of the Soft output Viterbi algorithm. Maximum a posteriori (MAP) soft decisions for each bit can be obtained by use of the BCJR (Bahl, Cocke, Jelinek and Raviv) algorithm.

3.2.1 Maximum Likelihood (ML) Decoding

Maximum likelihood decoding involves searching the entire code space and is generally impractical because of the large associated computational burden. However, a decoding algorithm due to Viterbi provides a maximum likelihood decoding procedure that is practical for use with short-constraint-length convolutional codes.

As with ML Decoding, it assumes that all input message sequences are equal. The decoder chooses $\hat{m}_k = \arg \max_{m_k} \Pr \{ \hat{X} | m_k \}$, where $\hat{m}_k$ is a Likelihood function, $\hat{X}$ is received sequence, and $m_k$ is one of the possible transmitted sequences. The
maximum likelihood decoder minimizes error probability and the Viterbi decoding algorithm performs ML decoding.

3.2.2 Viterbi Decoding Algorithm

The Viterbi algorithm makes use of the highly repetitive structure of the code tree to dramatically reduce the number of computations required to search the entire code space.

3.2.2.1 Hard-Decision Viterbi Decoding Algorithm

If we decode this received bit sequence, the decoding process is termed hard decision decoding based on Hamming distance. In hard-decision, it is assumed that, before it reaches the decoder, the coded sequence is demodulated, sampled and then quantized. The quantization is done in two levels (0: absence of a signal, 1: presence of a signal) which means hard decision.

Theoretically, the soft-output Viterbi algorithm (SOVA) gets the same hard decision as the max-Log-MAP algorithm in [10].

3.2.2.2 Soft-Decision Viterbi Decoding Algorithm

Besides “hard” symbol decisions, SOVA provides reliability information, which can be interpreted as an approximation of the LLRs. It determines the surviving path and competing path with largest metric.

\[
M(path) = \sum_{k=1}^{N} \left[ \log \Pr \left\{ \tilde{u}_k \mid m_k(path) \right\} + \log \Pr \left\{ \tilde{v}_k \mid r_k(path) \right\} + \log \Pr \left\{ s_k \mid s_{k-1} \right\} \right]
\]

If we decode the voltage samples directly before digitizing them, we term the process soft decision decoding. In the case of the soft decision, the variations of the
signal at the output of the demodulator are sampled and quantified. For a Gaussian channel with additive white noise (AWGN), the hard quantification of the received signal, compared with the fine quantification, gives losses of about 2dB in S/N ratio.

Moreover, 8-level quantification reduces this loss to a value of less than 0.25dB, compared to the fine quantification. This means that the 8-level quantification is adequate to this kind of decoding. However, lots of memory is required. It is similar to the ordinary Viterbi algorithm except using soft information. The decoding of \( m_k \) is performed after the surviving path is determined. The competing path used to decode \( m_k \) may differ from that of MAX-Log-MAP algorithm.

3.2.3 Maximum a posteriori Probability (MAP)

The MAP can be used to obtain a point estimate of an unobserved quantity on the basis of empirical data. It is closely related to Fisher’s method of maximum likelihood (ML), but employs an augmented optimization objective which incorporates a prior distribution over the quantity one wants to estimate. MAP estimation can therefore be seen as a regularization of ML estimation.

Assume that we want to estimate an unobserved parameter \( \mathbf{X} \) on the basis of observations \( m_k \). Let \( f \) be the distribution of \( m_k \), so that \( f(m_k | \tilde{\mathbf{X}}) \) is the probability of \( m_k \) when the underlying population parameter is \( \mathbf{X} \).

\[
\hat{m}_k = \arg \max_{m_k} \Pr\{m_k | \tilde{\mathbf{X}}\}
\]

If \( m_k \) is uniform distribution, ML is equal to MAP.

Based on the Log-likelihood Ratio of a posteriori probability (APP)
MAP decoding is performed by:

\[ \hat{m}_k = \begin{cases} 
0 & \text{if } L_k > 0 \\
1 & \text{if } L_k < 0 
\end{cases} \]

3.2.3.1 BCJR Algorithm

The BCJR algorithm is the optimum algorithm to generate the sequence of APPs, but its computational complexity is large with respect to that of the Viterbi algorithm (VA).

The following steps are used while doing the BCJR decoding algorithm:

1. Initialize forward and backward recursions
2. Compute branch metrics
3. Carry out forward recursion
4. Carry out backward recursion
5. Compute L-values

3.2.3.1.1 Computing Branch Metrics

Let \( s_k \) be the state at time index \( k \), and \( X_k^n \) be the channel output from time index \( a \) to time index \( b \).

\[ \gamma_k \left( X_k, s_{k-1}, s_k \right) = \Pr \left\{ m_k = n, s_k, X_k | s_{k-1} \right\}, \text{ where } n = 0, 1 \]

3.2.3.1.2 Calculating Forward Metrics

\[ \alpha_k(s_k) = \frac{\sum_{s_{k-1}} \gamma_k \left( X_k, s_{k-1}, s_k \right) \alpha_{k-1} \left( s_{k-1} \right)}{\sum_{s_{k-1}} \sum_{n=0}^{1} \gamma_n \left( \tilde{X}_k, s_{k-1}, s_k \right) \alpha_{k-1} \left( s_{k-1} \right)} \]

\[ \alpha_0(s_0) = \begin{cases} 
1 & \text{for } s_0 = 0 \\
0 & \text{else} 
\end{cases} \]
3.2.3.1.3 Calculating Backward State Metrics

\[ \beta_k(s_k) = \frac{\sum_{s_{k-1}} \sum_{l=0}^{1} \gamma_n(\tilde{x}_{k+1}, s_{k-1}, s_k) \beta_{k+1}(s_{k+1})}{\sum_{s_{k-1}} \sum_{l=0}^{1} \gamma_n(\tilde{x}_{k+1}, s_{k-1}, s_k) \alpha_k(s_k)}, \]

\[ \beta_k(s_k) = \begin{cases} 1 & \text{for } s_k = 0 \\ 0 & \text{else} \end{cases} \]

3.2.3.1.4 Computing Extrinsic L-values

\[ L_h = \log \frac{\sum_{s_{k-1}} \sum_{l=0}^{1} \gamma_n(\tilde{x}_k, s_{k-1}, s_k) \alpha_{k-1}(s_{k-1}) \beta_k(s_k)}{\sum_{s_{k-1}} \sum_{l=0}^{1} \gamma_n(\tilde{x}_k, s_{k-1}, s_k) \alpha_{k-1}(s_{k-1}) \beta_k(s_k)}, \]

where \( \alpha_{k-1}(s_{k-1}) \) and \( \beta_k(s_k) \) are obtained recursively through \( \alpha \) process (forward recursion) and \( \beta \) process (backward recursion), respectively. For decoding each \( m_k \), we consider all paths.

3.2.3.2 Modifications of MAP Algorithm

3.2.3.2.1 Log-MAP Algorithm

The Log-MAP algorithm computes the MAP parameters by utilizing a correction function to compute the logarithm of sum of numbers.

\[ L_h = \log \frac{\sum_{s_{k-1}} \sum_{l=0}^{1} \exp\left\{ \gamma_n(\tilde{x}_k, s_{k-1}, s_k) \alpha_{k-1}(s_{k-1}) \beta_k(s_k) \right\}}{\sum_{s_{k-1}} \sum_{l=0}^{1} \exp\left\{ \gamma_n(\tilde{x}_k, s_{k-1}, s_k) \alpha_{k-1}(s_{k-1}) \beta_k(s_k) \right\}}, \]

\[ \log \alpha_k(s_k) = \log \frac{\sum_{s_{k-1}} \sum_{l=0}^{1} \exp\left\{ \log \gamma_n(\tilde{x}_k, s_{k-1}, s_k) + \log \alpha_{k-1}(s_{k-1}) \right\}}{\sum_{s_{k-1}} \sum_{l=0}^{1} \sum_{s_{k-1}} \exp\left\{ \log \gamma_n(\tilde{x}_k, s_{k-1}, s_k) + \log \alpha_{k-1}(s_{k-1}) \right\}}, \]

\[ \bar{\alpha}_k(s_k) = \log \frac{\sum_{s_{k-1}} \sum_{l=0}^{1} \exp\left\{ \bar{\gamma}_n(\tilde{x}_k, s_{k-1}, s_k) + \bar{\alpha}_{k-1}(s_{k-1}) \right\}}{\sum_{s_{k-1}} \sum_{l=0}^{1} \sum_{s_{k-1}} \exp\left\{ \bar{\gamma}_n(\tilde{x}_k, s_{k-1}, s_k) + \bar{\alpha}_{k-1}(s_{k-1}) \right\}}, \]
\[
\log \beta_i(s_i) = \log \frac{\sum_{s_i \in 1} \sum_{s_{i-1} \in 0} \exp \left\{ \log \gamma_i(\hat{X}_i, s_i, s_{i-1}) + \log \beta_{i+1}(s_{i+1}) \right\}}{\sum_j \sum_{s_{i-1}=1} \sum_{s_{i+1}=0} \exp \left\{ \log \gamma_i(\hat{X}_i, s_i, s_{i+1}) + \log \gamma_i(\hat{X}_i, s_i, s_{i-1}) + \log \alpha_i(s_i) \right\}}
\]

\[
\bar{\beta}_i(s_i) = \log \frac{\sum_{s_i \in 1} \sum_{s_{i-1} \in 0} \exp \left\{ \gamma_i(\hat{X}_i, s_i, s_{i-1}) + \bar{\beta}_{i+1}(s_{i+1}) \right\}}{\sum_j \sum_{s_{i-1}=1} \sum_{s_{i+1}=0} \exp \left\{ \gamma_i(\hat{X}_i, s_i, s_{i+1}) + \alpha_i(s_i) \right\}}
\]

For decoding each \( m_i \), all paths are considered. At each step, the logarithm of two added values by maximization operation is accommodated for by an additional correction value which is provided by a look-up table or a threshold detector in the Log-MAP algorithm. The Log-MAP parameters are very close approximations of the MAP parameters and therefore, the Log-MAP BER performance is close to that of the MAP algorithm. It is easier to implement and numerically more stable compared to MAP. Therefore, we applied Log-MAP in this thesis.

3.2.3.2.2 Max-Log-MAP Algorithm

A simple approximation to the logarithm of the sum of numbers is the logarithm of the maximum number.

\[
\log(1 + e^{x_1} + L + e^{x_p}) = \max_{1 \leq i \leq p} x_i
\]

Updating based on approximation:

\[
\bar{\alpha}_k(s_k) = \max_{s_{k-1} \in \{0,1\}} \left\{ \gamma_0(\hat{X}_k, s_{k-1}, s_k) + \bar{\alpha}_{k-1}(s_{k-1}) \right\} - \max_{s_{k-1} \in \{0,1\}} \left\{ \gamma_1(\hat{X}_k, s_{k-1}, s_k) + \alpha_{k-1}(s_{k-1}) \right\}
\]

\[
\bar{\beta}_k(s_k) = \max_{s_{k+1} \in \{0,1\}} \left\{ \gamma_0(\hat{X}_{k+1}, s_k, s_{k+1}) + \bar{\beta}_{k+1}(s_{k+1}) \right\} - \max_{s_{k+1} \in \{0,1\}} \left\{ \gamma_1(\hat{X}_{k+1}, s_k, s_{k+1}) + \alpha_k(s_k) \right\}
\]

\[
L_k = \max_{s_{k-1} \in \{0,1\}} \left\{ \gamma_0(\hat{X}_k, s_{k-1}, s_k) + \alpha_k(s_k) + \bar{\beta}_{k+1}(s_{k+1}) \right\}
\]

\[
- \max_{s_{k-1} \in \{0,1\}} \left\{ \gamma_1(\hat{X}_k, s_{k-1}, s_k) + \alpha_{k-1}(s_{k-1}) + \bar{\beta}_k(s_k) \right\}
\]
For decoding each \( m_k \), two best paths, one with \( m_k = 0 \), and another \( m_k = 1 \) are considered. Therefore, there is an approximation error in the computation of these two variables.

Since these two variables are computed recursively, this approximation error is propagated throughout the entire block of data. If the SNR requirement for a given BER performance is very high, then this approximation error is comparable to the noise and it will have a significant effect on the performance of the system. On the other hand, if the SNR requirement is not high, then this approximation error is much less than the noise power and this will not be a significant factor in performance degradation. The BER performance of the MAX-Log-MAP is always worse than that of the MAP algorithm.

3.2.4 Difference of Path Search

The differences between the (Log) MAP, Max-Log-MAP and SOVA are illustrated in Fig. 3.2. The MAP takes all paths into its calculation, but splits them into two sets: those that have information bit one at step \( j \) and those that have a zero; it returns the LLR of these two sets. All that changes from step to step is the classification of the paths into the respective sets. Due to the Markov properties of the trellis, the computation can be done relatively easily. In contrast, the Max-Log-MAP looks at only two paths per step: the best with bit zero and the best with bit one at transition \( j \); it then outputs the difference of the log-likelihoods. However, from step to step, one of these paths can change, but one will always be the maximum-likelihood (ML) path. The SOVA will always correctly find one of these two paths (the ML path), but not necessarily the other, since it may have been eliminated before merging with the ML path. There is no bias in the SOVA output when compared to that of the Max-
Log-MAP algorithm, and only the former will be noisier.

Figure 3.2 Comparisons between Log-MAP, Max-Log-MAP and SOVA. [10]

The MAP uses all paths in the trellis to optimally determine the reliability of bit $d_j$. The Max-Log-MAP makes its decision (and soft output) based on the best two paths with different $d_j$. The SOVA also takes two paths, but not necessarily both the same as for the Max-Log-MAP.

Figure 3.3 Relationship between MAP, Log-MAP, Max-Log-MAP and SOVA
Log-MAP is equivalent to the MAP algorithm in terms of performance, excluding its problems of implementation. The correction needs just an additional one-dimensional table look-up and an additional max-operation. The relationship between these algorithms is illustrated in Fig 3.3.

### 3.2.5 Complexity Comparisons for Each Algorithm [6]

As mentioned earlier, the correction function used by the Log-MAP can be implemented using a look up table, like in Table 3.3 below.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Max-Log-MAP</th>
<th>Log-MAP</th>
<th>Soft Output Viterbi</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Operation</td>
<td>$5 \times 2^M - 2$</td>
<td>$5 \times 2^M - 2$</td>
<td>$3(M + 1) + 2^M$</td>
</tr>
<tr>
<td>Additions</td>
<td>$10 \times 2^M + 11$</td>
<td>$15 \times 2^M + 9$</td>
<td>$2 \times 2^M + 8$</td>
</tr>
<tr>
<td>Multiplications by ± 1</td>
<td>8</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Bit comparisons</td>
<td></td>
<td></td>
<td>$6(M+1)$</td>
</tr>
<tr>
<td>Table look- ups</td>
<td></td>
<td></td>
<td>$5 \times 2^M - 2$</td>
</tr>
</tbody>
</table>

Table 3.3 Complexity Comparisons for Each Algorithm [10]

If we assume that the first four types of operations in the above table are of equal complexity, then this allows us to conclude that Max-Log-MAP algorithm is more than twice as complex as the SOVA for memory $M = 4$, and less than twice as complex for $M = 2$. The Log-MAP algorithm is three times as complex respectively as SOVA.

We selected both SOVA and Log-MAP in this thesis as system results comparison in order to choose optimal decoder case by case.
Chapter 4 Systems Comparison and Results Analysis

After detailed technical descriptions of all the designed system models in previous chapters, we are now able to simulate them by MATLAB, in order to compare and analyze the results. All these systems over AWGN channel were evaluated by required spectral efficiency, targeting BER performance in terms of Eb/N0 (SNR), the tradeoff among different coding rates and modulation schemes. Furthermore, the system behaviors over Rayleigh Fading channel were present.

4.1 System Selection for Determinate Spectral Efficiency

By using the calculation method mentioned in Chapter 2, the modulation efficiency of 64-QAM and 256-QAM cannot exceed 6 bits/symbol and 8 bits/symbol respectively. Thus, the upper bound of spectral efficiency of these two modulation schemes are 6 (bits/s)/Hz and 8 (bits/s)/Hz. When the coding rates are applied, a set of fixed spectral efficiency is yielded, as shown in Table 4.1.

<table>
<thead>
<tr>
<th>Coding Rates</th>
<th>Rate 1/2</th>
<th>Rate 2/3</th>
<th>Rate 3/4</th>
<th>Rate 5/6</th>
<th>Rate 7/8</th>
<th>Rate 8/9</th>
<th>Rate 9/10</th>
<th>Rate 13/14</th>
<th>Rate 15/16</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-QAM</td>
<td>3</td>
<td>4</td>
<td>4.5</td>
<td>5</td>
<td>5.25</td>
<td>5.33</td>
<td>5.4</td>
<td>5.571</td>
<td>5.625</td>
</tr>
<tr>
<td>256-QAM</td>
<td>4</td>
<td>5.33</td>
<td>6</td>
<td>6.67</td>
<td>7</td>
<td>7.11</td>
<td>7.2</td>
<td>7.429</td>
<td>7.5</td>
</tr>
<tr>
<td>512-QAM</td>
<td>4.5</td>
<td>6</td>
<td>6.75</td>
<td>7.5</td>
<td>7.875</td>
<td>8</td>
<td>8.1</td>
<td>8.357</td>
<td>8.438</td>
</tr>
<tr>
<td>1024-QAM</td>
<td>5</td>
<td>6.667</td>
<td>7.5</td>
<td>8.33</td>
<td>8.75</td>
<td>8.89</td>
<td>9</td>
<td>9.286</td>
<td>9.475</td>
</tr>
</tbody>
</table>

Table 4.1 Spectral Efficiency for Higher Order QAM at Different Coding Rate (bits/s)/Hz

Since in this part the same spectral efficiency will be a fixed condition to compare the system behaviors, 64-QAM with coding rate 2/3 and 256-QAM with coding rate 1/2 will be selected to compare first. They both provide the same spectral
efficiency as 4 (bits/z)/Hz.

![Graph](image.png)

Figure 4.1 System Result for Spectral Efficiency = 4(bits/s)/Hz

Obviously, if the same spectral efficiency can be achieved by both 64-QAM and 256-QAM with channel coding, 64-QAM with coding rate 2/3 has much better performance in terms of Eb/N0 at target BER 10e-3 than the other, regardless which decoder is applied.

Therefore, when high spectral efficiency is not required, lower-order modulation is preferred for better BER performance.

Another sets of results comparison are 64-QAM with coding rate 8/9 and 256-QAM with coding rate 2/3; they both achieved a higher spectral efficiency at 5.33 (bits/s)/Hz. Their performance is shown in Figure 4.2.
In Figure 4.2, simulation result shows the same conclusion as figure 4.1: higher-order modulation is not recommended to combine with low coding rates. In order to obtain the same spectral efficiency at 5.33 (bits/s)/Hz, 64-QAM with higher coding rate 8/9 should be considered instead of 256-QAM with lower coding rate 2/3. Furthermore, the MAP decoder is showing a slight SNR gain at target BER 10e-3 when a higher coding rate is applied.

4.2 Tradeoff among Different Coding Rates

Figure 4.3 presents the tradeoff among different coding rates for 256-QAM with SOVA decoder and MAP decoder.
Figure 4.3 Tradeoff among Different Coding Rates for 256-QAM

The lower coding rate shows better BER performance but provides much lower spectral efficiency. Besides, MAP decoder shows no performance advantage at a lower coding rate. Conversely, at higher coding rate of 9/10, the system provides 7.2 (bits/s)/Hz spectral efficiency, and the benefit brought by MAP decoder is observed.

After Figure 4.3, we decided to increase the coding rate and modulation order, in order to obtain the obvious difference between MAP decoder and Viterbi decoder in performance in Figure 4.4.
Figure 4.4 Performance for 1024-QAM with rate 15/16 over AWGN

From Figure 4.4, MAP decoder shows the behavior as an optimal decoder, and the BER performance gap between two decoders increases from 0.2 dB for 256-QAM with rate 9/10 to 1 dB for 1024-QAM with rate 15/16. It is a clear inference that coding rate is one of the main factors to show the significant outstanding performance from MAP decoder.

4.3 System over Rayleigh Fading Channel

Once we got the relative better performance from the system with 1024-QAM and 15/16 coding rate in AWGN channel, we applied it with a worse channel condition: Rayleigh Fading Channel. The channel model is described in Chapter 2.
Figure 4.5 Performance for System over Rayleigh Fading Channel

From Figure 4.5, MAP decoder shows the behavior as an optimal decoder in Rayleigh fading channel as expected, and the BER performance gap between two decoders increase from 1dB to about 2dB compared with Gaussian channel in Figure 4.4. In summary, from Figure 4.4, and Figure 4.5, for high order modulation with high coding rate in any channel environment, MAP decoder is better than Viterbi decoder in the aspect of BER performance. Especially when increasing coding rate, the decoding performance of MAP decoder is significant increased. As well as changing channel conditions, BER performance MAP decoder shows twice better then SOVA decoder.
Chapter 5 Conclusion

Channel coding is applied to decrease the data distortion brought by higher-order modulation schemes such as 64 QAM, 256 QAM, 512 QAM, 1024 QAM, etc. A coding rate close to 1 is highly preferred to ensure high spectral efficiency. However, there is a tradeoff between BER performance and spectral efficiency. Also, modulation order, coding rate and channel conditions are the main factors to affect the decoder performance.

As the contribution of thesis, we verified the MAP decoder as an optimal decoder especially when a high coding rate is considered with higher-order modulation and worse channel conditions. We also obtained the tradeoff between coding rate and BER performance for MAP and Viterbi decoders: at a lower coding rate, SOVA decoder performed better. However, when the coding rate increased, the performance gap between SOVA and MAP is decreased. In higher coding rate cases, MAP decoder shows better performance, and the BER gap increased if the coding rate was increased and close to 1. On the other hand, when channel condition degraded, MAP decoder performed better and the BER gap between MAP and SOVA decoders was also increased.
Bibliography


