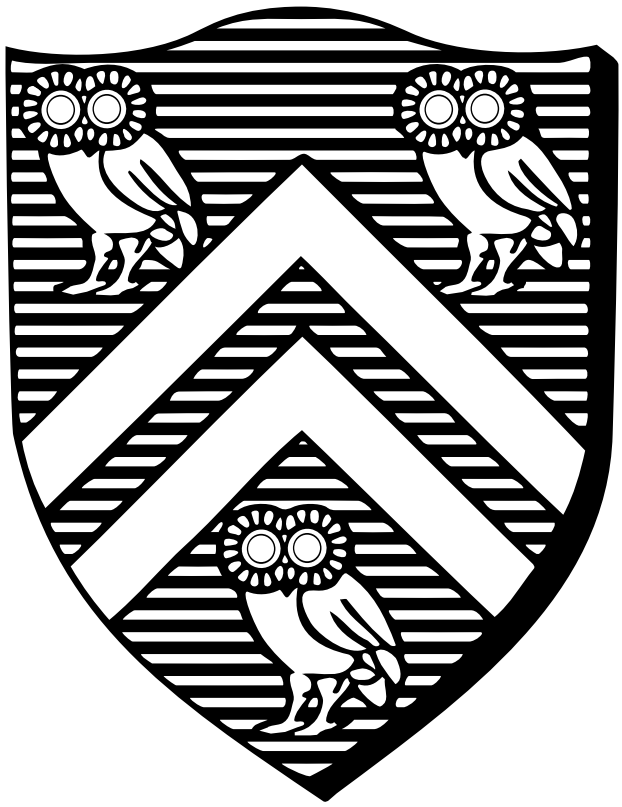
Multiuser Information Processing in Wireless Communication

Suman Das



Thesis: Doctor of Philosophy
Electrical and Computer Engineering
Rice University, Houston, Texas (September 2000)

RICE UNIVERSITY

Multiuser Information Processing in Wireless Communication

by

Suman Das

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

Doctor of Philosophy

Approved, Thesis Committee:

Joseph R. Cavallaro, Chair
Associate Professor
Electrical and Computer Engineering

Behnaam Aazhang
Professor
Electrical and Computer Engineering

Keith D. Cooper
Professor
Computer Science

Elza Erkip
Assistant Professor
Electrical and Computer Engineering
Brooklyn Polytechnic University, NY

David L. Applegate
Shannon Labs, AT&T, NJ

Houston, Texas

September, 2000

To Baba and Ma,

My greatest heroes.

Multiuser Information Processing in Wireless Communication

Suman Das

Abstract

Wireless channel is not very conducive towards error-free raw data transmission. On the other hand the tremendous growth in wireless services has made the channel bandwidth a scarce resource and effective utilization of this resource is mandatory. Thus it is instructive to know the limits of a wireless channel.

Shannon's theorems on channel capacity have been used so far to find the maximum rate at which data can be transmitted over any noisy channel. The theorem calculates the minimum signal to noise ratio (SNR) required to transmit data across a channel with *zero* probability of sequence error. However the result is practically inhibitive, as it requires encoding and decoding of *infinite length* code sequences. Practical finite codes never achieve this zero error limit.

For practical code design bit-error-rate is often a preferred metric over sequence error rate. However there is no satisfactory method to compare the Shannon's capacity results with the bit error rate performance of the practical codes. We introduce the notion of **distorted channel capacity** to bridge this gap. This measure defines the capacity of a channel when a particular bit-error-rate is allowed. It can also be used as a benchmark to measure the "goodness" of a code.

Our results show that most of the practical codes lie far beyond the capacity

limit. We see that *Turbo codes* and the *convolutional codes* come close to this achievable at a prohibitively large computational cost. Specifically, for the convolutional codes the performance improves with large *constraint length* codes. However the optimal decoding complexity of the convolutional codes grow exponentially with this parameter. We propose a suboptimal decoding technique that has linear complexity in the size of the constraint length and provides close to optimal performance.

We further extend our results to a multiuser environment. The optimal joint decoding complexity of multiple users data symbols is exponential in the number of users. Our proposed iterative joint interference cancellation and decoding technique provides computational gain without performance loss.

Acknowledgments

Graduate students have advisors. I had Joe and Behnaam. Through thick and thin, through good days and bad, with the delicate blend of pragmatism and encouragement, with the perfect mix of support, guidance and constructive criticism, with patience and perseverance, and above all with joy and laughter you have moulded me into what I am. Joe and Behnaam, thanks for believing in me and helping me believe in myself. Others have advisors. I have friends and mentors.

This is also the time to thank Elza, who introduced me to the wonderful world of information theory. I am grateful for the countless hours you spent listening to my problems and sharing your insights, for meticulously going through my papers and thesis, offering your elaborate suggestions to help improve my seriously handicapped technical writing skills and cheering me up on those gloomy days with your congenial and contagious smile.

Thanks are also due to my other two committee members, Keith Cooper and David Applegate, who spent several afternoons helping me straighten my thoughts. Thanks for all the pointers to the numerous relevant literature and all the stimulating discussions and thought-provoking comments.

Life in Rice wouldn't have been half as gratifying without all the great people around me. From faculty members like Rich, Rob and Ed who always had their doors open for me, to staff members like Emma, Mandy, Nora and Dee who never failed to help me and all the graduate students in ECE and beyond, who were there

for laughter and volleyball and musicals and “barrier-breaking” lunches - thanks for these memorable five years in Rice. A special mention goes to all my CMC collagues and my roommates SK, Karthick, Partha, Kalla and Kiran. Thanks for being an indispensable part of my life.

And last but not the least, I am grateful to my mom and dad and my relatives back home who were always there for me with their constant support and prayers. You are responsible for this.

Contents

Illustrations

Chapter 1

Introduction

1.1 Introduction

The last quarter of the year 1998 was marked by two watershed events. For the first time in the history of telecommunication, chips for wireless industry outsold all other silicon products. The convenience and popularity of tetherless connection has firmly established wireless communication as one of the fastest growing technologies.

The emergence of multimedia technology is also redefining the traditional face of communication. Since 1998, non-voice data is dominating the international telecommunication traffic and this trend is also expected to appear in the domestic lines. If both these trends continue, and there is no reason to believe otherwise, we can easily envision a future wireless channel with multimedia traffic streaming through it.

Wireless multimedia will open up a plethora of new applications in medicine, education, entertainment and a variety of other areas. However its success and proliferation will be dictated by the successful convergence of computing and communication. This is evident from the future trends of media applications that impose unprecedented demand on both communication bandwidth and processing power. Even with the significant improvement in both the wireless communication and the computing technology, implementation of a real-time wireless multimedia service still remains a monumental challenge.

This thesis deals with efficient wireless solutions for broadband applications. We

explore information theoretic bounds on the capacity of wireless channels; we develop computationally efficient receiver (specifically detection and decoding) algorithms to achieve the system capacity; and we discuss their effective implementations in a multiuser wireless system. But before we embark on the journey to the future of wireless systems, their problems and proposed solutions, let us take a look back at the genesis and evolution of wireless communication.

1.2 A brief history of wireless communication

The origin of wireless communication can be possibly dated back to the discovery of radio by Marconi in 1895. Initially the application of wireless devices was limited to communication between locations that cannot be connected by wires. Most of the technology was used in military applications especially during World War II [1], when mobile and portable radio played a significant role. However, until almost the middle of the twentieth century, wireless personal communication was not a part of traditional telecommunication. It was only in 1946, that mobile radios were first connected to the public switched telephony network (PSTN) [2].

Initially, the wireless communication was only point-to-point, half-duplex and could support only a few users. However the demand and congestion of mobile telephony soon revealed the inadequacy of the system. The first cellular concept was proposed in 1968 by AT&T. The idea behind cellular telephony is to divide the geographical region into several smaller sub-regions or *cells* and each cell is assigned a fraction of the entire available bandwidth [3]. Distant cells however can use the same spectrum. This concept of re-use of spectrum, greatly increases the capacity of the system. The size and distribution of the cells is determined by the demand of service [4].

Each cell has its own *base station* which is responsible for micro- and macro-management of subscribers requesting service from within the cell. They also control the *hand-over* service to next cell as the users move from one location to the next. The base-stations are inter-connected by the mobile telephone switching office that also acts as a gateway to the PSTN. Cellular systems have proven to be extremely successful. As mobile communication evolved from a niche item for the rich to a mass-market consumer product, newer wireless standards developed from analog systems to its modern day complex digital incarnations.

1.2.1 First generation wireless system

The first generation wireless services in the US were analog services provided by Analog Mobile Phone Service (AMPS) system [3] and predominantly offered voice services. The voice signal was frequency modulated onto carriers in the 800 MHz spectrum and each user was assigned a 30KHz bandwidth. Voice over air interface was the primary service provided. Special subscriber units supported limited data services like fax and modem. But these services were poor in quality and supported very low transmission rates. Moreover the system utilization was extremely inefficient (only one voice call in over 200 kHz). Rising demands for better services led to deployment of second-generation standards.

1.2.2 Second-generation wireless system

The second-generation wireless systems [5, 6, 7] are characterized by digital air-interface technology. They also started to offer data services along with voice and is presently the most widely deployed system. Table 1.1 provides a comparative study of the various 2G based systems currently in operation. The second-generation

Network	Frequency band	Multiple access technology	Description
Digital AMPS (D-AMPS)	800 MHz	FDMA-TDMA	Based on IS-54 standard and is compatible with the AMPS standard. Provides three subscribers per 30KHz channels.
Global System for Mobile communications (GSM)	900 & 1800 MHz	FDMA-TDMA	Primarily used in Europe. The uplink and downlink are broken into several 200KHz bandwidths and each channel is divided into 8 TDMA slots or *logical channels* for each user.
CDMA Digital Cellular	900 MHz	CDMA	Based on IS-95 standard. Data bit rate of 9600 bit/s is spread 128 times to a chip rate of 1.288 Mchip/s using spreading, interleaving and coding. Soft handover and multipath diversity combining improves performance at the edges.

Table 1.1 : Description of various 2G wireless systems

wireless system primarily supports two types of multiplexing techniques. TDMA (Time Division Multiple Access) is mainly a store and forward technique where the available time is divided into several slots. Data from each user is buffered and transmitted during the corresponding time slot assigned to the user. CDMA (Code Division Multiple Access) on the other hand is a spread spectrum technique [8, 9]. Data bits of each user are modulated by the unique spreading waveform associated with the user. Because CDMA spreads each call over the entire available frequency band, it is more immune to interference than TDMA and can support more users per channel in some situations. The service provided by the second-generation wireless system is far superior to the first generation systems, however the data rate supported

is still much lower than the demand.

1.2.3 2.5G wireless systems

The next true generation of wireless technologies will be called third generation (3G); however, an interim step called 2.5G [10] is planned for release until the 3G services become available. The 2.5G system characteristics are:

- High Speed Circuit Switched Data (HSCSD) increases the speed of the air channel by changing the error-correction codes and aggregating channels. It will be efficient for applications such as file transfer and video transmission.

- General Packet Radio Service (GPRS) is a packet-based data transmission technology that will initially provide data transfer rates of up to 115 Kbps. GPRS will work with CDMA and TDMA, and it supports both the IP and X.25 communication protocols.

- Enhanced Data rate for Global Evolution (EDGE) will increase data rate up to 384 Kbps, by using 8PSK (phase-shift-keying) modulation rather than the traditional BPSK.

1.2.4 Third generation wireless systems

The 3^{rd} generation wireless represents the next generation of wireless standards. The main goal is to harmonize all the available seemingly incompatible standards. The International Telecommunications Union (ITU) has made great strides towards a unified IMT-2000 (International Mobile Telecommunications) [11] standard through committees in Europe, Japan, South Korea and United States [12, 13, 14]. CDMA has been chosen as the multiple access technology for the 3G systems. There are

two forms of CDMA standards currently under consideration. While Europe and Asia have chosen W-CDMA (Wideband CDMA) as their standard, the US standard [15] is called CDMA2000. The main difference is that the standards are backward compatible with GSM and IS-95 networks respectively. (However recent agreement between Ericsson and Qualcomm may pave the way towards a unified solution).
The mission of 3G systems is to:

- Provide users with high bandwidth service anytime anywhere. Mobile data rates up to 384 Kbps and fixed wireless data rates up to 2 Mbps are expected to be available to the users.

- Support a full gamut of voice, data, audio, video and internet services.

With the worldwide acceptance and deployment of 3G systems, newer telecommunication services are also coming into prominence. We are seeing the explosion of internet services, rapid convergence of voice and data networks and the widespread use of media rich applications in the wireless systems. This is enabling a new genre of communication, aptly termed wireless multimedia.

1.3 Challenges of the next generation wireless systems

The new wireless generation however brings in new acute challenges. Due to the widespread deployment and popularity of wireless services the number of wireless customers has increased dramatically over the past few years. Moreover the multimedia traffic demands much higher data rate and hence wireless bandwidth than voice data. Since the available wireless bandwidth is not increasing at the same pace, the traffic density and multiple access interference level have increased tremendously. Since a CDMA based system is inherently interference limited, in the new situation,

with the current solution, the bit-error rate per transmission will be severely affected. Additionally, the multimedia data is lot less tolerant towards channel errors. Finally, if we want to provide real time services, retransmission of erroneous frames may not be an alternative. Error free one-time transmission becomes a priority. Efficient data transmission and reception techniques hence are needed to effectively use the scarce wireless bandwidth. It is obvious that we will need to incorporate forward error correcting (FEC) techniques to combat the errors in transmission. But before we deploy any such scheme, it is always helpful to know the limits/capacity of a wireless channel.

Shannon's theorem on channel capacity has pointed to the maximum rate at which data can be transmitted over any noisy channel. However the results are applicable only in restricted environment. Shannon's theorem provides the capacity bounds for systems that will not allow any sequence error, but the theorem is not constructive and does not tell us methods to find codes that will enable *reliable* transmission. Moreover the results are prohibitive for implementation as they require infinite length codewords for error-free transmission. All practical systems have finite length codewords and finite computational resources for decoding and as expected, almost none of the codes meet the zero-error criteria. Hence the sequence error rate is not an appealing metric to the designers. Most of the real-life codes are characterized by their bit error rate performance.

Ideally, in order to measure the "goodness" of a code we should compare the performance against the Shannon's rates. However the capacity results based on sequence error are too pessimistic. In fact, we contend that if we allow a limited fraction of the information bits to be in error (low bit error rate), we should be able to transmit at a rate higher than that dictated by Shannon capacity. The first

part of our thesis is aimed at augmenting the results of Shannon's channel capacity theorem to incorporate bit-error rate and provide a basis to compare the bit error rate performance of various codes against the best possible code.

A natural extension of the above result is to actually find codes and efficient decoding techniques that perform close to the optimal bounds. Convolutional codes have been widely used as a FEC technique in a noisy communication system. The strength of the convolutional code depends on the rate of the code and the constraint length of the code. Since the rate of the code determines the throughput of information across a channel, lowering the rate is not an attractive alternative. We want to maximize the throughput rate to use the wireless bandwidth most efficiently. Naturally, in order to achieve the optimal performance we will have to use large constraint length codes.

The most popular method of decoding convolutional codes is the Viterbi algorithm. Unfortunately the complexity of this algorithm grows exponentially with the constraint length. For any real-time service, the Viterbi algorithm cannot be used to decode convolutional codes with large constraint lengths. If we want to provide wireless communication services that satisfy stringent QOS requirement it will be imperative to develop algorithms that can decode "strong" convolutional codes in real time.

Finally, in a practical system, a wireless channel will support a multitude of users, each using convolutional codes or other FEC schemes to protect against error. The optimal decoding technique that estimates the transmitted data of all users simultaneously requires exponential complexity in the number of users. Any practical wireless system would require a real-time solution for the receiver. Hence there is a strong and urgent need for the design of computationally efficient multiuser receivers.

1.4 Contribution

Our contributions in this area are threefold:

- We have developed a new measure of channel capacity, that calculates the maximum rate at which data can be transferred across a Gaussian channel when we can allow a certain amount of error in the recovered messages. This measure acts as a benchmark for the performance of practical codes. Before this expression for capacity, we could find the maximum rate at which we can transfer information across the channel when there is no error in the recovered sequence. If we consider the performance at very small bit error rate we probably can still live with the expression of Shannon's capacity, but for audio data the accepted bit error rate is of the order of 10^{-3}. At this high rate of errors, we find Shannon's expression for capacity of channel is very pessimistic and in fact it is possible to transfer data at higher rates.

  This result will act as a lower bound for bit error rate for all the forward error correcting codes, so researchers can continue their search for good codes until they hit this lower bound. We also study the performance of turbo codes and convolutional codes and observe that the performance of these codes come close to this bound at very high computational complexity.

- We propose a computationally efficient algorithm to decode convolutional codes. We observe that only the performance of convolutional codes of large constraint length come close to the optimal bounds. Unfortunately the traditional Viterbi algorithm is too complex to decode codes with large constraint length. Our proposed algorithm has a complexity quadratic in the constraint length. The performance of this suboptimal algorithm is close to that of the Viterbi al-

gorithms. This new decoding technique will enable us to decode codes with larger constraint length. Even if we lose marginally in performance because of the suboptimal nature of the decoding technique, since we will be able to decode much stronger codes than is possible with the Viterbi algorithms under the real-time constraint, we will ultimately gain in absolute performance.

- We finally investigate a multiuser wireless communication system. In this system, the sources of error are not only the background noise but also the signals of other users. The data of each individual user are coded by some FEC codes. In case of a convolutionally encoded system the optimal decoding strategy is to simultaneously decode the information stream of all users at the base station. However, the complexity of this decoding scheme is exponential in the number of users. We propose an iterative interference cancellation technique combined with maximum-a-posteriori (MAP) decoding of convolutional codes that reduce the complexity of the decoding algorithm. Our simulation study shows that the performance of this algorithm is close to that of the optimal algorithm.

The rest of the document is organized as follows. In chapter 2, we talk about *distorted channel capacity*. In that chapter, we also study the performance of the practical codes. In chapter 3, we propose the maximal basis decoding of convolutional codes. In chapter 4, we investigate joint iterative multiuser decoding techniques in a CDMA system. Finally, we present some future research directions.

Chapter 2

Distorted channel capacity

"*The fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.*" Claude E. Shannon.

In almost all practical communication systems, the raw information signals can never be reproduced perfectly at the receiver. In order to protect against communication errors, we introduce redundancy in the transmitted signal. However added redundancy will reduce the *rate* of message transfer. A critical problem is to find a trade-off between the rate of communication and the error associated with the process for a given communication channel. In this chapter we quantify the optimal rate of message transfer across a channel when we allow a particular error rate in the recovered message.

2.1 Capacity of a communication channel

Wireless communication has become an integral part of our life. But we have not yet reached the pinnacle of this technology and there is a clear demand from the users and eagerness among the providers to support other data types over and above the traditional voice traffic. Wireless services are going to be enriched by multimedia data. However these additional services require support for more stringent quality of service (QOS) and higher data rates than the typical voice data. Unfortunately,

wireless bandwidth is not growing at the same pace as the demand for the service, and hence efficient use of the wireless channel is becoming mandatory.

In order to optimize the usage of the available bandwidth, we need to study the characteristics of the wireless channel. It is worthwhile to remember that even though in our future discussions we will mostly concentrate on the data transmitted across wireless channel, the scope of our results extends well beyond that. Communication of information can be as remote as a radio link between earth and an orbiting satellite, or it can be as close as transferring data from memory to a disk drive. "Channel" is a concept that introduces distortion to the data being transferred from source to destination. The type of distortion depends on the nature of the channel. It can

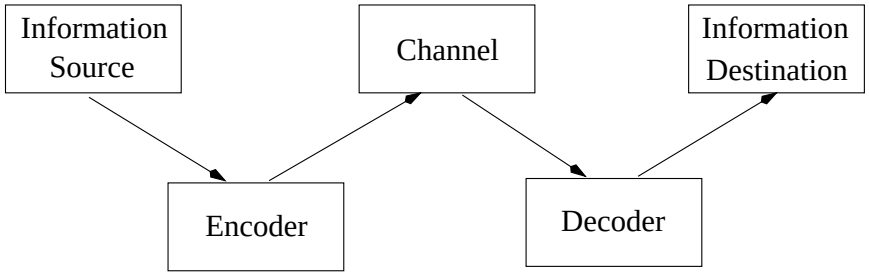


Figure 2.1 : Information transfer over a channel

be a crosstalk over a cable telephone line, radiation from cosmic sources in satellite channels or thermal or magnetic noise for a disk drive. Whatever be the cause, its presence is unavoidable and will eventually lead to erroneous recovery of transmitted signal.

There are two conventional ways to combat this distortion:

- Better channel design, and
- Redundant data transmission.

Better channel design usually strains the resources, bumps up the cost of the service and is sometimes precluded by physical limitations. The second approach is mathematically more challenging. In this mode, we accept the given noisy channel and incorporate methods to *detect* and *correct* the errors introduced by the channel. This is achieved by the addition of encoder and decoder blocks (Figure 2.1) to the communication system. In contrast to the physical solution, with additional computations, this solution can turn noisy channels into reliable systems.

The solution also involves addition of redundancy in the transmitted message. Since one of the primary goals of wireless communication is to transmit data as quickly and by using as little resource as possible, it is essential that the encoder introduce the least amount of redundancy. A significant amount of research has been directed towards the design of good encoders and decoders. The deployed encoder and decoder dictate the data transfer rate and of course the goal is to achieve the best possible performance without introducing additional redundancy.

It was a commonly held belief that the errors in the communication system increase as the redundancy in the transmitted signal decreases. But what is the least amount of redundancy to be added for reliable communication? Shannon's theorem [16] showed that for every channel there is a *threshold rate*, below which there exists an encoder and decoder pair that will ensure error-free transmission. Shannon's results quantified the theoretical limitations and potentials of any given channel. Without this result, the search for good codes would be rudderless. In fact, Shannon's result tells us the maximum rate that we can expect from a channel and when to stop looking for a "better" code. Shannon's theorem provides the lower bound on the redundancy to be introduced in the data. Efforts from then on have been geared towards designing codes that achieve this lower bound.

Apart from the traditional channel capacity defined by Shannon, several other researchers have proposed other measures of capacity. A commonly used approximate metric to determine the maximum rate of data transfer is the *cut-off rate* [17]. This measure is used when the real data is quantized into M level for transmissions. In order to take into account the finite decoding delay of practical codes Tse and Hanly introduced the concept of *delay limited capacity* [18]. In case of a fading channel a commonly used metric is the *probability of outage* [19].

It should be noted that Shannon's result specifies the maximum rate at which information can be transmitted if we want *error free* communication. This seems to be too strong a condition, since most of the wireless signals can tolerate some form of error. However Shannon's result does not specify the maximum rate at which we can transmit information if we are willing to tolerate a certain amount of error. In the following sections we will seek an answer to that problem. Before answering the question, we will briefly review Shannon's results on channel capacity.

2.1.1 Entropy

Information transmission is error-free when the recovered signal agrees perfectly with the transmitted signal. It is essential to identify the maximum number of signals that can be distinguished for n uses of the channel to ensure successful transmission. This number not only signifies the maximum amount of information that can be transferred across the channel in n attempts, but also determines the minimum amount of redundancy that should be associated with the transmission. Shannon showed that the number depends solely on the channel and termed it the **capacity** of the channel.

The channel induces a probability distribution on the output sequence for every

transmitted input sequence and the receiver has to recover the input sequence from this output sequence. If there is an one-one relationship between the input and the output sequences, the recovery of the signal would be error free. However it is conceivable that for some input sequences the corresponding output sequences might overlap. This will lead to ambiguity during the recovery and, in some cases, to errors. Since the transmitter has control over the input sequences to transmit and since the output sequences depend on the input signals, the key to error free transmission is to avoid input sequences whose output sequences overlap. This can be achieved by mapping (encoding) the original information signals to appropriate points in the input signal space, based on the channel characteristic, and then reconstructing (decoding) the original signal from the unambiguous received signal. This method is known as *joint typicality encoding and decoding* technique. In order to achieve this signal separation, it is essential to characterize the output sequences induced by a channel.

Let us assume that the source alphabet $\mathcal{X}$ is finite. **Entropy** is the measure of average information content or the uncertainty of the source. Specifically, if the input signal assumes the discrete values x with probabilities $p(x)$, then the entropy of the source is defined as

$$H(X) = -\sum_{x \in calX} p(x) \log p(x).$$

The entropy is the expected value of $\log 1/p(x)$. In fact it is a measure of the amount of average information required to describe the random variable. We know that the channel induces, for each input signal x, a distribution $p(y|x)$ on the output. We

can similarly define the *conditional entropy* as

$$\begin{aligned} H(Y|X) &= \sum_{x\in\mathcal{X}} p(x)H(Y|X=x) \\ &= -\sum_{x\in\mathcal{X}} p(x)\sum_{y\in\mathcal{Y}} p(y|x)\log p(y|x) \\ &= -\sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log p(y|x). \end{aligned}$$

The *relative entropy* or the *Kullback Leibler distance* between two probability mass function is defined by

$$D(p \parallel q) = \sum_{x\in\mathcal{X}} p(x)\log\frac{p(x)}{q(x)}.$$

This distance measure is used to define the mutual information of two random variables. This can be thought of as a measure of how accurately the distribution $p(x)$ matches the unknown distribution $q(x)$.

In the decoding process we have to estimate the transmitted signal from the received signal. The *mutual information* is defined as the relative entropy between the joint distribution $p(x,y)$ and the product distribution $p(x)p(y)$.

$$\begin{aligned} I(X;Y) &= \sum_{x\in\mathcal{X},y\in\mathcal{Y}} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \\ &= -\sum_{x,y} p(x,y)\log p(x) - \sum_{x,y} p(x,y)\log p(x|y) \\ &= H(X) - H(X|Y). \end{aligned}$$

We can interpret mutual information as the reduction in uncertainty of the source due to the knowledge of the output signal. The *channel capacity* is defined as the maximum value of this mutual information:

$$C = \max_{p(x)} I(X;Y).$$

Shannon's theorem showed that it is possible to transfer information across a channel with as small a frequency of error as desired by appropriately choosing the encoder

and the decoder when the rate is lower than the capacity of the channel. He further proved that no matter how "good" the code is, error free information transfer is not possible at a rate higher than the channel capacity.

2.1.2 The typical set

In this section, we will show that the maximum mutual information actually represents the channel capacity. First, we will introduce the concept of *asymptotic equipartition property* (AEP). It is essentially the application of weak law of large numbers, which states that if $X_1, X_2, \cdots X_N$ are independent and identically distributed random variables, then $-1/n \log p(X_1, \cdots, X_n)$ approach the entropy $H(X)$ of the system.

Based on the notion of AEP, we define the *typical set* A_ϵ^n, with respect to the given distribution $p(x)$, as the set of n-sequences $(x_1, \cdots, x_n)$ whose joint distribution is given by

$$2^{-n(H(X)+\epsilon)} \leq p(x_1, \cdots, x_n) \leq 2^{-n(H(X)-\epsilon)}.$$

It can be shown that for any given ϵ, we can always find a sufficiently large n, so that the probability of any n-sequence not being a typical sequence is less than ϵ, i.e. $\Pr\{A_\epsilon^n\} > 1 - \epsilon$. It can also be shown that all the typical elements of the typical set have equal probabilities and the cardinality of the typical set is nearly $2^{nH(X)}$.

Let us consider these definitions in light of information transmission across a channel. As before, let the source alphabet be $\mathcal{X}$ with the associated probability $p(x)$. Thus any n-sequence of the information bits will be a subset of $\mathcal{X}^n$. The channel induces an output distribution $p(y|x)$ for each input sequence. The output, which is also a n-sequence, will have the marginal distribution $p(y)$. Associated with both the input and output distribution will be the corresponding typical sets denoted

by X_ϵ^n and Y_ϵ^n respectively. We will use these notations in our following discussions.

$\mathbf{x} \in \mathcal{X}^n$: n-bit transmitted code sequence

$\mathbf{y} \in \mathcal{Y}^n$: n-bit received signal sequence

Code space $\mathcal{X}_* \subset \mathcal{X}^n = \{\mathbf{x} \mid \mathbf{x} \in X_\epsilon^n\}$

$\mathcal{Y}_* \subset \mathcal{Y}^n = \{\mathbf{y} \mid \mathbf{y} \in Y_\epsilon^n\}$,

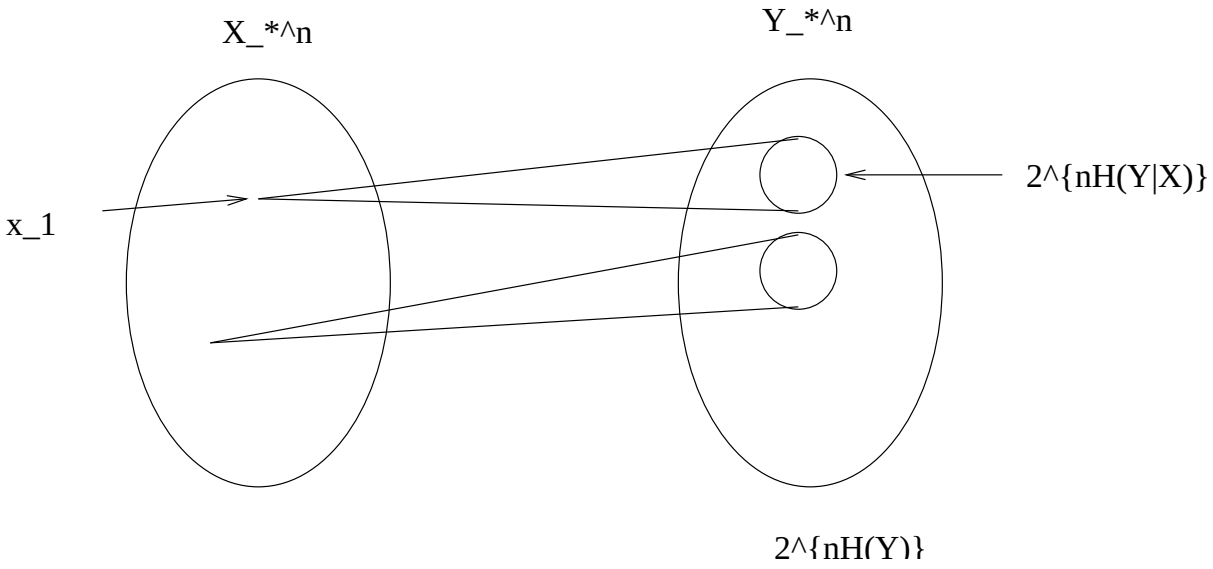


Figure 2.2 : Distortionless coding

From the properties of the typical set, for sufficiently large n, the size of the typical output space is given by $2^{nH(Y)}$. The same property holds for the input space. While considering the input or output sequences, we need to be concerned only with the corresponding typical set and henceforth we will drop the asterisk subscript from the notation when we talk about the space. For every typical $\mathbf{x}$, the channel will induce the output distribution based on $p(y|x)$. Following similar arguments as above, we can define a typical set of output sequences for each input sequence. We say that

there will be several $\mathbf{y} \in \mathcal{Y}$ which is jointly typical with the given $\mathbf{x}$. The number of such jointly typical $\mathbf{y}$ is given by $2^{nH(Y|X)}$. The whole typical space $\mathcal{Y}$ consists of $2^{nH(Y)}$ points. We want to chose our codeword from the typical space $\mathcal{X}$ in such a way that there aren't two $\mathbf{x_1}$ and $\mathbf{x_2}$ such that there exists a $\mathbf{y} \in \mathcal{Y}$ which is jointly typical with both $\mathbf{x_1}$ and $\mathbf{x_2}$. This is equivalent to saying that we want to chose the codewords in such a way that their corresponding jointly typical $\mathbf{y}$-sequence spaces are disjoint. Under this constraint, the maximum number of codewords that can be selected determine the capacity of the channel. In our case the maximum number of codewords that can be chosen is given by $2^{n[H(Y)-H(Y|X)]} = 2^{nI(X;Y)}$.

The result shows that the capacity of the channel is given by the maximum value of the mutual information. Shannon showed that there is at least one encoder-decoder pair that can achieve this rate and approach error free information transfer. He also showed that if the transmission rate is higher than the channel capacity then there is no hope of error free transmission.

2.2 Distorted channel capacity

2.2.1 Motivation

As we have seen so far, traditional channel capacity calculation ([16]) has been restricted to asymptotic *error free* transmission and retrieval of information across the channel. In this scheme an n-bit sequence is transmitted containing nR bits of information, where R is the rate of the code, and in the receiver end, the nR bits of information can be recovered exactly *iff* R does not exceed the capacity of the channel. The capacity result also suggests that if the information bits are coded at a rate higher than the capacity of the channel, the estimated nR bit information

sequence *definitely* will not match the corresponding transmitted bit sequence. Thus there is a sharp boundary between the viable and non-viable code rates for *reliable* transmission across a channel.

The error criteria considered in this calculation is the probability of sequence error. However none of the available practical codes achieve zero error rates. For example transmission over an additive white Gaussian noise (AWGN) channel is characterized by the signal to noise ratio (SNR) of the transmission. The capacity of the channel is a direct function of the SNR of the transmission. However for practical codes it is hardly ever known, at what value of SNR the code achieves zero rate. So it is not possible to find out how far the SNR for zero error rate is away from the optimal SNR defined by Shannon's theorem. Since zero error rate codes are not common, the usual metric to measure and compare the performance of the codes ([20]) is the bit error rate (percentage of recovered bits in error over the number of transmitted information bits).

For a given transmitted information sequence σ_1, if the estimated bit sequence is σ_2 and if $\sigma_1 \neq \sigma_2$, in the traditional channel capacity calculation this error event has an associated measure equal to 1. However in case of a bit-error rate calculation, instead of associating a uniform measure, 1, to all incorrect bit sequences, we associate a *distortion measure* proportional to the *distance* of the estimated bit sequence from the transmitted bit sequence. We want to find the capacity of the channel under this measure. More specifically given a distortion measure, and a maximum allowable distortion we want to calculate the achievable code rates.

2.2.2 System notations

We will use the concept of joint typicality encoding and decoding to calculate the capacity of the distorted channel. As before let $\mathbf{x}$ and $\mathbf{y}$ represent the n-bit long input and output sequences respectively. It should be remembered that $\mathbf{x}$ is the coded bit sequence. We will use $\mathbf{b}$ to represent the k-bit original information bit sequence. If R is the rate of the code, $k = nR$. In addition, there is a one-one relationship between the codeword sequence and the information sequence. In other words, given an information bit sequence we can uniquely determine the transmitted coded bit sequence and vice-versa. Let $\hat{\mathbf{b}}$ represent the estimated information bit sequence and $\hat{\mathbf{x}}$ be the corresponding estimated coded bits.

We will also introduce a notation to represent the distortion between the transmitted and received information sequences. If $\mathbb{B}^k$ represent the k-bit binary information bit sequences, then the distortion function is represented by

$$d(.,.) : \mathbb{B}^k \times \mathbb{B}^k \mapsto \mathbb{R}.$$

For our purposes we will use the *average Hamming distance* as the distortion measure. This is essentially the fraction of number of positions in which the estimated bits differ from the transmitted bits over the length of the transmitted sequence, k.

Given an acceptable distortion rate α, we want to design our decoding function

$$g : \mathcal{Y} \mapsto \mathbb{B}^k$$

to be such that

$$d(\mathbf{b}, \hat{\mathbf{b}}) \leq \alpha,$$

for all transmitted and corresponding received sequence $\mathbf{x}$ and $\mathbf{y}$ respectively and $\hat{\mathbf{b}} = g(\mathbf{y})$.

2.2.3 Geometric Interpretation of Joint Typicality Encoding and Decoding

In the previous section we have provided a geometric proof of Shannon's capacity results. We will use similar arguments to extend the results to allow for distortion. We term this new metric the *distorted channel capacity*. Figure 2.3 denotes the

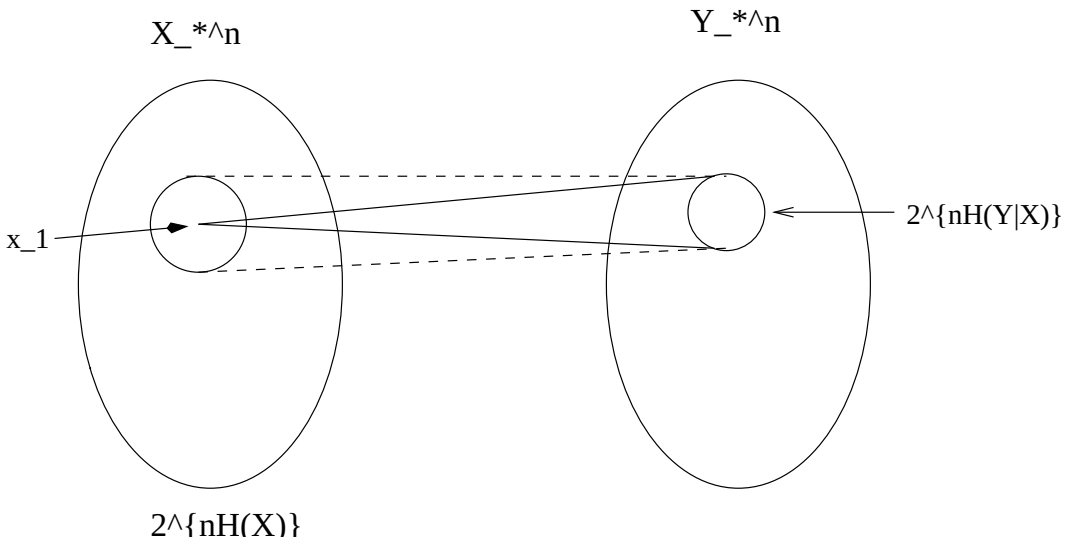


Figure 2.3 : Codes allowing distortions

encoding and decoding process. Let us assume that the codeword $\mathbf{x}_1$ is sent. We know there are $2^{nH(Y|X)}$ $\mathbf{y}$-sequences which are typical with $\mathbf{x}_1$. Let us call this typical set $\mathcal{Y}(x_1)$. Upon receiving any $\mathbf{y} \in \mathcal{Y}(x_1)$, we would be using joint typicality decoding to retrieve the transmit sequence. Let $S \subset \mathcal{X}$ be the set of elements in the typical input sequences such that for any $\mathbf{x} \in S$ there exists at least one element $\mathbf{y} \in \mathcal{Y}(x_1)$ which is jointly typical with $\mathbf{x}$. That is S is the set of possible input sequences that can be decoded as codewords using the joint typicality decoding method, when $\mathbf{x}_1$ is transmitted. Under the concept of Shannon's capacity results

only one codeword, $\mathbf{x}_1$, is allowed to be part of the transmit codebook for each S, to avoid any decoding error. However in the new scenario, we will allow more than one codeword to reside in the set S if it satisfies certain restrictions. If $\mathbf{x}_2$ is any other codeword we want to ensure that $d(\mathbf{b}_1, \mathbf{b}_2) \leq \alpha$, where $\mathbf{b}_1$ and $\mathbf{b}_2$ are the information sequences corresponding to $\mathbf{x}_1$ and $\mathbf{x}_2$ respectively.

Lemma 1 *The capacity of such a system is given by the maximum value of* $I(X;Y)/\big(1-H(\alpha)\big)$.

Proof: Let us give another geometrical interpretation of the distorted channel capac-

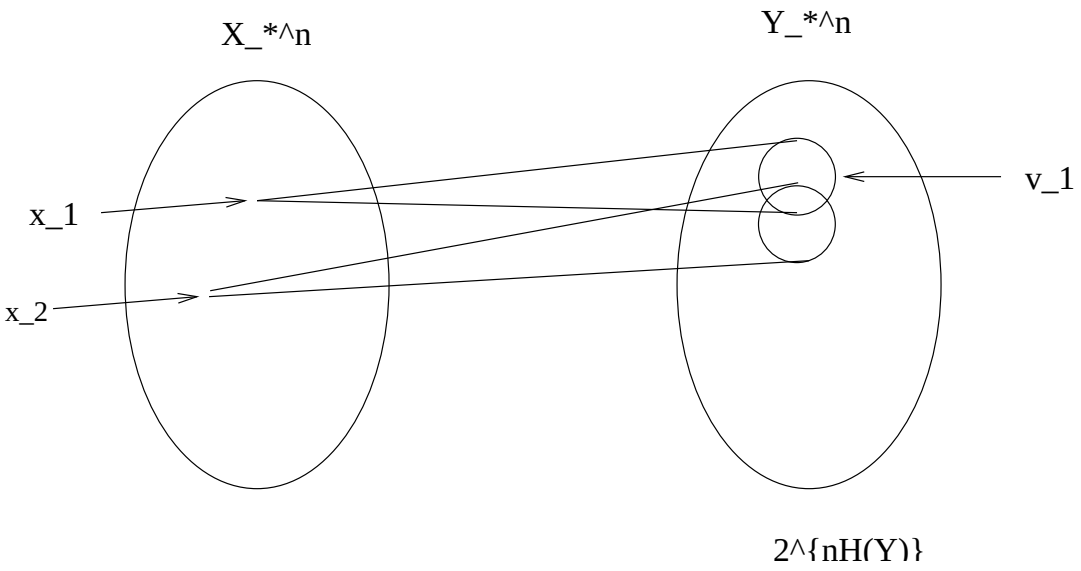


Figure 2.4 : Codes with overlapping jointly typical spaces

ity. It is shown earlier, that in case of Shannon's capacity results the jointly typical spaces corresponding to the codes are non-overlapping (Figure 2.2). This ensures that the decoding is error free. However since we are going to allow some error in the recovered sequences, we will allow the typical sequence spheres to overlap. If

the jointly typical space corresponding to two codewords $\mathbf{x}_1$ and $\mathbf{x}_2$ overlap then we must ensure that the information bits corresponding to the codewords satisfy the distortion constraint.

This problem is equivalent to filling a volume $V = 2^{nH(Y)}$ with spheres of volume $v_1 = 2^{nH(Y|X)}$ spaced uniformly. If the spheres are non-overlapping then the maximum number of spheres that can be packed in the space is $N_1 = V/v_1$ spheres. In case of distorted channel capacity, the problem is to find the number of spheres that can be packed in the space without violating the distortion constraint. Let there be at most N_2 spheres each of volume v_1 which can be packed in the volume V.

Let us consider a sphere of volume v_1 corresponding to the codeword σ. Since the N_2 spheres are distributed uniformly over the space V, there will be $n_2 = (N_2 v_1)/V$ center points in this space. Each of the spheres corresponding to these n_2 center points will overlap with the original sphere. We have to ensure that the information bits corresponding to all the n_2 spheres satisfy the distortion constraint.

Since each of the spheres correspond to a k-bit information sequence we must be able to enumerate each of the N_2 spheres by the k bits. Without loss of any generality let us assume that the index of sphere σ is $0, 0 \cdots, 0$. Our distortion constraint restricts that the number of places at which any of the n_2 sphere can differ from σ should not exceed $k * \alpha$. Since all the information bits are binary, and $\sigma = 0, 0 \cdots, 0$, it implies that none of the n_2 information bits can have more than $D = k\alpha$ "ones" in them. The total number of indices which are at a Hamming distance at most D apart from σ is given by $\sum_{i \leq D} \binom{k}{i}$ and to meet our distortion constraint n_2 should be less than this summation.

Unfortunately there is no closed form solution to this sum for any general k and D. We will use some approximations to find a closed form expression. Since α is the

allowable distortion or the bit error rate, we are interested in values of α which are close to zero and much less than 1/2. We note that $\forall i \leq D = k\alpha$, we get

$$\begin{aligned}\binom{k}{i-1}/\binom{k}{i} &= \frac{i}{k-i+1} \\ &\leq \frac{k\alpha}{k-k\alpha+1} \\ &< \frac{\alpha}{1-\alpha}\end{aligned}$$

and similarly $\binom{k}{i-2}/\binom{k}{i} \leq \left(\alpha/(1-\alpha)\right)^2$. Hence for a large k we get that

$$\begin{aligned}\sum_{i\leq D}\binom{k}{i} &< \binom{k}{D}\left(1+\frac{\alpha}{1-\alpha}+\frac{\alpha}{1-\alpha}^2+\cdots\right) \\ &= \binom{k}{D}\frac{1-\alpha}{1-2\alpha} \\ &= \binom{k}{D}O(1).\end{aligned}$$

We only need to estimate is the value of $\binom{k}{\mathcal{D}}$. Using Stirling's approximation ([21]) we get

$$\begin{aligned}\lg\binom{k}{\mathcal{D}} &\approx -\frac{\lg k}{2} - \alpha k \lg\alpha - (1-\alpha)k\lg(1-\alpha) \\ &= -\frac{\lg k}{2} + kH(\alpha)\end{aligned}$$

The total number of spheres (n_2) of volume v_1 which can be packed in a volume V under the Hamming distance restriction should satisfy the following restriction

$$n_2 \leq 2^{-\frac{\lg k}{2}+kH(\alpha)}.$$

From our definition of code rate we also know that $k = nR$. Thus the maximum number of typical sequences allowed to as codewords is

$$\begin{aligned}N_2 &= \frac{n_2 V}{v_1} \\ &\leq \frac{2^{nH(\alpha)}2^{nH(X)}}{2^{nH(X|Y)}} \\ &= 2^{n(I(X;Y)+RH(\alpha))}.\end{aligned}$$

Since all these N_2 codewords are enumerated by k bits we have, $N_2 = 2^{nR}$. From the above two relationship we get

$$nR \leq nI(X;Y) + nRH(\alpha)$$

This gives the maximum achievable code rate or the distorted channel capacity as $I(X;Y)/\left(1 - nH(\alpha)\right)$.

The above argument shows the upper bound of the rate of transmission. In order to complete the capacity argument we need to design a system that can achieve this upper bound. We will consider a system with separate source and channel coding [22]. If we deploy a lossy source coding technique [23] to compress the source till it achieves a distortion of α and then transmit this compressed lossy source across the channel with perfect channel coding then the rate of transmission will be $I((X;Y)/\left(1 - nH(\alpha)\right)$. Thus we can design a system, which can achieve the upper bound described above, and this completes the capacity result. □

2.2.4 Numerical results for a Gaussian channel

To investigate the properties of the above result let us consider a numerical example. We will investigate a single user system transmitting binary symbols over a channel. The binary information bits $\mathbf{b}$ are mapped to binary antipodal coded symbols $\mathbf{d}$ and transmitted through an AWGN channel. The received signal is given by

$$\mathbf{y} = \mathbf{x} + \eta,$$

where $\mathbf{x} = \sqrt{\mathcal{E}}\mathbf{d}$ is the transmitted signals and η is the white Gaussian noise with mean 0 and variance σ^2. Usually for a transmitter the input average power is restricted. In case of a static channel this can be modeled as the constant received

energy $\mathcal{E}$. The received signal is $\pm\sqrt{\mathcal{E}}$ corrupted by the noise. The optimum decoding rule is to decide that $d = 1$ was transmitted when y is positive. In the rest of the chapter unless mentioned otherwise, normal face symbols (like d or y) would represent a single bit variable while the same symbols in the bold face will represent the entire sequence. The probability of error with such a decoding scheme is

$$\begin{aligned} P_e &= \frac{1}{2}\Pr(y < 0|d = +1) + \frac{1}{2}\Pr(y > 0|d = -1) \\ &= \frac{1}{2}\Pr(z < -\sqrt{\mathcal{E}}|d = +1) + \frac{1}{2}\Pr(z > \sqrt{\mathcal{E}}|d = -1) \\ &= \Pr(z > \sqrt{\mathcal{E}}) \\ &= 1 - \Phi(\sqrt{\frac{\mathcal{E}}{\sigma^2}}), \end{aligned} \tag{2.1}$$

where $\Phi(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$ is the error function. Thus we can model a binary

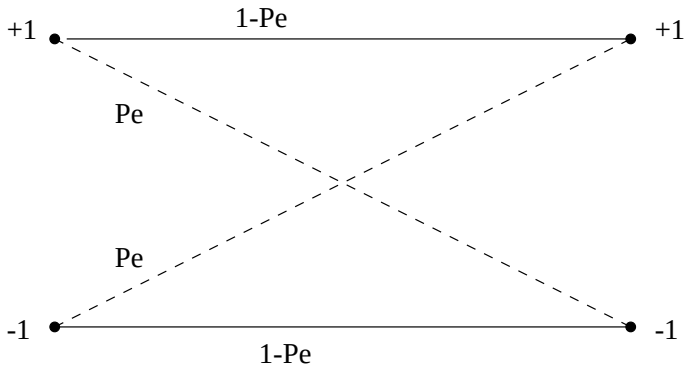


Figure 2.5 : Binary symmetric channel

transmission in an AWGN channel with power constraint by a binary symmetric channel with the crossover probability given by (2.1). The capacity of a binary symmetric channel is given by $1-H(P_e)$. So the maximum rate at which information bits can be transmitted with bit error rate α is given by

$$R = \frac{1 - H(P_e)}{1 - H(\alpha)}$$

If the signal to noise ratio of the system is 1.77dB, then the corresponding Shannon

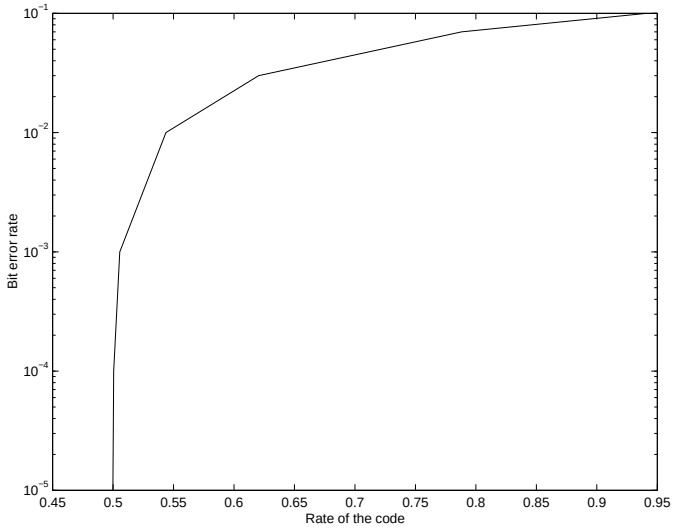


Figure 2.6 : Achievable rates of code and the corresponding performance

capacity of the channel is almost 1/2. In Figure 2.6 we plot the rates of the optimal codes under various acceptable bit error rates at that particular SNR. For a rate below 1/2 the optimal codes can achieve zero error. As predicted before, the bit error rate increases as we increase the code rates. Now that we have derived the expression for distorted channel capacity we are ready to study the "goodness" of the practical codes. As noted before, without knowing the performance of the optimum codes it is not possible to know when the search for good codes should end. If we fix the code rate, we want to study the SNR versus bit-error rate curve for the optimum codes. For most of the practical codes this study is used to compare the relative performance. Previously, Shannon's theorem told us at what signal to noise ratio we can hope to achieve zero error rate. Since no practical code achieves zero error rate, it is not possible to find out at what SNR any practical code would achieve similar performance. As a convention, the SNR corresponding to a small bit error rate

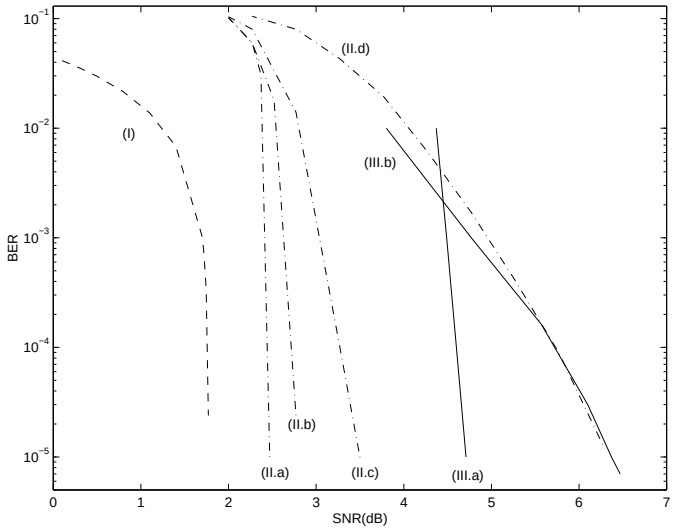


Figure 2.7 : Goodness of the practical rate 1/2 codes (I) Optimal code (II) represents the performance of turbo codes for (a) 18 iteration (b) 6 iteration (c) 3 iteration and (d) 1 iteration and (III) represent the performance of convolutional codes with (a) $\kappa = 41$ and (b) $\kappa = 7$

(10^{-5}) is used for the comparative study. However the distorted channel capacity results will now tell us at what SNR the optimum codes achieves the bit-error rate of 10^{-5}. We will also be able to compare the performance of all practical codes with the optimum code for other values of bit error rate.

In Figure 2.7 we study the performance of turbo codes and convolutional codes against the optimum code. We consider codes of rate 1/2. The performance of turbo codes is plotted after 1,3,6 and 18 iterations. We also study the performance of rate 1/2 convolutional codes with constraint length 7 and 41. We see that performance of the practical codes come close to that of the optimal codes but at the expense of high computational complexity. In the next chapter we will address this issue in greater detail.

2.2.5 Extension to multiuser case

In this section we present some preliminary results for a system with multiple users. A wireless channel is usually shared by a number of users and we would like to find out the maximum rate at which data can be transmitted in such a system when we allow certain amount of error in the recovered bits. Let us first consider

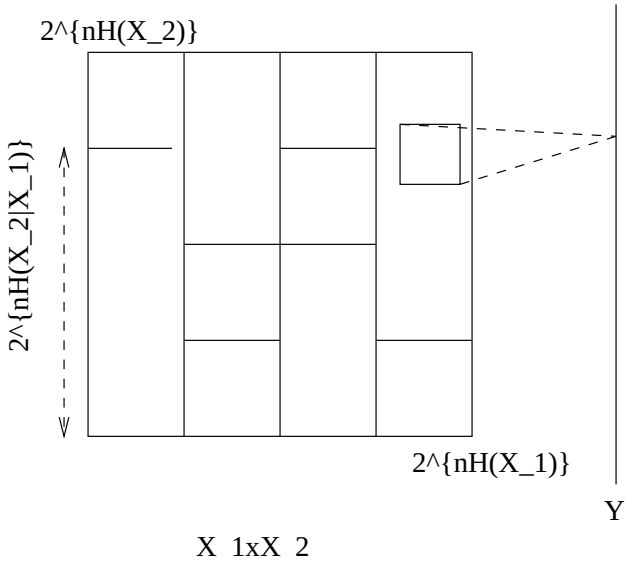


Figure 2.8 : Multiuser case

the two user situation and then extend the results to n-user scenario. There are $2^{nH(X_1)}$ typical sequences for user1 and $2^{nH(X_2)}$ typical sequences for user2. It can be shown that the joint code space consist of only $2^{nH(X_1,X_2)}$ elements. However $H(X_1, X_2) = H(X_1) + H(X_2)$ only when X_1 and X_2 are independent. This implies that if the users are not independent, then for each typical X_1 all the possible $2^{nH(X_2)}$ X_2 are not jointly typical, rather only $2^{nH(X_2|X_1)}$ will be jointly typical (Figure 2.9). In other words if we look at the $\mathcal{X}_1 \times \mathcal{X}_2$ space there will be holes in the allowable space.

We will consider cases where the distortion allowed for user1 is α_1, for user2 is α_2 and the total amount of distortion allowed is α. We considered two relationships

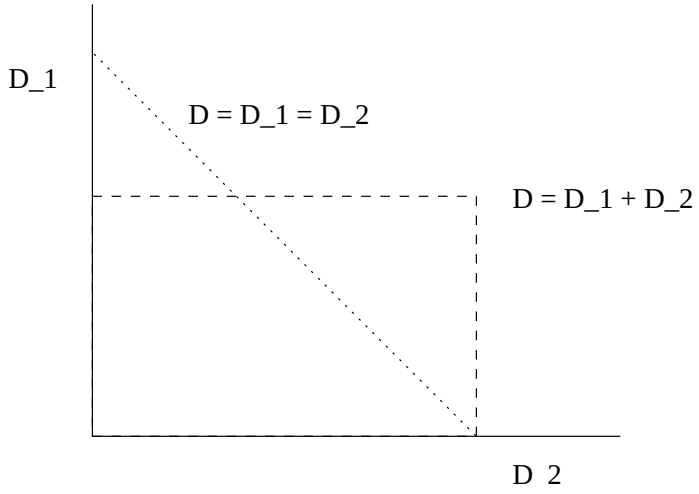


Figure 2.9 : Different types of distortion

among the individual distortions α_i and the total distortion α:

- $\alpha = \alpha_1 + \alpha_2$.

  In this case there is no further restriction on the total distortion allowed. The restrictions are on the distortions of the individual users only and the total allowable distortion is the sum of the maximum allowable distortion for individual user.

- $\alpha = \alpha_1 = \alpha_2$

  This case pertains to an upper bound on the total number of errors only. There is no other bound on the individual distortions as long as the total distortion constraint is met.

Our preliminary results in the two user scenario shows that the capacity of the

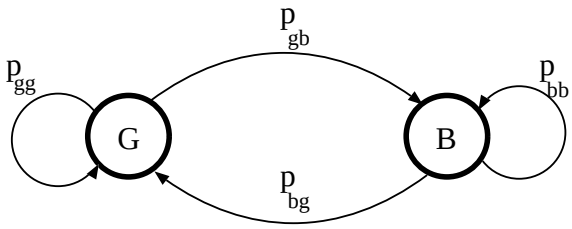


Figure 2.10 : The Gilbert-Elliott model

channel is given by the convex hull of:

$$\begin{aligned} R_1 &\leq I(X_1;Y|X_2)/\big(1-H(\alpha_1)\big) \\ R_2 &\leq I(X_2;Y|X_1)/\big(1-H(\alpha_2)\big) \\ R_1+R_2 &\leq I(X_1,X_2;Y)/\hat{N}\big), \end{aligned}$$

where $\hat{N}$ is either $\prod_{k=1}^{2} 1-H(\alpha_k)$ or $1-2H(\frac{\alpha}{2})$ depending on the relation between α, α_1 and α_2. These results are part of our future research work.

2.3 Achievable rates for fading channel

So far we have considered a static channel for our discussion. However a static channel is a rather simplistic model of a practical wireless system. Due to the relative motion of the source and the destination and the reflecting bodies in the path, the wireless channel exhibits fading characteristic. This implies that the attenuation of the source energy and the background noise level in the system varies over time. There are various statistical models that have been used to characterize the dynamic wireless system. We use the two-state Gilbert-Elliot model for the fading channel.

2.3.1 Channel model

The wireless fading channel can be modeled as a finite state Markov process [24]. A popular model is the two-state Gilbert-Elliott model, as shown in Figure 2.10. In each state, the channel is described by an underlying AWGN background noise with standard deviations denoted by σ_g for the good state and σ_b for the bad state. As before we can model each of these Gaussian noise channels as the binary symmetric channels and let the Shannon capacities associated with these channels be given by C_g and C_b respectively.

The Markov model is specified by transition probabilities P_{gb} and P_{bg}. Let us assume that in the steady state the system is in the good state with probability p_g and in the bad state with probability $p_b = 1 - p_g$. A simple encoder and decoder design is to design for average channel capacity $C = p_g C_g + p_b C_b$. This is mandatory if the receiver and the transmitter are not aware of the underlying changes in the channel. The maximum allowable distortion α will determine the maximum rate R at which the information bits can be transmitted and is given by

$$R = \frac{C}{1 - H(\alpha)}.$$

However if the transmitter and the receiver is aware of the underlying channel then a second option of transmission would be to transmit at a rate R_g during the good state and R_b during the bad state incurring distortion α_g and α_b respectively. We want to maximize the average rate of transfer of data $R_{av} = p_g R_g + p_b R_b$ under the average distortion constraint ($\alpha = p_g \alpha_g + p_b \alpha_b$). For a given C_g and C_b, the solution lies at the equilibrium of the Lagrangian:

$$J = \frac{p_g C_g}{1 - H(\alpha_g)} + \frac{p_b C_b}{1 - H(\alpha_b)} - \lambda(p_g \alpha_g + p_b \alpha_b - \alpha).$$

Taking the derivatives with respect to α_g and α_b and equating them to zero we get

$$\frac{C_g \log\left(\frac{\alpha_g}{1-\alpha_g}\right)}{\left(1-H(\alpha_g)\right)^2} = \lambda = \frac{C_b \log\left(\frac{\alpha_b}{1-\alpha_b}\right)}{\left(1-H(\alpha_b)\right)^2}.$$

Together with the average distortion constraint $\alpha_g + \alpha_b = \alpha$ this determines the maximum rate at which information can be transferred across the fading channel. It is interesting to note that for $\alpha_g = \alpha_b$ the average distortion constraint is satisfied and

$$\begin{aligned} R_{av} &= p_g R_g + p_b R_b \\ &= \frac{p_g C_g}{1-H(\alpha_g)} + \frac{p_b C_b}{1-H(\alpha_b)} \\ &= \frac{p_g C_g + p_g C_b}{1-H(\alpha)} \\ &= R \end{aligned}$$

Thus the maximum rate at which information can be transferred is never going to be less than the case when we transmit based on average channel characteristic.

2.4 Summary

In this chapter we have proposed a new capacity measure for AWGN channel. The traditional Shannon capacity deals with exact recovery of the information bits by the receiver. However most of the practical systems can tolerate variable amount of errors in the recovered information sequence. We have proposed a new expression of such a system, which can tolerate variable amounts of error. This new measure also enables us to compare the performance of practical codes at various levels of errors. Specifically we can study how closely these practical codes can achieve the performance of optimal codes in a Gaussian channel. We have also studied a fading channel approximated by a multi-state Gilbert-Elliott model. If the average bit error

rate is fixed, we study the rates at which information bits should be transmitted at each state to maximize the overall throughput of the system.

Chapter 3

Maximal weight basis decoding of convolutional codes

With the emergence of mobile multimedia applications, the design of reliable broadband wireless networks has become mandatory. Unfortunately, the presence of interfering users, the multipath fading characteristic of the channel and the background thermal noise can adversely affect the transmission of information. Forward error-correction codes [25] can allow system designers to improve the reliability of the channel. In the previous chapter we introduced the concept of distorted channel capacity as a benchmark to study the "goodness" of the practical codes and studied the behavior of some popularly used coding schemes.

Convolutional code is one of the most popular error-control schemes deployed by the wireless system designers. The popularity of this code is not only because of its effectiveness in recovering from transmission error, but also because of the availability of a computationally efficient optimal decoding algorithm proposed by Viterbi [26]. Viterbi's algorithm is the maximum likelihood decoding algorithm, i.e., it minimizes the probability of decoding a sequence of received symbols to an incorrect codeword. Unlike the decoding algorithms for many block codes, the complexity of the Viterbi algorithm grows only linearly with the block length of the code. This makes it an attractive option for physical implementation.

The rate of the code, which measures the amount of redundancy added to the raw transmission bits, is usually a good indicator of the strength of the error-correcting

codes. Unlike a block code, the strength of the convolutional codes is also determined by the *constraint length* of the code, which is the number of past information bits that affect the current coded bit. The rate of generation of source bits and the available bandwidth usually upper bounds the rate of the code. From Figure 2.7 it is apparent that the performance of the convolutional code approaches the optimal bounds for large constraint length codes. Hence it is desirable to have a large constraint length convolutional code to improve the reliability of the system. However, the complexity of Viterbi algorithm grows exponentially with the constraint length. This precludes large constraint length convolutional codes from being practically implemented, especially in a system with real time demands or limited processing power. In this chapter, we investigate methods to reduce the complexity of decoding large constraint length convolutional codes to be able to bridge the gap of optimal bound and practically implemented codes.

3.1 Convolutional codes

While Shannon's work laid the foundation of information theory, it is Richard Hamming who is often credited with the invention of first *error-correcting codes* [27] and thereby the initiation of *coding theory*. Hamming's work was followed by the discovery of several other schemes by Golay [28], Reed-Muller [29, 30], Reed-Solomon [31], Bose-Chaudhari-Hocquenghem [32, 33] and others who led the development of *block codes*. Even though block codes have enjoyed tremendous success there are several fundamental drawbacks to their use in practical systems. Due to their frame oriented nature, the entire codeword must be received before the decoding can be completed and similar restriction applies for the encoding process also. Secondly, most of the algebraic based decoders for block codes work on the output of hard bit

decisions as opposed to soft decisions, while Shannon's results favor codes that can assume continuous valued channel outputs. So for the low SNR channels, the performance of the algebraic linear block codes can sometimes be rather poor, because of its associated hard-decision decoding.

However these drawbacks can be easily overcome by considering a *convolutional code*. Convolutional code is a non-block code invented by Peter Elias in 1955 [34]. A convolutional code is characterized by three parameters:

- The *alphabet* over which the code is defined. Generally a convolutional code is defined on a finite field. We will restrict our discussion to convolutional codes defined on binary alphabets $\{0, 1\}$.
- The *rate* of the convolutional code is a measure of the redundancy introduced in the information stream for recovery from error. A rate $R = (n, k)$ code implies that for every k information bits n coded bits are transmitted. It should be remembered that any (n, k) linear code over a field F is usually a k-dimensional subspace of the n-dimensional space F^n.
- The above two parameters are necessary for any error correcting codes. However what sets convolutional code apart from the block codes is the *memory* or *constraint length* (κ) of the code. In case of a block code, the n-bit block of a coded message depends on only the current k-bit block of the information bits. However in case of a convolutional code the current coded bits depend not only on the present k bits but also on the k-bit sequences from the past κ time.

A convolutional code is usually implemented with a set of shift registers or memory elements (Figure 3.1). In our example the bits $d(i)$ are dependent on the information

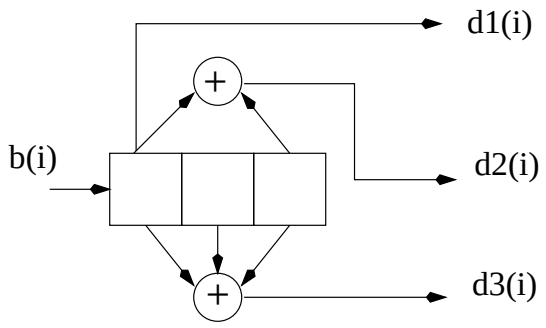


Figure 3.1 : Example of a (3,1) convolutional code with constraint length 2

bits $b(i), b(i-1)$ and $b(i-2)$. A convolutional code is said to be *systematic* if the first k bits of the n coded bits are exact replicas of the information bits. The rest of the $n-k$ bits are called the *parity* bits.

It is evident that two convolutional codes of the same rate and constraint length differ by the functions used to determine the parity bits. Even though there are multiple ways of describing the convolutional codes, we will only describe the *trellis diagram* method for our example. A convolutional code can be described as a *finite state machine*. The current state and the input uniquely determine the next state and the output of the machine. The current state is given by the content of the inputs that are present in the shift registers. The trellis diagram is just an enumeration of the outputs of this finite-state machine.

We assume that the initial state (also called state zero) has all zeroes in the registers. The memory length of the system determines the number of states. In our example of (3,1) convolutional code with $\kappa = 2$, there are 4 possible states. The solid transition is due to input '0' and the dashed line shows transition due to input '1'. The numbers along the transition describe the output of the decoder due to that

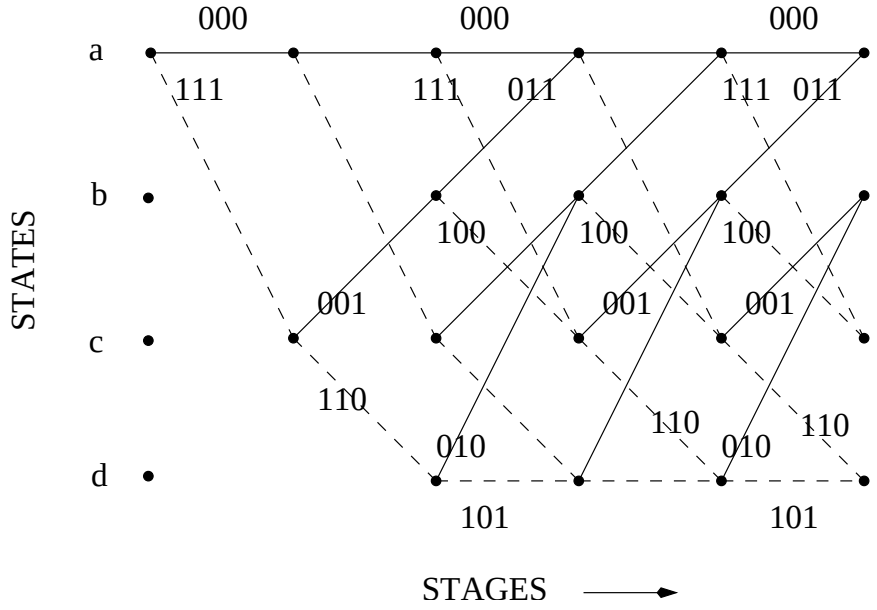


Figure 3.2 : Trellis diagram of the convolutional code

transition. This trellis diagram extends till the input stream is exhausted and the output is the corresponding convolutionally encoded bit-stream.

3.2 System description and Viterbi decoding

We consider a system where raw information bits, $\mathbf{b}$, are encoded by a convolutional encoder. Usually for transmission over a channel the binary bits are BPSK modulated. Thus the convolutionally encoded $\{0, 1\}$ bit sequences are mapped to antipodal symbols ± 1. The received signal consists of the original symbols corrupted by noise and is given by

$$\mathbf{r} = \sqrt{\mathcal{E}}\mathbf{d} + \eta, \tag{3.1}$$

where $\mathcal{E}$ is the received energy of the signal, $\mathbf{d}$ is the coded bit sequence and η is the noise. We assume that the noise is additive white Gaussian in nature with zero mean and variance σ^2. We next consider how one can optimally estimate the information

bits from the received symbols.

Wozencraft and Reiffen [35] proposed the first practical decoding algorithm for the convolutional codes. In 1967 Viterbi [26] proposed his seminal decoding technique. It was later shown by Omura [36] that the Viterbi algorithm was a dynamic programming solution to finding the shortest path through a weighted graph. Forney [37] established that Viterbi's decoding technique is the maximum likelihood decoding method and can be equally adapted for estimating transmitted sequence in a system that introduces inter-symbol interference.

The maximum likelihood decoding technique is also optimal in the sense that it minimizes the probability of error and is given by

$$\hat{\mathbf{b}} = \arg\max_{\mathbf{b}} p(\mathbf{b}|\mathbf{r}).$$

Since there is a one-to-one correspondence between the information bits and the codewords, this maximum likelihood algorithm computes the codeword that has the maximum conditional probability given the received signal. For an additive white Gaussian noise channel this optimization criteria is equivalent to finding a codeword sequence that has minimum *Euclidean distance* from the received signal. Essentially we should explore the code trellis to generate all the possible codeword sequences and enumerate the associated likelihood of transmission of that codeword when the received sequence is $\mathbf{r}$. If the block length of the code is N, and the code rate is R, there are 2^{NR} possible codewords that should be considered.

Viterbi however showed that it is indeed possible to describe an optimal decoder whose complexity is linear in the codeword length. For his algorithm, Viterbi associated *metrics* to *branches* and *nodes* of the trellis. The metric associated with a particular branch at a particular stage or level i, is the probability of receiving r_i, when the output corresponding to that branch is transmitted. A *path* is defined by a

sequence of branches at consecutive levels so that the terminal node of a branch ends in the source node of the next branch. The metric associated with a path is the sum of the metrics associated with the branches in the path. The metric associated with any node is the minimum metric associated with any path starting from the start node to that node. It is evident that the maximum likelihood codeword corresponds to the path that has the lowest metric from the start node to the final node. If the start level is termed 0, the end level is termed N and we know both the start and the terminal states are 0, then for $0 < l < N$, the defining equation in the optimization problem is

$$metric(0, N) = \min_{m \in states} \left(metric(0, l_m) + metric(l_m, N) \right),$$

where $metric(i, j)$ is the minimum metric of any path originating from node i and ending in node j and l_m represent the m^{th} node in level l. Once we know the metric associated with all the nodes in level l, the metric associate with the m^{th} node in level $l + 1$ can be calculated by

$$metric\big(0, (l+1)_m\big) = \min_{i \in states} \big(metric(0, l_i) + metric(l_i, (l+1)_m \big).$$

If there is no branch between the node i in state l and node m in state $l + 1$ then the metric associated with that branch is assumed to be infinitely large (∞).

This iterative method of calculating the optimal code reduces the complexity of the decoder to be linear in the code-word length. However at every stage of the trellis, Viterbi algorithm has to calculate the likelihood of each state. The number of states is exponential in the size of the constraint length of the code. Thus Viterbi algorithm for a convolutional code of constraint length κ and codeword length of N requires $N2^{\kappa+1}$ operations indicating that the complexity becomes prohibitively large for large constraint lengths. As mentioned in the previous section, larger constraint

length convolutional codes lead to better protection against channel errors. Hence for real time implementations, we need decoding algorithms that have more tractable computational complexity.

3.2.1 Alternative decoding techniques

Apart from the Viterbi algorithm, several other decoding algorithms have been proposed over the years for the convolutional codes. In this section we will briefly describe a few of them for the sake of completeness.

Sequence decoding

This decoding technique was first proposed by Wozencraft [35]. The major steps of the algorithm are:

1. *Initialization*: Load the start node in the stack and initialize its metric to be zero.

2. Compute the metrics of all the successor branches coming out of the node.

3. Delete the top node and compute the new path metrics, (metric of the node and the metrics of all the outgoing branches). Insert the new metrics in appropriate places in the stack maintaining an ascending order in the stack.

4. If the path at the top of the stack is the terminal node, then the procedure terminates or else go back to step 2.

The original sequential decoding technique has been modified by Fano [38] and Jelinek [39]. An alternative stack based approach to the sequential algorithm for efficient implementation was originally proposed by Zigangirov and Jelinek [40, 41].

Even though the performances of these various implementations of the sequential decoding algorithm are similar to that of Viterbi Algorithm, practical considerations can often affect the performance. A general problem with sequential decoding based methods is that the decoding complexity is non-deterministic, making the decoding technique unacceptable for real-time applications. Furthermore, specific practical limitations exist with different implementations. In the stack based methods, depending on the data arrival rate and the decoding rate, the buffer that stores the incoming blocks can often get filled up. This will lead to *erasure* of bits and often limits the performance of the decoder. The problem is especially acute in case of low signal to noise ratio (SNR) values. Several suboptimal approaches to alleviate this problem have been proposed, such as dynamic programming techniques [42] or pruning of the search space based on dynamically computed thresholds [43]. Moreover even if the data rate and decoding rate are compatible, there is always a possibility of stack-overflow since the buffer size to store the incompletely explored paths is finite. The solution is to make the buffer much larger than the block length. Alternatively, multiple stacks [44, 45], each with different priorities can be used. However, multiple stack based methods require reordering of stack elements when elements are moved across the stacks and this involves complex operations. Furthermore, a large memory space is necessary for implementation. Coupled with the requirement that the size of the stacks should be several times the size of the block length of the code, the possibility of practical implementation of stack based algorithm is limited. Fano's algorithm [38], on the other hand, requires a much smaller memory at the expense of being slower than the stack based implementations.

Majority logic decoding

This technique is proposed by Massey [46]. We can also describe a convolutional code by a *generator matrix* G ($\mathbf{d} = G^{\top}\mathbf{b}$). The *parity check* matrix, H, of a convolutional code is defined as one that satisfies $HG^{\top} = 0$. For a binary systematic code the *parity check* matrix H can be easily found. In majority logic decoding the *syndrome*, $\mathbf{s} = H\mathbf{r}$, is calculated for the received signal $\mathbf{r}$. If the received signal is a code word the syndrome is zero. This syndrome is used to form a set of orthogonal check-sums for each error bit. An error bit is supposed to be "true" if at least half of the orthogonal checksums corresponding to it is equal to 1. From these error bits and the received signal the original codeword is reconstructed. In this approach, irrespective of the length of the received sequence, the decoding window is based only on the constraint length of the code. The computational complexity is much lower than the Viterbi algorithm for codes with moderate constraint lengths. However the performance of majority logic decoding is suboptimal as decision is based only on one constraint length and not the entire sequence. Moreover the original majority logic decoding algorithm can only accept hard value inputs. Extensions that incorporate soft inputs have been developed [47] but at considerable increase in computational complexity. More recently, table look-up based decoding [48] of convolutional codes that reduces the decoding time at the expense of added memory requirements has also been investigated. However the performance of these techniques for large constraint length codes are much worse than the Viterbi algorithm.

Among all the available decoding techniques, the Viterbi algorithm is the most popular because of its superior bit error rate performance. Moreover the complexity of algorithm is predictable and the regular *butterfly* structure of the Viterbi decoder lends itself to parallel implementation. However its complexity grows exponentially

with the constraint length and as such large constraint length convolutional codes are practically not feasible.

In the subsequent sections we propose a new decoding algorithm for convolutional codes based on the *maximal weight basis* of the code. The decoding principle is similar to the generalized Dijkstra's algorithm [**?**] used for finding the minimum spanning tree of a weighted graph. In any error correcting code of rate R, for every sequence of N coded bits at most NR bits can assume unconstrained values and these NR bits uniquely determine the values of the remaining $N(1-R)$ coded bits. An optimal decoding rule searches for the "best" choice of these NR unconstrained bits. The MWB decoding algorithm approximates this search by identifying a "good" (not necessarily the best) set of NR independent bits, also called the maximal weight basis, and then generates the entire code-word from this maximal weight basis in a computationally efficient manner. The choice of the maximal weight basis is based on likelihoods of individual bits. Similar techniques for block-codes have been investigated in [49, 50]. However, a systematic method to compute the ordered statistics for convolutional codes is not provided. For improved performance, the MWB decoding technique is augmented by incorporating the list decoding [51, 52, 53] principles.

The complexity of MWB algorithm only grows quadratically with the constraint length, as opposed to the exponential complexity for the Viterbi algorithm . This reduction in complexity is achieved without significant performance loss. Furthermore, some disadvantages of other low complexity decoding techniques are avoided. Unlike the sequential algorithms, the amount of computation is deterministic making it particularly suitable for real-time applications. The storage space required does not grow significantly with the block length of the code. Elaborate storage, as required in the table look-up techniques or in the stack based sequential decoding,

is not necessary. Unlike majority logic decoding, the MWB algorithm can work on soft inputs and therefore is expected to result in better performance. An added advantage of MWB based decoding is that it is not restricted to BPSK modulation schemes, or convolutional codes. The MWB decoding idea can easily be adapted to work with M-ary symbols and decode most of the linear block-codes. Even though most of our discussions in this chapter will be restricted to hard decision decoding of convolutional codes, we will briefly describe how to extend the ideas to soft-decision decoding and turbo codes.

3.3 Motivation of the maximal weight basis

Let us consider an example of a simple rate 2/3 even-parity-check code. For every two bits of information, three coded bits are produced where the first two bits are the original information bits and the third bit is the modulo-2 sum of the two information bits. Suppose the information bits are $\mathbf{b} = (1, 0)$. Then the transmitted codeword is $\mathbf{d} = (1, -1, 1)$. Suppose the noisy version of the received signal determined according to (3.1) is $\mathbf{r} = (7.5, 0.5, 7.5)$.

If we use the maximal likelihood sequence estimation (MLSE) method to estimate the information bits, the decoder would correctly recognize the transmitted bits as $(1, -1, 1)$ in this particular example and from there extract the original information bits as $(1, 0)$. However, we will have to consider all the possible 2^2 codewords, compute their respective likelihood values and select the one that has the largest likelihood. In general for a large block-length and even moderate constraint length, MLSE is a computationally expensive decoding technique.

On the other hand, if the decoder ignores the fact that the transmitted bit sequence is a codeword and assumes that the received signal results from an un-

constrained binary sequence corrupted by noise, then the estimate is given by the unconstrained bit sequence that has the least Euclidean distance from the received signal. This is simply equivalent to the sign estimation of the received sequence $\hat{\mathbf{d}} = sgn(\mathbf{r})$ and the decoding has trivial computational complexity. Obviously in this method of decoding it is not always guaranteed that the estimated bit sequence will be a codeword. To summarize, the decoding complexity becomes non-trivial when we have to restrict our search to codewords only, while reduction in decoding complexity can be achieved if we can make our search unconstrained. We want to achieve a tradeoff between these two extreme options.

It should be noted that in the maximum likelihood sequence estimation not only do we estimate the codeword that is most likely, but also compute the likelihood value corresponding to that codeword. We define the likelihood corresponding to the j^{th} bit of the codeword $\hat{d}_j$ as

$$\phi_j = \frac{\Pr(r_j|d_j = \hat{d}_j)}{\Pr(r_j|d_j = -\hat{d}_j) + \Pr(r_j|d_j = \hat{d}_j)},$$

where we assume that $d_j = \pm 1$ can occur with the same probability. Then the likelihood corresponding to the estimated codeword $\hat{\mathbf{d}}$ becomes

$$\phi(\hat{\mathbf{d}}) = \prod_{j=1}^{N} \phi_j. \tag{3.2}$$

The MLSE algorithm identifies the codeword for which this likelihood value is maximum. Additionally, the numerical value of the likelihood function measures our confidence in the estimated codeword based on the received signal. The larger the likelihood value corresponding to a given codeword, the greater are the chances that it was actually transmitted. We can rewrite our original decoding problem as

$$\hat{\mathbf{d}} = \arg\max_{\mathbf{d}\in\mathcal{C}} \phi(\mathbf{d}), \tag{3.3}$$

where $\mathcal{C}$ represents the codebook.

For simplicity of discussion, let us consider a systematic code. The systematic codeword has two parts - the *information bits*, that can take any arbitrary binary values, and the *parity bits*, which are uniquely determined by the information bits. Let I represents the locations corresponding to the independent information bits and D represents the locations corresponding to the parity bits that depend on the information bits. We can partition the codeword $\mathbf{d}$ as $\mathbf{d}_I$, representing the information bits, and $\mathbf{d}_D$, the parity bits. We can rewrite (3.3) as

$$\begin{aligned}\hat{\mathbf{d}} &= \arg \max_{\{\mathbf{d}_I, \mathbf{d}_D\} \in \mathcal{C}} \prod_{j=1}^{N} \phi_j \\ &= \arg \max_{\{\mathbf{d}_I, \mathbf{d}_D\} \in \mathcal{C}} \prod_{i \in I} \phi_i \prod_{j \in D} \phi_j\end{aligned}$$

At this point we would like to distinguish between two types of likelihood values we will consider. The likelihood function defined in (3.2) will be referred to as the *total likelihood* of the codeword, while the product corresponding to a few bits in the codeword will be referred to as *partial likelihood* of the codeword. The positions of the bits will be obvious from the context. For example in our systematic codes, $\prod_{i \in I} \phi_i$ is the partial likelihood of the codeword corresponding to the information bits.

For optimal decoding, we need to maximize the total likelihood of the codeword, but as we have observed this constrained optimization is computationally expensive, while a completely unrestricted optimization will not guarantee a valid codeword. In order to guarantee a valid codeword, we can approximate the optimization problem in (3.3) by maximizing the partial likelihood corresponding to the information bits. Recall that the information bits can assume any unconstrained binary sequence. Referring to the earlier example, where the received signal is $(7.5, 0.5, 7.5)$, we need

to maximize the likelihood corresponding to the first two bits and the decoded bits are $(1, 1)$. Of course, as in this example, this approximation technique may not always give the correct values as we have ignored the likelihoods $\prod_{j \in D} \phi_j$ corresponding to the parity bits. We can obtain further information about our decision if we calculate the total likelihood of the codeword corresponding to the information bit sequence $(1, 1)$. We see that the total likelihood value is close to zero, which usually indicates an erroneous decision. We now have an approximate decoding algorithm, which is computationally simple and an associated metric, which can notify us when we arrive at wrong decisions.

It is not enough to just know when we arrive at a wrong decoded codeword. We will make two adjustments to rectify this situation. When we consider only the partial likelihood, we are trying to optimize the whole problem by only looking at the partial problem. The basis of this assumption is that the codeword with total largest likelihood will also have the largest partial likelihood corresponding to the information bits. This may not be entirely true, but it is quite likely that the optimal codeword will have a "large" partial likelihood value corresponding to the information bits. So if we consider all the codewords that have "large" partial likelihoods, it is quite likely that the optimal codeword will be in that set. We will then calculate the total likelihoods corresponding to this set of codewords only and select the one that has the maximum total likelihood. Again going back to the example, the information bit sequence with the second largest partial likelihood is $(1, 0)$. In fact, this corresponds to the correct decision. Thus, instead of looking at only the most likely information sequence we will consider several information bit sequences that have large partial likelihoods and create a *feasible set*. The final decision will be the codeword sequence from this feasible set that has the largest

total likelihood.

We can further refine this idea as follows. If we carefully consider the codeword space of the 2/3 parity check code, we observe that not only the information bits, but any two out of the three code bits can assume unrestricted values. These two *independent bits* uniquely determine the third bit. We also observe that our confidence in the decision is not the same for all the bits. In fact, we have a greater confidence in making correct decisions about the first and the third bits than the second bit. So if we select our feasible sets based on the received bits for which the likelihood values are large, we have a larger probability of including the optimal codeword in our feasible set. The estimated unconstrained values corresponding to the first and the third codeword positions are $(1, 1)$, which corresponds to the correct codeword $(1, 0, 1)$.

Generalizing the above idea to a code of rate R and block length N, we can say that there are NR independent locations. In other words at most NR of N bits can be unconstrained. To borrow a term from linear algebra, these NR bits act as a basis for the codeword space. However, which set of NR values can be independent, depends on the type of the code. We would like to make decisions on those NR bits, about which we are most confident, i.e., those NR bits with largest likelihoods. This NR bit long codeword with partial likelihoods will later be defined as the maximal weight basis of the codeword. We want to generate M possible *partial codewords* that have large partial likelihoods and from them select the one with largest total likelihood.

Our sub-optimum decoding rule is based on using the sign detector for only NR independent bits of the N coded bits. The remaining $N - NR$ bits of the codeword can be obtained through the structure of the codebook, $\mathcal{C}$. The choice of these

NR independent bits will be based on the *weight* of the location $i \in \{1, \ldots, N\}$ denoted by w_i which is equal to $|r_i|$, the absolute value of the received vector at that location. Since large weight corresponds to lower probability of error in the sign detector, the estimates for these NR bits will be reliable. Effectively, we want to select an independent set of bits that has the largest reliability. However, the problem is more challenging in the case of a convolutional code since any set of NR bits are not independent. It is essential to find not only the independent bases in a computationally efficient manner but also the basis with the largest weight. In the rest of this chapter we will formulate this suboptimal algorithm.

3.4 Maximal weight basis decoding of convolutional codes

In this section we illustrate how the discussion in Section 3.3 can be applied to decode convolutional codes and provide the formal algorithm in terms of a convolutional code $\mathcal{C}$ of rate R and constraint length κ. Even though almost all of our results hold equally well for all types of convolutional codes, we will develop our theory mostly for systematic codes.

3.4.1 Preliminaries

Let $\mathbf{d} = (d_1, \cdots, d_N)$ represent a block of coded bits of length N, and $\mathbf{r} = (r_1, \cdots, r_N)$ the corresponding received signal. We want to devise an algorithm that finds the codeword, which has the least probability of decoding error. To reduce the complexity of decoding we will make decisions about NR independent bits in an unconstrained manner and then reconstruct the codeword from these NR bits. We would also like to make decisions about those NR bits which can be estimated with largest reliability out of the length N received vector. Simply selecting the largest valued

NR locations will not work, as all subsets of NR bits are not independent. Let $\mathcal{N} = \{1, \cdots, N\}$ represent the set of locations of the coded bits.

Definition 3.4.1 *A set $I \subset \mathcal{N}$ is defined to be an* independent subset *of $\mathcal{N}$ if the bits corresponding to locations in I can be chosen independently without violating any of the conditions for being part of a codeword in $\mathcal{C}$.*

Definition 3.4.2 *A set J is said to be* maximally independent*, if it is an independent subset of $\mathcal{N}$ and there does not exist any element e, which is in $\mathcal{N}$ but not in J such that $J + e$ is also an independent subset.*

In our subsequent discussion we will use I, J, $\mathcal{N}$ to represent not only the set of locations but also the coded bits corresponding to those locations when there is no ambiguity. Also as set operators '+' will denote the set union and '-' the set difference operation.

In case of a convolutional code, the maximally independent subset of coded bits represent those bits which can assume unconstrained binary values and can be determined independently without violating the properties of convolutional code. Additionally, given the values of the bits in the maximally independent subset, the entire codeword can be *uniquely* determined. The bits in the maximally independent subset of a convolutional code form a *basis* for the codeword space. Obviously, for a systematic codeword, the set of information bits forms a basis for the codewords. In fact a convolutional code is described by a set of relations that define the parity bits in terms of the information bits.

In order to illustrate the discussions in the following sections, let us consider a numerical example of a systematic convolutional code of rate $R = 1/2$, constraint

length $\kappa = 3$ with the parity bits being defined by the following set of equations,

$$\begin{bmatrix} d_2 \\ d_4 \\ d_6 \\ d_8 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 0 & 0 & \cdots & 0 \\ 1 & 1 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 1 & 1 & \cdots & 0 \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} d_1 \\ d_3 \\ d_5 \\ d_7 \\ \vdots \\ d_{N-1} \end{bmatrix} \tag{3.4}$$

where the bits $d_1, d_3, \cdots, d_{N-1}$ are the information bits and the system is described in $GF(2)$. The vector $\mathbf{d} = (d_1, d_2, \cdots, d_N)$ represents a codeword.

In this example the information bits form a basis of the codeword space. In fact the systematic convolutional encoder is a function to generate the entire codeword from the information bits. We claim that the basis for a convolutional code is not unique. We will propose an algorithm to calculate the *maximal weight basis* and we will show it is easy to regenerate the codewords from this basis, given the original representation of the convolutional code in terms of the information bits. But instead of only one maximal weight basis we will create a feasible set of codewords with M largest partial likelihoods and from it identify the codeword with the largest total likelihood. We will first provide a brief outline of our decoding algorithm. The details of these steps, their derivation, proof of correctness and the computational complexity will be explored in detail in subsequent sections.

Outline of Maximal weight basis decoding algorithm

- After receiving all the N symbols, the codeword corresponding to the maximal weight basis is computed using *Algorithm* II of Section 3.4.2.

- In order to find the feasible set of M largest partial likelihood codewords,

rather than recomputing from scratch, we make use of the maximal weight basis. *Lemma* 4 and *Theorem* 2 of Section 3.4.4 illustrate that the m^{th} largest basis differs from the maximal weight basis in only a few locations. *Theorem* 3 and 4 of Section 3.4.5 describe how to identify these locations.

- Once these M largest weight bases are identified, the total likelihoods corresponding to each of the codewords are calculated. The codeword with the largest total likelihood is the output of the MWB algorithm.

In order to describe our algorithm, we will need to formally define the concepts of "weight" and "maximal weight basis". But before that, we will prove a few properties of the independent subset and maximally independent subset of a code, concepts that we have introduced in the first part of this section. Though many of these results are true for any linear codes we will study them particularly in the context of convolutional codes.

Lemma 2 *Any independent subset satisfies the following properties.*

1. *If $I' \subset I$, and I is an independent subset of $\mathcal{N}$, then so is I'.*
2. *For a convolutionally coded system of block length N and rate R there are at most NR locations whose values can be chosen independently. In other words the* rank *of a convolutional code of block-length N and rate R is NR.*
3. *The cardinalities of all maximally independent subsets are equal.*

Proof: : The first property is true from the definition of the independent subset. We know that convolutional code is a *linear code*, i.e., it is defined by a linear system of equations. The number of independent locations in a convolutional code is given

by the rank of the linear system. For a systematic convolutional code there are $N(1-R)$ equations involving N variables and rank of the system is NR. Thus at most NR locations can be chosen independently. In fact the third property says that exactly NR locations can be chosen independently, which is true because any basis of a linear system will have the same number of elements. □

3.4.2 Maximal weight basis

Definition 3.4.3 *We define the* weight *of a particular location i, $wt(i)$, to be a measure of reliability of making decision on that particular bit location. Quantitatively we define, $wt(i) =\mid r_i \mid$ and it is non-negative. The weight of a set I is given the sum of the weights of each individual member of the set $wt(I) = \sum_{i \in I} wt(i)$.*

It should be remembered that this definition of weight is applicable for binary antipodal symbols. We will briefly discuss how to change our algorithm to handle the M-ary symbol in our concluding section. When the transmitted bits are corrupted by additive white Gaussian noise, the likelihood, ϕ_j, is a monotonically increasing function of the weight $wt(i)$. We want to find an independent subset of bit positions whose partial likelihood is the largest among all possible independent subsets.

Definition 3.4.4 *The* maximal weight basis *(MWB), denoted by $J^\star$, of $\mathcal{N}$ with respect to the received signal $\mathbf{r}$ is an independent subset whose weight is maximum.*

Since the partial likelihood is given by $\prod_i \phi_i$ and log is a monotonic function the maximal weight basis is,

$$J^\star = \arg\max_I \prod_{i \in I} \phi_i = \arg\max_I \sum_{i \in I} wt(i),$$

where I is an independent subset of $\mathcal{N}$. The maximal weight basis gives us an independent subset of bits whose partial likelihood is largest among all the independent

subsets. It is clear that since all the weights $wt(i) \geq 0$, our chosen set must be a maximally independent subset; otherwise we could add elements to it without violating the independence condition and thereby form another independent subset whose weight is larger than that of our set. We now present an algorithm to find the MWB $J^\star$ of $\mathcal{N}$ with respect to $\mathbf{r}$.

Algorithm I: Maximally independent subset

. Set $I = \emptyset$

. Sort the weights $wt(i)$ of elements in $\mathcal{N}$ based on the received vector $\mathbf{r}$.

. While $|I| < NR$

- Select the location $e \notin I$ from $\mathcal{N}$ with the largest weight such that $I + e$ is still an independent subset of $\mathcal{N}$
- $I = I + e$

This is essentially a greedy algorithm. We keep on adding elements to our independent subset in decreasing order of their weight unless they violate the independence relation. Since the block-length is finite the algorithm terminates after a finite number of steps. The following lemma shows that the greedy algorithm actually finds the maximal weight basis.

Theorem 1 *The above greedy algorithm identifies the MWB of* $\mathcal{N}$.

Proof: Suppose the theorem is false. Let $J_1 = \{e_1, e_2, \cdots, e_k\}$ be the maximal subset given by the above algorithm and let $J = \{q_1, q_2, \cdots, q_k\}$ be a maximally independent subset with total weight larger than J_1. Since J_1 and J are both maximally independent subsets of $\mathcal{N}$ their cardinalities are the same. Let this cardinality be k.

Without loss of generality, we assume $wt(e_1) \geq wt(e_2) \geq \cdots \geq wt(e_k)$ and $wt(q_1) \geq wt(q_2) \geq \cdots wt(q_k)$. Of course there might be some elements common in J_1 and J. Select the least index m such that $wt(q_m) > wt(e_m)$. Thus the element added to the set I at the m^{th} step of the algorithm is not one of $q_1, q_2, \cdots, q_m$ and for $1 \leq i \leq m$, either $q_i \in \{e_1, \cdots, e_{m-1}\}$ or $\{e_1, \cdots, e_{m-1}, q_i\}$ is not an independent subset of $\mathcal{N}$.

In other words the set $\{e_1, \cdots, e_{m-1}\}$ of cardinality $(m-1)$ is a maximally independent subset of $\{e_1, \cdots, e_{m-1}, q_1, \cdots q_m\}$. But the set $\{q_1, \cdots, q_m\}$ is an independent subset of $\{e_1, \cdots, e_{m-1}, q_1, \cdots q_m\}$ and is of cardinality m. This contradicts the third property of *Lemma* 2, that the cardinality of all maximally independent subsets are equal, and proves that the above algorithm gives the desired maximally independent subset. □

However it is not enough to just identify the maximal weight basis corresponding to a given received sequence. This information will only allow us to compute the partial likelihood. In order to have an idea of our confidence in our decision we need to compute the total likelihood. For that we will need to generate the entire codeword and not only the basis for the codeword. In general, the MWB may not be the basis corresponding to only the information bits. If the basis corresponded to the information bits only we can generate the rest of the codeword from the convolutional encoder which provides us with the necessary functions. To compute the total likelihood we will also need to find an algorithm to find the codeword corresponding to the MWB. For that we note that a convolutional code can be represented by a system of linear equations relating the coded bits to the information bits. For any equation involving n variables, as soon as $n-1$ of these variables have been determined, the last variable can be found uniquely.

Based on these observations, we next present an algorithm that selects the maximum weight independent subset for a convolutional code along with the associated codeword.

Algorithm II: Codeword corresponding to maximal weight basis

- Set $I = \emptyset$
- Sort the weights $wt(i)$ of elements in $\mathcal{N}$ based on the received vector $\mathbf{r}$.
- While $|I| < NR$
  - Select the location $e \notin I$ from $\mathcal{N}$ with the largest weight such that $I + e$ is still an independent subset of $\mathcal{N}$. The received signal corresponding to the location e is r_e.
  - Set $\hat{d}_e = sgn(r_e)$.
  - Consider the convolutional code equations. In all the equations that location e appears, reduce the number of unknowns. If any of the equations has only one unknown left, solve for that unknown.
  - $I = I + e$

3.4.3 M largest weight bases

So far we have designed an algorithm to calculate the MWB of a convolutional code for a given received signal. At this point we would like to re-emphasize that our final aim is to find an approximate solution to the optimization problem described by (3.3). The idea is that the solution obtained by finding the MWB will be close to the optimal solution. However it is intuitive that rather than finding just one basis

that has a large *reliability* if we form a *feasible set* of a number of large weight bases and then optimize on that particular set, our probability of not finding the optimal codeword will be reduced. In other words, we want to find a set of M bases that have large associated weights and then find the one that gives the highest likelihood among these M bases.

So our first goal is to find the set, $\mathcal{M}_M$, of M bases such that there exists no other basis which is not in $\mathcal{M}_M$ and whose weight is larger than any basis in $\mathcal{M}_M$. We call this set the *M largest weight bases*. It is evident that $\mathcal{M}_1 = \{J^\star\}$. We have already found an algorithm to find the MWB $J^\star$ and now we want to find the other large weight bases. We will show that the bases in $\mathcal{M}_M$ differ from $J^\star$ in a limited number of positions and we can obtain the elements of $\mathcal{M}_M$ by modifying the MWB. This will save us some computational effort that needs to be spent to compute each of the M-largest bases from scratch.

Equation (3.4) is representative of the linear equations that describe any convolutional code. This relation shows that for any systematic convolutional code we can express all the parity bits in terms of the information bits. If $\mathbf{d}^P$ are the parity bits and $\mathbf{d}^I$ are the information bits, then the original convolutional code relationship can be expressed as $I_0\mathbf{d}^P = G\mathbf{d}^I$, where I_0 is an identity matrix and G is the generator matrix describing the convolutional encoder. Let the bits of the MWB be $\mathbf{d}^J$ and the rest of the dependent bits be $\mathbf{d}^D$. The original bits $\mathbf{d}^P$ and $\mathbf{d}^I$ can each be divided into two sets $\mathbf{d}_1^P, \mathbf{d}_2^P$ and $\mathbf{d}_1^I, \mathbf{d}_2^I$, where the subscripts 1 and 2 represent the bits in $\mathbf{d}^D$ and $\mathbf{d}^J$ respectively. Thus exchanging the columns we can represent the new set

of relationships as

$$\begin{bmatrix} I_1 & G_1 \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{d}_1^P \\ \\ \mathbf{d}_1^I \end{bmatrix}}_{\mathbf{d}^D} = \begin{bmatrix} I_2 & G_2 \end{bmatrix} \underbrace{\begin{bmatrix} \mathbf{d}_2^P \\ \\ \mathbf{d}_2^I \end{bmatrix}}_{\mathbf{d}^J}$$

We can reduce this relationship as

$$d_D = R^J d^J \tag{3.5}$$

using the algorithm described above. In fact, any Gaussian elimination scheme using $GF(2)$ arithmetic can achieve this simplification. This set of equations uniquely express the dependent bits in terms of the MWB and can also be used to find the values of the parity bits when the values of the bits corresponding to the MWB is obtained.

3.4.4 Distance of $\mathbf{m}^{th}$ largest basis from maximal weight basis

Let us recall that our suboptimal decoding algorithm approximates the calculation of the codeword with the largest likelihood by calculating the codeword with largest partial likelihood. Our belief is that the codeword with largest total likelihood will have a very large partial likelihood corresponding to any set of bits and hence a set of bits which form an independent basis. The advantage is that the largest independent basis can be calculated very efficiently. However the optimal codeword with largest total likelihood does not necessarily correspond to the maximal weight basis, but intuitively, we should be close to the correct solution.

Thus we need to compute not one but a number of codewords which have very large partial likelihoods and then we will be able to find from them the codeword

with the largest total likelihood. If each of these bases were very different from the maximal weight basis then it would make sense to compute them from scratch. But we will show that these other bases differ from the MWB in only a few places. Hence computation efforts can be saved if we start from the MWB and try to compute the other bases. We will first quantify the number of places the m^{th} largest weight basis differs from the MWB. We will then give a constructive proof towards our claim.

Lemma 3 *For a convolutional code of block-length N and rate R let $J^\star$ be the MWB. If e and f are two locations in the codeword such that $e \notin J^\star$, $f \in J^\star$ and $(J^\star - f + e)$ is also a basis, where $(J^\star - f + e)$ is the set $J^\star$ with the element f replaced by e, then $wt(e) \leq wt(f)$.*

Proof: $J^\star$ is the MWB and $J = (J^\star - f + e)$ is also another basis. This implies

$$wt(J^\star) \geq wt(J) = wt(J^\star) + wt(e) - wt(f).$$

Since all the weights are non-negative $wt(e) \leq wt(f)$ □

This simple yet instructive lemma will be used in our next result, which establishes a *partial ordering* among the bases and shows that the MWB acts as a *least upper bound* for each partial ordering. This result will be finally used to quantify the distance between the m^{th} largest weight basis and the MWB.

Lemma 4 *If J_m is a basis of $\mathcal{N}$, there is a monotone sequence consisting of single exchanges from J_m to the MWB. In other words if $J_0 = J^\star = \{e_1, e_2, \cdots, e_k\}$ is the MWB and J_m is another basis which differs from J_0 in m places, then we can find a sequence of bases $J_1, J_2, \cdots, J_{m-1}$ such that J_i differs from J_{i-1} in only one place and $wt(J_m) \leq wt(J_{m-1}) \leq \cdots \leq wt(J_1) \leq wt(J_0)$.*

Proof: From the results of *Lemma* 2 we know that the number of elements in all the bases are equal. Let us first consider the bases J_0 and J_m. In general, there will be

some elements that are common to both the bases. Let S be the set of these common elements. We will use S_0 to represent the set of elements $\{e_1, e_2, \cdots e_m\}$ which are exclusively in J_0 and not in J_m and $S_m = \{f_1, f_2, \cdots, f_m\}$ to represent the elements only in J_m. We can write

$$J_0 = S \cup S_0 \quad J_m = S \cup S_m.$$

J_m is a basis and so the whole codeword can be represented *uniquely* by the elements of J_m only. This implies that all the elements of the basis J_0 can also be represented by the basis J_m. All the elements in S of J_0 are also elements of J_m and can be expressed solely by the corresponding elements in J_m, but since the elements of S_0 are independent of S they require elements from S_m in their representations.

Let $T_0 \subset S_0$ be the set of elements that use f_1 for their representation. We claim that T_0 is nonempty. Let us assume that this is not true. This implies that all the elements of S_0 can be expressed by S and $S_m - f_1$. The elements of S can be trivially represented solely by elements of S. Combining, all the elements of J_0 can be represented by $S + S_m - f_1$. That is $S + S_m - f_1$ is a basis of J_0 which is also a basis of $\mathcal{N}$. But $S + S_m - f_1$ has one less element than J_m. Thus we have two bases of $\mathcal{N}$, J_m and $J_m - f_1$ with unequal number of elements. This contradicts the third proposition of *Lemma* 2 and hence our assumption is false. This implies that T_0 has at least one element. Without loss of generality let us assume that $T_0 = \{e_1, \cdots, e_q\}$.

Let us consider the representation of $e_i \in T_0$ in terms of the basis set J_m. This is essentially an equation involving e_i, f_1 and zero or more elements from J_m. This relationship also suggests how f_1 can be expressed in terms of the other elements in J_m and e_i. Thus all the elements in J_m can be expressed by elements of $J_m - f_1$ and e_i. In other words for all $e_i \in T_0$, $J_m - f_1 + e_i$ forms a basis. All these bases are obtained by a single exchange of elements in J_m and differ from the MWB, J_0, at

$m-1$ locations. We need to prove that at least one of these bases have weight no less than J_m to prove the lemma.

We now try to express f_1 in terms of the basis elements in J_0. We claim in this representation there will be at least one element from T_0. If this is not true, it implies that, f_1 can be represented by elements from $S + S_0 - T_0$. Now T_0 is the subset of elements in J_0 that need f_1 for their representation in terms of the elements of the basis J_m. That is all the elements in $S+S_0-T_0$ can be represented solely by J_m-f_1. Thus f_1 can be represented in terms of elements in $J_m - f_1$. But f_1 is a member of the basis J_m and as such we should not be able to represent f_1 by other elements in J_m. So our assumption is false and in the representation of f_1 in terms of elements of J_0, there is at least one element from T_0. Let that element be e_1'. Using a similar argument as above, we can now show that $J_0 + f_1 - e_1'$ is a basis.

That is, we have shown that along with J_0 and J_m, both $J_0+f_1-e_1'$ and $J_{m-1} = J_0-f_1+e_1'$ are also bases of $\mathcal{N}$. Now since J_0 is the MWB and $J_0+f_1-e_1'$ is also a basis, from *Lemma* 3, we can say that the weight of e_1' is no less than the weight of f_1. Thus we have created a new single exchange basis $J_{m-1} = (J_m - e_1' + f_1)$, whose weight is no less than that of J_m and which has $(m-1)$ elements different from J_0. Since m is finite, proceeding in this way, we will be able to generate the monotone sequence. □ .

Lemma 5 *If $\mathcal{M}_m$ is the set of m maximal weight bases and the basis $J, (\notin \mathcal{M}_m)$, is the $(m+1)^{th}$ largest weight basis, then there exists at least one basis $J' \in \mathcal{M}_m$ such that J differs from J' in exactly one location.*

Proof: Let us assume that the claim is not true. From the previous lemma it follows that we can find a monotone sequence of single exchanges from J to the MWB $J^\star$. Let $\tilde{J}$ be the basis that is obtained from J by the first single exchange and so

$wt(\tilde{J}) \geq wt(J)$. Since we have assumed the claim is not true and $\tilde{J}$ differs from J in one position, it is not an element of $\mathcal{M}_m$. This implies there exists a basis $\tilde{J}$, not in $\mathcal{M}_m$, whose weight is greater than or equal to J. This contradicts our assumption and asserts the lemma. □

From the discussions in the previous section it is evident that for any basis J_m which differs from the MWB $J^\star$ at m locations there are at least m bases whose weight are greater than or equal to J_m. So the m^{th} largest basis cannot differ from the MWB at more than m locations. However we will prove a tighter bound on the number of locations in which the m^{th} largest basis differs from the MWB.

Theorem 2 *The m^{th} largest weight basis differs from the maximal weight basis in at most $\lceil \log_2 m \rceil$ locations.*

Proof: In order to prove the lemma let us look at a basis J_m which differs from the MWB at m locations. Let a chain of single exchange monotone sequence be $J_m, J_{m-1}, \cdots, J_0 = J^\star$. As before let us assume that the common elements in J_0 and J_m be S and the differing elements be the sets $S_0 = \{e_1, e_2, \cdots, e_m\}$ and $S_m = \{f_1, f_2, \cdots, f_m\}$ respectively.

Without loss of generality, we order the elements in S_0 and S_m in such a way that J_{m-i} is obtained from J_{m-i+1} by exchanging f_{m-i+1} with e_{m-i+1}. In particular $J_1 = J_0 - e_1 + f_1$ and $J_2 = J_1 - e_2 + f_2 = J_0 - e_1 - e_2 + f_1 + f_2$ and so on. Following the arguments from *Lemma* 3 and 4, we can show that $wt(e_i) \geq wt(f_i)$ and that $J_0 - e_2 + f_2$ is also a basis. We can conclude that $wt(J_0 - e_2 + f_2) \geq wt(J_2)$. A generalization of this statement is that for every basis J_m which differs from the MWB at m locations there are $1 + 2 + \cdots + 2^{m-1} = 2^m - 1$ bases which have weight at least as large as J_m. In other words the m^{th} largest weight basis differs from the MWB in at most $\lceil \log_2 m \rceil$ locations. □

3.4.5 Replacement elements

Even though none of the bases in $\mathcal{M}_M$ differ from the MWB $J^\star$ in more than $\lceil \log_2 M \rceil$ locations, the locations that they differ from the maximal weight basis may not be the same $\lceil \log_2 M \rceil$ locations. For example it may so happen that all the bases differ from the MWB in one location only but that location is different for each basis. However using *Lemma* 5, we can safely claim that the total number of locations in which the MWB differs from the collection of other bases in $\mathcal{M}_M$ will not exceed M. However there are NR possible locations in the codeword, and as such we might need to consider $\binom{NR}{M}$ locations. Similarly these locations should be filled up with elements from $\mathcal{N} - J^\star$. There are $N - NR$ such elements and thus there are $\binom{N-NR}{M}$ choices. This can be quite large for large N. We would like to prune down the options that should be verified to obtain the desired M-largest weight bases.

Definition 3.4.5 *A pair of elements* $[e, f]$*, where* $e \in J$ *and* $f \in \mathcal{N} - J$*, is defined to be a* J-exchange*, if* $(J - e + f)$ *is a basis when* J *is a basis.*

Definition 3.4.6 *For every element* $e \in J^\star$*, there is a set of elements*

$$R(e) = \{f | [e, f] \text{ is a } J\text{-exchange}\}.$$

We call this set the replacement set *of* e*. The* replacement element*,* $r(e) = f \in R(e)$*, is defined as that element for which* $wt(e) - wt(f)$ *is the minimum.*

Just as we have defined the replacement set for the elements in the MWB we can also define a replacement set, $\tilde{R}(f)$, and a replacement element, $\tilde{r}(f)$ for the element $f \in \mathcal{N} - J^\star$.

We are now ready to find the elements of the M maximal weight bases $\mathcal{M}_M$. We will first claim that there are a number of locations in the MWB that will remain

invariant in all the M bases which are elements of $\mathcal{M}_M$. Similarly there will be some elements in $\mathcal{N} - J^\star$ that cannot appear in any of the bases. We will also find methods to identify these locations.

Theorem 3 *Given a maximal weight basis $J^\star$ of $\mathcal{N}$ and the replacement elements $r(e)$ for all elements in $J^\star$, the set of $NR - M$ elements that will remain unchanged can be computed in linear time in M.*

Proof: For each element $e \in J^\star$ let $w'(e) = wt(e) - wt(r(e))$. In other words $w'(e)$ is the minimum weight that we lose by replacing the element e from the MWB. We can find the $M-1$ smallest values of w' using a linear time (in M) selection algorithm [**?**]. Let Q be the set of the remaining $NR - M$ elements.

Then for each element $e \in Q$ there are at least $M-1$ elements e' with $wt\big(J^\star - e' + r(e')\big) \leq wt\big(J^\star - e + r(e)\big)$, and therefore together with $J^\star$ there are at least M bases better than $(J^\star - e + r(e))$. Therefore every basis in the M maximal weight basis must contain the element e. □

Similarly for every element $f \in \mathcal{N} - J^\star$ once we know the replacement element $\tilde{r}(f) \in J^\star$, we can show a similar result:

Theorem 4 *Given a maximal weight basis $J^\star$ of $\mathcal{N}$ and the replacement elements $\tilde{r}(f)$ for all elements in $\mathcal{N} - J^\star$, the set of $(N - NR - M)$ elements that cannot be a part of any basis of the M maximal weight basis can be calculated in linear time in M.*

We have so far shown how to efficiently calculate the MWB $J^\star$ of $\mathcal{N}$ and given an optimal weight basis and replacement elements for both the basis and non basis elements how to calculate the set of M elements in $J^\star$ that can be potentially replaced

and the set of M elements in $\mathcal{N} - J^\star$ which are the corresponding potential replacement candidates. However we have not yet shown how to calculate the replacement elements and in general the replacement set for every element. We will use two properties to calculate the replacement set for each element both in $J^\star$ and $\mathcal{N} - J^\star$.

Lemma 6 *For all the non basis elements, the relationship $d_D = R^J d^J$ in (3.5), defines the replacement set.*

Proof: Each variable in the non-basis element is uniquely defined in terms of a set of basis variables. We claim that these basis variables constitute the replacement set. If this is not true then let f_i be a non-basis element which is defined by the basis elements $e_1, \cdots, e_k$. If e_{k+1} is also a member of the replacement set then it implies that $\{e_1, \cdots, e_k, f_i, e_{k+2} \cdots, e_n\}$ is a basis, where we assumed the rank is n. We should be able to describe e_{k+1} in terms of this new basis. Now since e_{k+1} is a part of the MWB it cannot be described exclusively by the basis elements e_i and needs f_i. From this relation we have two descriptions of f_i in terms of the MWB, one that includes e_{k+1} and the other that doesn't. Thus the description is non-unique which is contradictory to the definition of basis. This proves that the replacement set can be defined by (3.5). □

We also claim that the replacement set relation is *symmetric*. By that we mean if f_i is a member of the replacement set of e_j, then e_j is a member of the replacement set of f_i. So once we have constructed the replacement set of all the non-basis elements we can automatically calculate the replacement set of all basis elements. In terms of (3.5) the non-zero entries in the columns give the replacement set of the basis elements.

It should be noted that the two replacement sets, along with the MWB $J^\star$, form a *necessary and sufficient statistic* for the calculation of the M largest weight bases $\mathcal{M}_M$. The set of M elements from $J^\star$ and the corresponding replacement elements alone are not enough for the computation as there might be two elements e and e' in this set such that there exists two elements e_1, e_2 in the replacement set of e with $wt(e) - wt(e_1)$ and $wt(e) - wt(e_2)$ both smaller than $wt(e') - wt(r(e'))$. In that case we have to consider both e_1 and e_2 before replacing e'.

Once we have calculated the MWB $J^\star$, the M potential replaceable candidates in $J^\star$ and $\mathcal{N} - J^\star$, we prune the replacement sets to include only these candidates. Thus for each of the M basis elements e_i that can be replaced, we calculate $wt(e_i) - wt(f_j)$ where f_j is both a member of the replacement set of e_i and is also one of the M elements in the non basis set that are candidates for replacement.

3.4.6 Complexity of maximal weight basis decoding algorithm

Let us once again recall the various steps for the maximal weight basis decoding algorithm. We then analyze the computation cost of each step resulting in the computational complexity of the algorithm.

1. Upon receiving N symbols sort them in the decreasing order of their absolute values. Using *Algorithm* II of Section 3.4.2, compute the maximal weight basis from the sorted list and the dependence relationship among the various coded bits as in (3.5). Also compute the first candidate codeword from this maximal weight basis.

2. For each element in the basis set calculate the corresponding replacement element using *Lemma* 5. Using the symmetric relationship of the replacement

set, for each element in the non-basis set compute the replacement elements using (3.5).

3. Using *Theorem* 3 and 4, select M elements from both the basis and non-basis sets that have the least amount of replacement penalty.

4. Calculate the table of replacement penalties for these two sets of elements.

5. Find the M largest weight bases from the maximal weight basis and the table of replacement penalties.

6. Determine the codewords corresponding to these M largest weights. Compute the total likelihood corresponding to each codeword and select the one with the largest likelihood.

The first step of the algorithm requires us to sort N absolute values for the received symbols. Using a radix sort algorithm [**?**] this computation can be done in $\mathrm{O}(N)$ operations. In order to calculate the maximal weight basis in each step of *Algorithm* II, we will need to find whether an element added to an independent subset still keeps it independent. Since each equation has κ variables and each variable can appear in at most κ equations this check can be accomplished in κ^2 operations. The cardinality of the maximal weight basis is NR, so we will need to perform each of these operations $\mathrm{O}(N)$ times. Thus we may need at most $N\kappa^2$ operations to calculate the maximal weight basis and the corresponding codeword. This particular computation also enables us to compute the relationship given by (3.5). The computation of replacement elements thus can be done in $\mathrm{O}(N)$ steps.

From *Theorem* 3 it is apparent that we will need to select M top replacement elements from the list of replacement elements. A linear selection algorithm can

be used to achieve the top M replacement elements in $\mathrm{O}(M)$ steps. The table of replacement penalties has M^2 entries and they can be calculated in M^2 steps. We then have to sort these entries using another radix sort algorithm. The computation of each of the other M bases and their codes from the MWB and the replacement sets will require at most N operations. Therefore the total complexity of the algorithm is $\mathrm{O}(N\kappa^2 + MN + M^2)$.

It should be recalled that Viterbi's algorithm has a decoding complexity given by $\mathrm{O}(N2^{\kappa+1})$. The complexity of the MWB decoding algorithm also grows linearly with the block length N as Viterbi's algorithm. But the complexity of our algorithm increases only *quadratically* with the constraint length. This feature will enable us to decode convolutional codes of large constraint length in real time.

3.5 Simulation results

We have shown that the proposed maximal weight basis algorithm requires much fewer operations than Viterbi's decoding rule. However, since MWB is not necessarily optimal, we will study the performance loss from the optimal Viterbi algorithm in an AWGN channel via simulations. Figure 3.4 shows the comparison for systematic convolutional codes of rate 1/2 and 2/3 and constraint length 7. We decode 1000 information bits at a time. For the MWB algorithm, we have used $M = 6$, that is, we only kept a list of size 6 in the suboptimal algorithm. The simulation results show there is very little performance loss when using the suboptimal MWB algorithm over the optimal decoding algorithm. Since MWB provides a suboptimal decoding method, it results in performance loss when compared to the probability of Viterbi algorithm on the *same* code. However, the computational complexity of MWB is much lower than the Viterbi algorithm. We now can afford to decode a

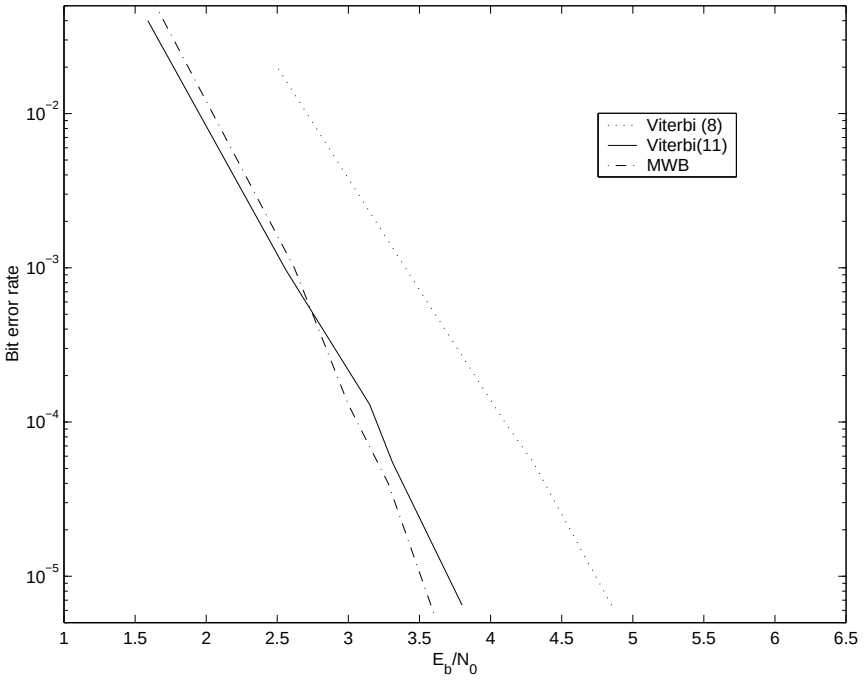


Figure 3.3 : Performance comparison of the MWB decoding algorithm for a convolutional code of constraint length 18 and the Viterbi algorithm for convolutional codes with constraint length 8 and 11. All codes are of rate 1/2 and $M = 10$.

stronger code with larger constraint length using the same number of computations (or spending the same amount of processing time) as the Viterbi algorithm uses on a code with smaller constraint length. Thus what we have gained in *complexity* could be traded in for improvement in *performance*. We study such a situation in Figure 3.3. We compare the performance of Viterbi decoder on a convolutional code of rate 1/2 and constraint length 11 with MWB decoder on a code of same rate and constraint length 18. The parameter M is chosen to be 10 for the MWB decoder. The Viterbi algorithm requires $(2^{12}N)$ operations while MWB uses only $(18^2+10)N+100$ operations. Yet the performance is almost indistinguishable. If we consider a convolutional code of same rate and constraint length 8, we see that the computational complexity of the Viterbi decoder (2^9N) is comparable to that of the

MWB algorithm, but MWB outperforms Viterbi. Even though MWB is a suboptimal decoding method, the better error correcting capability of the code with larger constraint length is the dominant factor. It should however be noted we haven't counted the exact number of operations but merely the order and such a study of implementation aspects is planned for the future, but the potential of our system is apparent.

3.6 Conclusions and future work

The optimal Viterbi decoding algorithm for convolutional codes has a complexity that grows exponentially with the constraint length. This prohibits implementation of "strong" convolutional codes for practical systems. In this chapter we have proposed a maximal weight basis algorithm that has complexity quadratic in the constraint length and yet performs close to Viterbi algorithm. The computational complexity of the MWB algorithm grows only linearly with the block-length, same as the Viterbi decoder. The MWB decoding approximates the computation of the codeword with largest total likelihood by significantly pruning the search space in a computationally efficient manner. Added advantages of MWB are the deterministic complexity unlike stack based algorithms, which makes it suitable for real-time applications, and insignificant storage space when compared to the stack-based or table look-up based algorithms. The benefits of MWB based decoding have also been illustrated in a multiuser environment [**?**] where we combined MWB with a multiuser iterative interference cancellation technique.

In this chapter we have restricted most of our discussions to systematic convolutional codes. The MWB decoding described in this chapter uses a block decoding approach to convolutional codes and does not utilize the particular structure of the

systematic convolutional code generator matrix. Therefore it can easily be generalized to non-systematic convolutional codes and linear block codes. The MWB algorithm can also handle $M-ary$ alphabets, the only adjustment is that one needs to redefine the *weights* as $w_i = (r_i - \alpha)^2$ for each $M-ary$ value α.

The discussion in this chapter is mostly restricted to hard decisions on coded bits. As reported by several researchers, better decoding results can be expected if soft decision output decoders are considered. It should be noted that for each dependent bit in the codeword we have a linear relationship expressing the dependent bits in terms of the independent bits. We can calculate the soft likelihood values for these bits from the corresponding received signal. We can compute the soft decisions corresponding to the dependent bits from the likelihoods of the independent bits and the linear relationship between them using likelihood algebra [54]. These results can also be extended to turbo codes also which are usually defined by two constituent convolutional codes. These likelihood values can be used as approximations to maximum a-posteriori estimates at every iteration. These issues are currently explored as future research directions.

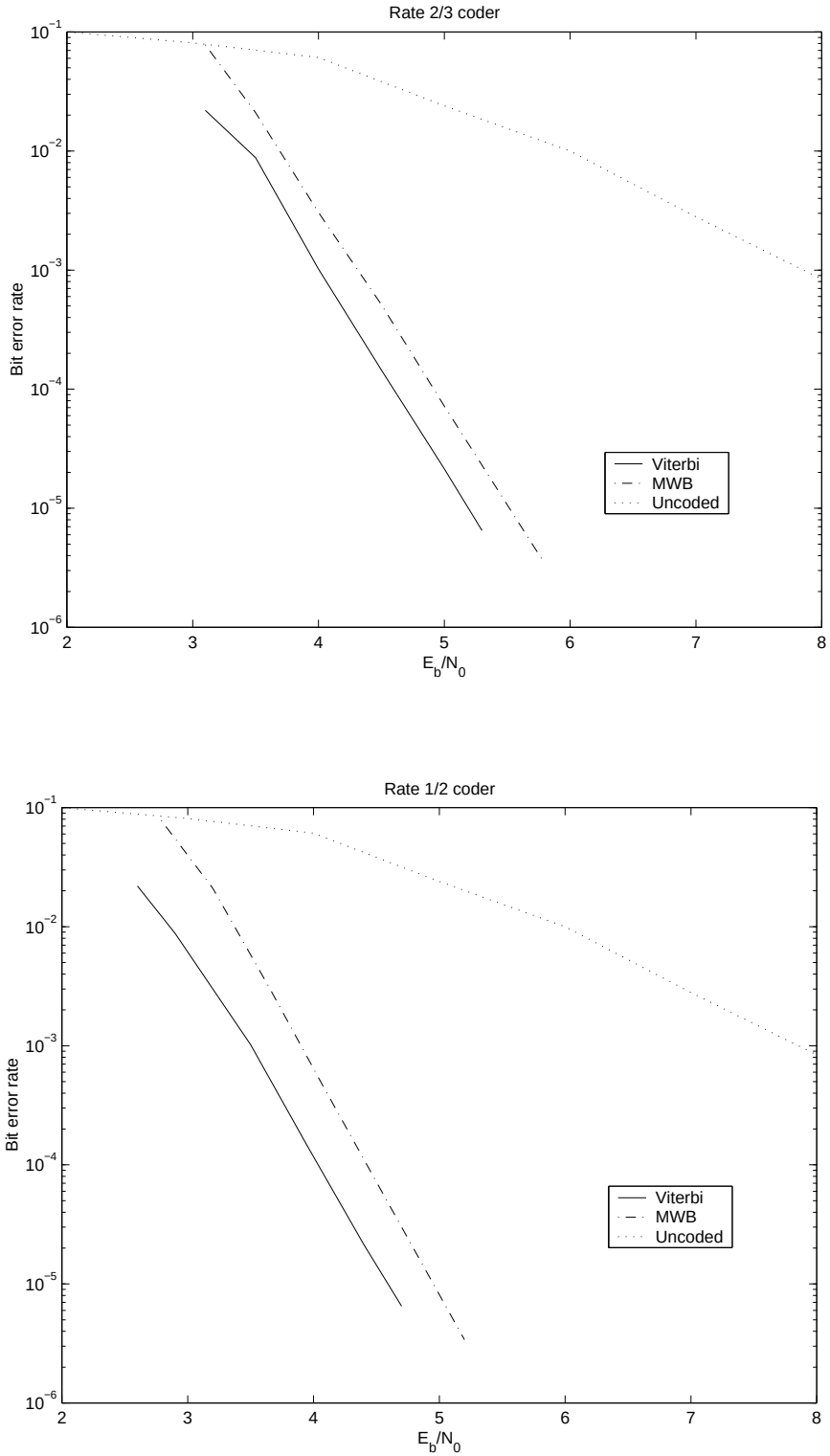


Figure 3.4 : Comparison of performance of MWB decoding algorithm versus Viterbi decoding for a rate 2/3 and 1/2 convolutional code with constraint length 7. M is chosen as 6.

Chapter 4

Joint multiuser detection and decoding

In the previous sections we have first studied the theoretical limits of a Gaussian channel and the maximum rate at which data can be transmitted when we can tolerate a certain amount of error in the recovered information bits if we employ the optimal coding techniques. We also compared the performance of popularly used codes with respect to this new benchmark and realized that in case of a convolutional code the optimal bit-error rate performance can be obtained at high computational costs. We then proposed a suboptimal algorithm to reduce the complexity of decoding convolutional codes.

So far most of our results are primarily targeted towards the single user environment. However in a practical wireless system a number of users use a common channel. In this chapter we will present a low complexity multiuser decoding technique that can be implemented in real-time for a convolutionally coded direct sequence code division multiple access system. The main idea, which we call *iterative prior update* consists of iterative interference cancellation and prior updates on sequences of coded bits, combined with list decoding.

4.1 Introduction

Direct-sequence code-division multiple-access (DS-CDMA) systems assign mobile users different signature waveforms over which information bearing signals are mod-

ulated. One important purpose of these signature waveforms is for the base station to be able to distinguish among different users. The asynchronous nature of the mobile transmissions together with impossibility of designing mutually orthogonal signature waveforms for all possible delays results in an interference limited uplink cellular system. Multiuser detection techniques have been very effective in combating this multiple access interference [55]. The optimal multiuser detector, investigated by Verdú [56], has exponential complexity in the number of users. This observation has led to a number of suboptimal alternatives with lower complexity [57, 58, 59].

Apart from the multiple access interference, thermal noise is another important source of error for mobile signals. Channel coding strategies are essential for protection of information against various sources of error and convolutional coding in conjunction with DS-CDMA has been shown to provide sufficient protection against errors in a wireless system [20, 9].

The optimal detection and decoding strategy for a convolutionally coded uplink DS-CDMA system consists of combining the trellises of all the users. Like the optimal detector, this has exponential complexity in the number of users [60] and thus the feasibility of implementation in real-time systems is severely limited. A straightforward low complexity alternative is to have a suboptimal multiuser detector, which makes hard decisions on the coded bits, followed by single user decoders. Although this strategy manages to bring the complexity down to linear in the number of users, it results in a considerable loss in performance. Our goal in this chapter is to balance the load, in terms of computational complexity, between the detector and the decoder and achieve acceptable error and delay performance along with real-time implementation speed. This necessitates passing information back and forth in an iterative fashion between the multiuser detector and decoder.

Several suboptimal joint detection and decoding techniques [61, 62] have already been proposed in the literature. A number of recent studies are on iterative, or *turbo* methods [63, 64, 65, 66] named after the iterative principle successfully used in the turbo codes [54]. These turbo methods are based on the maximum-a-posteriori (MAP) rule of Bahl et. al. [67] that aims to minimize the probability of symbol error in a convolutional code. The Viterbi algorithm, as described in t previous chapter, on the other hand, minimizes the probability of sequence (or codeword) error. The turbo principle consists of successfully applying the MAP rule and passing prior distributions on symbols iteratively along the component codes of the particular system being examined.

In this chapter, we discuss a low complexity iterative detection and decoding strategy that also makes use of the maximum-a-posteriori decision rule, but this time on *sequences* of symbols unlike the turbo methods. Our scheme, which we call *iterative prior update*, consists of iterative interference cancellation on coded sequences of bits using MAP rule for decoding these sequences along with prior updates at every iteration using the Viterbi algorithm. We argue, through real system examples, that this strategy provides a receiver structure that has low computational complexity, delay and buffer size requirements and thus can be implemented in real time. An important benefit is that our iterative prior update based solution uses the already existing hardware solutions for the Viterbi algorithm; thus can be implemented with today's systems only with minor modifications.

The rest of the chapter is organized as follows: Section 4.2 provides a description of the optimal multiuser sequence detection and decoding rule for a convolutionally coded DS-CDMA system. Section 4.3 describes our proposed iterative prior update algorithm. Section 4.4 discusses computational complexity and delay issues, and

provides modifications that enable real time implementation of the iterative prior update scheme. Section 4.5 includes comparisons with other suboptimal joint sequence detection and decoding techniques both by simulations and by providing an analytical framework. The convergence of the iterative prior update algorithm can be shown to be fast and storage requirements for keeping track of priors along iterations is low. This section also contains a numerical example illustrating the speed of the iterative prior update strategy establishing it as an important alternative for real-time implementations.

4.2 Optimum joint detection and decoding

4.2.1 System model

We consider an asynchronous convolutionally coded uplink DS-CDMA system with K users. User k has the normalized spreading code s_k ($< s_k, s_k >= 1$) for $k = 1, \ldots, K$ that extends over a symbol period of length T and consists of N_c chips. We let the vector $\mathbf{b}_k$ denote the uncoded information bits of user k, and the vector $\mathbf{d}_k$ denote the corresponding coded bits. We observe the received signal $r(t)$ at the base station for N symbol periods, so $\mathbf{d}_k$ is of length N for all k. The baseband signal $r(t)$ can be written as

$$r(t) = \sum_{k=1}^{K} \sum_{i=1}^{N} A_k d_k(i) s_k(t - iT - \tau_k) + z(t),$$

where A_k is the amplitude and τ_k is the delay of user k at the receiver. The signal $z(t)$ denotes the additive white Gaussian noise.

When $r(t)$ is passed through a chip matched filter [20], the discrete output $\mathbf{r}$, a vector of length $N_c(N+1)$, can be expressed as

$$\mathbf{r} = \mathbf{SAd} + \mathbf{z}, \tag{4.1}$$

where $\mathbf{S}$ is the $N_c(N+1) \times NK$ matrix of the signature waveforms repeated over N bits, $\mathbf{A}$ is the $NK \times NK$ diagonal matrix of user amplitudes, $\mathbf{d}$ is the $NK \times 1$ vector of the coded bits of all users over N symbol periods and $\mathbf{z}$ is the $N_c(N+1) \times 1$ noise vector. Details about this model of the uplink can be found in [20]. If $\mathbf{r}$ is further passed through a bank of code matched filters, the output is an $NK \times 1$ vector

$$\mathbf{y} = \mathbf{R_N A d} + \eta, \tag{4.2}$$

where $\mathbf{R_N} = \mathbf{S'S}$ is the asynchronous code correlation matrix and η is the resulting additive noise vector. Since the output of the bank of code-matched filters provides a sufficient statistic for the estimation of the vector $\mathbf{d}$, either $\mathbf{r}$ or $\mathbf{y}$ can be used to find these estimates.

4.2.2 MAP sequence decoding

The goal of the receiver is to obtain reliable estimates of the information bit sequences of all the K users simultaneously. If we let vector $\mathbf{b} = [\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_K]'$ denote the information bit sequences transmitted by the K mobile users and $\hat{\mathbf{b}}$ its estimate at the receiver, the aim is to minimize the probability of error $\Pr(\hat{\mathbf{b}} \neq \mathbf{b})$. It is well known that this is achieved by choosing the maximum-a-posteriori estimate of $\mathbf{b}$ given $\mathbf{r}$, that is

$$\hat{\mathbf{b}} = \arg\max_{\mathbf{b}} p(\mathbf{b}|\mathbf{r}).$$

The same maximization can be written in terms of the coded bits of all the users by noting that for any useful error correction code, there is a one-to-one relationship between the information bit sequences and coded bit sequences. Then the equivalent coded bit estimate is given by

$$\hat{\mathbf{d}} = \arg\max_{\mathbf{d} \in \mathcal{C}} \log p(\mathbf{d}|\mathbf{r}), \tag{4.3}$$

where $\mathcal{C}$ denotes the collection of all users' codebooks. We have used the fact that logarithm is an increasing function of its argument and thus does not change the maximization problem. Note that we now have a constrained optimization as the codewords have to be part of their respective codebooks.

The optimal decoding rule necessitates a joint search through the codebooks of all K users. This search has exponential complexity in K. The rest of this chapter will be on approximating the optimization in (4.3) by an iterative interference cancellation, MAP sequence decoding and prior update scheme. We will observe that this strategy brings the complexity of joint detection and decoding down to linear in K.

Before we continue with the proposed iterative prior update algorithm, we would like to elaborate on (4.3). It is possible to rewrite it as

$$\begin{aligned}\hat{\mathbf{d}} &= \arg\max_{\mathbf{d}\in\mathcal{C}}[\log p(\mathbf{d},\mathbf{r}) - \log p(\mathbf{r})] \\ &= \arg\max_{\mathbf{d}\in\mathcal{C}} \log p(\mathbf{d},\mathbf{r}) \\ &= \arg\max_{\mathbf{d}\in\mathcal{C}}[\log p(\mathbf{r}|\mathbf{d}) + \log p(\mathbf{d})]. \end{aligned} \tag{4.4}$$

Above derivations make use of Bayes' rule. Also, we removed $\log p(\mathbf{r})$ as it is common for all $\mathbf{d} \in \mathcal{C}$ and does not affect the optimization process.

The well-known maximum likelihood sequence detection rule is a special case of (4.4) when we have uniform priors on the coded bit sequences of all the users. While this assumption is very reasonable for optimum detection and decoding, we will observe that for a suboptimal iterative scheme it is beneficial to keep track of priors along iterations. This notion will be clarified in the next section where we describe the proposed algorithm.

4.3 Iterative multiuser detection and decoding

Let us first focus on a particular mobile, say k. Let $I_k = \{1, \ldots, k-1, k+1, \ldots K\}$ be the set of interferers for mobile k. We can rewrite (4.1) as

$$\mathbf{r} = \mathbf{S}_k \mathbf{A}_k \mathbf{d}_k + \mathbf{S}_{I_k} \mathbf{A}_{I_k} \mathbf{d}_{I_k} + \mathbf{z},$$

where $\mathbf{S}_k$, $\mathbf{A}_k$ and $\mathbf{d}_k$ denote the spreading code, amplitude and coded bit matrices of user k spanning N symbol periods, and $\mathbf{S}_{I_k}$, $\mathbf{A}_{I_k}$ and $\mathbf{d}_{I_k}$ are the corresponding quantities for the set of interferers.

If we had accurate estimates of $\hat{\mathbf{d}}_{I_k}$, the coded bit sequences of the interferers, then the signal $\hat{\mathbf{y}}_k$ given by

$$\hat{\mathbf{y}}_k = \mathbf{S}_k'(\mathbf{r} - \mathbf{S}_{I_k} \mathbf{A}_{I_k} \hat{\mathbf{d}}_{I_k}) \tag{4.5}$$

would provide a sufficient statistic for estimation of the coded bit sequence $\mathbf{d}_k$. The minimum probability of error estimate would then be given by the maximum-a-posteriori sequence rule as

$$\begin{aligned} \hat{\mathbf{d}}_k &= \arg \max_{\mathbf{d}_k \in \mathcal{C}_k} p(\mathbf{d}_k | \hat{\mathbf{y}}_k) \\ &= \arg \max_{\mathbf{d}_k \in \mathcal{C}_k} [\log p(\hat{\mathbf{y}}_k | \mathbf{d}_k) + \log p(\mathbf{d}_k)]. \end{aligned} \tag{4.6}$$

Here $\mathcal{C}_k$ denotes the codebook of user k. With uniform priors on all the codeword sequences, this results in maximum likelihood sequence estimates, or single user Viterbi decoders for all the K users.

However, for any practical system there will be a nonzero probability of error associated with the interferer estimates $\hat{\mathbf{d}}_{I_k}$. This suggests that $\hat{\mathbf{y}}_k$ in (4.5) will contain residual interference and will no longer be sufficient for the estimation of $\mathbf{d}_k$. It was shown in [61] that one can get satisfactory performance by an iterative

interference cancellation scheme. For user k, $k \in \{1, \ldots, K\}$, at iteration step i, coded bit sequence estimates $\hat{\mathbf{d}}_{I_k}$ from iteration step $i-1$ are used to obtain the semi-interference-free soft signal $\hat{\mathbf{y}}_k$ which in turn is used to obtain the maximum likelihood sequence estimate of $\mathbf{d}_k$. Thus the prior $p(\mathbf{d}_k)$ in (4.6) is assumed to be uniform for all the iteration steps and $p(\hat{\mathbf{y}}_k|\mathbf{d}_k)$ is approximated assuming complete interference cancellation. In [62] the performance of this strategy was analyzed for a DS-CDMA system combined with trellis coded modulation.

In the iterative scheme described above, only hard decisions on the coded bit sequences $\hat{\mathbf{d}}$ are passed along iterations. However, while obtaining those estimates, we in fact calculate the *posterior* distributions $p(\mathbf{d}_k|\hat{\mathbf{y}}_k)$ for all the users. The MAP sequence estimate is simply the mode of this posterior distribution, so it carries less information than the distribution itself. The knowledge of the distribution along with the coded sequence estimate provides robustness against estimation errors. Hence we would like to pass these posteriors along iterations as well. In our proposed *iterative prior update* algorithm, for every user we set the posterior of iteration step $i-1$ as the *prior* distribution of step i. This prior and the signal $\hat{\mathbf{y}}_k$ at step i are both used to calculate the posterior and the MAP sequence estimate for the current iteration according to (4.6). Iterations are carried on until we reach a reliable estimate of the information bit sequences. The proposed algorithm is summarized below.

Iterative prior update (IPU) algorithm

1. Set initial coded sequence estimates $\hat{\mathbf{d}}_k^0 = \mathbf{0}$, and initial priors $p^0(\mathbf{d}_k)$ uniform for $k \in \{1, \ldots, K\}$. This corresponds to setting the first iteration estimates based on matched filter outputs.

2. In iteration step i, for $k \in \{1, \ldots, K\}$

- Set

$$\hat{\mathbf{y}}_k^i = \mathbf{S}_k'(\mathbf{r} - \mathbf{S}_{I_k}\mathbf{A}_{I_k}\hat{\mathbf{d}}_{I_k}^{i-1}),$$

  where $I_k = \{1, \ldots, k-1, k+1, \ldots K\}$.

- Set $\hat{\mathbf{y}}_k = \hat{\mathbf{y}}_k^i$ and $p(\mathbf{d}_k) = p^i(\mathbf{d}_k)$ in (4.6). Calculate the posterior $p^i(\mathbf{d}_k|\hat{\mathbf{y}}_k^i)$ and the current estimate $\hat{\mathbf{d}}_k^i$.

- Set the prior for the next iteration equal to the posterior for this iteration, that is

$$p^{i+1}(\mathbf{d}_k) = p^i(\mathbf{d}_k|\hat{\mathbf{y}}_k^i).$$

3. Iterate until there is no further change in successive codeword estimates.

Figure 4.6 shows a block diagram representation of a particular iteration step of this algorithm for user 1. The "Spread" and "Amplitude" blocks correspond to multiplying by $\mathbf{S}_{I_k}$ and $\mathbf{A}_{I_k}$ respectively. "Estimate coded bits" block performs MAP sequence decoding using the updated prior and the signal $\hat{\mathbf{y}}_k$. Note that it is possible to speed up the algorithm further by immediately using the estimates $\hat{\mathbf{d}}_1^i, \ldots, \hat{\mathbf{d}}_{k-1}^i$ in obtaining the estimate $\hat{\mathbf{d}}_k^i$.

Having motivated and mathematically stated the iterative prior update algorithm, we will next provide a complexity analysis of the proposed scheme. We will also incorporate practical constraints such as decoding delay and storage requirements into the algorithm making it suitable for real-time implementations.

4.4 Complexity and efficient implementation

The optimum multiuser detection and decoding algorithm is computationally expensive in terms of the number of users K because it optimizes jointly over the

codebooks of all users. The algorithm described above uses K single user decoders per iteration step and thus brings the complexity down to linear in K rather than exponential. We will later on show via simulations that a few iterations are enough for convergence.

Viterbi algorithm can be used to provide the likelihood value (and thus the posterior) for the codeword with largest likelihood - essentially the mode of the distribution. However our goal is to carry out the information about the entire distribution along iterations. A straightforward modification of Viterbi algorithm that calculates the likelihoods for all codewords would require keeping track of partial codewords at each state of the trellis, and would result in a large number of computations. Furthermore, the algorithm in Section 4.3 requires that the receiver store the posterior distribution for all $\mathbf{d} \in \mathcal{C}$ along iterations. This means we need a storage space exponential in the block length N.

However, we conjecture that except for a few codewords, for most of the coded sequences the corresponding posteriors are insignificant. In fact, calculating and maintaining the posteriors for only these few codeword sequences can very well approximate the distribution. Based on this conjecture our strategy, akin to list decoding [53], will only calculate and store the probabilities of L codewords that have the highest posterior distribution thus keeping the computational complexity and storage requirements low. By properly choosing L, we can ensure that the total probability of these top L codewords is close to one. Through simulations, we show that the above conjecture is true and that small values of L give satisfactory performance.

Another important issue is the decoding delay encountered by the single user decoders used within the iterative prior update algorithm. The M-algorithm [68]

provides a standard method for reducing the decoding delay for a convolutional code. A window of size 5κ, where κ is the constraint length of the convolutional code, is used to make a decision on the first branch of the trellis. The window is then slid by one symbol period to continue decoding. However, when combining list decoding with the M-algorithm, storing the top L paths up to level m in the trellis does not necessarily lead to the top L paths up to any subsequent levels. The next lemma shows how to overcome this problem in a storage-efficient manner without the need of an elaborate sorting scheme at each stage.

Theorem 5 *For a convolutional code of constraint length κ, to evaluate the top L paths at any level, or depth, of the trellis we need to store at most $L2^{\kappa}$ path metrics.*

Proof: We start with a few words on the notation. For a convolutional code of constraint length κ, we have 2^{κ} states. The trellis branch connecting state i to j is denoted by $b(i \rightarrow j)$. A path up to level t is a $t \times 1$ vector of branches. Note that not all combinations of branches can form a path on the trellis, the terminal state of the branch at time $t-1$ should be the originating state of the branch at time t.

We claim that if we have the L most probable paths into each state at a particular level of the trellis, then we will be able to calculate the top L most probable paths in the next level. Since the number of states is given by 2^{κ}, the total storage requirement will then be $L2^{\kappa}$ proving the theorem.

In order to show that the above claim is true, we use induction on the level of the trellis. The claim is true for level 1, since all the paths originate from the same node at level 0. Let us assume the claim is true for level $(t-1)$. We have to prove that if we have the top L paths ending at each state in level $(t-1)$, we will be able to get the top L paths at level t.

Let us assume the contrary. Suppose there exists a path at level t that is one of the top L most probable paths ending at a particular state but its predecessor at level $t-1$ does not correspond to one of the top L paths stored for each state at level $t-1$. Note that we calculate the probabilities conditioned on the received signal $\mathbf{y}^t = (\mathbf{y}_1, \ldots, (\mathbf{y}_t)$. We denote this path by the vector $P_0^t = (b(s_0 \rightarrow s_1), \ldots, b(s_{t-1} \rightarrow s_t)$ and its predecessor by P_0^{t-1} where s_i denotes the state that the path goes through at level i. Since P_0^{t-1} is not one of the top L paths, there exist L paths $P_1^{t-1}, \ldots, P_L^{t-1}$ at level $t-1$ that end at state s_{t-1} and that have probabilities larger than P_0^{t-1}.

Using Bayes' rule we can write

$$\log p(P_0^t|\mathbf{y}^t) \propto \log p(\mathbf{y}^t|P_0^t) + \log(P_0^t)$$

where we have ignored $\log p(\mathbf{y}^t)$ since it contributes equally to all the paths. Also

$$\log p(\mathbf{y}^t|P_0^t) = \log p(\mathbf{y}^{t-1}|P_0^{t-1}) + \log p(\mathbf{y}_t|b(s_{t-1} \rightarrow s_t))$$

and

$$\log p(b(s_{t-1} \rightarrow s_t)|P_0^{t-1}) = \log p(b(s_{t-1} \rightarrow s_t))$$

as the channel is memoryless.

Combining above relations, we have

$$\begin{aligned}
\log p(P_0^t|\mathbf{y}^t) &\propto \log p(\mathbf{y}^{t-1}|P_0^{t-1}) + \log p(\mathbf{y}_t|(b(s_{t-1} \rightarrow s_t)) + \log p(P_0^t) \\
&\propto \log p(P_0^{t-1}|\mathbf{y}^{t-1}) + \log p(\mathbf{y}_t|b(s_{t-1} \rightarrow s_t)) + \log p(b(s_{t-1} \rightarrow s_t)) \\
&< \log p(P_j^{t-1}|\mathbf{y}^{t-1}) + \log p(\mathbf{y}_t|b(s_{t-1} \rightarrow s_t)) + \log p(b(s_{t-1} \rightarrow s_t)),
\end{aligned}$$

for all $i = 1, \ldots, L$. The last inequality is based on the above definition of P_i^{t-1} as the top L paths into state s_{t-1} at level $t-1$.

The above chain of inequalities suggest that by appending P_i^{t-1} with $b(s_{t-1} \rightarrow s_t)$, we can get L paths a level t that end at state s_t and that have probabilities larger

than the probability of P_0^t. This is a contradiction to our assumption and proves the original claim and the theorem. □

The above theorem also shows that elaborate sorting is not necessary to obtain top L paths of the trellis and a simple extension of the Viterbi decoding algorithm, where we keep the top L paths rather than just one into every state, is sufficient.

Using the efficient combination of list decoding with the M-algorithm, we now need a smaller amount of storage, of size $L2^{\kappa}$, and the decoding delay is reduced to 5κ. We next provide a modified version of the original iterative prior update algorithm to incorporate these delay and buffer size reduction techniques.

Delay and buffer efficient (DBE) iterative prior update algorithm

1. For all of the N coded bits, set initial coded sequence estimates $\hat{\mathbf{d}}_k = \mathbf{0}$, and initial priors $p(\mathbf{d}_k)$ uniform for $k \in \{1, \ldots, K\}$.

2. Set $n = 1$.

3. In order to decode the n^{th} coded bit for all the users, take a block length of size 5κ starting from the n^{th} bit. Note that updated starting priors and starting coded bit estimates for time instants $n-1, n, \ldots, n+5\kappa-1$ are already determined from the decoding window corresponding to bit number $n-1$.

   Working with the current block of coded bits iterate as follows:

   (a) In iteration step i, for $k \in \{1, \ldots, K\}$

      (i) Set

$$\hat{\mathbf{y}}_k^i = \mathbf{S}_k'(\mathbf{r} - \mathbf{S}_{I_k}\mathbf{A}_{I_k}\hat{\mathbf{d}}_{I_k}^{i-1}),$$

where $I_k = \{1, \ldots, k-1, k+1, \ldots K\}$.

(ii) Set $\hat{\mathbf{y}}_k = \hat{\mathbf{y}}_k^i$ and $p(\mathbf{d}_k) = p(\mathbf{d}_k^i)$ in (4.6). Calculate the posterior $p^i(\mathbf{d}_k|\hat{\mathbf{y}}_k^i)$ and the current estimate $\hat{\mathbf{d}}_k^i$. For the posterior calculation, use Theorem 1 to calculate the likelihood for only the top L paths in the trellis. For all other paths use a uniform likelihood that is smaller than the probability of the L^{th} smallest path. Combine likelihood with the prior to find the posterior.

(iii) Set the prior for the next iteration equal to the posterior for this iteration, that is

$$p^{i+1}(\mathbf{d}_k) = p^i(\mathbf{d}_k|\hat{\mathbf{y}}_k^i).$$

(b) Iterate until there is no further change in successive estimates for the first bit of the block for all the users.

4. Increase n by 1. If $n \leq N$, go back to step 3 to repeat the procedure to decode the next coded bit. Otherwise stop.

This modified algorithm requires $O(5\kappa[K - 1 + 2^{\kappa+1}])$ operations per coded bit per user per iteration, slightly higher than the original algorithm of Section 4.3 that uses $O(K - 1 + 2^{\kappa+1})$. The added benefit of the modified algorithm, as explained above, is to reduce the decoding delay to 5κ and storage requirements to $L2^{\kappa}$ where κ is the constraint length of the convolutional code and L is the number of paths kept in the DBE iterative prior update scheme.

We next provide a simulation analysis of the proposed algorithm and a system example illustrating real-time decoding capabilities.

4.5 Numerical Studies

4.5.1 Performance analysis

We first analyze the results of a simulation study on the performance of the proposed algorithm. We also compare it with other low complexity multiuser sequence detection/decoding schemes. A number of iterative techniques that aim to minimize the probability of symbol error have recently been investigated in the literature [63, 64]. Since our goal is to reduce the probability of codeword error, or sequence error, we will not compare our algorithm with the symbol based MAP techniques.

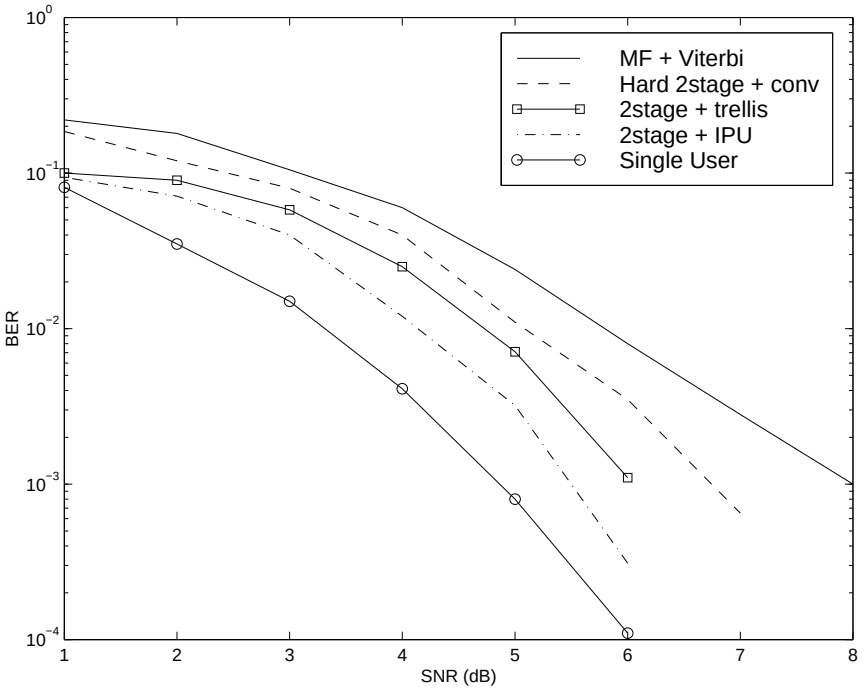


Figure 4.1 : Comparative study of various joint detection and decoding algorithms. "2stage+IPU" refers to the algorithm described in Section 4.4. Number of users(K)=4, Spreading gain (N_c) = 7.

For the first set of simulations we use Gold code sequences of length 7. Our system has 4 users with the amplitude of all other users being twice that of user

1. For error protection, the users have a convolutional code of rate 2/3 and depth 2. The delays of the users were assumed to be distributed uniformly over the bit period. Figure 4.1 provides the bit error rate performance of user 1 for various multiuser detection/decoding algorithms.

The curve labeled "MF+Viterbi" is for a system where hard decisions on coded bits are made based on matched filter outputs, then passed through K single user Viterbi decoders. "Hard2stage+conv" has two-stage hard output multistage detector followed by single user decoders. Since the multistage detector is better in mitigating multiple access interference, the performance of "Hard2stage+conv" is superior to "MF+Viterbi". "2stage+trellis" refers to the algorithm described in [62]. A multistage detector in conjunction with K single user soft Viterbi decoders is used. We have iterated the algorithm two times. Since their algorithm combines detection and decoding, performance is better than "Hard2stage+conv". Finally, "2stage+IPU" refers to the DBE iterative prior update algorithm described in Section 4.4 with two iteration steps. We observe that the iterative prior update consistently outperforms all the other schemes of comparable complexity. It provides about 0.5dB gain over the best algorithm ("2stage+trellis") for a bit error rate of 10^{-3}. It also comes to within 0.5dB of the single user bound for this loaded system.

Simulation results for a system with 12 users, spreading gain of 31 and a convolutional code of rate 2/3, constraint length 5 follow a similar trend and can be found in Figure 4.2. We also illustrate how the normalized probability distribution for the top 10 paths ($L = 10$) evolves over iterations in Figure 4.3. Initially, at iteration step 0, all paths are equally likely. However, as the number of iteration increases, it is possible to distinguish the most likely path with higher reliability. Note that we can get a good estimate of the top path after 2-3 iterations.

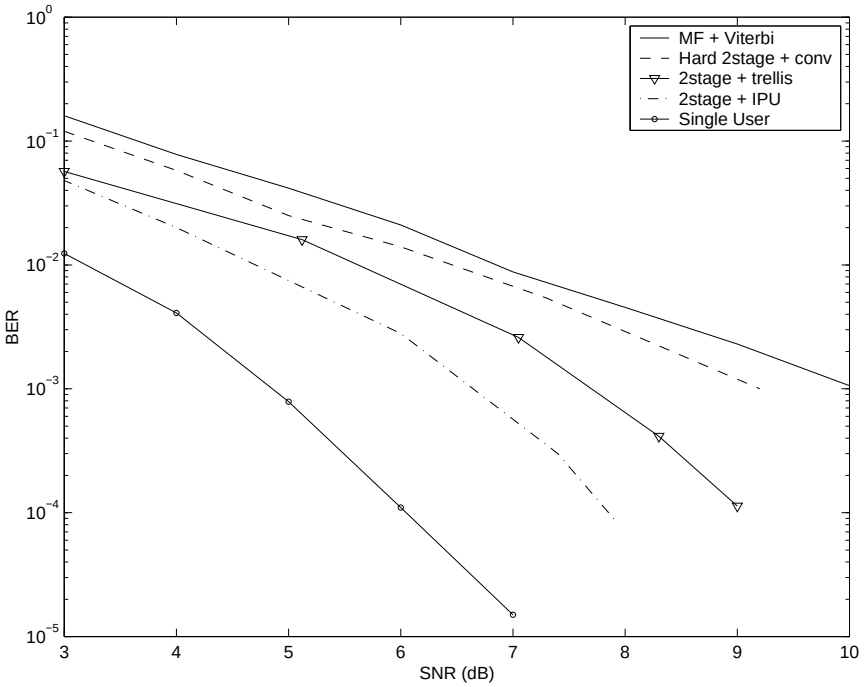


Figure 4.2 : Comparative study of various joint detection and decoding algorithms with a 12 user system and spreading gain 31.

4.5.2 Framework for comparison with other algorithms

We now provide a systematic way of investigating the differences and similarities between the strategies compared in Figure 4.1. We will consider the original iterative prior update algorithm in Section 4.3.

In the i^{th} iteration of the algorithm, coded bit sequence for user k is estimated using (4.6). This optimization can be rewritten as follows:

$$\begin{aligned}\hat{\mathbf{d}}_k &= \arg\max_{\mathbf{d}_k \in \mathcal{C}_k} [\log p(\hat{\mathbf{y}}_k^i | \mathbf{d}_k) + \log p^i(\mathbf{d}_k)] \\ &= \arg\max_{\mathbf{d}_k} \{[\log p(\hat{\mathbf{r}} | \mathbf{d}_k) + \log p^i(\mathbf{d}_k)] \mathcal{I}(\mathbf{d}_k \in \mathcal{C}_k) \mathcal{I}(\mathbf{d}_{I_k} = \hat{\mathbf{d}}_{I_k}^{i-1})\}, \qquad (4.7)\end{aligned}$$

where $\mathcal{I}$ denotes the indicator function. The indicator functions ensure we restrict our sequence search to codewords of the k^{th} codebook and we cancel interference

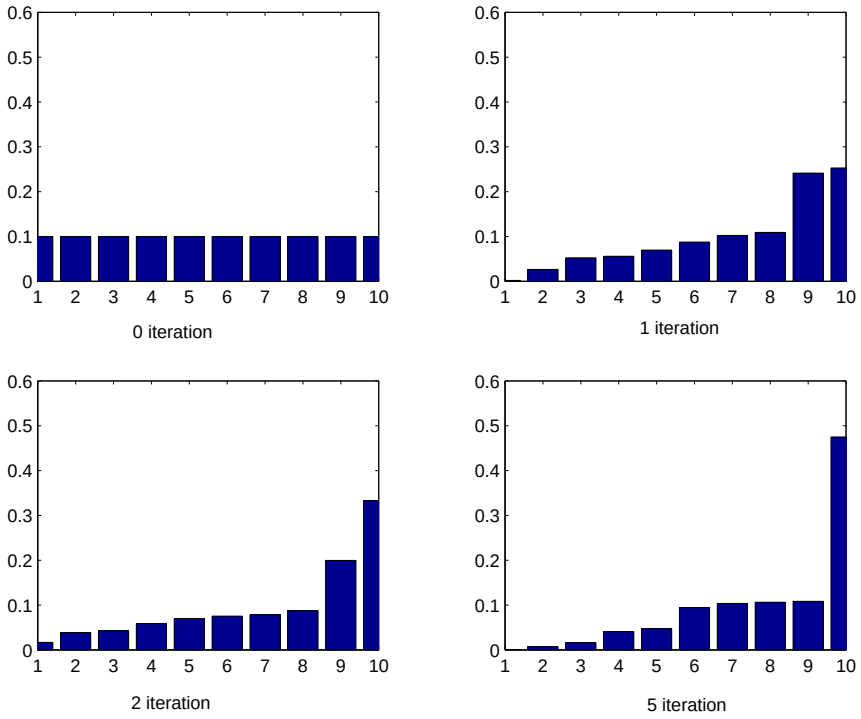


Figure 4.3 : Evolution of the normalized probability distribution of the top 10 paths with the number of iterations

using estimates of interferers from the previous iteration. The algorithm also updates priors $p^i(\mathbf{d}_k)$ using posteriors from the previous iteration step.

The "2stage+trellis" type scheme assumes uniform prior distribution $p^i(\mathbf{d}_k)$ for all the iteration steps i. We observed from the simulation results that updating the prior and passing along extra information improves the performance. The "Hard2stage+conv" strategy ignores the constraint $\mathcal{I}(\mathbf{d}_k \in \mathcal{C}_k)$ in (4.7) and uses a uniform prior $p^i(\mathbf{d}_k)$ to obtain a hard decision on the coded bits. Then the information bit sequence is extracted using single user Viterbi decoders. The "MF+Viterbi" scheme ignores both the constraints $\mathcal{I}(\mathbf{d}_k \in \mathcal{C}_k)$ and $\mathcal{I}(\mathbf{d}_{I_k} = \hat{\mathbf{d}}_{I_k}^{i-1})$ and assumes a uniform prior in making hard decision on the coded bits. Similar to "Hard2stage+conv", information is then extracted using Viterbi decoding. Since no information is fed

back, there are no iterations.

4.5.3 Storage and computational cost

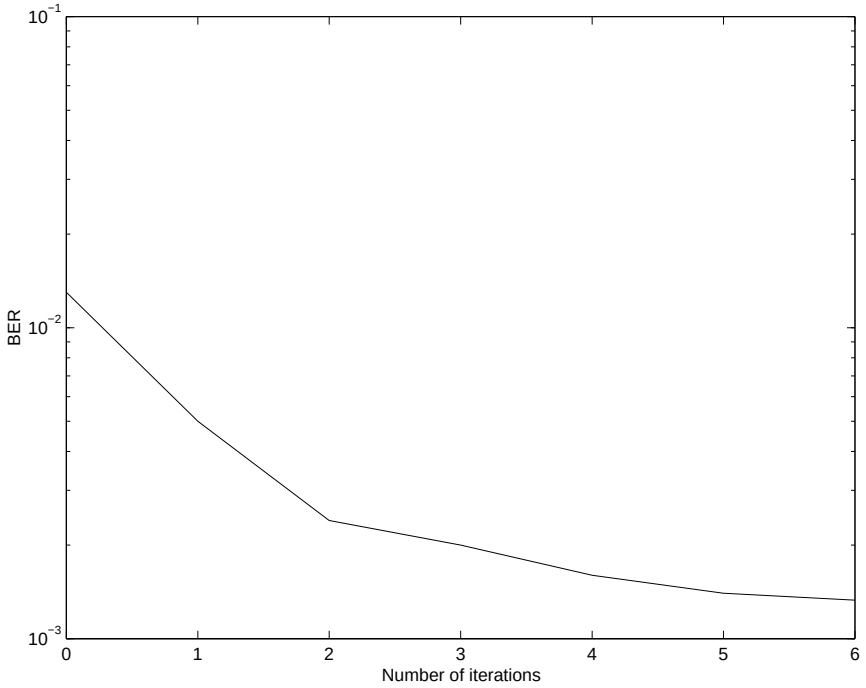


Figure 4.4 : Convergence study of the iterative prior update algorithm, $K = 12$, $N_c = 31$, $L = 6$, convolutional code of rate $R = 2/3$, $\kappa = 5$

Each iteration of the IPU algorithm improves the performance but introduces extra computational cost. Figure 4.4 illustrates the sensitivity of the performance of the IPU algorithm to the number of iterations. The simulation parameters are the same as in Figure 4.1 and the SNR is fixed at 6 dB. We observe that IPU achieves its near optimal performance only after 2-3 iterations. This was also observed in Section 4.5.1. Thus relatively few iterations are needed and the complexity is essentially linear in the number of users.

We also study the sensitivity of DBE-IPU to L, the number of paths stored. By

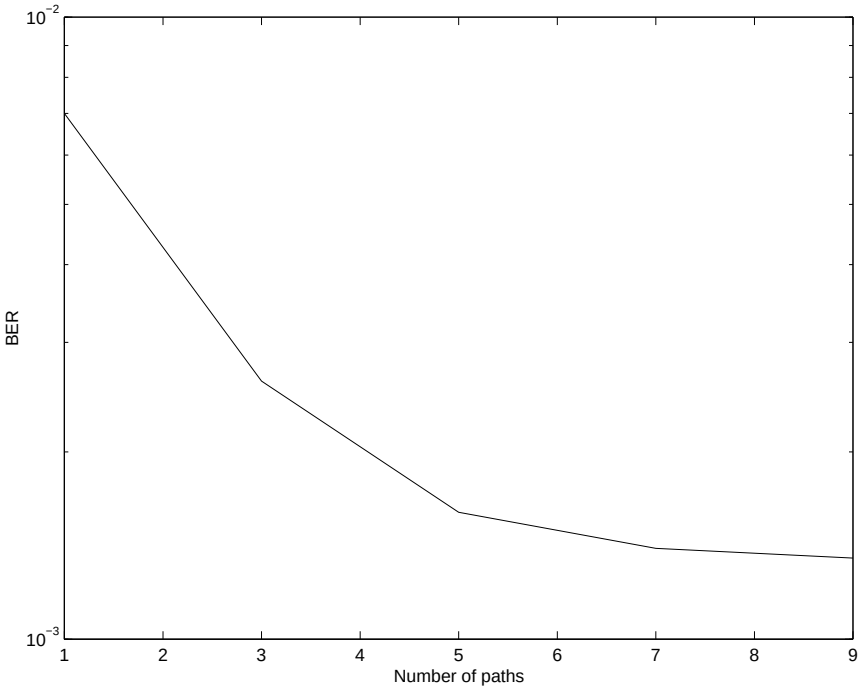


Figure 4.5 : Sensitivity study of the DBE iterative prior update algorithm to storage space, $K = 12$, $N_c = 31$, $R = 2/3$, $\kappa = 5$.

Theorem 1, L is directly related to the storage requirements of the system. Our simulation results of Figure 4.5 show that by storing probabilities for only a few (5-6) number of paths, we can achieve the limiting performance. Hence as claimed, the algorithm has low storage requirements. However we should note that both of the above sensitivity analyzes depend on other system parameters such as the number of users in the system, the spreading gain N_c and the constraint length of the code κ.

4.5.4 Real-time implementation

We now consider a numerical example to illustrate the real-time capabilities of the iterative prior update algorithm. We consider a system with 15 users each transmit-

ting at 20 Kbits/sec. Each user has a convolutional code of rate 1/2 and constraint length 5. If we consider a block-length of 1024 data bits the receiver will have 0.05 seconds to decode all the information bits of all the users.

For the optimal joint multiuser trellis decoding technique, the complexity of the algorithm per user is given by $N2^{K+\kappa+1}/K$, where N is the number of coded bits in a block, K is the number of users and κ is the constraint length. Using $N = 1024$, $K = 15$, $\kappa = 5$ as above, we need a total of 128×10^6 operations per user. If we use a state of the art TI DSP processor (TMS320C67) running at 200Mhz per user, it will require 0.64 seconds to complete all the operations. Even if the processor can exploit the maximum possible 8-way parallelism, which is in most applications not possible, we still would need 0.08 seconds.

On the other hand the DBE iterative prior update algorithm requires $N(K - 1 + 5\kappa 2^{\kappa+1})$ operations per user resulting in a total of $12 * 10^6$ operations which can be completed in 0.06 seconds. If we assume a very realistic 2-way parallelism, the total decoding time becomes 0.03 seconds which is well within the real-time bound. Moreover the decoding delay is only 25 (5κ) bit periods (i.e., only 1.25msec) and we need less than 1 Kbyte of storage space. Coupled with the fact that the performance is better than other low complexity multiuser schemes, we argue that the iterative prior update algorithm is an attractive alternative for real-time implementations.

4.6 Conclusions

In this chapter, we have described a low complexity multiuser joint detection and decoding algorithm for a convolutionally coded DS-CDMA system. Our main goal was to show that this algorithm is suitable for systems requiring real-time communications, such as cellular voice. The optimal joint detector/decoder has exponential

complexity in the number of mobile users and is not practical for such real-time applications. The proposed iterative prior update algorithm is based on iterative interference cancellation together with MAP decoding for *sequences* of coded bits and prior updates at every iteration. It requires a small storage space for storing priors along iterations, has low decoding delay and converges fast. The complexity can be shown to be linear in the number of users. It also has the added benefit of utilizing the already existing hardware for Viterbi decoding. Through simulations, we show that the performance is superior to other low complexity sequence detection strategies. We also provide a numerical real-life example illustrating that the applicability of the algorithm in real-time applications.

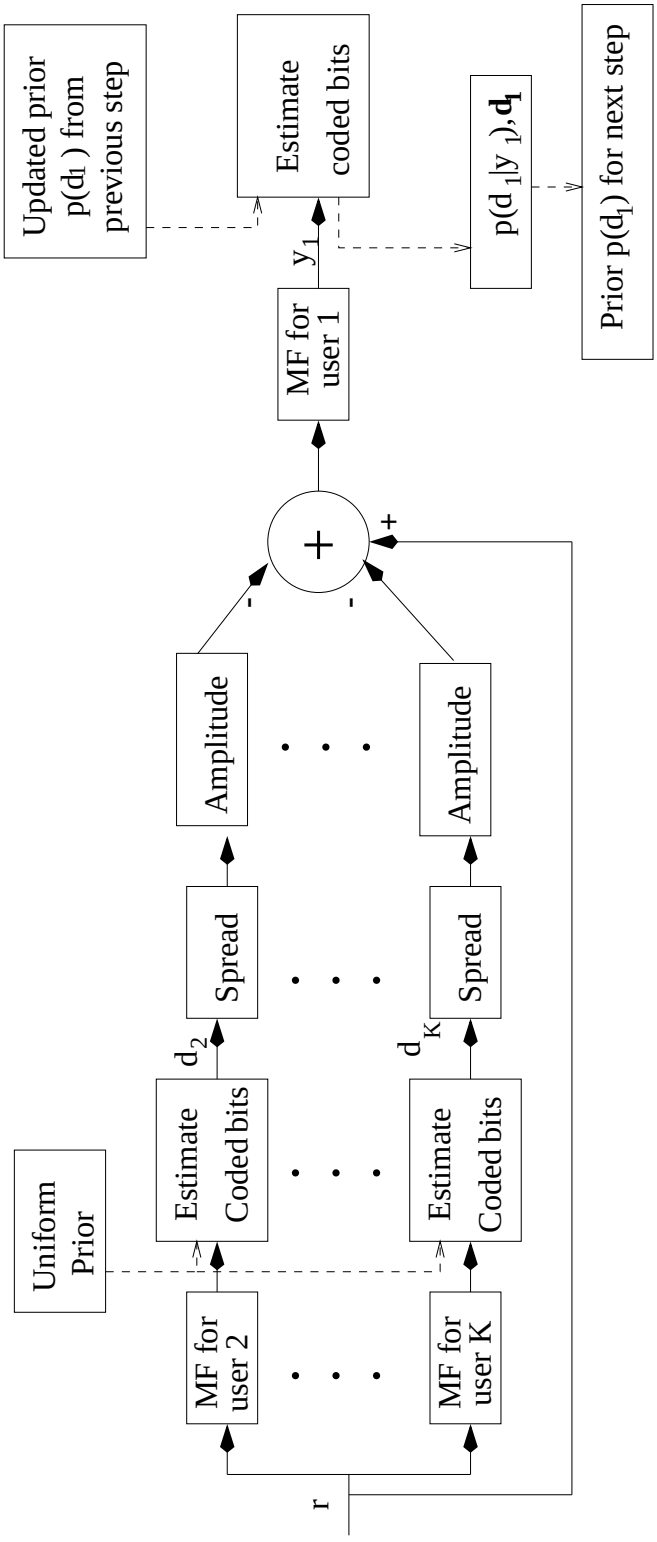


Figure 4.6 : Block diagram of the iterative prior update algorithm for user 1

Chapter 5

Future work

In this chapter we present some unresolved issues that need to be addressed for the completion of our thesis. We also present our initial approach towards these problems.

5.1 Future work on distorted channel capacity

In this chapter we have presented a new capacity measure for an information bits transmitted over a system that can tolerate some amount of errors. We have actually presented an upper bound of the capacity of such a system. However any capacity result is completed only when we can prove not only the upper bound but also the achievability of the capacity. In other words we need to propose a coding and decoding system that can achieve this capacity.

Fortunately the achievability proofs are usually easier to arrive at. As a simplistic approach we can employ a lossy source coding technique over the information sequence and transmit this source coded information stream in an error free manner over the channel. The available results will prove the achievability of the capacity region. However we would also like to come up with some geometrical solution to the achievability problem.

We have primarily looked at distorted channel capacity of the single user system. However a wireless channel is usually shared by a number of users and it would be

interesting to extend the results to multiple user scenarios. We have presented some preliminary results about multiple users, but some further study is necessary in this direction.

Finally we investigated the optimal rate distribution in a multi-state fading channel scenario. However in practical implementation another related problem is the optimal power distribution to maximize the throughput of the system. The SNR of the channel is determines the Shannon capacity of the channel at that particular state. The SNR expression appears in the distorted channel capacity expression also. The SNR of the channel is determined by the noise variance, which is a channel characteristic, and the signal power which is controlled by the user. Usually the average power of the system is fixed. So far we have considered scenarios where the SNR of the system at each state is fixed, but now we will have an additional degree of freedom and this will lead to another interesting optimization problem.

5.2 Future work on maximal basis decoding

Most of the future work in this direction is in simulation of practical systems. We have proposed a new algorithm and we have studied the performance of the algorithm for low constraint length codes. However we will need to study the performance of convolutional codes with large constraint length. An interesting study would be to find out the number of partial codewords we need to consider as the constraint length increases.

Finally we need to put a constraint on the maximum amount of allowable computation and study the performance of optimal Viterbi decoding for convolutional codes of low constraint length with that of convolutional codes of larger constraint length but decoded by our algorithm.

In this chapter we have primarily focussed on the systematic convolutional codes. It would be natural to extend the results to non-systematic convolutional codes. Also in our algorithms we have made "hard-decisions" for each of the basis bits and then the convolutional codes. Instead of hard decisions we can decide to use soft decisions at each stage. This result will enable us to use our algorithm to decode turbo codes also.

5.3 Future work on joint detection and decoding

The results in this section are mostly mature. We would also like to use our maximum basis decoding technique in conjunction with iterative interference cancellation methods. Our initial results are very encouraging. Finally once we have the distorted channel capacity results for multiple users, we would like to compare the performance of the iterative detection and decoding algorithms with that described by the theoretical limits.

Bibliography

[1] I. Brodksy, *The Revolution in Personal Telecommunications*. Artech House Publishers, 1995. Boston, MA.

[2] T. S. Rappaport, *Wireless Communications: Principle and Practice*. Prentice Hall Publishing, 1996. Upper Saddle River, NJ.

[3] Y. C. Lee, "Smaller Cells for Greater Performance," *IEEE Communications Magazine*, pp. 19–23, Feb. 1991.

[4] R. Steele, J. Whitehead, and W. C. Wong, "System Aspects of Cellular Radio," *IEEE Communications Magazine*, vol. 33, pp. 80–86, Jan. 1995.

[5] M. Rahnema, "Overview of the GSM System and Protocol Architecture," *IEEE Communications Magazine*, vol. 31, pp. 92–100, Apr. 1993.

[6] Telecommunications Industry Association, *TIA/EIA/IS-95 Mobile Station-Base Station Compatibility Standard for Dual-Mode Wideband Spread Spectrum Cellular System*, July 1993. http://www.tiaonline.org/.

[7] J. C. Liberti and T. S. Rappaport, *Smart Antennas for Wireless Communications: IS-95 and Third Generation CDMA Applications*. Prentice Hall Publishing, 1999. Upper Saddle River, NJ.

[8] R. L. Pickholtz, D. L. Schilling, and L. B. Milstein, "Theory of Spread-Spectrum Communications - a Tutorial," *IEEE Transactions on Communica-*

tion, vol. COM-30, no. 5, pp. 129–158, 1982.

[9] A. J. Viterbi, *CDMA Principles of Spread Spectrum Communication*. Addison-Wesley Publishing Company, 1995.

[10] M. C. Oliphant, "The Mobile Phone Meets the Internet," *IEEE Spectrum Magazine*, vol. 36, Aug. 1999.

[11] E. Berruto, M. Gudmundson, R. Menolascino, W. Mohr, and M. Pizarroso, "Research Activities on UMTS Radio Interface, Network Architectures, and Planning," *IEEE Communications Magazine*, vol. 36, pp. 82–95, Feb 1998.

[12] E. T. S. Institute, "The ETSI UMTS Terrestial Radio Access (UTRA) ITU-R RTT Candidate Submission," *http://www.etsi.org/smg/UTRA/utra.pdf*, May/June 1998.

[13] T. I. Association, "The CDMA2000 ITU-R RTT Candidate Submission," *http://www.t1.org/index/0506.htm/8p110690.pdf*, April 1998.

[14] A. of Radio Industries and Businesses, "Japan's Proposal for Candidate Radio Transmission Technology on IMT-2000:W-CDMA," *http://www.itu.int/imt/2-radio-dev/proposals/*, June 1998.

[15] D. N. Knisely, S. Kumar, S. Laha, and S. Nanda, "Evolution of Wireless Data Services: IS-95 to CDMA2000," *IEEE Communications Magazine*, vol. 36, pp. 140–149, Oct. 1998.

[16] C. E. Shannon, "A Mathematical Theory of Communication," 1948.

[17] J. M. Wozencraft and R. S. Kennedy, "Modulation and Demodulation for Probabilistic Decoding," *IEEE Transactions on Information Theory*, pp. 291–297,

July 1966.

[18] S.V.Hanly and D. Tse, "Multi-access Fading Channels: Part II: Delay-Limited Capacities," *IEEE Transactions on Information Theory*, 1997.

[19] L.H.Ozarow, S. Shamai, and A. D. Wyner, "Information Theoretic Considerations for Cellular Mobile Radio," *IEEE Transactions on Vehicular Technology*, vol. 43, pp. 359–38, May 1994.

[20] J. G. Proakis, *Digital Communications*. McGraw-Hill, 1995.

[21] R. L. Graham, D. E. Knuth, and O. Patashnik, *Concrete Mathematics*. Addison-Wesley Publishing Company, 1989.

[22] T. M. Cover and J. A. Thomas, *Elements of Information Theory*. A Wiley-Interscience Publication, 1991.

[23] T. Berger, *Rate Distortion Theory: A Mathematical basis for Data Compression*. Prentice-Hall, 1971.

[24] H. Wang and N. Moayeri, "Finite State Markov Channel - a Useful Model for Radio Communication Channels," *IEEE Transactions on Vehicular Technology*, vol. 44, pp. 163–171, 1995.

[25] R. E. Blahut, *Theory and Practice of Error Control Codes*. Addison-Wesley Publishing Company, Massachusetts, 2 ed., 1984.

[26] A. Viterbi, "Error Bound on Convolutional Codes and an Asymptotically Optimum Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. 13, pp. 260–269, April 1967.

[27] R. W. Hamming, "Error Detecting and Correcting Codes," *Bell Systems Technical Journal*, vol. 29, pp. 147–160, 1950.

[28] M. J. E. Golay, "Notes on Digital Coding," *Proceedings of IRE*, p. 657, 1949.

[29] I. S. Reed, "A Class of Multiple Error Correcting Codes and a Decoding Structure," *IEEE Transactions on Information Theory*, pp. 38–49, sep 1954.

[30] D. E. Muller, "Application of Boolean Algebra to Switching Circuit Design," *IEEE Transactions on Information Theory*, pp. 6–12, sep 1954.

[31] I. S. Reed and G. Solomon, "Polynomial Codes over Certain Finite Fields," *Journal of SIAM*, pp. 300–304, jun 1960.

[32] R. C. Bose and D. K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," *Information Control*, pp. 68–79, mar 1960.

[33] A. Hocquenghem, "Codes Corecteurs d'Erreurs," *Chiffres*, pp. 147–156, 1959.

[34] P. Elias, "Coding for Noisy Channels," *IRE Convention Record*, pp. 37–47, 1955.

[35] J. M. Wozencraft and B. Reiffen, *Sequential Decoding*. MIT Press, Cambridge, MA, 1961.

[36] J. K. Omura, "On the Viterbi Decoding Algorithm," *IEEE Transactions on Information Theory*, pp. 177–179, Jan 1969.

[37] G. D. Forney, Jr., "Convolutional Codes II: Maximum Likelihood Decoding," *IEEE Transactions on Information Theory*, pp. 363–378, May 1972.

[38] R. M. Fano, "A Heuristic Discussion of Probabilistic Decoding," *IEEE Transactions on Information Theory*, vol. 9, pp. 64–74, 1963.

[39] F. Jelinek, "An Upper Bound on Moments of Sequential Decoding Effort," *IEEE Transactions on Information Theory*, pp. 464–468, Jul 1969.

[40] K. Zigangirov, "Some Sequential Decoding Procedures," *Probl. Peredachi Inf.*, pp. 13–25, 1966.

[41] F. Jelinek, "A Fast Sequential Decoding Algorithm Using a Stack," *IBM J. Res and Dev.*, pp. 675–685, Nov 1969.

[42] D. Haccoun and M. J. Ferguson, "Generalized Stack Algorithms for Decoding Convolutional Codes," *IEEE Transactions on Information Theory*, vol. 21, pp. 638–651, 1975.

[43] J. M. Geist, "Search Properties of Some Sequential Decoding Algorithm," *IEEE Transactions on Information Theory*, vol. 19, pp. 519–526, 1973.

[44] P. R. Chevillat and D. J. Costello, Jr, "A Multiple Stack Algorithm for Erasure Free Decoding of Convolutional Codes," *IEEE Transactions on Communications*, vol. 25, pp. 1460–1470, 1977.

[45] H. H. Ma, "The Multiple Stack Algorithm Implemented on a Zilog Z-80 Microcomputer," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1876–1882, 1980.

[46] J. L. Massey, *Threshold Decoding*. MIT Press, 1998. Cambridge, MA.

[47] L. D. Rudolph, "Generalized Threshold Decoding of Convolutional Codes," *IEEE Transactions on Information Theory*, vol. 16, pp. 739–745, 1970.

[48] M. A. V. A. Dhalokia and D. L. Bitzer, "Table-driven Decoding of Binary One-half Rate Non Systematic Convolutional Codes," in *IEEE International*

Symposium on Information Theory, p. 270, 1993.

[49] Y. S. Han, C. R. P. Hartmann, and C. Chen, "Efficient Priority-First Search Maximum-Likelihood Soft-Decision Decoding of Linear Block Codes," *IEEE Transactions on Information Theory*, vol. 39, pp. 1514–1523, 1993.

[50] M. P. C. Fossorier and S. Lin, "Soft-Decision Decoding of Linear Block Codes Based on Ordered Statistics," *IEEE Transactions on Information Theory*, vol. 41, pp. 1379–1396, 1995.

[51] V. B. Balakirskii and B. D. Kudryashov, "List Decoding of Convolutional Codes," in *Problems of Information Transmission*, pp. 11–17, 1989.

[52] K. Zigangirov and H. Osthoff, "Analysis of List Decoding for Convolutional Codes," in *Proceedings of 1993 International Symposium on Information Theory (ISIT'93)*, p. 269, 1993.

[53] K. R. Narayanan and G. L. Stuber, "List Decoding of Turbo Codes," *IEEE Transactions on Communications*, vol. 41, pp. 754–762, 1998.

[54] C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit Error Correcting Coding and Decoding: Turbo Codes," in *Proceedings of 1993 International Conference on Communications (ICC'93)*, pp. 1064–1070, 1993.

[55] S. Verdú, *Multiuser Detection*. Cambridge University Press, 1st ed., 1998. New York, NY.

[56] S. Verdú, "Minimum Probability of Error for Asynchronous Gaussian Multiple-Access Channels," *IEEE Transactions on Information Theory*, vol. 32, no. 1, pp. 85–96, 1986.

[57] A. Duel-Hallen, "Decorrelating Decision Feedback Multiuser Detector for Synchronous Code-Division Multiple-Access Channel," *IEEE Transactions on Communications*, vol. 41, pp. 285–290, Feb. 1993.

[58] R. Lupas and S. Verdú, "Linear Multiuser Detectors for Synchronous Code-Division Multiple-Access Channels," *IEEE Transactions on Information Theory*, vol. 41, no. 4, pp. 123–136, 1989.

[59] M. K. Varanasi and B. Aazhang, "Multistage Detection in Asynchronous Code Division Multiple Access Communications," *IEEE Transactions on Communications*, vol. 38, pp. 509–519, Apr. 1990.

[60] T. Giallorenzi and S. Wilson, "Multiuser ML Sequence Estimator for Convolutional Coded Asynchronous System," *IEEE Transactions on Communications*, vol. 44, no. 8, pp. 997–1008, 1996.

[61] T. Giallorenzi and S. Wilson, "Suboptimum Multiuser Receivers for Convolutionally Coded Asynchronous DS-CDMA Systems," *IEEE Transactions on Communications*, vol. 44, no. 9, pp. 1183–1196, 1996.

[62] U. Fawer and B. Aazhang, "Multiuser Receivers for Code-Division Multiple-Access Systems with Trellis-Based Modulation," *IEEE Journal on Selected Areas in Communications*, vol. 14, no. 7, pp. 1602–1609, 1996.

[63] P. Alexander, M. Reed, J. Asenstorfer, and C. Schlegel, "Iterative Multiuser Interference Reduction: Turbo CDMA," *IEEE Transactions on Communication*, vol. 47, no. 7, pp. 1008–1014, 1999.

[64] X. Wang and H. Poor, "Iterative (Turbo) Soft Interference Cancellation and

Decoding for Coded CDMA," *IEEE Transactions on Communication*, vol. 47, no. 7, pp. 1046–1061, 1999.

[65] M. Moher, "An Iterative Multiuser Decoder for Near-capacity Communications," *IEEE Transactions on Communications*, vol. 46, pp. 870–880, July 1998.

[66] A. Hafeez and W. E. Stark, "Combined Decision Feedback Multiuser Detection/Soft-decision Decoding for CDMA Channels," in *IEEE 46th Vehicular Technology Conference*, pp. 382–386, 1996.

[67] L. Bahl, J. Cocke, F. Jelinek, and J.Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error Rate," *IEEE Transactions on Information Theory*, vol. 20, pp. 284–287, 1974.

[68] J. Anderson and S. Mohan, "Sequential coding algorithms: A survey and cost analysis," *IEEE Transactions on Communications*, vol. 32, pp. 169–176, 1984.