IMPROVING ERROR PERFORMANCE IN BANDWIDTH-LIMITED BASEBAND CHANNELS

Juan W. Alfaro Zavala

June 2012

Master’s Thesis in Electronics

Master’s Program in Electronics/Telecommunications
Examiner: Niclas Björsell, Ph.D.
Supervisor: Prof. Sašo Tomažič
IMPROVING ERROR PERFORMANCE IN BANDWIDTH-LIMITED BASEBAND CHANNELS
Preface

Claude E. Shannon, known as “father of the information theory” and perhaps one of the most influential characters of the twentieth century in the IT world, established with his landmark paper in 1948, a research agenda for the next 50 years.

Powerful coding schemes were developed with the purpose of approaching transmission of information at the highest possible data rate, the Shannon limit. Error-correcting codes usually allow for improved performance at the expense of bandwidth. This thesis deals with the problem of coding for the bandwidth-limited channel where no bandwidth expansion is allowed. By now the principles of construction of the best codes for the bandwidth-limited channel are well understood. However, there is always a need to investigate and propose systems with less complexity and comparable performance.

The present work evaluates the performance of a simple system based on convolutional code and multilevel transmission that achieves a coding gain comparable to a more sophisticated scheme. The simulations are performed in Matlab. This thesis has a considerable amount of theory which I consider important and appropriate, especially for those whom this document is mostly intended for; students willing to learn the main concepts of coding.

This thesis was carried out at the University of Ljubljana in Slovenia, and completed at the University of Gävle. I would like to thank my supervisor Prof. Sašo Tomažič for allowing me to join the Laboratory of Communication Devices (LKN) and for his continuous support during the time I worked on this thesis. I would like to thank as well my examiner Niclas Björsell for revising this document and for suggesting relevant corrections. I also want to thank the LKN team of Ph.D. students and researchers I worked along during my stay in Ljubljana: Sara, Goran, Grega, Anton, Jaka, and Tonček; they all were my motivation to keep working. During my Erasmus exchange period in Slovenia I spent very enjoyable moments with my Croatian friends to whom I also want to thank: Alen, Vedran, Jasna, Senka and Aleksandra. Likewise, I want to thank my Slovenian and Latvian friends from Dom IV: Dejan, Andrej, Darja, Mojca, Kristīne and Aija. They are all good friends with whom I would love to meet again and remember old Erasmus times. I am also grateful to my Peruvian and international friends in Gävle; for almost three years we shared not only lectures and academic matters but also unforgettable experiences while living either in Sätra, Triangln or Midgård, or travelling, or simply helping each other, so I want to express my special thanks to Demet, Paul, Danial, Cristina and Buket for their close friendship, support and advice before and after I started working on this thesis. Finally, I want to thank my Peruvian friends Jessica, Ulises and Gustavo, without them it would not have been possible to get the scholarship for studies in Sweden. Thank you all my good friends.

Lastly, I would like to mention that this work would not have been completed without the unceasing support of my family: my beloved mother and father, Eva and Willy and my dear brothers, Dalton, Cónif and José. You, my dearly loved family, are my engine. Los quiero mucho!
IMPROVING ERROR PERFORMANCE IN BANDWIDTH-LIMITED BASEBAND CHANNELS
Abstract

Channel coding has been largely used for the purpose of improving error performance on a communications system. Typical methods based on added redundancy allow for error detection and correction, this improvement however comes at a cost of bandwidth. This thesis focuses on channel coding for the bandwidth-limited channel where no bandwidth expansion is allowed.

We first discuss the idea of coding for the bandwidth-limited channel as seen from the signal space point of view where the purpose of coding is to maximize the Euclidian distance between constellation points without increasing the total signal power and under the condition that no extra bits can be added. We then see the problem from another angle and identify the tradeoffs related to bandwidth and error performance.

This thesis intends to find a simple way of achieving an improvement in error performance for the bandwidth-limited channel without the use of lattice codes or trellis-coded modulation.

The proposed system is based on convolutional coding followed by multilevel transmission. It achieved a coding gain of 2 dB on $E_b/N_0$ or equivalently, a coding gain of approximately 2.7 dB on $SNR_{norm}$ without increase in bandwidth. This coding gain is better than that obtained by a more sophisticated lattice code Gosset $E_8$ at the same error rate.
IMPROVING ERROR PERFORMANCE IN BANDWIDTH-LIMITED BASEBAND CHANNELS
Table of Contents

Preface .......................................................................................................................... iii
Abstract ......................................................................................................................... v
Abbreviations ............................................................................................................... ix
List of Figures .............................................................................................................. x
1. Introduction ............................................................................................................. 1
2. A Brief Review of Channel Coding ........................................................................ 5
   2.1. Shannon's channel capacity theorem .................................................................. 5
   2.2. Continuous-time and discrete-time AWGN channels ......................................... 6
       2.2.1. Continuous-time AWGN channel model ..................................................... 6
       2.2.2. Continuous time to discrete time ................................................................. 7
       2.2.3. Orthonormal PAM and QAM modulation .................................................... 7
   2.3. The uncoded performance and the Shannon limit .............................................. 9
       2.3.1. Discrete-time AWGN channel model .......................................................... 9
       2.3.2. Normalized measures of SNR .................................................................... 10
       2.3.3. Power-limited and bandwidth-limited channels ........................................... 10
       2.3.4. Performance of M-PAM and (M × M)-QAM .............................................. 11
   2.4. Coding for the AWGN channel ......................................................................... 15
       2.4.1. Encoder design ............................................................................................ 15
       2.4.2. Decoder goal ................................................................................................ 17
       2.4.3. Convolutional coding ................................................................................... 19
   3.1. Coding for the bandwidth-limited AWGN channel ........................................... 25
   3.2. Spherical lattice codes ....................................................................................... 28
   3.3. Trellis-coded modulation ................................................................................... 29
4. Coding with no Bandwidth Expansion .................................................................. 30
   4.1. Research problem .............................................................................................. 30
   4.2. Justification ....................................................................................................... 30
5. Process and Results ............................................................................................... 33
   5.1. Simulation environment .................................................................................... 34
   5.2. Simulations of uncoded systems ....................................................................... 34
5.2.1. Validation of 2-PAM.................................................................................................34
5.2.2. Validation of 4-PAM.................................................................................................35
5.3. Simulations of convolutional code.................................................................................37
5.4. Simulations of proposed system ....................................................................................39
5.4.1. Proposed system with hard decisions.................................................................39
5.4.2. Proposed system with soft decisions.................................................................40
5.5. Discussion....................................................................................................................42
6. Conclusions......................................................................................................................43
References...........................................................................................................................45
Appendix..............................................................................................................................A1
Abbreviations

1D one dimension
2D two dimensions
3D three dimensions
AWGN additive white Gaussian noise
BER bit error rate
C.T. continuous time
D.T. discrete time
iid independent and identically distributed
ISI intersymbol interference
IT information technology
LDPC low density parity check
MAP maximum a posteriori
MDD minimum distance decision
ML maximum likelihood
MPE minimum probability of error
PAM pulse amplitude modulation
PDF probability density function
PSD power spectral density
QAM quadrature amplitude modulation
SER symbol error rate
SNR signal to noise ratio
TCM trellis-coded modulation
List of Figures

Figure 1.1. Typical layered model of a digital communications system.................................3
Figure 2.1. Orthonormal PAM system [9]; ..................................................................................8
Figure 2.2. $P_b(\varepsilon)$ vs. $E_b/N_0$ for uncoded 2-PAM system [9]. ........................................13
Figure 2.3. $P_b(\varepsilon)$ vs. $SNR_{\text{norm}}$ for uncoded $(M \times M)$-QAM system [9]............................15
Figure 2.4. Encoder design........................................................................................................15
Figure 2.5. Left: 3D representation of the Cartesian-product $B = \mathcal{A}^k$; Right: A subset of signal points from $B$ with increased $d_{\text{min}}$ .............................................17
Figure 2.6. Decoder goal...........................................................................................................18
Figure 2.7. $Pr(\varepsilon)$ is invariant to translations........................................................................19
Figure 2.8. $Pr(\varepsilon)$ is invariant to orthonormal rotations......................................................19
Figure 2.9. Convolutional encoder for $K=3$, $n=2$, $b=1$............................................................19
Figure 2.10. Tree-code representation for encoder of Figure 2.9..................................................20
Figure 2.11. One stage of the trellis-diagram for encoder of Figure 2.9........................................20
Figure 2.12. State-diagram representation for encoder of Figure 2.9...........................................21
Figure 2.13. Split-state diagram for the encoder of Figure 2.9.......................................................21
Figure 2.14. Trellis labeled with distances from all-0 path and showing smallest codewords..................................................................................................................22
Figure 2.15. Illustration of Basic Viterbi algorithm with hard-decision decoding .......................23
Figure 2.16. Illustration of Basic Viterbi algorithm with soft-decision decoding .........................24
Figure 3.1. 2-cube and 2-sphere with $M=4$ and same area......................................................26
Figure 3.2. 16-QAM signal sets. (a) $(4 \times 4)$-QAM; (b) V.29 standard; (c) hexagonal. [9]...........................................................27
Figure 3.3. 16-QAM signal sets of Figure 3.2 illustrating 2-sphere packings [9]............................27
Figure 3.4. Shaping gains of $n$-spheres for $n \leq 24$, [9]...............................................................27
Figure 3.5. $P_b(\varepsilon)$ vs. $SNR_{\text{norm}}$ for Gosset Lattice $E_8$ and Leech lattice $\Lambda_{24}$ with no shaping [9]..............................................................................................................28
Figure 3.6. $P_b(\varepsilon)$ vs. $SNR_{\text{norm}}$ for 8-state 2D and 16-state 4D Wei Trellis codes with no shaping [9]..............................................................................................................29
Figure 4.1. Mapping 2 bits into 2 symbols....................................................................................31
Figure 5.1. Proposed system with convolutional code and 4-PAM.............................................34
Figure 5.2. Uncoded 2-PAM system............................................................................................35
Figure 5.3. Uncoded 4-PAM system............................................................................................35
Figure 5.4. Error rates for the uncoded 2-PAM and 4-PAM systems........................................36
Figure 5.5. Convolutional code with 2-PAM system..................................................................37
Figure 5.6. Error rates for the convolutional code with 2-PAM system.....................................38
Figure 5.7. Error rates for the proposed system of Figure 5.1 with hard-decoding....................39
Figure 5.8. Error rates for the proposed system of Figure 5.1 with soft-decoding.......................41
Chapter 1

Introduction

In the early 1950s, an international telephone call was a striking event, and television was just beginning to become widely available. Today, by contrast, an international phone call and downloading big files immediately from the internet have become ordinary events. The replacement of analog by digital communications has been the basis of this revolution where the most important factor has definitely been the astonishing progress of microelectronics and optical fiber technology. For wireline and wireless radio transmission, another fundamental factor has been the advancement of channel coding, data compression and signal processing algorithms [1].

Any communications channel, i.e. a telephone line, a radio band or a fiber-optic cable, could be characterized by two factors: bandwidth and noise. Bandwidth is the range of frequencies that can be used to transmit a signal; and noise is anything that may disturb that signal. Because of the noise, the information may get distorted and present defects. This phenomenon is the fundamental problem of communications. Any communications system aims at reproducing at one point (usually the receiver) either exactly or approximately a message selected at another point (usually the transmitter).

The digital revolution has led to a pervasive utilization of bits for the purpose of representing information. In communications, a message (containing some kind of information) can be represented as a bit stream which is meant to be carried over a channel. The noise may cause errors in the transmission, thus, for a certain bit stream being sent, the ratio of erroneously received bits to all transmitted bits is called the Bit Error Rate (BER) and is used as a measure of performance of the communications system. Certainly, in order to reproduce the original message as perfectly as possible, we have to minimize the BER. At the same time, we want to send as much information as possible, i.e. we want to increase the speed at which the bits are being sent, i.e. the transmission rate, also called bit rate or data rate, which can be measured in bits per second. But, how fast can bits be sent over a channel? This question has already been answered in 1948 by Claude E. Shannon, known as “the father of the information theory”, in his landmark paper “A mathematical theory of communication” [2]. He showed that the maximum transmission rate (“the Shannon limit”) for reliable communications (i.e. keeping the BER as small as required) over a certain channel is determined by its bandwidth and noise characteristics.

Bits are typically transmitted as waveforms over the channel. The process of mapping bits into waveforms is called modulation. The simplest model of a transmitter in a communications system consists of a modulator which takes the message bits in and transforms them into appropriate waveforms that can be carried over the channel. The simplest modulator maps each
message bit into one of two possible waveforms, one of them representing a ‘0’ and the other representing a ‘1’. However, as we will see later, with this scheme we are far from reaching the Shannon limit. Some improvements may be done by using “better waveforms” for transmission over the channel; this is a study area that deals with waveform or signal design and will not be considered in this work. Rather, we will focus on the type of channel coding techniques being used extensively in the communications arena, sometimes called “structured sequences” which deal with transforming data sequences into “better sequences” having structured redundancy to better resist the effects of noise in the channel.

In a noisy environment, adding redundancy is a powerful way to detect or correct errors. If for example we want to transmit a message with only 1 bit, let us say ‘1’; we could set up the system so that the bit is sent twice: ‘11’. If because of the noise, the receiver got ‘10’ instead, she would know that an error has occurred, since the receiver is ‘expecting’ either of the two codewords ‘00’ or ‘11’. However, in this case, there is no way for the receiver to decide whether ‘0’ or ‘1’ was sent. If we now set up the system so that the bit is sent three times: ‘111’ and because of the noise the receivers gets ‘101’; the receiver would not only know that an error has occurred (since the receiver is expecting either of the two codewords ‘000’ or ‘111’), but also would be able to decide (by majority vote) that a ‘1’ was sent. Therefore, this (repetition) code can detect and also correct, one error, because ‘101’ is closer to the codeword ‘111’ than to the codeword ‘000’. This notion of distance is the key to error correction [3].

The code described above belongs to the group of block codes. Sending a message three times is actually a terrible code, since it requires the transmission of two extra (redundant) bits per each information bit; the transmission rate gets then reduced by two-thirds. If, alternatively, we wanted to keep the transmission rate the same as without coding, we would need to use more bandwidth in order to carry the extra two bits. Additionally, if two errors occurred in the proper places, the receiver would never recover the original message. Shannon knew and proved that better error-correcting codes must exist, codes that enable transmissions close to the Shannon limit. However, he did not explain how to construct such codes.

On the road to channel capacity (approaching the Shannon limit), many error-correcting codes have been developed from the very beginning of the field of channel coding [4] (which started with Shannon’s 1948 paper), since then, most coding schemes were motivated by the deep-space communications channel, which can actually be considered to have unlimited bandwidth and therefore allow for the use of codes with long sequences and high redundancy. The telephone channel is another practical channel that motivated the development of coding and modulation schemes and even gave rise to a combination of both of them (coded modulation); this channel, unlike the deep-space channel, is band-limited and therefore does not allow for the use of redundancy in the terms described above.

This thesis deals with the most important aspects of channel coding for a typical layered model of a digital communications system as depicted in Figure 1.1. From this perspective, we study the possibilities of obtaining an improvement in error performance for the bandwidth-limited channel.
More precisely, we first consider the simplest communications system without coding. Then, in order to improve the error performance of the system, we use coding under the strict condition that no bandwidth expansion is allowed. This, of course, assumes that the signal power is the same before and after coding and also that the noise characteristics remain the same. We want to find out whether an improvement can be achieved, if so, we want to quantify the improvement in terms of coding gain and explain the methods utilized during the process.

When designing error-correcting codes, the notion of distance is a key factor as we already mentioned. In fact, we want to design codes that make distances between codewords larger. The research problem we are trying to solve originated from this notion. If no bandwidth expansion is allowed, then the encoder should not produce at the output more symbols than bits at the input, i.e. a block of \( k \) bits will get mapped into a block of \( k \) symbols. For example, let us take blocks of 1 bit. If the signal power is subject to an average signal power constraint of 1, then the largest distance between logical ‘1’ and ‘0’ is obtained when ‘1’ is coded as 1 and ‘0’ as -1. This can be seen by solving the problem: Maximize \((a_1 - a_2)\) under the constraint \(a_1^2 + a_2^2 = 1\).

This analysis can be extended to two or more dimensions where each dimension represents an input bit. At first sight, it seems unlikely to find an arrangement of signal points that increases their Euclidian distances without increasing the average signal power. Our task is to test the viability of this hypothesis.

Notice however that we do not want to go into the mathematical complexities of typical coding schemes for the bandwidth-limited channels, namely lattice codes or trellis-coded modulation, since they have already been widely developed and studied by others [5]– [8]. We rather want to contribute to the literature on performance evaluation of channel coding for the bandwidth-limited channel with a setup based on a convolutional code followed by multilevel transmission. For a probability of error of \(10^{-5}\), the use of soft decisions at the input of the decoder allowed us to get a coding gain of approximately 2 dB on \(E_b/N_0\) or equivalently, a coding gain of approximately 2.7 dB on \(SNR_{\text{norm}}\) with no increase in bandwidth.

The following chapter will show a brief review of the most important concepts behind channel coding; these concepts will help us prepare the land over which this thesis work will be developed. Chapter 3 intends to show briefly two of the coding schemes typically used for
In Chapter 4, we state our research problem and explain the justification for this work. Chapter 5 shows the methods used and results obtained which reveal that it is actually possible to achieve an improvement in error performance in the bandwidth-limited channel with methods other than lattice codes or trellis-coded modulation. However, further analysis is needed regarding the distance properties of the code in order to give a better assessment of the viability of the hypothesis. Finally we present important remarks and the conclusions of this research problem.
Chapter 2

A Brief Review of Channel Coding

This chapter attempts to give the necessary background information for the purpose of gaining an insight into the main motivations for coding; it explains the Shannon limit, and gives important reasons why we can reduce the continuous-time channel model to a discrete-time channel model. It introduces the terminology regarding channel coding. It measures the performance of uncoded systems and establishes the baseline systems for both, the power-limited regime and the bandwidth-limited regime. The design of an encoder and goal of a decoder is briefly treated and the convolutional code is presented. Most of the theory shown here was taken from [9].

2.1. Shannon’s channel capacity theorem

The field of information theory and coding started with Shannon’s 1948 groundbreaking paper [2]. His most celebrated result was his channel capacity theorem. For the next half century, researchers struggled to find practical coding schemes that could approach channel capacity ("The Shannon’s limit") on well-understood channels such as the additive white Gaussian noise (AWGN) channel. Performance of channel coding for the point-to-point AWGN channel is the standard benchmark for comparison between different coding schemes since most of the advances in practical channel coding have taken place in this arena.

Shannon showed that every channel has an upper limit on the rate at which information can be transmitted reliably through the channel. He proved the existence of codes that enable information to be transmitted through a noisy channel such that the probability of error is as small as required, provided that transmission rate is smaller than channel capacity. If transmission rate is greater than channel capacity, it is not possible to achieve error-free transmission. For a band-limited AWGN channel, the capacity $C$ in bits per second [b/s] depends on only two parameters, the channel bandwidth $W$ in Hertz and the ratio between average signal power and average noise power, namely the signal-to-noise ratio (SNR), as follows:

$$C_{[\text{b/s}]} = W \log_2(1 + \text{SNR}).$$  \hspace{1cm} (2.1)

The proof that reliable transmission is possible at any rate less than capacity is based on randomly selected codes and optimal (maximum likelihood) decoding. The channel capacity theorem is basically an application of various laws of large numbers. The theorem also uses a fundamental result of large-deviation theory.
Throughout the history of channel coding, most of the advances on the road to channel capacity have been made for two different practical channels; the main characteristics of these two channels are described in Table 2.1.

Table 2.1. The deep-space channel and the telephone channel.

<table>
<thead>
<tr>
<th>The deep-space channel</th>
<th>Telephone channel</th>
</tr>
</thead>
<tbody>
<tr>
<td>The only noise in the receiver front end is AWGN; Unlimited-bandwidth, power-limited; Receiver (decoding) complexity is regarded as unlimited; Fractions of a dB have great scientific and economic value; Binary codes with binary modulation are appropriate.</td>
<td>Well modeled as a band-limited AWGN channel; Bandwidth-limited; Low enough data rates to allow for considerable amount of processing per bit; One dB has a significant commercial value; Multilevel modulation must be used. It is important to use as much of the available bandwidth as possible.</td>
</tr>
</tbody>
</table>

2.2. Continuous-time and discrete-time AWGN channels

Our aim in this section is to show that the continuous-time (C.T.) AWGN channel model may be reduced to an equivalent discrete-time (D.T.) AWGN channel model, without loss of generality or optimality. The sampling theorem and the theorem of irrelevance allow us to prove this equivalence. The D.T. model can be obtained with more practical methods like orthonormal Pulse Amplitude Modulation (PAM) or Quadrature Amplitude Modulation (QAM), which use an arbitrarily small amount of excess bandwidth. Some key parameters such as $SNR$, spectral efficiency and capacity are transferred from C.T. to D.T. on the condition that the bandwidth is taken to be the nominal (Nyquist) bandwidth.

2.2.1. Continuous-time AWGN channel model

A C.T. AWGN channel can be described by the equation:

$$ Y(t) = X(t) + N(t), \quad (2.2) $$

where $Y(t)$ is the output waveform, $X(t)$ is the input waveform, both of them considered as real random processes. $N(t)$ is a real white Gaussian noise process independent of $X(t)$ with single-sided noise power density $N_0$. $X(t)$ is regarded to be both band-limited and power-limited with average power $P$. The channel band is a positive-frequency interval with bandwidth $W$ Hz. The $SNR$ of this channel model is

$$ SNR = \frac{P}{N_0W}. \quad (2.3) $$

For digital communications purposes, the channel is completely characterized by $W$ and $SNR$ and it is not affected by the absolute scale of $P$ and $N_0$, nor the location of the band.
A digital communications scheme that transmits $R$ [b/s] over such a channel has a spectral efficiency $= R/W$ [b/s/Hz]. The Shannon limit on spectral efficiency is then

$$C_{[b/s/Hz]} = \log_2(1 + SNR);$$  \hspace{2cm} (2.4)

when $\rho < C_{[b/s/Hz]}$, reliable transmission is possible.

### 2.2.2. Continuous time to discrete time

The theory on signal space, the sampling theorem and the theorem of irrelevance which is fundamental for the detection of signals that lie in some signal space in the presence of AWGN, lead us to prove that we may without loss of optimality reduce the output $Y(t)$ to the sequence

$$Y = X + N,$$  \hspace{2cm} (2.5)

Therefore, the sequence $Y$ is a set of sufficient statistics for detection of $X(t)$ from $Y(t)$. Note that the representation of vectors or sequences is made with font-style regular (not italic).

A detailed analysis of the equivalence between the C.T. model and the D.T model can be found in reference [9]. Here we show only the most important conclusions. The characteristics of a D.T. model $Y = X + N$ are carried over from the C.T. model $Y(t) = X(t) + N(t)$ from which it was derived:

- Symbol interval: $T = 1/2W$, symbol rate: $2W$ symbols per second [symbols/s];
- Average signal energy per symbol: $PT = P/2W$;
- $N$ is a sequence of independent and identically distributed (iid) zero-mean (white) Gaussian random variables with variance $N_0/2$ per symbol;
- $SNR$ is the same as for the C.T. model

$$SNR = \frac{(P/2W)}{(N_0/2)} = \frac{P}{N_0 W};$$  \hspace{2cm} (2.6)

- A data rate of $\rho$ bits per two-dimensions [b/2D] translates to a data rate of $R = W\rho$ [b/s], or equivalently to a spectral efficiency of $\rho$ [b/s/Hz].

Note that in the Euclidian space, a real symbol can be represented in one-dimension (1D) and a complex symbol can be represented in two-dimensions (2D); therefore, 1D refers to one real symbol, and 2D refers to either two real symbols or one complex symbol.

For the passband case, it can be shown that the passband discrete-time model is effectively the same as the baseband model [9].

### 2.2.3. Orthonormal PAM and QAM modulation

The equivalence between C.T. systems and D.T. systems can more generally be developed considering an orthonormal PAM system architecture as shown in Figure 2.1.
Figure 2.1. Orthonormal PAM system [9].

X is a sequence of symbols coming in at the modulator’s input. X(t) is the output waveform that the modulator produces, \{p(t - kT)\} is an orthonormal sequence of time shifts of a basic modulation pulse p(t). The channel is AWGN. Y(t) is the noisy waveform at the input of the receiver. A canonical receiver structure consists of a matched filter p(-t) followed by sampling.

The theorem of irrelevance again shows that \( Y_k \) is a set of sufficient statistics for detection of \( X_k \) given that Y(t) is received at the front end of the receiver. If we project N(t) which is WGN onto orthonormal waveforms, the corresponding noise samples will be iid, the noise will be independent of everything which is out of band and so there is no correlation among the noise samples and then \( Y_k \) is a set of sufficient statistics to detect \( X_k \).

It can be shown that no orthonormal PAM system can have bandwidth less than the Nyquist bandwidth [9], only a system using pulses with autocorrelation function \( p(t) \ast p(-t) = sinc_T(t) \) can have exactly the Nyquist bandwidth. However, an orthonormal PAM system may use arbitrarily small excess bandwidth beyond the Nyquist bandwidth \( W = 1/2T \), or put it differently, the power in the out-of-band frequency components may be made to be arbitrarily small. So, if we let \( W \) denote the Nyquist bandwidth rather than the actual bandwidth, then again we can reduce the C.T. channel model \( Y(t) = X(t) + N(t) \) to a D.T. channel model \( Y = X + N \) for any orthonormal PAM system, not only for a system with modulation pulse \( p(t) = \frac{1}{\sqrt{T}} sinc_T(t) \), and all the characteristics mentioned in the above section remain the same.

A similar architecture also holds when the C.T. system operates in passband rather than baseband. The main difference is that instead of PAM modulation, we use QAM modulation and the sequences X and Y are complex numbers rather than real numbers.

The relationship between C.T. systems and D.T. systems can be made more precise by relating some parameters in C.T. with those in D.T. See Table 2.2.

By definition, SNR in D.T. is equal to the energy per two-dimensions over the noise variance per two-dimensions. From the table above, we see that this definition is equivalent to the definition of SNR in C.T.

The spectral efficiency in D.T. can be seen from the point of view of the design of an encoder, as shown in the above table. We see that the definitions in C.T. and D.T. are also equivalent.

We conclude that in continuous time, the key parameters of the band-limited AWGN channel are \( W \) in Hz and SNR, regardless of the location of the band (baseband or passband), the absolute scale of the signal, etc. In discrete time, the key parameters of the AWGN channel are its symbol rate \( W \) in 2D real or 1D complex symbols per second and its SNR. These key parameters are preserved whether orthonormal PAM or QAM is used and regardless of the modulation pulse. The nominal spectral efficiency \( \rho \) in [b/s/Hz] or in [b/2D] is also preserved.
Table 2.2. Parameters of a Continuous-time and Discrete-time AWGN channel models.

<table>
<thead>
<tr>
<th>C.T. AWGN model</th>
<th>D.T. AWGN model</th>
</tr>
</thead>
<tbody>
<tr>
<td>$Y(t) = X(t) + N(t)$</td>
<td>$Y = X + N$</td>
</tr>
<tr>
<td>C.T. PARAMETERS</td>
<td>D.T. PARAMETERS (PAM system)</td>
</tr>
<tr>
<td>Bandwidth $W$</td>
<td>Symbol interval $T = \frac{1}{2W}$</td>
</tr>
<tr>
<td>(By Nyquist ISI criterion)</td>
<td>(1 complex symbol occupies 2D)</td>
</tr>
<tr>
<td>Power $P$</td>
<td>Energy per two-dimensions $E_s$</td>
</tr>
<tr>
<td>$E_s = P(2T) = \frac{P}{W}$</td>
<td>$(T: 1$ symbol, $2T: 2$ symbols)</td>
</tr>
<tr>
<td>Noise AWGN</td>
<td>$N_k$ iid</td>
</tr>
<tr>
<td>PSD is $N_0$ flat, positive-support of bandwidth</td>
<td>zero-mean Gaussian, variance $N_0/2$ per dimension</td>
</tr>
<tr>
<td>$SNR = \frac{P}{N_0 W}$</td>
<td>$SNR \triangleq \frac{E_s}{N_0} = \frac{P}{N_0 W}$</td>
</tr>
<tr>
<td>$\rho_{[b/2D]} = \frac{2b}{n} = \frac{R}{2W} = \frac{R}{W}$</td>
<td>$\rho_{[b/2D]} = \frac{b}{n} = \frac{R}{W}$</td>
</tr>
<tr>
<td>$b$ bits, $n$ symbols, code rate: $(b$ bits$)/(n$ symbols$)$</td>
<td>$b$ bits, $n$ complex symbols, code rate: $(b$ bits$)/(n$ complex symb$)$</td>
</tr>
<tr>
<td>PAM: 1 symbol/dimension</td>
<td>QAM: 1 complex-symb./2D</td>
</tr>
</tbody>
</table>

2.3. The uncoded performance and the Shannon limit

In this section, we introduce two normalized expressions for $SNR$, namely $SNR_{\text{norm}}$ and $E_b/N_0$. And we show the error performance of the baseline systems for both, the power-limited regime and the bandwidth-limited regime which utilize $E_b/N_0$ and $SNR_{\text{norm}}$ respectively. We identify the gap to capacity for a probability of error of $\approx 10^{-5}$ for both channels.

2.3.1. Discrete-time AWGN channel model

In the previous section we have seen the equivalence between the C.T. AWGN channel model and the real or complex D.T. AWGN channel model. We can obtain the D.T. model by using orthonormal PAM or QAM architecture. The D.T. AWGN channel model is stated in (2.5) where $Y$ is the output sequence, $X$ is the random input signal point sequence, and $N$ is an iid zero-mean Gaussian noise sequence with variance $\sigma^2 = N_0/2$ per real dimension. We have also seen that there is no essential difference between the real and complex versions of this model. Therefore, from now on we will only consider the real model.
2.3.2. Normalized measures of SNR

The Shannon limit on spectral efficiency $C_{[b/2D]} = \log_2(1 + SNR)$, suggests that for reliable transmission the spectral efficiency is upperbounded by $\rho < C_{[b/2D]}$ or, given a spectral efficiency $\rho$, the SNR needed for reliable transmission is lower bounded by

$$SNR > 2^{\rho} - 1.$$  \hspace{1cm} (2.7)

This motivates the definition of a normalized SNR parameter

$$SNR_{\text{norm}} = \frac{SNR}{2^{\rho - 1}},$$  \hspace{1cm} (2.8)

so, the Shannon limit may be expressed as

$$SNR_{\text{norm}} > 1 \text{ (0 dB).}$$  \hspace{1cm} (2.9)

The "gap to capacity" is given by the value of $SNR_{\text{norm}}$ in dB and it is a measure of how far a coding scheme is operating from the Shannon limit.

Another well-known normalized measure of SNR is $E_b/N_0$, where $E_b$ is the average signal energy per information bit and $N_0$ is the noise variance per two-dimensions. If $E_s$ is the average signal energy per two-dimensions, then $E_b = E_s/\rho$ and

$$\frac{E_b}{N_0} = \frac{E_s}{\rho N_0} = \frac{SNR}{\rho}.$$  \hspace{1cm} (2.10)

The Shannon limit on $E_b/N_0$ can be found from $SNR > 2^{\rho - 1}$

$$\frac{E_b}{N_0} > \frac{2^{\rho - 1}}{\rho}.$$  \hspace{1cm} (2.11)

From the last expression, we find that when $\rho = 2$, $E_b/N_0 > 3/2$ (1.76 dB). When $\rho = 1$, $E_b/N_0 > 1$ (0 dB). The ultimate Shannon limit on $E_b/N_0$ can be found when $\rho \to 0$, in this case $E_b/N_0 \to \ln 2 \approx 0.69$ (−1.59 dB).

2.3.3. Power-limited and bandwidth-limited channels

The upperbound on spectral efficiency is given by $\rho < C_{[b/2D]} = \log_2(1 + SNR)$, the Shannon limit on rate is given by $C_{[b/g]} = W \log_2(1 + SNR)$. As shown by these equations and as we saw before, $SNR$ and $W$ are two fundamental parameters to the communications system. An operational meaning of the Shannon limit on spectral efficiency is that if we look at the encoder, then it gives an upperbound on the number of bits that can be carried per each symbol. The criterion that must be satisfied is that the probability of error $Pr(\varepsilon) \to 0$.

It can be shown that the Shannon limit on spectral efficiency behaves differently for small values of $SNR$ and for large values of $SNR$. When $SNR$ is small (the power-limited regime), $\log_2(1 + SNR) \approx SNR \log_2 e$ which means that the achievable spectral efficiency increases approximately linearly with $SNR$. When $SNR$ is large (the bandwidth-limited regime), $\log_2(1 + SNR) \approx \log_2(SNR)$ which means that the achievable spectral efficiency increases logarithmically with $SNR$. 

10
There is no strict dividing point between the power-limited regime and the bandwidth-limited regime but from an engineering point of view we take $\rho = 2 \, b/2D$ as a dividing point between the two regimes. This value was chosen since it corresponds to the maximum spectral efficiency we can get from binary modulation. If we want to have $\rho > 2 \, b/2D$, we have to use multilevel modulation.

In the power-limited regime, if we double the SNR then the achievable spectral efficiency also doubles. Doubling the bandwidth has no effect on the achievable rate. In the bandwidth-limited regime, if we double the SNR then the achievable spectral efficiency increases by $1 \, b/2D$ or $1 \, b/s/Hz$. By doubling the bandwidth, the achievable rate also doubles. These two regimes differ in almost every way. Table below shows these differences.

<table>
<thead>
<tr>
<th></th>
<th>Power-limited regime</th>
<th>Bandwidth-limited regime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Doubling $P$</td>
<td>$C_{[b/2D]}$ doubles, or $C_{[b/s/Hz]}$ doubles</td>
<td>$C_{[b/2D]}$ increases by $1 , b/2D$, or $C_{[b/s/Hz]}$ increases by $1 , b/s/Hz$</td>
</tr>
<tr>
<td>Doubling $W$</td>
<td>$C_{[b/s]}$ not affected</td>
<td>$C_{[b/s]}$ doubles</td>
</tr>
<tr>
<td>Baseline system</td>
<td>2-PAM</td>
<td>M-PAM</td>
</tr>
<tr>
<td></td>
<td>Binary modulation and coding suffices since $\rho &lt; 2 , b/2D$</td>
<td>Non-binary (&quot;multilevel&quot;) modulation is appropriate since we want $\rho &gt; 2 , b/2D$</td>
</tr>
<tr>
<td>Normalization</td>
<td>&quot;per information bit&quot;</td>
<td>&quot;per two-dimensions&quot;</td>
</tr>
<tr>
<td>Performance measure</td>
<td>$P_b(\varepsilon)$ vs $E_b/N_0$</td>
<td>$P_s(\varepsilon)$ vs $SNR_{\text{norm}}$</td>
</tr>
<tr>
<td></td>
<td>$P_b(\varepsilon)$ probability of error per information bit</td>
<td>$P_s(\varepsilon)$ probability of error per 2D</td>
</tr>
<tr>
<td>Ultimate limit</td>
<td>$E_b/N_0 = -1.59 , \text{dB}$</td>
<td>$SNR_{\text{norm}} = 0 , \text{dB}$</td>
</tr>
</tbody>
</table>

### 2.3.4. Performance of M-PAM and $(M \times M)$-QAM

The probability of error analysis will be done for both of the regimes with the simplest possible uncoded systems, starting from 2-PAM which is the typical baseline system for the power-limited regime and continuing with M-PAM which is the typical baseline system for the bandwidth-limited regime. We will see that the calculations can be easily extended to $(M \times M)$-QAM.
Uncoded 2-PAM

Let $\mathcal{A}$ be a 2-PAM constellation

$$\mathcal{A} = \{-\alpha, +\alpha\},$$

(2.12)

where $\alpha$ is chosen such that $E_b = \alpha^2$ satisfies the average signal energy constraint. As we saw before, without coding, the nominal spectral efficiency in this case is $\rho = 2 \text{ b/2D}$.

We use the discrete-time AWGN channel model from (2.5). Recall that $Y$ is the sequence of received symbols, $X$ is the sequence of symbols being transmitted over the channel where and for this case each symbol $x_k \in \mathcal{A}$, and $N$ is a sequence of samples of a zero-mean Gaussian random variable $N \sim \mathcal{N}(0, \sigma^2)$ with $\sigma^2 = N_0/2$. The typical symbol by symbol detector (optimum) makes an independent decision on each received symbol $y_k$. The standard detection problem at the receiver is to find out whether $x_k$ was $+\alpha$ or $-\alpha$ given that $y_k$ was received. For this binary case then, given that each symbol in the constellation is equiprobable, the decision boundary is $(-\alpha + \alpha)/2 = 0$, i.e. if $y_k < 0$, then the detector will decide that $-\alpha$ was sent; similarly, if $y_k > 0$, then $+\alpha$ was sent.

The probability of error given that $-\alpha$ was sent is

$$Pr(\varepsilon) = Pr(\text{error} | x_k = -\alpha)$$
$$= Pr(y_k > 0 | x_k = -\alpha)$$
$$= Pr(N > \alpha)$$

(2.13)

$$Pr(\varepsilon) = Q(\alpha/\sigma).$$

(2.14)

The result in (2.14) comes from the fact that $N$ is zero-mean Gaussian with variance $\sigma^2$, where

$$Q(x) = \int_x^\infty \frac{1}{\sqrt{2\pi}} e^{-\frac{y^2}{2}} dy$$

(2.15)

The probability of error when $+\alpha$ was sent is evidently the same. Since each symbol represents one bit, then $Pr(\varepsilon) = P_b(\varepsilon)$. The probability of bit error can also be expressed in terms of $E_b/N_0$ by using $E_b = \alpha^2$ and $\sigma^2 = N_0/2$,

$$P_b(\varepsilon) = Pr(\varepsilon) = Q\left(\frac{\alpha}{\sigma}\right)$$

(2.16)

$$P_b(\varepsilon) = Q\left(\frac{2E_b}{N_0}\right)$$

(2.17)

The probability of bit error as a function of $E_b/N_0$ for the uncoded 2-PAM system is shown in Figure 2.2.
Power-limited baseline and the Shannon limit

Uncoded binary PAM is the typical baseline system for the power limited regime. The rate of 2-PAM is $\rho = 2 \, b/2D$. At this rate, we found before that the ultimate Shannon limit on $E_b/N_0$ is $E_b/N_0 > 1.76 \, \text{dB}$. At $P_b(\varepsilon) = 10^{-5}$, the required $E_b/N_0 \approx 9.6 \, \text{dB}$. If a system can tolerate a maximum bit error probability of $P_b(\varepsilon) = 10^{-5}$, then we could achieve a “coding gain” of $9.6 - 1.76 \approx 7.84 \, \text{dB}$ while keeping the rate $\rho = 2 \, b/2D$ the same. If there is no limit on increasing the bandwidth, as in the case of deep-space communications, we could let $\rho \to 0$ and achieve a maximum “coding gain” of $9.6 - (-1.59) \approx 11.2 \, \text{dB}$. See Figure 2.2.

“Coding gain” therefore is a measure of performance of a coded system as compared to the uncoded system. It represents the reduction in required $\text{SNR}$ achieved by coding while maintaining the same BER.

Uncoded M-PAM and $(M \times M)$-QAM

Let $\mathcal{A}$ be an M-PAM constellation,

$$\mathcal{A} = \{- (M-1)\alpha, ..., -3\alpha, -\alpha, +\alpha, +3\alpha, ..., +(M-1)\alpha\}, \quad (2.18)$$

where $\alpha$ is chosen to satisfy the average signal energy constraint. Without coding, the nominal spectral efficiency is

$$\rho = 2 \log_2 M \, [b/2D]. \quad (2.19)$$
Let $E(A)$ be the average energy per $M$-PAM symbol. The average energy per two-dimensions is then $E_s = 2E(A)$, where

$$E(A) = \frac{a^2(M^2-1)}{3},$$

therefore $E_s$ is

$$E_s = \frac{2a^2(M^2-1)}{3}.$$  \hfill (2.20)

The probability of error on the detection of each of the $(M-2)$ intermediate symbols is given by $2Q(a/\sigma)$, since each of them has two neighboring symbols; and the probability of error for the two end-symbols is simply $Q(a/\sigma)$. Assuming again that all the symbols in the constellation are equally likely, then the average probability of error per $M$-PAM symbol is

$$Pr(\varepsilon) = \frac{1}{M}[(M-2)2Q\left(\frac{a}{\sigma}\right) + 2Q\left(\frac{a}{\sigma}\right)].$$

$$Pr(\varepsilon) = \frac{2(M-1)}{M}Q\left(\frac{a}{\sigma}\right).$$ \hfill (2.22)

Note that this expression also holds for $M = 2$. For $M \geq 4$, $Pr(\varepsilon) \approx 2Q(a/\sigma)$.

In the case of $(M \times M)$-QAM transmission which can be regarded as two independent $M$-PAM transmissions, we see that the previous calculations of $\rho$ and $E_s$ hold, since they were obtained in "per two-dimensions". The probability of error per two-dimensions $P_s(\varepsilon)$, however, has to be calculated.

$$P_s(\varepsilon) = 1 - (1 - Pr(\varepsilon))^2 = 2Pr(\varepsilon) - (Pr(\varepsilon))^2 \approx 2Pr(\varepsilon) \approx 4Q(a/\sigma).$$ \hfill (2.23)

**Bandwidth-limited baseline and the Shannon limit**

In the bandwidth-limited regime, we take everything normalized in "per two-dimensions". We take $(M \times M)$-QAM with $M \geq 4$ as the baseline uncoded system and measure the error performance as a function of $SNR_{norm}$. From (2.8), (2.19) and with $SNR = E_s/N_0$, and $\sigma^2 = N_0/2$, we can easily derive that

$$SNR_{norm} = \frac{a^2}{3\sigma^2}.$$ \hfill (2.24)

And combining this with $P_s(\varepsilon) \approx 4Q(\alpha/\sigma)$, we finally get the probability of error $P_s(\varepsilon)$ per $(M \times M)$-QAM symbol as a function of $SNR_{norm}$,

$$P_s(\varepsilon) \approx 4Q(\sqrt{3SNR_{norm}}).$$ \hfill (2.25)

which shows that since $SNR_{norm}$ is independent of $M$, $P_s(\varepsilon)$ is also independent of $M$. This is an important result because it says that when we increase or decrease $M$, we certainly obtain different spectral efficiencies but we do not gain anything in terms of coding gain (the "gap to capacity" does not change), this is what motivates $M$-PAM and $(M \times M)$-QAM to be considered as "uncoded systems".

By using (2.25), the "gap to capacity" at $P_s(\varepsilon) = 10^{-5}$ turns out to be $SNR_{norm} \approx 8.4$ dB. The curve of $P_s(\varepsilon)$ vs. $SNR_{norm}$ for the uncoded $(M \times M)$-QAM system is shown in Figure 2.3.
2.4. Coding for the AWGN channel

Here we will deal with the encoder design parameters. We will study the properties of constellations and the benefits and drawbacks of coding. We will also deal with the many criteria that the decoder takes into account in the detection process. Finally, we will introduce a very effective coding scheme that has been largely used specially in the power-limited regime, namely the convolutional code and the Viterbi algorithm.

2.4.1. Encoder design

As we saw before in the context of spectral efficiency, a typical encoder design takes $b$ bits in and maps them into $n$ symbols as depicted in Figure 2.4 below.

\[ M = 2^b \]

---

Figure 2.3. $P_e(x)$ vs $SNR_{norm}$ for uncoded $(M \times M)$ QAM system [9].

Figure 2.4. Encoder design.
We can represent the input bits by the vector $X$ and the output symbols by the vector $X$. The output sequence of symbols $X$ is not any arbitrary sequence but it lies in a setup of all possible sequences $\mathbb{C}$ which is essentially a set of $M$ permissible output symbol sequences $C_j$ with each $C_j$ being a vector in $\mathbb{R}^n$ since there are $n$ symbols being produced. This could be summarized by

$$X \in \mathbb{C} = \{C_j \in \mathbb{R}^n | 1 \leq j \leq M\}$$  \hspace{1cm} (2.26)

$\mathbb{C}$ is known as a codebook and each $C_j$ is called a codeword.

For the specific cases when $n=1$ and $n=2$, which correspond to a PAM and QAM constellation respectively, we will use the following notation:

$$\mathcal{A} = \{a_j \in \mathbb{R}^n | 1 \leq j \leq M\}$$  \hspace{1cm} (2.27)

Where $\mathcal{A}$ is the signal constellation and each vector $a_j$ is a symbol or signal point in the constellation.

The signal constellation has the following properties:

- $n$ is known as its dimension,
- $M$ is its size (number of signal points),
- $E(\mathcal{A})$ is its average energy given by $E(\mathcal{A}) = \frac{1}{M} \sum_{j=1}^{M} \|a_j\|^2$,
- $d_{\text{min}}(\mathcal{A})$ is its minimum distance given by $d_{\text{min}}(\mathcal{A}) = \min_{i,j} \|a_i - a_j\|$,
- $K_{\text{min}}(\mathcal{A})$ is its average number of nearest neighbors (number of signal points in the constellation located at a distance $d_{\text{min}}(\mathcal{A})$)

Additionally, there are some normalized parameters:

- $\rho_{[b/2D]} = \frac{2b}{n} = \frac{2 \log_2 M}{n}$, \hspace{1cm} (2.28)
- $E_s = \left(\frac{2}{n}\right) E(\mathcal{A})$, \hspace{1cm} (2.29)
- $E_b = \frac{E_s}{\rho} = \frac{E(\mathcal{A})}{\log_2 M}$, \hspace{1cm} (2.30)

Example:

We have a 2-PAM system with $\mathcal{A} = \{-\alpha, +\alpha\}$. We want to study the properties of a constellation $\mathcal{B}$, such that,

$$\mathcal{B} = \mathcal{A}^k = \{(x_1, x_2, \ldots, x_k) | x_i \in \mathcal{A}\},$$  \hspace{1cm} (2.31)

known as the $k$-fold Cartesian-product of $\mathcal{A}$.

We first identify the normalized parameters for constellation $\mathcal{A}$:

$$\rho_{[b/2D]} = 2, \ E_s = 2\alpha^2, \ E_b = \alpha^2.$$  \hspace{1cm} (2.32)

The properties of the signal constellation $\mathcal{B}$ can be easily calculated: $n=k; \ M= 2^k; \ E(\mathcal{B}) = k\alpha^2; \ d_{\text{min}}(\mathcal{B}) = 2\alpha; \text{ and } K_{\text{min}}(\mathcal{B}) = k$. Then, the normalized parameters for constellation $\mathcal{B}$ are

$$\rho_{[b/2D]} = \frac{2 \log_2 M}{n} = 2, \ E_s = \left(\frac{2}{n}\right) E(\mathcal{B}) = 2\alpha^2, \ E_b = \frac{E_s}{\rho} = \alpha^2$$  \hspace{1cm} (2.33)
What we see is that the normalized parameters are the same for the Cartesian-product \( B \) as for its corresponding original constellation \( A \). Therefore, we conclude that with the Cartesian-product we are not really doing any coding. In the original constellation we have 1 bit coming in and we are mapping it into 1 symbol. In the Cartesian-product we have \( k \) bits coming in and we are mapping them into \( k \) symbols. So we still have the same spectral efficiency; the noise is iid, then it is optimal for the detector to make decisions for each of the coordinates independently and determine whether that coordinate corresponds to \(+\alpha\) or \(-\alpha\). Therefore, there is nothing gained by doing this Cartesian-product. The probability of error expression depends on the normalized parameters, so there is no gain in terms of \( P_b(\epsilon) \) vs. \( E_b/N_0 \) through Cartesian-product. The error performance is the same for \( B \) and \( A \).

We will now consider the case when \( k=3 \), then \( B = A^3 \) so, all the constellation points are located at the vertices of a three-dimensional cube as depicted in Figure 2.5 (left). Note that \( d_{\text{min}}(A) = d_{\text{min}}(B) = 2\alpha \). Let \( B' \) be a constellation consisting of four signal points from \( B \) chosen so that we increase \( d_{\text{min}} \) as shown in Figure 2.5 (right). Since \( d_{\text{min}} \) is larger, then the probability of error will be smaller.

\[
d_{\text{min}}(B) = 2\alpha \\
d_{\text{min}}(B') = 2\sqrt{2}\alpha
\]

Figure 2.5. Left: 3D representation of the Cartesian-product = \( A^k \); Right: A subset of signal points from \( B \) with increased \( d_{\text{min}} \).

So, by simply selecting a subset of signal points we have been able to increase \( d_{\text{min}} \) and hence decrease the probability of error. This came at a price of reduced spectral efficiency. For constellation \( B', M=4 \) so the spectral efficiency becomes \( \rho_{[b/2D]} = (2 \log_2 M)/n = 4/3 \), which is smaller than \( \rho_{[b/2D]} = 2 \) for the original constellation.

Consequently, there is a trade-off between minimum distance and spectral efficiency. We want to reduce spectral efficiency in order to increase minimum distance and thus reduce the probability of error. This trade-off has in fact been a dominant design principle for a large number of codes that have come up in coding theory.

2.4.2. Decoder goal

At the decoder’s input we have the received sequence \( Y = X + N \). From \( Y \) which is a noisy version of the original sequence \( X \), we want to estimate the sequence \( \hat{X} \), where each of its elements belongs to \( A \). See Figure 2.6.
For a single transmitted symbol, represented by the vector \( x \), the goal of the decoder is to minimize the probability of error

\[
\min \ Pr(\varepsilon) = Pr(x \neq \hat{x}).
\]  

(2.34)

We will show that the minimum-probability-of-error (MPE) criterion is equivalent to a bunch of other criteria. We can write

\[
Pr(\varepsilon) = \int Pr(\varepsilon|y) P(y) dy,
\]

(2.35)

where the vector \( y \) represents a received symbol and \( P(y) \) is the probability-density-function (PDF) of \( y \); if we want to minimize \( Pr(\varepsilon) \), then we have to minimize each term in the integral, which implies to minimize \( Pr(\varepsilon|y) \), \( \forall y \). Now, suppose that \( y \) is received and the decision is made that \( a_j \) was sent, then

\[
Pr(\varepsilon|y) = 1 - Pr(x = a_j|y);
\]

(2.36)

therefore, in order to minimize \( Pr(\varepsilon|y) \), \( a_j \) has to be chosen so that \( Pr(x = a_j|y) \) is maximized. This is known as the maximum-a-posteriori-probability (MAP) rule. The idea behind this rule is then to choose a symbol in the constellation that maximizes the posterior probability given a received symbol. Under the assumption that all points in the constellation are equally likely and by using Bayes’ theorem, we can show that

\[
Pr(a_j|y) \propto Pr(y|a_j);
\]

(2.37)

accordingly, \( a_j \) has to be chosen so that \( Pr(y|a_j) \) is maximized. This is known as the maximum-likelihood (ML) rule. Hence, when the signal points in the constellation are equally probable, the MAP rule is equivalent to the ML rule.

Finally, since the noise is AWGN then

\[
P(y|a_j) \propto e^{-\frac{\|y-a_j\|^2}{\sigma_0^2}};
\]

(2.38)

Consequently, \( a_j \) has to be chosen so that \( \|y - a_j\| \) is minimized. This is known as the minimum-distance-decision (MDD) rule.

In summary, under the assumption of equiprobable inputs and iid Gaussian noise, the MPE rule is equivalent to the MDD rule. The MDD rule gives important geometrical insights which make the detection problem easier to understand e.g. under MDD rule, we can divide the signal space into decision regions. The minimum-distance regions are also called Voronoi-regions. The probability of error given that certain symbol was sent can be expressed in terms of its corresponding Voronoi-region and the probability density function of the noise. From this, we can derive some interesting properties about the probability of error:
Pr(ε) is invariant to translations, since distances remain unchanged. If we have a constellation \( \mathcal{A} \) with a certain mean, we can subtract off the mean and get another constellation \( \mathcal{A}' \) with the same probability of error but with smaller average energy. This implies that any optimal constellation will have zero-mean. See Figure 2.7.

![Figure 2.7. Pr(ε) is invariant to translations.](image)

Pr(ε) is invariant to orthonormal rotations (without scaling), since orthonormal rotations preserve the distances. See Figure 2.8.

![Figure 2.8. Pr(ε) is invariant to orthonormal rotations.](image)

### 2.4.3. Convolutional coding

According to [10], a convolutional encoder is a linear finite-state machine consisting of a \( K \)-stage shift register (\( K \) is called the constraint-length of the encoder). The input data which is usually binary is shifted along the register \( b \) bits at a time. The output data consists of \( n \) bits per each shift and it is usually produced by \( n \) linear algebraic function generators. Note that \( b \) and \( n \) were previously introduced in Section 2.4.1. An example with \( K=3, n=2, b=1 \) is shown in Figure 2.9.

![Figure 2.9. Convolutional encoder for K=3, n=2, b=1.](image)

Note that for the input sequence indicated in Figure 2.9, the output of the encoder follows the highlighted path in the tree-code representation shown in Figure 2.10. If the input bit is a zero, the coded bits at the encoder’s output are those shown on the upper branch, while if the input bit is a one, the output is shown on the lower branch.
From the diagram it is clear that after the first three branches the structure becomes repetitive. The reason is that as the fourth input bit enters the encoder, the first data bit falls off and no longer influences the output. Note that the input sequences 011xy... and 111xy generate the same output after the third branch. Hence, both nodes labeled d can be joined together. This gives rise to the trellis diagram shown in Figure 2.11.

Since the convolutional encoder is a finite-state machine, it can also be represented by a state-diagram as illustrated in Figure 2.12.
The transfer function of a convolutional code

Any codeword of a convolutional encoder corresponds to a path through the trellis that starts from the all-0 state and returns to this state. For each convolutional code, the transfer function gives information about the various paths through the trellis that start from the all-0 state and return to this state for the first time. To obtain the transfer function, we split the all-0 state into two states. See Figure 2.13. Corresponding to each branch connecting two states, a function of the form $W^\alpha L^\beta$ is defined, where $\alpha$ indicates the number of 1’s in the output and $\beta$ is the number of 1’s in the input for that branch.

The transfer function $T(W, I, L)$ is then found from the flow graph. Each term of $T(W, I, L)$ corresponds to a path through the trellis starting from the all-0 state and ending at the all-0 state. The exponent of $L$ indicates the number of branches spanned by that path, the exponent of $W$ shows the number of 1’s in the codeword corresponding to that path (or, equivalently, the Hamming distance of the codeword with the all-0 codeword) and the exponent of $I$ indicates the number of 1’s in the input information sequence. Note that in deriving the transfer function, any self-loop at the all-0 state is ignored. The transfer function of the encoder in Figure 2.9 is given by

$$T(W, I, L) = \frac{W^5 I L^3}{1 - W I L - W I L^2}$$

(2.39)

$$T(W, I, L) = W^5 I L^3 + W^6 I^2 L^4 + W^6 I^2 L^5 + W^7 I^3 L^5 + \ldots$$

(2.40)

This transfer function tells that there exists one codeword with Hamming weight 5, two codewords with Hamming weight 6, four codewords with Hamming weight 7 and so on. The codeword with Hamming weight 5 corresponds to an input sequence of Hamming weight 1 and
length 3. One of the codewords with Hamming weigh 6 corresponds to an input sequence of Hamming weigh 2 and length 4, the other codeword with Hamming weigh 6 corresponds to an input sequence of Hamming weigh 2 and length 5. These codewords are shown in Figure 2.14. Each branch is labeled with the corresponding Hamming weigh.

The smallest power of $W$ is called the free distance of the convolutional code and is denoted by $d_{\text{free}}$. For this encoder, $d_{\text{free}} = 5$.

**Decoding convolutional codes**

As stated in [11], Convolutional code decoding algorithms infer the values of the input information sequence from the stream of received distorted coded symbols. There are three main families of decoding algorithms used for convolutional codes: sequential, Viterbi, and maximum a posteriori (MAP). Wozencraft [12], Fano [13], and Johannesson and Zigangirov [14] proposed and developed sequential decoding. Viterbi [15] originally described the decoding algorithm that bears his name. See also Forney's work [16, 17] introducing the trellis structure and showing that Viterbi decoding is maximum-likelihood in the sense that it selects the sequence that makes the received sequence most likely.

Bahl et al. [18] proposed MAP decoding, which explicitly minimizes bit (rather than sequence) error rate. Compared with Viterbi, MAP provides a negligibly smaller bit error rate (and a negligibly larger sequence error rate). These small performance differences require roughly twice the complexity of Viterbi, making MAP unattractive for practical decoding of convolutional codes. However, MAP decoding is crucial to the decoding of Turbo codes. See the original paper on Turbo codes by Berrou et al. [19].

When convolutional codes are used in the traditional way (not as constituents in Turbo codes), they are almost always decoded using some form of the Viterbi algorithm, and the rest of this section focuses on describing it. The goal of the Viterbi algorithm is to find the transmitted sequence (or codeword) that is closest to the received sequence. As long as the distortion is not too severe, this will be the correct sequence.
The basic Viterbi Algorithm

The Viterbi algorithm is a maximum-likelihood decoding algorithm, which upon receiving the channel output, searches through the trellis to find the path that is most likely to have generated the received sequence. For the AWGN channel, 2-PAM commonly maps binary 1 to 1.0 and binary 0 to -1.0. These two transmitted values are distorted by AWGN, so that the received values will usually be neither 1.0 nor -1.0. At the receiver, prior to Viterbi decoding, a decision can be made on the received symbols; each received symbol will be chosen to be binary 1 if its value is closer to 1.0, or binary 0 if its value is closer to -1.0. This method of decoding is called hard-decision decoding. When hard decisions are not being made, the decoding method is called soft-decision decoding. If hard-decision decoding is used, this algorithm finds the path that is at the minimum Hamming distance from the received sequence, and if soft-decision decoding is employed, it finds the path that is at the minimum Euclidean distance from the received sequence.

As an example, consider the encoder in Figure 2.9 and hard decision decoding; assume that the receiver knows that the encoder begins in state 00. Figure 2.15 shows the basic Viterbi algorithm for the received sequence 01 01 10. Branch metrics label each branch, indicating the Hamming distance between the received symbols and the corresponding output for each branch. The path metric for each destination state is the sum of the branch metric of the incident branch and the path metric at the root of the incident branch. From the third column, only the path with minimum path metric is the survivor path (shown with thicker arrows.) When two incident paths have the same path metric, a surviving path may be arbitrarily selected.

The decoding process starts by examining the last column; following the survivor branches from the state with the smallest path metric, i.e. state 11 backwards to the beginning of the trellis identifies the path of the maximum likelihood sequence. For this example it can be found that the maximum likelihood output symbol sequence is 11 01 10, which differs in exactly one bit from the received sequence as indicated by its path metric. The input information sequence is decoded to be 111.

With soft decision decoding, the branch and path metrics are determined by the squared Euclidian distance. Figure 2.16 works an example analogous to that of Figure 2.15 for the case in which 1.0 and -1.0 are transmitted (as with antipodal signaling) over the AWGN channel.
Figure 2.16. Illustration of Basic Viterbi algorithm with soft-decision decoding
Chapter 3

State of the Art in the Bandwidth-Limited Channel

From the theory presented in Section 2.3, we know that for the bandwidth-limited channel, multilevel coding schemes must be used since the desired spectral efficiency is greater than $2 \frac{b}{2D}$. We also saw that in the bandwidth-limited regime, it is more appropriate to use $SNR_{norm}$ since it gives more information in this regime. In practice, coding for the power-limited regime and coding for the bandwidth-limited regime are considerably different.

The first attempts were mainly theoretical and were focused on lattice codes, which are similar in many ways to binary linear block codes. But the invention of trellis-coded modulation by Ungerboeck, which is similar to convolutional coding, has definitely been a practical breakthrough in coding for the bandwidth-limited channel. See the original paper on trellis-coded modulation by Ungerboeck [7].

3.1. Coding for the bandwidth-limited AWGN channel

Most of the coding schemes for this bandwidth-limited AWGN channel use two-dimensional QAM. By the Nyquist limit, if the channel has a bandwidth of $W$, then we can transmit over the channel at a maximum rate of $W$ QAM symbols per second. An uncoded baseline system is based on an $(M \times M)$-QAM constellation, where $M$ is typically a power of 2. The average energy of this constellation can be derived from (2.21) as

$$E_s = \frac{d^2(M^2-1)}{6}, \quad (3.1)$$

where $d$ is the minimum distance between constellation points. From (2.25) in Section 2.3 we also recall that the probability of error per QAM symbols was found to be $P_s(\varepsilon) \approx 4Q(\sqrt{3SNR_{norm}})$.

In the bandwidth-limited regime we should consider a new term called shaping. See [4] and [9]. The set of all $n$-tuples of constellation points from a square QAM constellation is the set of all points on a $d$-spaced rectangular grid that lie within a $2n$-cube in real $\mathbb{R}^{2n}$. If instead the constellation consisted of all points on the same grid that lie within a $2n$-sphere of the same volume, the average energy of this $2n$-dimensional constellation could be reduced. This reduction is called shaping gain of a $2n$-sphere. When $n \to \infty$, the shaping gain approaches $\pi e/6$ (1.53 dB).
To understand *shaping*, consider the case when \( n = 1 \), a 2-cube signal set is the set of all odd-integer sequences of length 2 within a 2-cube of side \( 2M \) centered on the origin. For example, the signal set of red points in Figure 3.1 is a 2-cube signal set with \( M = 4 \).

A 2-sphere signal set is the set of all odd-integer sequences of length 2 within a 2-sphere of squared radius \( r^2 \) centered on the origin. For example, the signal set of red points in Figure 3.1 is also a 2-sphere signal set for any squared radius \( r^2 \) in the range \( 18 \leq r^2 < 25 \). In particular, it is a 2-sphere signal set for \( r^2 = 64/\pi = 20.37 \), \( r = 4.51 \), where the area \( \pi r^2 \) of the 2-sphere (circle) equals the area \((2M)^2 = 64\) of the 2-cube (square) of the previous paragraph. Both 2-cube and 2-sphere signal sets therefore have minimum squared distance between signal points \( d^2 = 4 \), and 2-cube decision regions of side 2. It can be shown [9] that when \( M \) is large, the ratio of the average energy of the 2-cube to that of the 2-sphere is

\[
\frac{E_{2\text{-cube}}}{E_{2\text{-sphere}}} = \frac{\pi}{3} = (0.2 \text{ dB}).
\]

(3.2)

Thus, in two dimensions, using a large circular rather than square constellation saves only about 0.2 dB in power efficiency. If this analysis is extended to higher dimensions, namely \( n \to \infty \), it can be shown that

\[
\frac{E_{2n\text{-cube}}}{E_{2n\text{-sphere}}} = \frac{\pi e}{6} = (1.53 \text{ dB}).
\]

(3.3)

Therefore, the greatest possible shaping gain in any number of dimensions is 1.53 dB [9].

The total coding gain in the bandwidth-limited regime is the sum of the shaping gain plus the coding gain due to a denser packing than the baseline \( M \)-PAM scheme. To illustrate the idea of coding gain due to packing, see Figure 3.2 which shows three 16-point 2-dimensional QAM (16-QAM) signal sets, all have \( d_{\text{min}}^2 = 4 \); the average energy of each signal set can be calculated, assuming equiprobable signals, \( E_{(a)} = 10 \) dB, \( E_{(b)} = 11.3 \) dB, and \( E_{(c)} = 9.42 \) dB, which shows that (c) is the most power-efficient packing.
This can also be seen if we sketch 2-spheres (circles) of radius 1 about each signal point, since $d_{\min} = 2$. The densest (most power-efficient) packing is found to be again (c). See Figure 3.3.

For big signal constellations, shaping and coding can be implemented almost independently of each other. Therefore their gains are also almost independent of each other. In the bandwidth-limited regime, with coding we can get close to about 1.53 dB, if we want to reach the Shannon limit 0 dB, then this could be done only by shaping. Obtaining shaping gains of about 1 dB turned out to be not so hard and that is why most of today’s practical schemes for the bandwidth-limited channel include shaping. Figure 3.4 shows the shaping gain of an $n$-sphere for dimensions until 24.

Figure 3.2. 16-QAM signal sets. (a) (4 × 4)-QAM; (b) V.29 standard; (c) hexagonal. [9].

Figure 3.3. 16-QAM signal sets of Figure 3.2 illustrating 2-sphere packings [9].

Figure 3.4. Shaping gains of $n$-spheres for $n \leq 24$, [9].
If we look at Figure 2.3, we see that the uncoded system at $P_s(\varepsilon) = 10^{-5}$ requires $SNR_{\text{norm}} \approx 8.4$ dB. The Shannon limit on $SNR_{\text{norm}}$ is 0 dB. As we mentioned before, the maximum coding gain we can get with shaping is 1.53 dB, therefore the maximum coding gain we can get with denser packing is about 6.9 dB.

### 3.2. Spherical lattice codes

For a bandwidth-limited AWGN channel and from the proof of Shannon’s capacity theorem we infer that an optimal code consist of a dense packing of signal points within an $n$-sphere in a high-dimensional Euclidean space $\mathbb{R}^n$. Finding the densest packings for different constellation sizes, has been for many years a continuous mathematical problem. The densest known packings found are lattices. Lattices are packings that have a group property. Among the notable lattice packings we have: the integer lattice $\mathbb{Z}$ in one-dimension, the hexagonal lattice $A_2$ in two-dimensions, the Gosset Lattice $E_8$ in eight-dimensions and the Leech lattice $\Lambda_{24}$ in 24-dimensions. In the mid-1970s, Lang proposed an $E_8$ lattice code for telephone-line modems and in the late 1980s, a Leech lattice modem was built [9].

There union bound estimate for the probability of error for lattice codes, depends on parameters like the kissing number (number of nearest neighbors), the nominal coding gain, and the shaping gain of an $n$-sphere. Figure 3.5 shows the real coding gain for the Gosset lattice $E_8$ at $P_s(\varepsilon) = 10^{-5}$ which is about 2.2 dB. It also shows the real coding gain for the Leech lattice $\Lambda_{24}$ which at $P_s(\varepsilon) = 10^{-5}$ is about 3.6 dB.

![Figure 3.5. $P_s(\varepsilon)$ vs. $SNR_{\text{norm}}$ for Gosset Lattice $E_8$ and Leech lattice $\Lambda_{24}$ with no shaping [9].](image)
3.3. Trellis-coded modulation

Trellis-coded modulation (TCM) was originally conceived in 1970; its inventor was Gottfried Ungerboeck but this breakthrough in practical coding for bandwidth-limited channels was not published until 1982. Ungerboeck realized that the redundancy needed for coding should be obtained by expanding the signal constellation while keeping the bandwidth fixed. Unlike the common way of adding redundancy as in the power-limited AWGN channel, by increasing the bandwidth while keeping the signal constellation fixed. Ungerboeck showed that doubling the signal constellation should suffice if we want to reach capacity. He invented clever trellis codes for such expanded constellations and instead of using Hamming distance, he used minimum Euclidean distance as the design criterion. These codes can be optimally decoded by the Viterbi algorithm decoder. The decoding complexity is proportional to its number of states. Effective coding gains of 3 to 4 dB can be achieved with simple 4 to 8 states trellis codes, without bandwidth expansion.

The error performance curve for an 8-state 2D QAM trellis code was incorporated into the V.32 voice grade telephone line modem standard. Its real coding gain is about 4 dB. Later, V.34 used a 16-state 4D QAM trellis code of Wei and achieved a real coding gain of 4.2 dB. See Figure 3.6.

In terms of performance versus complexity, trellis codes turned out to be more attractive, in the same way as convolutional codes have been preferred to block codes. Nevertheless, trellis codes constellations have been based on simple lattices and their "set partitioning" is best understood as being based on a sublattice chain.

![Figure 3.6. $P_\epsilon$ vs. $SNR_{norm}$ for 8-state 2D and 16-state 4D Wei Trellis codes with no shaping [9].](image)
Chapter 4

Coding with no Bandwidth Expansion

The ever-increasing demand for more bandwidth which comes among others as a result of the increased use of wireless communication systems, has led to the development of a vast amount of techniques dedicated to improve the performance of the systems in order to better utilize the available bandwidth.

In the communications arena, engineers and theorists have been using channel coding to solve the design problem of transmitting as much information as possible over a channel, they have devised channel coding schemes that allow for higher data rates, a reduction in the transmission power requirements, better spectral efficiencies, smaller error rates, or some other kind of benefit. However, these benefits cannot be obtained all at the same time. An improvement in performance related to one of the parameters implies the deterioration in performance related to another parameter. One of the fundamental tradeoffs of channel coding is that of error performance versus bandwidth.

In Chapter 2, we described the basic idea of channel coding which is to introduce controlled redundancy into the transmitted signals that is subsequently exploited at the receiver for error correction. If we want to keep the information bit rate the same before and after coding, and if the modulation scheme is not changed, this added redundancy implies an increase in required bandwidth. When error correction is performed at the receiver, the result is a reduction in BER. The utilization of channel coding techniques, then, allows for an improvement in error performance which has to be paid with bandwidth.

4.1. Research problem

This thesis considers the typical layered model of a digital communications system as depicted in Figure 1.1. From this perspective, we study the possibilities of channel coding (no modulation) for the purpose of obtaining an improvement in error performance for the bandwidth-limited AWGN channel where no bandwidth expansion is allowed. Since coding schemes for the bandwidth-limited channel have already been developed, we restrict the scope of this thesis to the use of coding schemes other than those presented in Chapter 3. Performance analysis of the proposed system is to be simulated in Matlab.

4.2. Justification

The channel coding techniques for the bandwidth-limited regime, which were introduced in Chapter 3, are highly effective in higher dimensions and with large signal constellations, Lattice codes use shaping and sphere packing and TCM combines coding and modulation. In terms of
performance versus complexity, TCM was found to be more attractive than lattice codes in the same way as convolutional codes have been preferred to block codes. However, since TCM combines coding and modulation, it takes us slightly away from the canonical layered model of Figure 1.1 where the two layers are clearly separated. It is important to preserve the layered model, since it is the basis on which the most important concepts of coding have been developed.

The idea for this thesis work started from a simple analysis of coding: Suppose that we are using the orthonormal PAM system of Figure 2.1 with binary transmission, suppose that ‘1’ is being mapped into 1 and ‘0’ is being mapped into -1. In this case we are evidently transmitting at 1 bit per symbol and if we use the terminology suggested in Chapter 2, the spectral efficiency turns out to be $\rho = 2 \text{ b/2D}$ which as we saw before, corresponds to an uncoded system. We have shown the error performance curves for this system and said that through the use of coding we could get some improvement at the expense of bandwidth (recall from Chapter 2 that the main purpose of coding is that of increasing the distances between codewords) but if we are not allowed to increase the bandwidth, nor the average signal energy, we might think of taking blocks of $k$ bits and mapping them into blocks of $k$ symbols and try to find an arrangement of constellation points with increased distances between signal points.

An attempt of this type of mapping is shown in Figure 4.1 for $k=2$, i.e. when 2 bits are mapped into 2 symbols. The average signal energy of both constellations is 2. However, because of the energy constraint, with this kind of mapping it was not possible to increase the distances. It can be seen from the figure that the distance between signal points is still 2. We have basically performed an orthonormal rotation. Therefore the probability of error is also the same, there is no improvement. The problem of coding with no bandwidth expansion, consequently, can be seen as a maximization problem. We want to maximize the Euclidian distance between signal points in the constellation while keeping the average energy of the constellation fixed.

From the above analysis, the first impression is that, for a bandwidth-limited channel, there is no way to find an arrangement of constellation points that increases the distances between signal points with the same average signal energy, if we cannot increase these distances then we cannot improve the error performance of the system. This is a hypothesis that we will try to verify or deny.

Another reason why it is important to consider the problem of coding with no bandwidth expansion under the conditions mentioned in Section 4.1 is that it was not possible to find sufficient material on the performance analysis of conventional codes applied in the bandwidth-limited regime. This thesis intends to make its contribution to this field by evaluating the
performance of a simple system which could probably obtain improved performance versus complexity as compared to lattice or trellis codes; and, in turn, allow for a better understanding of the tradeoffs in the bandwidth-limited channel.

The simulation results of this thesis could open up possibilities for implementation of the codes on hardware for obtaining error performance improvements both in wireless and wireline systems.
Chapter 5

Process and Results

A coding scheme for the AWGN channel considers two important parameters: SNR and $p$. If this coding scheme transmits $R$ [b/s] over an AWGN channel of bandwidth $W$ Hz, then it has a spectral efficiency $\rho = R/W$ [b/s/Hz]. Such a coding scheme usually maps a sequence of bits at a rate $R$ [b/s] to a sequence of real symbols at a rate of $R_s$ [symbols/s]. The discrete-time code rate is then $r = R/R_s$ bits per symbol. The sequence of real symbols is modulated using PAM or QAM and transmitted over the channel. According to the Nyquist criterion, the symbol rate cannot exceed $2W$. If $R_s \approx 2W$, then $\rho = R/W \approx R/(R_s/2) = 2r$. Therefore, the nominal spectral efficiency (i.e. spectral efficiency when $R_s \approx 2W$) of a discrete-time coding scheme is $2r$ – the discrete-time code rate in bits per two symbols.

As mentioned in Section 4.1, we want to study the possibilities of channel coding for the bandwidth-limited regime where no bandwidth expansion is allowed. This restriction suggests studying the simplest case when the required bandwidth is chosen to be the same before and after coding, under these circumstances we want to find out whether channel coding allows for an improvement in error performance. Notice that if the symbol rate is not changed by coding, then bandwidth requirements do not change either, this implies $R = R_s$, which makes $r = 1$ bits per symbol, and $\rho = 2$ bits per two symbols.

When using binary transmission (where each signal waveform carries 1 bit), the use of coding leads to an inherent increase in required bandwidth; therefore, in order to compensate for this increment, multilevel transmission must be used (as suggested in Chapter 2). Recall that in multilevel signal transmission each signal waveform conveys multiple bits. So, for a given information bit rate, the utilization of multilevel signal transmission yields a reduction in required bandwidth, however, because of the signal energy constraint, it will be more difficult for the receiver to distinguish among the possible waveforms which means that under the same channel conditions, there will be more errors. The use of multilevel signal transmission, then, allows for a reduction in required bandwidth which has to be paid with an increase in BER.

From the previous paragraph, we infer that the condition $R = R_s$ could be fulfilled by using coding in conjunction with multilevel transmission, it is clear though, that the improvement in error performance achieved by coding will be diminished by the use of multilevel transmission. We therefore want to quantify the net improvement in error performance (coding gain), if any, achieved by coding plus multilevel transmission.

By choosing a coding scheme in which 1 bit at its input generates 2 bits at its output, e.g. a convolutional code with $b=1$ and $n=2$, the bandwidth requirements doubles. To compensate for this, we might use multilevel transmission so that each level represents 2 bits, e.g. 4-PAM which
halves the bandwidth requirements. In this way we can effectively fulfill the condition \( R = R_s \) [b/s] and \( \rho = 2 \text{ b}/2D \). This system is depicted in Figure 5.1. The following pages will show the simulations of error performance and results.

![Diagram of proposed system with convolutional code and 4-PAM.](image)

Figure 5.1. Proposed system with convolutional code and 4-PAM.

### 5.1. Simulation environment

Since the simulations of uncoded performance both for 2-PAM and 4-PAM as well as for the convolutional code are already well-known [20], we start by validating our simulation environment by obtaining the theoretical and simulation curves for the cases mentioned before, and then we simulate the proposed system shown in Figure 5.1. Note that for direct comparison with the curves obtained in [20] we prefer to use the \( P_b(\epsilon) \) vs. \( E_b/N_0 \) plane rather than the \( P_b(\epsilon) \) vs. \( SRN_{\text{norm}} \) plane.

Simulations were performed in Matlab 7.0.1 (R14). A pseudorandom sequence of bits is generated where the occurrence of ‘0’ and ‘1’ have equal probability; then, the D.T. AWGN channel model shown in (2.5) is used. Each element in vector \( X \) belongs to a constellation \( \mathcal{A} \) corresponding to a 2-PAM or 4-PAM constellation. The sequence of bits is mapped to signal points in the constellation and transmitted over the AWGN channel. Each element in vector \( N \) is a pseudorandom value drawn from a normal (Gaussian) distribution with variance chosen according to the current simulation setup so that \( E_b/N_0 \) is maintained unaltered in all of the cases. The detector uses MDD rule as explained in Section 2.4.2. For a given \( E_b/N_0 \), the number of errors are computed by comparison between the sequence of bits obtained at the output of the detector/encoder and the original pseudorandom sequence of bits generated at the transmitter’s side. Finally, the BER is calculated. This process is repeated for increasing values of \( E_b/N_0 \), starting from 0. The BER values are recorded for different \( E_b/N_0 \) and in the end the error performance curves are plotted.

### 5.2. Simulations of uncoded systems

The 2-PAM and 4-PAM systems are simulated in order to validate the simulation environment.

#### 5.2.1. Validation of 2-PAM

A pseudorandom sequence of \( 10^6 \) bits is generated and transmitted over the D.T. AWGN channel using 2-PAM with a constellation \( \mathcal{A} = \{-\alpha, +\alpha\} \) as shown in (2.11). For simplicity, \( \alpha \) is chosen to be equal to 1, therefore \( E(\mathcal{A}) = 1 \). Since each single bit in the input sequence is mapped to one signal point in the constellation, the spectral efficiency is 1 bit per symbol or equivalently \( \rho = 2 \text{ b}/2D \). Figure 5.2 shows the uncoded 2-PAM system.
The theoretical error performance curve is plotted from (2.16) and it is shown by the blue line in Figure 5.4. It can be seen that the BER curve obtained from simulations which is shown by the green line in the same figure, follows closely the theoretical curve. Notice that these two curves coincide with that shown in Figure 2.2 which corresponds to the $P_b(\epsilon)$ vs. $E_b/N_0$ for the uncoded 2-PAM system. We therefore conclude that the simulation setup for the uncoded 2-PAM system is valid.

### 5.2.2. Validation of 4-PAM

In the same way as before, a pseudorandom sequence of $10^6$ bits is generated and transmitted over the D.T. AWGN channel but using 4-PAM with a constellation $A = \{-3\alpha - \alpha, +\alpha, +3\alpha\}$ which can be derived from (2.18) when $M=4$. Because of the signal energy constraint, the same average signal energy as for the 2-PAM system described in the previous section has to be maintained, i.e. $E(A) = 1$, and $\alpha$ can be found from (2.20) to be equal to $(1/\sqrt{5})$.

Since each pair of bits in the input sequence is mapped to one signal point in the constellation, the spectral efficiency is 2 bits per symbol or equivalently $\rho = 4 \text{ b}/2\text{D}$. Figure 5.3 shows the uncoded 4-PAM system.

If $+\alpha$ is sent, the detector might make a decision error if it chooses a symbol different from $+\alpha$. The most probable symbols which the detector may choose in case of error are those adjacent to $+\alpha$. Since for 4-PAM, each symbol represents two bits, there could be different ways of mapping. In Gray mapping, adjacent symbols are represented by groups of bits which differ in only one bit, so that the occurrence of a symbol error translates to a single bit error. Table 5.1 shows two ways of mapping two bits into one 4-PAM symbol.
Table 5.1. Mapping two bits into one 4-PAM symbol.

<table>
<thead>
<tr>
<th>No-Gray</th>
<th>Gray</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>10</td>
<td>+3(\alpha)</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>+(\alpha)</td>
</tr>
<tr>
<td>01</td>
<td>01</td>
<td>-(\alpha)</td>
</tr>
<tr>
<td>00</td>
<td>00</td>
<td>-3(\alpha)</td>
</tr>
</tbody>
</table>

The theoretical error performance curve is plotted from (2.22) and it is shown by the red line in Figure 5.4. Notice that this is the theoretical symbol error rate (SER) and it is closely followed by the black line representing the SER obtained from simulations. These two curves are in accordance with [20].

Figure 5.4. Error rates for the uncoded 2-PAM and 4-PAM systems.

Figure 5.4 also shows the BER curves. The BER obtained with Gray mapping is shown by the pink line and is in average half of the SER. This is because when Gray mapping is used, if one symbol out of 4 is in error, then, one bit out of 8 is in error; hence the BER is half the SER. As expected, when No-Gray mapping is used, the BER is slightly higher as shown by the cyan line. From these results we therefore conclude that the simulation setup for the uncoded 4-PAM system is also valid.
5.3. Simulations of convolutional code

The convolutional encoder of Figure 2.9 is simulated with the 2-PAM system of Section 5.2.1. The simulation setup is shown in Figure 5.5. At the decoder's side, the 2-PAM detector produces hard decisions based on MDD rule or soft decisions either with 3-bit quantization or “infinitely” fine quantization. The convolutional decoder uses the Viterbi algorithm described in Section 2.4.3.

![Convolutional code with 2-PAM system.](image)

The optimum decoder performance is achieved when the path of the maximum likelihood sequence is found from the last stage in the trellis to the beginning of the trellis; however, in order to avoid decoding delay and to reduce decoder memory requirements, the path history length (traceback length) has to be reduced. Negligible degradation from optimum decoder performance can be achieved when the traceback length is chosen to be 5 times the constraint length of the encoder [21, 22]. Therefore, the convolutional code is simulated with traceback length equal to 15 (since the encoder’s constraint length is $K=3$). In order to evaluate the degradation from optimum decoder performance, the code is also simulated when the path history length corresponds to the entire input bit sequence.

When soft decisions are used in practical systems, a quantizer is needed. A Viterbi decoder with soft decision data inputs quantized to three bits of precision can perform about 2 dB better than one working with hard decision inputs. The usual quantization precision is three bits. More bits provide little additional improvement [21, 22]. The quantizer considered in these simulations was taken from [21].

Figure 5.6 shows the error curves for the convolutional code. As a reference, the uncoded 2-PAM system is also plotted, see blue curve. Recall that for the uncoded 2-PAM system, $\rho = 2 \text{ b/2D}$. As discussed before, the convolutional encoder used in this work, doubles the bandwidth requirements which means that the spectral efficiency gets reduced by half, i.e. $\rho = 1 \text{ b/2D}$. 

37
Matlab provides the built-in function *vitdec* where most of the parameters of the convolutional code can be specified for both, hard and soft decisions. When hard decisions are used with traceback length equal to 15, the BER is obtained and shown by the red curve. It can be seen that the code produces an improvement in error performance for values of $E_b/N_0 > 4$ dB. When soft decisions are used with traceback length equal to 15 and 3-bit quantization, the BER is obtained and shown by the green curve; as expected, this error performance curve is about 2 dB better than the curve obtained with hard decisions. Finally, the system is simulated with an ideal Viterbi decoder with no restrictions on path history length and with infinitely fine quantization. For this purpose, the Matlab code for soft input Viterbi decoder given by [23] was adapted to our system and the BER curve was obtained. This is shown by the pink curve. As expected, the ideal performance is slightly better than the performance obtained with the 3-bit soft decision decoder.

It was seen from the simulations that coding allows for an improvement in error performance but this requires a reduction in spectral efficiency. The spectral efficiency for the uncoded system is $\rho = 2$ b/2D; but after coding, the spectral efficiency becomes $\rho = 1$ b/2D. Since, the simulation results obtained for the convolutional code with 2-PAM system are in accordance with [21], [22] and [23]; we conclude that the simulation setup for the convolutional code to be used for performance evaluation of the proposed system, is valid.
5.4. Simulations of proposed system

In the previous sections, the 2-PAM and 4-PAM systems implemented in Matlab were shown to be valid, since the error performance curves obtained from simulations followed closely the theoretical curves. The convolutional code was also shown to be valid, since the simulations yielded similar results to those obtained by others [21–23]. In this section, the proposed system with convolutional code and 4-PAM system as shown in Figure 5.1 is to be evaluated. With this system, the spectral efficiency is kept unchanged as compared to the uncoded 2-PAM system, i.e. $\rho = 2\, b/2D$. This allows for a fair comparison of error performance between the uncoded system and the proposed system for the bandwidth-limited channel.

5.4.1. Proposed system with hard decisions

The performance of the convolutional code in 2-PAM system with hard decisions was shown to be worse than the one with soft decisions by about 2 dB. However, it still allowed for an improvement in error performance for $E_b/N_0 > 4$ dB. In this section, the performance of the convolutional code with hard decisions in 4-PAM system is evaluated. Notice that the 4-PAM detector in Figure 5.1 uses MDD rule to distinguish among the noisy symbols and uses either Gray mapping or no-Gray mapping as shown in Table 5.1 to transform the detected symbols into bits which are then decoded with the hard-decoding Viterbi algorithm.

![Figure 5.7. Error rates for the proposed system of Figure 5.1 with hard-decoding.](image-url)
Figure 5.7 shows the simulation results. The blue curve represents the error performance of the uncoded 2-PAM baseline system. The green curve was obtained when no-Gray mapping was used and the pink curve when Gray mapping was used. As expected, the system with Gray mapping has a better error performance than the system with no-Gray mapping. Neither of the coded systems, however, showed improvement over the uncoded 2-PAM system.

We therefore conclude that when hard decisions are used, the improvement in error performance offered by the convolutional code is less significant than the deterioration in error performance caused by the use of multilevel transmission. Consequently, the proposed system with hard decisions failed at enabling improved error performance for the bandwidth-limited channel.

5.4.2. Proposed system with soft decisions

The proposed system of Figure 5.1 is simulated with soft decisions. In Section 5.3, it was seen that the error performance of the convolutional code in 2-PAM system with soft decisions was about 2 dB better than the system with hard decisions. When a 3-bit quantizer was used at the receiver, the improvement in error performance was slightly less than the improvement in error performance of the ideal Viterbi decoder which has no restrictions on path history length and uses infinitely fine quantization. Consequently, and since our focus is a theoretical study rather than implementation, in this section the simulations are performed with the ideal Viterbi decoder.

For the 2-PAM case, The Viterbi decoder utilizes soft inputs where each input value corresponds to one of the levels produced by the quantizer and represents certain degree of confidence, e.g. the lowest level produced by the quantizer represents the most confident ‘0’ and the highest level represents the most confident ‘1’. In Matlab, the built-in function \texttt{vitdec} allows for straightforward utilization of this function for the purpose of obtaining the decoded sequence from soft inputs when binary transmission is used (2-PAM); however, when multilevel transmission is used, there is no way of using \texttt{vitdec} without modifying the C-function \texttt{vit} called by \texttt{vitdec}. For 4-PAM and with reference to Table , the lowest level of the quantizer should represent the most confident ‘00’ and the highest level should represent either the most confident ‘11’ if no-Gray mapping is used, or the most confident ‘10’ if Gray mapping is used. Accordingly, the \texttt{vitdec} function could be modified; but it was found to be easier to modify the codes provided by [23] for appropriate work with the 4-PAM system. The Matlab code used for simulation of the proposed system with soft decisions and Gray mapping is given in the Appendix of this document.

The simulation results are shown in Figure 5.8. The blue curve represents the error performance of the uncoded 2-PAM baseline system. The green curve represents the error performance curve when no-Gray mapping was used and the pink curve was obtained when Gray mapping was used. As expected, the system with Gray mapping has a better error performance than the system with no-Gray mapping. The coded system with no-Gray mapping did not show any improvement over the uncoded 2-PAM system. However, the coded system with Gray mapping did show an improvement in error performance over the uncoded 2-PAM system for values of $E_b/N_0 > 4.5$ dB. By extrapolating the pink curve up to BER of $10^{-5}$, it can be estimated that $E_b/N_0 \approx 7.5$ dB and the coding gain in terms of $E_b/N_0$ can be found to be about 2 dB. This coding gain can also be expressed in terms of $SNR_{nrm}$ by using (2.8) and (2.10), and with $\rho = 2b/2D$, $b/2D$.
it can be found that $SNR_{norm} = (\rho E_b/N_0)/(2^\rho - 1) \approx 5.7 \text{ dB}$. Consequently, the coding gain in terms of $SNR_{norm}$ is about $8.4 \text{ dB} - 5.7 \text{ dB} = 2.7 \text{ dB}$.

Therefore, a coding gain of 2 dB on $E_b/N_0$ or equivalently 2.7 dB on $SNR_{norm}$ was obtained with the proposed system. This coding gain is larger than that obtained with the Gosset lattice $E_b$ code which is about 2.2 dB on $SNR_{norm}$. See Figure 3.5. We conclude that when the proposed system uses soft decisions and Gray mapping, the improvement in error performance offered by the convolutional code is more significant than the deterioration in error performance caused by the use of multilevel transmission. The net improvement in coding gain at BER of $10^{-5}$ achieved by this system was better than a Gosset lattice $E_b$ code by about 0.5 dB. Hence, the use of soft decisions and Gray mapping with the proposed system (which is simpler than those introduced in Chapter 3, i.e. lattice codes and TCM) not only allows for improved error performance for the bandwidth-limited channel, but also comparable performance to the more sophisticated codes of Chapter 3.
5.5. Discussion

From the simulations results, it can be said that for the bandwidth-limited channel it is, in fact, possible to achieve an improvement in error performance. However, since the simulations did not deal in a straight manner with the maximization problem mentioned in Section 4.2, it cannot be asserted yet that it was possible to find an arrangement of constellation points with increased distances between constellation points and with the same average signal energy. Further mathematical analysis of the distance properties of the proposed system is therefore needed.

The simulations performed in this thesis were not made over the recommended “playing field” for the bandwidth-limited regime: $P_e(\varepsilon)$ vs. $SNR_{\text{norm}}$. Instead, it was preferred to use the $P_e(\varepsilon)$ vs. $E_b/N_0$ “playing field” because of two reasons: First, it enabled the validation of the simulation setup of the uncoded systems and the convolutional code by straightforward comparison of error performance curves with those obtained by others [20–23]; and second, since the spectral efficiency of the system was set up to $\rho = 2 b/2D$, there was enough freedom to choose any of the “playing fields”. Recall from Chapter 2 that $\rho = 2 b/2D$ is considered as the dividing point between the bandwidth-limited regime and the power-limited regime.
Chapter 6

Conclusions

In this thesis, coding for the bandwidth-limited AWGN channel was treated considering the typical layered model of a digital communications system where coding and modulation are separate units. The main task was to verify whether an improvement in error performance can be achieved without bandwidth expansion with coding methods different from those presented in Chapter 3, i.e. lattice codes and TCM. This research problem was derived from the basic idea of coding as the process of increasing Euclidian distances. A simple analysis in the Euclidian space suggested that it is unlikely to find an arrangement of constellation points with the same average signal energy but with increased distances between constellation points. A first assumption was made that if, while keeping the signal energy and the symbol rate constant, an improvement in error performance could be achieved; then, it would be possible to find such an arrangement. This thesis focused on achieving an improvement in error performance under these circumstances rather than on finding such an arrangement.

A system was proposed with a conventional convolutional code with code rate 1/2 and using multilevel transmission (4-PAM). With this system, the spectral efficiency was fixed to $\rho = 2 \text{ b/2D}$, the same as for the uncoded 2-PAM baseline system. Performance analysis of the proposed system was simulated in Matlab for different combinations of Gray or no-Gray mapping and hard- or soft-decoding. When soft-decoding and Gray mapping was used, at BER of $10^{-5}$, a coding gain of approximately 2 dB on $E_b/N_0$ or equivalently, a coding gain of approximately 2.7 dB on $SNR_{\text{norm}}$ was obtained with no increase in bandwidth. It can be therefore concluded that the use of multilevel transmission and soft decoding brings up interesting opportunities for achieving improved performance over the bandwidth-limited channel. When the Viterbi decoder is simulated with soft decisions and the multilevel symbols are Gray mapped, the improvement in error performance caused by the convolutional code is more significant than the deterioration in error performance caused by the use of multilevel transmission. The proposed system (which is less sophisticated than lattice codes or TCM) not only showed that it was possible to achieve improved error performance without bandwidth expansion, but also it turned out to yield a coding gain better than that of a Gosset lattice $E_8$ code. This is an important result of this thesis, since it opens up interesting possibilities for implementation of simpler coding schemes for the bandwidth-limited channel, but in order to increase the coding gain, further simulations are required with higher number of transmission levels and with more sophisticated convolutional codes or perhaps with other types of codes, e.g. capacity approaching codes like LDPC codes or turbo codes.

Future work includes not only the simulation of different types of codes with higher number of transmission levels, but also the mathematical analysis of the proposed system and future improved systems. Since, this thesis did not deal in a straight manner with the maximization
problem mentioned in Section 4.2. It could not be asserted yet that it was possible to find an arrangement of constellation points with increased distances between constellation points and with the same average signal energy. Therefore, further mathematical analysis of the distance properties of the proposed system is needed. It also remains to find the theoretical bounds of the proposed system.
References


Appendix

Matlab code for the simulation of the proposed system with soft decisions and Gray mapping

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BER performance for the bandwidth-limited AWGN channel
% Proposed system: convolutional code + 4-PAM with soft-decoding & Gray mapping
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This system achieves a coding gain of 2 dB on EbNo at BER=10^-5
% without bandwidth expansion.
% Spectral efficiency fixed at 2 b/2D before and after coding.
% Conv. code rate: 1/2, constraint length: 3, generator polynomials: [7,5] octal
% The Viterbi decoder uses soft decisions with infinitely fine quantization
% Symbols are mapped using Gray mapping:
% 10 -> +3*alpha
% 11 -> +1*alpha
% 01 -> -1*alpha
% 00 -> -3*alpha
% where alpha is chosen to satisfy the average signal energy constraint
% This script was checked with Matlab 7.0.1 (R14)
% Author        : Juan W. Alfaro Zavala
% Email         : jualza33@hotmail.com
% Date          : 4th July 2012
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clear all, close all,

% data generation
N=4e5;
bias=randint(1,N,2,123);
bias(end-1:end)=0; % to force the encoder to end up in the 00 state

% convolutional coding
t=poly2trellis(3,[7 5]);
bias_cod = convenc(bias,t); % Encode the input bits

% mapping coded bits into 4-PAM symbols
alpha=sqrt(1/5); % for average signal energy of 1 (same as uncoded 2-PAM)
bias_cod=bias_cod(:,);
temp=bias_cod;
temp1=reshape(temp,2,length(temp)/2).';
temp2=bi2de(temp1,'left-msb');
gray_mapping = [0 1 3 2];
temp3=gray_mapping(temp2+1);
level4_mapping= [-3 -1 1 3];
temp4=level4_mapping(temp3+1);
sym4=temp4*alpha; % 4-PAM symbols to be transmitted over the channel
Es4=mean(abs(sym4).^2); % Average energy per 4-PAM symbol, should approach 1
% EbNo = Es4No
EbNodB=0:10; EbNo=10.^(EbNodB/10);
Es4NodB=EbNodB; Es4No=EbNo;

% transmitting symbols over AWGN channel at different EbNo
BER=zeros(1,length(EbNo)); nerrors=zeros(1,length(EbNo)); count=1;
for I=1:length(EbNodB)
    % NumErr=10; % required number of errors for a minimum confidence of 95%
    % count=0; %to count the number of blocks sent through the channel
    % while nerrors(I)<NumErr
    %     generation of AWGN with zero-mean and variance = sigma^2
    %     sigma=sqrt(Es4/(2*Es4No(I)));
    %     n=sigma*randn(1,N);
    %     sym4_rcv=sym4+n;
    %     bits_dec = vitsoft4(sym4_rcv); % Viterbi decoding
    %     nerr=sum(abs(bits_dec-bits)); % finding errors
    %     nerrors(I)=nerrors(I)+nerr;
    %     count=count+1;
    % end
    BER(I)=nerrors(I)/(N*count);
end
BER_theo_uncoded=0.5*erfc(sqrt(EbNo));
figure,
semilogy(EbNodB,BER_theo_uncoded,'-b'); hold on;
semilogy(EbNodB,BER,'*r'); grid on;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function bits_detected = vitsoft4(symbolsrcv)
% This function utilizes the Viterbi Algorithm to perform soft decoding of
% the data produced by the proposed system: 4-PAM + convolutional code,
% code rate: 1/2, constraint length: 3, generator polynomials: [7,5] octal
% The average signal energy must be 1.
% This script is based on that provided by:
% slight modifications were made for correct decoding of 4-PAM symbols
alpha=sqrt(1/5); % for average signal energy of 1
refSoftm= alpha*[ -3 -3; -1 -1; 3 3; 1 1 ];
ipLUTm=[ 0 0 0 0;...
        0 0 0 0;...
        1 1 0 0;...
        0 0 1 1 ];
cipSoftm=symbolsrcv ;
N = length(cipSoftm) ; % number of symbols
cipSoftm=[cipSoftm; cipSoftm];
cipSoftm=cipSoftm(:).';

% Viterbi decoding
pmSoftm  = zeros(4,1); % soft path metric
svSoft_vm = zeros(4,N); % soft survivor path
for ii = 1:N
    rSoftm = cipSoftm(2*ii-1:2*ii); % taking two copies of a received symbol
    % computing the euclidean distance
    rSoftvm = kron(ones(4,1),rSoftm);
    euclideanDistm = sum(abs(rSoftvm-refSoftm),2);
if (ii == 1)||(ii == 2)

    % branch metric and path metric for state 0
    bm1Softm = pmSoftm(1,1) + euclideanDistm(1);
    pmSoft_nm(1,1) = bm1Softm;
    svSoftm(1,1) = 1;

    % branch metric and path metric for state 1
    bm1Softm = pmSoftm(3,1) + euclideanDistm(3);
    pmSoft_nm(2,1) = bm1Softm;
    svSoftm(2,1) = 3;

    % branch metric and path metric for state 2
    bm1Softm = pmSoftm(1,1) + euclideanDistm(4);
    pmSoft_nm(3,1) = bm1Softm;
    svSoftm(3,1) = 1;

    % branch metric and path metric for state 3
    bm1Softm = pmSoftm(3,1) + euclideanDistm(2);
    pmSoft_nm(4,1) = bm1Softm;
    svSoftm(4,1) = 3;

else

    % branch metric and path metric for state 0
    bm1Softm = pmSoftm(1,1) + euclideanDistm(1);
    bm2Softm = pmSoftm(2,1) + euclideanDistm(4);
    [pmSoft_nm(1,1) idx] = min([bm1Softm,bm2Softm]);
    svSoftm(1,1) = idx;

    % branch metric and path metric for state 1
    bm1Softm = pmSoftm(3,1) + euclideanDistm(3);
    bm2Softm = pmSoftm(4,1) + euclideanDistm(2);
    [pmSoft_nm(2,1) idx] = min([bm1Softm,bm2Softm]);
    svSoftm(2,1) = idx+2;

    % branch metric and path metric for state 2
    bm1Softm = pmSoftm(1,1) + euclideanDistm(4);
    bm2Softm = pmSoftm(2,1) + euclideanDistm(1);
    [pmSoft_nm(3,1) idx] = min([bm1Softm,bm2Softm]);
    svSoftm(3,1) = idx;

    % branch metric and path metric for state 3
    bm1Softm = pmSoftm(3,1) + euclideanDistm(2);
    bm2Softm = pmSoftm(4,1) + euclideanDistm(3);
    [pmSoft_nm(4,1) idx] = min([bm1Softm,bm2Softm]);
    svSoftm(4,1) = idx+2;

end

pmSoftm = pmSoft_nm;
svSoft_vm(:,ii) = svSoftm;

end

% trace back unit
currSoftStatem = 1;
ipHatSoft_vm = zeros(1,N);

for jj = N:-1:1
    prevSoftStatem = svSoft_vm(currSoftStatem,jj);
    ipHatSoft_vm(jj) = ipLUTm(currSoftStatem,prevSoftStatem);
    currSoftStatem = prevSoftStatem;
end

bits_detected = ipHatSoft_vm;